Table of Contents

Part I Введение 1

1 Техническая поддержка ... 1
2 Ограничения пробной версии ... 3
3 Информация в интернете и ссылки ... 3

Part II Распространение 5

1 Библиотеки пакета Reports.WEB ... 7
2 Библиотеки пакета Reports.NET ... 8
3 Скрипты пакета Reports.JS ... 9
4 Библиотеки пакета Reports.WPF ... 10
5 Скрипты пакета Reports.PHP ... 12
6 JAR файлы пакета Reports.JAVA ... 13
7 Библиотеки пакета Dashboards.WEB ... 14
8 Библиотеки пакета Dashboards.WIN ... 15
9 Скрипты пакета Dashboards.JS ... 17
10 Скрипты пакета Dashboards.PHP ... 18
11 Файлы приложения Stimulsoft Designer ... 19
12 Файлы приложения Stimulsoft Designer.JS ... 24
13 Файлы приложения Stimulsoft Demo ... 25

Part III Reports and Dashboards for WebForms 28

1 HTML5 вьювер ... 28

Как это работает ... 31
Активация ... 32
Отображение отчета и дашборда ... 33
Подключение данных ... 37
Локализация вьювера ... 43
Использование тем ... 44
Основные возможности ... 47
Печать отчета ... 48
Экспорт отчета и дашборда ... 50
Режимы отображения ... 55
Работа с параметрами ... 58
Работа с закладками ... 60
Динамическое сворачивание, сортировка и детализация ... 62
Редактирование построенного отчета ... 66
Отправка отчета по Email ... 68
Вызов дизайнера из вьювера ... 71
Кэширование ... 72
Экспорт и печать из кода ... 75
События вьювера ... 76
Время ожидания ... 78
Настройки вьювера ... 80

© 2003-2026 Stimulsoft

2 HTML5 дизайнер 102
Как это работает 104
Активация 105
Редактирование отчета и дашборда 106
Создание нового отчета и нового дашборда 109
Просмотр 110
Дополнительные возможности просмотра 113
Сохранение отчета и дашборда 114
Локализация дизайнера 118
Использование тем 120
Кэширование 121
События 124
Время ожидания 126
Добавление пользовательских функций 128
Настройки 129

Part IV Reports and Dashboards for ASP.NET MVC 155

1 HTML5 вьювер 155
Как это работает 158
Активация 158
Отображение отчета и дашборда 159
Подключение данных 163
Локализация вьювера 170
Использование тем 171
Основные возможности 174
Печать отчета 175
Экспорт отчета и дашборда 178
Режимы отображения 183
Работа с параметрами 186
Работа с закладками 189
Динамическое сворачивание, сортировка и детализация 191
Редактирование построенного отчета 195
Отправка отчета по Email 197
Вызов дизайнера из вьювера 200
Кэширование 201
Вспомогательные методы 204
Экспорт и печать из кода 208
Время ожидания 210
Настройки вьювера 212
2 HTML5 дизайнер 235
Как это работает 238
Активация 238
Редактирование отчета и дашборда 239
Создание нового отчета и нового дашборда 242
Просмотр 243
Дополнительные возможности просмотра 246
Сохранение отчета и дашборда 248
Локализация дизайнера 252
Использование тем 254
Кэширование 255
Вспомогательные методы 259
Время ожидания 262
Добавление пользовательских функций 264

© 2003-2026 Stimulsoft

Настройки ... 265

Part V Reports and Dashboards for ASP.NET Core MVC 292

1 HTML5 вьювер ... 293

Как это работает ... 295
Активация ... 296
Отображение отчета и дашборда ... 297
Подключение данных ... 301
Локализация ... 309
Использование тем ... 310
Основные возможности ... 313
Печать отчета ... 314
Экспорт отчета и дашборда ... 317
Режимы отображения ... 322
Работа с параметрами ... 325
Работа с закладками ... 328
Динамическое сворачивание, сортировка и детализация ... 330
Редактирование отчета ... 334
Отправка отчета по Email ... 336
Вызов дизайнера из вьювера ... 339
Кэширование ... 340
Вспомогательные методы ... 344
Экспорт и печать из кода ... 349
Время ожидания ... 350
Настройки вьювера ... 352

2 HTML5 дизайнер ... 375

Как это работает ... 378
Активация ... 378
Редактирование отчета и дашборда ... 379
Создание нового отчета и нового дашборда ... 382
Просмотр ... 383
Дополнительные возможности просмотра ... 386
Сохранение отчета и дашборда ... 387
Локализация дизайнера ... 392
Использование тем ... 393
Кэширование ... 395
Вспомогательные методы ... 398
Время ожидания ... 402
Добавление пользовательских функций ... 404
Настройки ... 405

Part VI Reports and Dashboards for ASP.NET Core Razor 433

1 HTML5 вьювер ... 433

Как это работает ... 436
Активация ... 436
Отображение отчета и дашборда ... 437
Подключение данных ... 441
Локализация ... 449
Использование тем ... 450
Основные возможности ... 453

© 2003-2026 Stimulsoft

Печать отчета ... 454
Экспорт отчета и дашборда ... 457
Режимы отображения ... 462
Работа с параметрами ... 465
Работа с закладками ... 468
Динамическое сворачивание, сортировка и детализация ... 470
Редактирование отчета ... 474
Отправка отчета по Email ... 476
Вызов дизайнера из вьювера ... 479
Кэширование ... 480
Вспомогательные методы ... 483
Экспорт и печать из кода ... 487
Время ожидания ... 488
Настройки вьювера ... 490

2 HTML5 дизайнер ... 513

Как это работает ... 515
Активация ... 516
Редактирование отчета и дашборда ... 517
Создание нового отчета и нового дашборда ... 520
Просмотр ... 522
Дополнительные возможности просмотра ... 525
Сохранение отчета и дашборда ... 526
Локализация дизайнера ... 531
Использование тем ... 533
Кэширование ... 534
Вспомогательные методы ... 537
Время ожидания ... 540
Добавление пользовательских функций ... 542
Настройки ... 543

Part VII Reports and Dashboards for Blazor 571

1 Вьювер ... 571

Активация ... 573
Отображение отчета ... 574
Подключение данных ... 576
Локализация вьювера ... 584
Использование тем ... 585
Основные возможности ... 588
Печать отчета ... 589
Экспорт отчета ... 592
Режимы отображения ... 597
Работа с параметрами ... 601
Работа с закладками ... 604
Динамическое сворачивание, сортировка и детализация ... 607
Редактирование построенного отчета ... 612
Отправка отчета по Email ... 613
Вызов дизайнера из вьювера ... 615
Экспорт и печать из кода ... 616
События вьювера ... 618
Настройки вьювера ... 619

2 Дизайнер ... 639

Активация ... 640
Редактирование отчета ... 641

© 2003-2026 Stimulsoft

Создание нового отчета ... 643
Просмотр отчета ... 645
Дополнительные возможности просмотра ... 647
Сохранение отчета ... 649
Локализация дизайнера ... 652
Использование тем ... 653
События дизайнера ... 654
Настройки ... 656

Part VIII Reports and Dashboards for JS 680

1 Быстрый старт ... 681
Чистый JavaScript ... 681
JavaScript и CDN сервисы ... 683
Angular JS ... 687
React JS ... 691
Vue JS ... 694
Node JS ... 697

2 HTML5 вьювер ... 699
Как это работает ... 701
Отображение отчета ... 702
Использование тем ... 706
Печать отчета ... 709
Экспорт отчета ... 712
Режимы отображения ... 715
Работа с параметрами ... 718
Работа с закладками ... 720
Динамическое сворачивание, сортировка и детализация ... 722
Редактирование построенного отчета ... 726
Отправка отчета по Email ... 728
Вызов дизайнера из вьювера ... 730
События вьювера ... 731
Настройки вьювера ... 747

3 HTML5 дизайнер ... 762
Как это работает ... 765
Редактирование отчета ... 765
Создание нового отчета ... 769
Просмотр ... 771
Дополнительные возможности предварительного просмотра ... 773
Сохранение отчета ... 774
Использование тем ... 776
События дизайнера ... 778
Пользовательская настройка дизайнера ... 790
Настройки ... 793

4 Engine ... 811
Активация ... 811
Подключение файлов данных ... 812
Подключение SQL данных ... 814
Локализация ... 821
Загрузка и сохранение отчета ... 823
Сохранение построенных отчетов ... 824
Получение доступа к страницам ... 824
События отчета ... 825

© 2003-2026 Stimulsoft

Добавление пользовательских функций ... 832

Part IX Reports and Dashboards for PHP 833

1 Генератор отчетов ... 833

Использование ... 835
Уменьшение времени загрузки скриптов ... 841
Активация лицензии ... 843
Загрузка и сохранение отчета ... 846
Построение отчета ... 850
Построение отчета на стороне сервера ... 852
Обработчик событий PHP ... 855
Работа с данными ... 861
Подключение файлов данных ... 865
Подключение SQL адаптеров данных ... 870
Работа с переменными отчета ... 878
Подключение пользовательских шрифтов ... 884
Печать отчета из кода ... 885
Экспорт отчета из кода ... 886
События генератора отчетов ... 894

2 HTML5 вьювер ... 909

Использование ... 911
Активация лицензии ... 914
Отображение отчета ... 914
Локализация ... 915
Печать отчета ... 915
Экспорт отчета ... 919
Режимы отображения ... 925
Работа с переменными отчета ... 928
Работа с закладками ... 930
Динамическое сворачивание, сортировка и детализация ... 932
Редактирование построенного отчета ... 936
Отправка отчета по Email ... 936
Вызов дизайнера ... 939
Визуальное оформление ... 940
События вьювера ... 942
Настройки вьювера ... 967

3 HTML5 дизайнер ... 989

Использование ... 990
Активация лицензии ... 993
Создание и редактирование отчета ... 994
Сохранение отчета ... 996
Локализация ... 1000
Просмотр ... 1002
Визуальное оформление ... 1004
Добавление пользовательских функций ... 1004
События дизайнера ... 1006
Настройки ... 1016

Part X Reports and Dashboards for Python 1043

1 Генератор отчетов ... 1043

Использование ... 1044
Уменьшение времени загрузки скриптов ... 1054

© 2003-2026 Stimulsoft

Активация лицензии ... 1056
Загрузка и сохранение отчета ... 1058
Построение отчета ... 1063
Построение отчета на стороне сервера ... 1065
Обработчик событий ... 1067
Работа с данными ... 1073
Подключение файлов данных ... 1078
Подключение SQL адаптеров данных ... 1083
Работа с переменными отчета ... 1090
Подключение пользовательских шрифтов ... 1097
Печать отчета из кода ... 1098
Экспорт отчета из кода ... 1098
События генератора отчетов ... 1106

2 HTML5 вьювер ... 1121

Использование ... 1122
Активация лицензии ... 1127
Отображение отчета ... 1128
Локализация ... 1129
Печать отчета ... 1129
Экспорт отчета ... 1132
Режимы отображения ... 1137
Работа с переменными отчета ... 1140
Работа с закладками ... 1143
Динамическое сворачивание, сортировка и детализация ... 1144
Редактирование построенного отчета ... 1148
Отправка отчета по Email ... 1149
Вызов дизайнера ... 1152
Визуальное оформление ... 1153
События вьювера ... 1155
Настройки вьювера ... 1175

3 HTML5 дизайнер ... 1197

Использование ... 1198
Активация лицензии ... 1203
Создание и редактирование отчета ... 1203
Сохранение отчета ... 1206
Локализация ... 1209
Просмотр ... 1210
Визуальное оформление ... 1212
События дизайнера ... 1213
Настройки ... 1223

Part XI Reports and Dashboards for Net 1250

1 Активация ... 1251

2 WinForms вьювер ... 1252

Как отобразить отчет ... 1252
Dot-Matrix режим ... 1254

3 Добавление пользовательских функций ... 1258

Part XII Reports.JAVA 1259

1 Активация ... 1259

2 HTML5 дизайнер ... 1260

© 2003-2026 Stimulsoft

Установка и описание дизайнера ... 1260
Шаблоны соединений ... 1264

Part XIII Reports.WPF 1266

1 Активация ... 1266
2 WPF вьювер ... 1267
Как отобразить отчет ... 1268
Dot-Matrix режим ... 1268

Part XIV Генератор отчетов 1272

1 Данные ... 1273
Бизнес-объекты в NET, Web ... 1273
Получение данных при помощи протокола OData ... 1283
Использование значения выражения в переменной ... 1284
2 Наследование отчетов ... 1286
Основные подходы ... 1286
3 Режим "справа налево" ... 1287
Вьювер отчетов для WinForms ... 1287
Вьювер и дизайнер отчетов WPF ... 1288
Значки ... 1289
4 Экспорт отчетов ... 1297
Экспорт отчетов из кода ... 1300
Использование метода ExportDocument ... 1300
Использование сервисов экспорта ... 1302
Форматы с фиксированной разметкой страницы ... 1304
PDF ... 1304
ZUGFeRD ... 1319
Особенности экспорта PDF/A ... 1321
Microsoft Power Point ... 1323
XPS ... 1325
Web-документы ... 1326
HTML ... 1327
MHT ... 1329
Текстовые форматы ... 1331
TXT ... 1331
RTF ... 1334
Word ... 1337
ODT ... 1340
Электронные таблицы ... 1342
Excel ... 1343
Excel 2007/2010 ... 1345
ODS ... 1348
Данные ... 1350
CSV ... 1350
DBF ... 1351
XML ... 1354
DIF ... 1356
SYLK ... 1357
Изображения ... 1358
5 Скрипты ... 1359
Язык программирования отчета ... 1359

© 2003-2026 Stimulsoft

Код отчета 1360
6 Отчеты как сборки 1363
7 Работа со сборками 1364
8 Автономные отчеты 1366

Part XV Forms.Web 1366

1 Интеграция с ASP.NET Core 3.1 1366

Index 0

© 2003-2026 Stimulsoft

1 Введение

Рады приветствовать Вас в онлайн-версии документации **Stimulsoft Reports**, которая посвящена использованию **API** для наших продуктов. Здесь мы рассмотрим, как из кода осуществлять передачу данных в отчет, экспортировать отчеты в различные типы файлов, наследовать отчеты, работать с компонентами и другое:

Добро пожаловать в **Stimulsoft**:

- Узнать о технической поддержке
- Ограничения пробной версии
- Информация в интернете и ссылки
- Посмотреть триал версию
- Engine генератора отчетов

- Reports and Dashboards for ASP.NET WebForms
- Reports and Dashboards for ASP.NET MVC
- Reports and Dashboards for ASP.NET Core MVC
- Reports and Dashboards for ASP.NET Core Razor
- Reports and Dashboards for Blazor
- Reports and Dashboards for WinForms

- Reports and Dashboards for JS
- Reports and Dashboards for PHP
- Reports and Dashboards for Python
- Reports for Java
- Reports for WPF
- Forms for Web

Первая часть документации, содержит в себе описание по работе с визуальными частями продуктов **Stimulsoft**.

1.1 Техническая поддержка

Зарегистрированные пользователи и пользователи с активной подпиской, могут получить техническую поддержку.

© 2003-2026 Stimulsoft



- Для получения технической поддержки, обращаться следует по адресу электронной почты: support@stimulsoft.com
- Для решения вопросов оплаты, программы подписки, обращаться по адресу электронной почты: sales@stimulsoft.com
- Для решения других вопросов, обращаться по адресу электронной почты: info@stimulsoft.com
- Если у вас возникли проблемы с нашими продуктами, Вы можете связаться с нами через форму обратной связи на https://www.stimulsoft.com/ru/support

Когда Вы обращаетесь к нам за технической поддержкой, используйте тот же адрес электронной почты, который использовался при покупке подписки нашего продукта. В противном случае, нам будет трудно определить Вас как зарегистрированного пользователя. Это может сделать наш ответ медленным. Пожалуйста, сообщайте нам, когда Ваш адрес электронной почты изменяется.

Чтобы решить проблему быстро, нам необходима следующая информация:

- Наименование продукта и его версия;

© 2003-2026 Stimulsoft

- Подробное описание проблемы и как ее воспроизвести;
- Ваша операционная система (98, ME, 2000, NT, XP, Vista, Windows7, Windows8 или Windows10), ее версия и локализация, установленные пакеты обновления;
- Версия Microsoft .NET Framework, его локализация и установленные пакеты обновления;
- Наименование Вашей среды разработки и ее версия;
- Дополнительная информация, которая может помочь нам решить проблему.

1.2 Ограничения пробной версии

Пробная версия

Бесплатная пробная версия продуктов Stimulsoft является полнофункциональной. Ознакомительная версия Stimulsoft, имеет следующие ограничения:

- Период использования пробной версии для компонентов составляет 60 дней;
- Водяной знак Trial, который будет напечатан на каждой странице отчета или на панели индикаторов (дашбордов).

Лицензионная версия

Если на страницах отчета отображается водяной знак TRIAL, значит Вы используете незарегистрированную версию. Приобретите подписку для необходимого продукта. Войдите в свой аккаунт и используя лицензионный ключ, активируйте продукт.

Ознакомьтесь как активировать

- ASP.NET HTML5 Viewer
- ASP.NET HTML5 Designer
- ASP.NET MVC HTML5 Viewer
- ASP.NET MVC HTML5 Designer
- ASP.NET Core HTML5 Viewer
- ASP.NET Core HTML5 Designer
- JavaScript компоненты
- WinForms компоненты
- Java компоненты
- WPF компоненты

1.3 Информация в интернете и ссылки

В этом разделе описано, как получить информацию о последних новостях и объявлениях программных продуктов, а также информацию об известных проблемах и вопросах, которые интересуют пользователей.

© 2003-2026 Stimulsoft

- Посетите официальный сайт нашей компании https://stimulsoft.com
  - Узнать о наших продуктах можно по ссылке https://stimulsoft.com/ru/products
  - Скачать последние версии продуктов можно по ссылке https://stimulsoft.com/ru/downloads
  - Ознакомиться с последними новостями компании можно по ссылке https://stimulsoft.com/ru/blog/news

- Кроме этого вы можете загрузить пакеты продуктов Stimulsoft:
  - Reports.Web, Reports.Blazor, Reports.Angular Reports.Net, Reports.Wpf, Reports.Web.NetCore, Dashboards.Blazor, Dashboards.Win, Dashboards.Web, Dashboards.Web.NetCore с NuGet по ссылке https://www.nuget.org/profiles/Stimulsoft
  - Reports.Java с Maven по ссылке http://central.maven.org/maven2/com/stimulsoft
  - Reports.JS и Dashboards.JS с npm по ссылке https://www.npmjs.com/search?q=stimulsoft
  - Reports.PHP и Dashboards.PHP с composer по ссылке https://packagist.org/?query=stimulsoft

- Ознакомиться с инструментами аналитики и отчетности можно по ссылке https://demo.stimulsoft.com

- Для создания, хранения и последующего развертывания отчетов в ваших приложениях отчетов используйте облачный сервис Stimulsoft https://cloud.stimulsoft.com

- Огромное количество видео-уроков доступны на нашем YouTube канале https://www.youtube.com/user/StimulsoftVideos

- Ознакомиться примерами для различных платформ можно:
  - На сайте GitHub по ссылке https://github.com/stimulsoft
  - На странице Примеры нашего сайта по ссылке https://www.stimulsoft.com/ru/samples

- Мы есть также в социальных сетях и мессенджерах:

X:
https://x.com/stimulsoft

Linked in:
https://www.linkedin.com/company/stimulsoft

© 2003-2026 Stimulsoft

Facebook:
https://www.facebook.com/Stimulsoft

RSS:
https://www.stimulsoft.com/ru/rss

GitHub:
https://github.com/stimulsoft

YouTube:
https://www.youtube.com/user/StimulsoftVideos

Telegram:
https://t.me/stimulsoft

Teams:
Teams - stimulsoft

WhatsApp:
WhatsApp - Stimulsoft

Instagram:
https://www.instagram.com/stimulsoft/

Также посетите наш Форум - http://forumru.stimulsoft.com/index.php
Здесь вы сможете прочитать и обсудить различные темы, касающиеся инструментов для создания отчетов. Для получения более подробной информации о продукте в других Интернет-ресурсах, пожалуйста, используйте поисковые системы.

2 Распространение

Информация

Пакет **Stimulsoft Ultimate** содержит набор библиотек, скриптов и других файлов, необходимых для работы с отчетами и панелями индикаторов в WinForms, ASP.NET, ASP.NET MVC, .NET Core, WPF, JavaScript, PHP и Java приложениях. В продукте доступны ядро отчетов и анализа данных, дизайнер и вьювер отчетов и панелей индикаторов для всех поддерживаемых платформ.

В данном разделе рассматриваются различные варианты распространения отчетов, библиотек, файлов поставляемых в комплекте продуктов Stimulsoft.

Наименование	Описание
Reports.WEB	Пакет содержит набор необходимых библиотек для работы с отчетами в ASP.NET, ASP.NET MVC и .NET Core

© 2003-2026 Stimulsoft

	приложениях.
Reports.NET	Пакет содержит набор необходимых библиотек для работы с отчетами в WinForms приложениях.
Reports.JS	Пакет содержит набор необходимых скриптов и стилей для работы с отчетами в JavaScript приложениях.
Reports.WPF	Пакет содержит набор необходимых библиотек для работы с отчетами в WPF приложениях.
Reports.PHP	Пакет содержит набор необходимых скриптов и стилей для работы с отчетами в клиент-серверных приложениях с использованием PHP.
Reports.JAVA	Пакет содержит набор необходимых JAR файлов для работы с отчетами в Java приложениях.
Dashboards.WEB	Пакет содержит набор необходимых библиотек для работы с панелями индикаторов (дашбордами) и отчетами в ASP.NET, ASP.NET MVC и .NET Core приложениях.
Dashboards.WIN	Пакет содержит набор необходимых библиотек для работы с панелями индикаторов (дашбордами) и отчетами в WinForms и WPF приложениях.
Dashboards.JS	Пакет содержит набор необходимых скриптов и стилей для работы с панелями индикаторов (дашбордами) и отчетами в JavaScript приложениях.
Stimulsoft Designer	Приложение содержит набор необходимых файлов, библиотек для запуска standalone дизайнера отчетов.
Stimulsoft Designer.JS	Приложение содержит набор необходимых файлов, скриптов для запуска standalone JS дизайнера отчетов.
Stimulsoft Demo	Приложение содержит набор необходимых файлов, библиотек для запуска демонстрационных отчетов и дашбордов.

© 2003-2026 Stimulsoft

Информация

Поставляется dmg образ, в котором **Stimulsoft Designer for Mac** представлен как одно приложение. После установки, для запуска дизайнера отчетов, следует запустить файл **Stimulsoft Designer.app**.

2.1 Библиотеки пакета Reports.WEB

Пакет **Reports.WEB** поставляется в комплекте со следующими библиотеками:

Сборки	Описание
Stimulsoft.Base.dll	Основная библиотека, содержит общие базовые интерфейсы и классы для всех продуктов.
Stimulsoft.Data.dll	Библиотека включает все классы и методы для анализа данных, их преобразования и фильтрации.
Stimulsoft.Map.dll	Библиотека содержит ресурсы, необходимые для работы региональных карт.
Stimulsoft.Report.dll	Основная библиотека, содержит весь функционал генератора отчетов - построение, экспортирование, работа с данными.
Stimulsoft.Report.Check.dll	Библиотека содержит все необходимые методы для работы Инспектор отчетов (Report Checker).
Stimulsoft.Report.Design.dll	Библиотека содержит ресурсы и классы контрола дизайнера отчетов.
Stimulsoft.Report.Helper.dll	Библиотека содержит ресурсы расширенной локализации продуктов – описания свойств, действий, сообщений об ошибках.
Stimulsoft.Report.Mvc.dll	Библиотека содержит ресурсы и классы контролов вьювера и дизайнера для ASP.NET MVC.
Stimulsoft.Report.Web.dll	Библиотека содержит ресурсы и классы контрола вьювера для ASP.NET.
Stimulsoft.Report.WebDesign.dll	Библиотека содержит ресурсы и классы контрола HTML5 дизайнера для ASP.NET.
Stimulsoft.Repor	Библиотека содержит ресурсы и классы контролов вьювера

© 2003-2026 Stimulsoft

t.Mvc.NetCore.dll	и дизайнера для .NET Core MVC.
Stimulsoft.System.dll	Библиотека содержит системные интерфейсы и классы, необходимые для совместимости генератора отчетов с .NET Standard
Stimulsoft.System.Web.dll	Библиотека содержит системные интерфейсы и классы, необходимые для совместимости Web контролов с .NET Standard
Svg	Необязательная библиотека, содержит функционал для работы с SVG изображениями в отчетах.
LibExcel	Библиотека, которая необходимы для использования файлов **xls** в качестве источников данных.

2.2 Библиотеки пакета Reports.NET

Пакет **Reports.NET** поставляется в комплекте со следующими библиотеками:

Сборки	Описание
Stimulsoft.Base.dll	Основная библиотека, содержит общие базовые интерфейсы и классы для всех продуктов.
Stimulsoft.Controls.dll	Библиотека содержит дополнительные элементы управления, которые используются в дизайнере и вьювере для настольных приложений.
Stimulsoft.Controls.Win.dll	Библиотека содержит основные элементы управления, которые используются в дизайнере и вьювере для настольных приложений.
Stimulsoft.Data.dll	Библиотека включает все классы и методы для анализа данных, их преобразования и фильтрации.
Stimulsoft.Database.dll	Используется для создания и редактирования подключений к различным базам данных. В составе библиотеки находится SQL Query Builder.
Stimulsoft.Design.dll	Библиотека содержит классы для работы с отчетами в Design Time в среде редактирования Visual Studio.
Stimulsoft.Editor.dll	Библиотека содержит текстовый редактор, который используется в дизайнере отчетов.

© 2003-2026 Stimulsoft

Stimulsoft.Map.dll	Библиотека содержит ресурсы, необходимые для работы региональных карт.
Stimulsoft.Report.dll	Основная библиотека, содержит весь функционал генератора отчетов - построение, экспортирование, работа с данными.
Stimulsoft.Report.Check.dll	Библиотека содержит все необходимые методы для работы Инспектор отчетов (Report Checker).
Stimulsoft.Report.Design.dll	Библиотека содержит ресурсы и классы контрола дизайнера для WinForms.
Stimulsoft.Report.Helper.dll	Библиотека содержит ресурсы расширенной локализации продуктов – описания свойств, действий, сообщений об ошибках.
Stimulsoft.Report.Web.dll	Библиотека содержит ресурсы и классы контрола вьювера для ASP.NET.
Stimulsoft.Report.Win.dll	Библиотека содержит ресурсы и классы контрола вьювера для WinForms.
Svg	Необязательная библиотека, содержит функционал для работы с SVG изображениями в отчетах.

2.3 Скрипты пакета Reports.JS

Пакет **Reports.JS** поставляется в комплекте со следующими скриптами:

Скрипты	**Описание**
Общие скрипты *:	
stimulsoft.blockly.editor.js	Содержит визуальный редактор **Blockly** для создания скриптов событий в отчете. Сам обработчик событий встроен в движок отчетов.
stimulsoft.designer.js	Скрипт для работы с дизайнером отчетов.
stimulsoft.reports.js	Основной скрипт, который содержит весь функционал генератора отчетов - построение, экспортирование, работа с данными.

© 2003-2026 Stimulsoft

stimulsoft.reports.maps.js	Скрипт, который содержит ресурсы, необходимые для работы региональных карт.
stimulsoft.viewer.js	Скрипт для работы с вьювером отчетов.
Скрипт stimulsoft.reports.js можно разделить на *:	
stimulsoft.reports.engine.js	Основной необходимый скрипт генератора отчетов.
stimulsoft.reports.chart.js	Скрипт необходимый для работы с диаграммами.
stimulsoft.reports.export.js	Скрипт для экспорта отчетов в PDF или другие форматы, кроме экспорта в HTML.
stimulsoft.reports.import.xlsx.js	Скрипт импорта данные из XLSX файлов.
Другие файлы:	
stimulsoft.reports.d.ts	Заголовочный файл, который описывает синтаксис и структуру функций и свойств.

* - Также, вместо общих скриптов или частей stimulsoft.reports.js, можно использовать упакованные скрипты. Упакованные скрипты позволяют уменьшить размер файлов скриптов в несколько раз. Имена упакованных скриптов совпадают с именами скриптов, но содержат слово **pack** в имени перед расширением **js**. Например, упакованный аналог скрипта **stimulsoft.reports.js** будет **stimulsoft.reports.pack.js**.

2.4 Библиотеки пакета Reports.WPF

Пакет **Reports.WPF** поставляется в комплекте со следующими библиотеками:

Сборки	Описание
Stimulsoft.Base.dll	Исходная сборка генератора отчетов.
Stimulsoft.Client.Designer.dll	Библиотека содержит ресурсы и классы WPF контрола дизайнера отчетов V2.

© 2003-2026 Stimulsoft

Stimulsoft.Data.dll	Библиотека включает все классы и методы для анализа данных, их преобразования и фильтрации.
Stimulsoft.Database.dll	Используется для создания и редактирования подключений к различным базам данных. В составе библиотеки находится SQL Query Builder.
Stimulsoft.Editor.dll	Библиотека содержит текстовый редактор, который используется в дизайнере отчетов.
Stimulsoft.Map.dll	Библиотека содержит ресурсы, необходимые для работы региональных карт.
Stimulsoft.Report.dll	Основная библиотека, содержит весь функционал генератора отчетов - построение, экспортирование, работа с данными.
Stimulsoft.Report.Check.dll	Библиотека содержит все необходимые методы для работы Инспектор отчетов (Report Checker).
Stimulsoft.Report.Helper.dll	Библиотека содержит ресурсы расширенной локализации продуктов – описания свойств, действий, сообщений об ошибках.
Stimulsoft.Report.Wpf.dll	Библиотека содержит ресурсы и классы контрола вьювера для WPF.
Stimulsoft.Report.Wpf.BlackTheme.dll	Эта библиотека содержит тему оформления Black для использования в WPF вьювере и дизайнере отчетов.
Stimulsoft.Report.Wpf.Office2003BlueTheme.dll	Эта библиотека содержит тему оформления Office 2003 Blue для использования в WPF вьювере и дизайнере отчетов.
Stimulsoft.Report.Wpf.Office2003OliveGreenTheme.dll	Эта библиотека содержит тему оформления Office 2003 Olive Green для использования в WPF вьювере и дизайнере отчетов.
Stimulsoft.Report.Wpf.Office2003SilverTheme.dll	Эта библиотека содержит тему оформления Office 2003 Silver для использования в WPF вьювере и дизайнере отчетов.
Stimulsoft.Report.Wpf.Office2007BlackTheme.dll	Эта библиотека содержит тему оформления Office 2007 Black для использования в WPF вьювере и дизайнере отчетов.

© 2003-2026 Stimulsoft

Stimulsoft.Report.Wpf.Office2007BlueTheme.dll	Эта библиотека содержит тему оформления Office 2007 Blue для использования в WPF вьювере и дизайнере отчетов.
Stimulsoft.Report.Wpf.Office2007SilverTheme.dll	Эта библиотека содержит тему оформления Office 2007 Silver для использования в WPF вьювере и дизайнере отчетов.
Stimulsoft.Report.Wpf.Office2010BlueTheme.dll	Эта библиотека содержит тему оформления Office 2010 Blue для использования в WPF вьювере и дизайнере отчетов.
Stimulsoft.Report.Wpf.Office2010WhiteTheme.dll	Эта библиотека содержит тему оформления Office 2010 White для использования в WPF вьювере и дизайнере отчетов.
Stimulsoft.Report.WpfDesign.dll	Библиотека содержит ресурсы и классы WPF контрола дизайнера отчетов V1.

2.5 Скрипты пакета Reports.PHP

Пакет **Reports.PHP** поставляется в комплекте со следующими скриптами:

Скрипты	Описание
stimulsoft.blockly.editor.js	Содержит визуальный редактор **Blockly** для создания скриптов событий в отчете. Сам обработчик событий встроен в движок отчетов.
stimulsoft.designer.js	Скрипт, который содержит ресурсы и классы контрола дизайнера отчетов.
stimulsoft.designer.pack.js	Упакованный скрипт, который содержит ресурсы и классы контрола дизайнера отчетов.
stimulsoft.reports.js	Основной скрипт, который содержит весь функционал генератора отчетов - построение, экспортирование, работа с данными.
stimulsoft.reports.d.ts	Заголовочный файл, который описывает синтаксис и структуру функций и свойств.
stimulsoft.reports.maps.js	Скрипт, который содержит ресурсы, необходимые для работы региональных карт.

© 2003-2026 Stimulsoft

stimulsoft.reports.maps.pack.js	Упакованный скрипт, который содержит ресурсы, необходимые для работы региональных карт.
stimulsoft.reports.pack.js	Основной упакованный скрипт, который содержит весь функционал генератора отчетов - построение, экспортирование, работа с данными.
stimulsoft.viewer.js	Скрипт, который содержит ресурсы и классы контрола вьювера.
stimulsoft.viewer.pack.js	Упакованный скрипт, который содержит ресурсы и классы контрола вьювера.

2.6 JAR файлы пакета Reports.JAVA

Пакет **Reports.JAVA** поставляется в комплекте со следующими файлами:

JAR файлы	Описание
stimulsoft-reports-base	Основной файл, который содержит общие базовые интерфейсы и классы для продукта.
stimulsoft-reports-demo	Файл, который содержит демонстрационное приложение.
stimulsoft-reports-lib	Файл, который содержит библиотеку утилит.
stimulsoft-reports-report	Основной файл, который содержит весь функционал генератора отчетов - построение, экспортирование, работа с данными.
stimulsoft-reports-samples	Файл, который содержит примеры.
stimulsoft-reports-viewer	Файл, который содержит ресурсы и классы контрола Java вьювера.
stimulsoft-reports-web	Файл, который содержит общую часть web дизайнера и вьювера.
stimulsoft-reports-webdesigner	Файл, который содержит ресурсы и классы контрола web дизайнера.
stimulsoft-	Файл, который содержит ресурсы и классы контрола web

© 2003-2026 Stimulsoft

reports-webviewer	вьювера.

2.7 Библиотеки пакета Dashboards.WEB

Пакет **Dashboards.WEB** поставляется в комплекте со следующими библиотеками:

Сборки	Описание
Stimulsoft.Base.dll	Основная библиотека, содержит общие базовые интерфейсы и классы для всех продуктов.
Stimulsoft.Dashboard	Библиотека, которая содержит основные функциональные возможности дашбордов - основные задачи, классы, определения.
Stimulsoft.Dashboard.Drawing	Библиотека, которая используется для отрисовки элементов Stimulsoft Dashboards с помощью GDI технологии.
Stimulsoft.Dashboard.Export	Библиотека содержит все необходимые методы экспорта дашбордов.
Stimulsoft.Data.dll	Библиотека включает все классы и методы для анализа данных, их преобразования и фильтрации.
Stimulsoft.Map.dll	Библиотека содержит ресурсы, необходимые для работы региональных карт.
Stimulsoft.Report.dll	Основная библиотека, содержит весь функционал генератора отчетов - построение, экспортирование, работа с данными.
Stimulsoft.Report.Check.dll	Библиотека содержит все необходимые методы для работы Инспектор отчетов (Report Checker).
Stimulsoft.Report.Helper.dll	Библиотека содержит ресурсы расширенной локализации продуктов – описания свойств, действий, сообщений об ошибках.
Stimulsoft.Report.Mvc.dll	Библиотека содержит ресурсы и классы контролов вьювера и дизайнера для ASP.NET MVC.
Stimulsoft.Report.Web.dll	Библиотека содержит ресурсы и классы контрола вьювера для ASP.NET.

© 2003-2026 Stimulsoft

Stimulsoft.Report.WebDesign.dll	Библиотека содержит ресурсы и классы контрола HTML5 дизайнера для ASP.NET.
Svg	Необязательная библиотека, содержит функционал для работы с SVG изображениями в отчетах.
LibExcel	Библиотека, которая необходимы для использования файлов **xls** в качестве источников данных.

2.8 Библиотеки пакета Dashboards.WIN

Пакет **Dashboards.WIN** поставляется в комплекте со следующими библиотеками:

Сборки	Описание
Stimulsoft.Base.dll	Основная библиотека, содержит общие базовые интерфейсы и классы для всех продуктов.
Stimulsoft.Controls.dll	Библиотека содержит дополнительные элементы управления, которые используются в дизайнере и вьювере для настольных приложений.
Stimulsoft.Controls.Win.dll	Библиотека содержит основные элементы управления, которые используются в дизайнере и вьювере для настольных приложений.
Stimulsoft.Dashboard.dll	Библиотека, которая содержит основные функциональные возможности дашбордов - основные задачи, классы, определения.
Stimulsoft.Dashboard.Design.dll	Библиотека содержит ресурсы и классы контрола дизайнера дашбордов для WinForms.
Stimulsoft.Dashboard.Drawing.dll	Библиотека, которая используется для отрисовки элементов Stimulsoft Dashboards с помощью GDI технологии в winforms.
Stimulsoft.Dashboard.Drawing.Wpf.dll	Библиотека, которая используется для отрисовки элементов Stimulsoft Dashboards с помощью WPF технологии.
Stimulsoft.Dashboard.Export.dll	Библиотека содержит все необходимые методы экспорта дашбордов.
Stimulsoft.Dashboard.Viewer.dll	Библиотека содержит ресурсы и классы контрола вьювера дашбордов для WinForms.

© 2003-2026 Stimulsoft

Stimulsoft.Dashboard.Viewer.Wpf.dll	Библиотека содержит ресурсы и классы контрола вьювера дашбордов для WPF.
Stimulsoft.Data.dll	Библиотека включает все классы и методы для анализа данных, их преобразования и фильтрации.
Stimulsoft.Database.dll	Используется для создания и редактирования подключений к различным базам данных. В составе библиотеки находится SQL Query Builder.
Stimulsoft.Design.dll	Библиотека содержит классы для работы с отчетами в Design Time в среде редактирования Visual Studio.
Stimulsoft.Editor.dll	Библиотека содержит текстовый редактор, который используется в дизайнере отчетов.
Stimulsoft.Map.dll	Библиотека содержит ресурсы, необходимые для работы региональных карт.
Stimulsoft.Report.dll	Основная библиотека, содержит весь функционал генератора отчетов - построение, экспортирование, работа с данными.
Stimulsoft.Report.Check.dll	Библиотека содержит все необходимые методы для работы Инспектор отчетов (Report Checker).
Stimulsoft.Report.Design.dll	Библиотека содержит ресурсы и классы контрола дизайнера для WinForms.
Stimulsoft.Report.Helper.dll	Библиотека содержит ресурсы расширенной локализации продуктов – описания свойств, действий, сообщений об ошибках.
Stimulsoft.Report.Web.dll	Библиотека содержит ресурсы и классы контрола вьювера для ASP.NET.
Stimulsoft.Report.Win.dll	Библиотека содержит ресурсы и классы контрола вьювера для WinForms.
Stimulsoft.Report.Wpf.dll	Библиотека содержит ресурсы и классы контрола вьювера для Wpf.
Svg	Необязательная библиотека, содержит функционал для работы с SVG изображениями в отчетах.

© 2003-2026 Stimulsoft

2.9 Скрипты пакета Dashboards.JS

Пакет **Dashboards.JS** поставляется в комплекте со следующими скриптами:

Скрипты	Описание
Общие скрипты *:	
stimulsoft.blockly.editor.js	Содержит визуальный редактор **Blockly** для создания скриптов событий в отчете. Сам обработчик событий встроен в движок отчетов.
stimulsoft.dashboards.js	Скрипт для работы с панелями индикаторов (дашбордами).
stimulsoft.reports.maps.js	Скрипт, который содержит ресурсы, необходимые для работы региональных карт.
stimulsoft.designer.js	Скрипт для работы с дизайнером отчетов.
stimulsoft.viewer.js	Скрипт для работы с вьювером отчетов.
stimulsoft.reports.js	Основной скрипт, который содержит весь функционал генератора отчетов - построение, экспортирование, работа с данными.
Части скриптов *:	
stimulsoft.reports.engine.js	Основной необходимый скрипт генератора отчетов.
stimulsoft.reports.chart.js	Скрипт необходимый для работы с диаграммами.
stimulsoft.reports.export.js	Скрипт для экспорта отчетов в PDF или другие форматы, кроме экспорта в HTML.
stimulsoft.reports.import.xlsx.js	Скрипт импорта данные из XLSX файлов.
stimulsoft.reports.maps.js	Скрипт необходимый для работы с картами.

© 2003-2026 Stimulsoft

Другие файлы:	
stimulsoft.reports.d.ts	Заголовочный файл, который описывает синтаксис и структуру функций и свойств.

* - Также, вместо общих скриптов или частей stimulsoft.reports.js, можно использовать упакованные скрипты. Упакованные скрипты позволяют уменьшить размер файлов скриптов в несколько раз. Имена упакованных скриптов совпадают с именами скриптов, но содержат слово **pack** в имени перед расширением **js**. Например, упакованный аналог скрипта **stimulsoft.reports.js** будет **stimulsoft.reports.pack.js**.

2.10 Скрипты пакета Dashboards.PHP

Пакет **Dashboards.PHP** поставляется в комплекте со следующими скриптами:

Скрипты	Описание
Общие скрипты *:	
stimulsoft.blockly.editor.js	Содержит визуальный редактор **Blockly** для создания скриптов событий в отчете. Сам обработчик событий встроен в движок отчетов.
stimulsoft.dashboards.js	Скрипт для работы с панелями индикаторов (дашбордами).
stimulsoft.reports.maps.js	Скрипт, который содержит ресурсы, необходимые для работы региональных карт.
stimulsoft.designer.js	Скрипт для работы с дизайнером отчетов.
stimulsoft.viewer.js	Скрипт для работы с вьювером отчетов.
stimulsoft.reports.js	Основной скрипт, который содержит весь функционал генератора отчетов - построение, экспортирование, работа с данными.
Части скриптов *:	
stimulsoft.repor	Основной необходимый скрипт генератора отчетов.

© 2003-2026 Stimulsoft

ts.engine.js	
stimulsoft.reports.chart.js	Скрипт необходимый для работы с диаграммами.
stimulsoft.reports.export.js	Скрипт для экспорта отчетов в PDF или другие форматы, кроме экспорта в HTML.
stimulsoft.reports.import.xlsx.js	Скрипт импорта данные из XLSX файлов.
stimulsoft.reports.maps.js	Скрипт необходимый для работы с картами.
Другие файлы:	
stimulsoft.reports.d.ts	Заголовочный файл, который описывает синтаксис и структуру функций и свойств.

* - Также, вместо общих скриптов или частей stimulsoft.reports.js, можно использовать упакованные скрипты. Упакованные скрипты позволяют уменьшить размер файлов скриптов в несколько раз. Имена упакованных скриптов совпадают с именами скриптов, но содержат слово **pack** в имени перед расширением **js**. Например, упакованный аналог скрипта **stimulsoft.reports.js** будет **stimulsoft.reports.pack.js**.

2.11 Файлы приложения Stimulsoft Designer

Приложение **Stimulsoft Designer**, после установки включает следующие файлы:

Наименование	Описание
Папки:	
Localization	Содержит файлы локализаций интерфейса вьювера и дизайнера.
Themes	Содержит файлы предустановленных стилей оформления в отчетах.
.exe файлы	

© 2003-2026 Stimulsoft

Designer.exe	Файл запуска winforms дизайнера отчетов.
Designer.Wpf.exe	Файл запуска WPF дизайнера отчетов V1.
DesignerV2.Wpf.exe	Файл запуска WPF дизайнера отчетов V2.
Библиотеки Stimulsoft:	
Stimulsoft.Accounts.Wpf.dll	Библиотека, которая используется для авторизации пользователя в дизайнере отчетов.
Stimulsoft.Base.dll	Исходная сборка генератора отчетов.
Stimulsoft.Client.dll	Библиотека, которая содержит пользовательскую конфигурация дизайнера и вьювера отчетов.
Stimulsoft.Client.Designer.dll	Библиотека содержит ресурсы и классы контрола WPF дизайнера отчетов V2.
Stimulsoft.Controls.dll	Библиотека содержит дополнительные элементы управления, которые используются в дизайнере и вьювере для настольных приложений.
Stimulsoft.Controls.Win.dll	Библиотека содержит основные элементы управления, которые используются в дизайнере и вьювере для настольных приложений.
Stimulsoft.Dashboard	Библиотека, которая содержит основные функциональные возможности дашбордов - основные задачи, классы, определения.
Stimulsoft.Dashboard.Design	Библиотека содержит ресурсы и классы контрола дизайнера дашбордов для WinForms.
Stimulsoft.Dashboard.Drawing.dll	Библиотека, которая используется для отрисовки элементов Stimulsoft Dashboards с помощью GDI технологии в winforms.
Stimulsoft.Dashboard.Drawing.Wpf.dll	Библиотека, которая используется для отрисовки элементов Stimulsoft Dashboards с помощью WPF технологии.
Stimulsoft.Dashboard.Export.dll	Библиотека содержит все необходимые классы и методы экспорта дашбордов.

© 2003-2026 Stimulsoft

Stimulsoft.Dashboard.Viewer.dll	Библиотека содержит ресурсы и классы контрола вьювера дашбордов для WinForms.
Stimulsoft.Dashboard.Viewer.Wpf.dll	Библиотека содержит ресурсы и классы контрола вьювера дашбордов для WPF.
Stimulsoft.Dashboard.Viewer.Wpf.Design.dll	Библиотека необходимая для интеграции вьювера дашбордов WPF на инструментарий Visual Studio.
Stimulsoft.Data.dll	Библиотека включает все классы и методы для анализа данных, их преобразования и фильтрации.
Stimulsoft.Database.dll	Используется для создания и редактирования подключений к различным базам данных. В составе библиотеки находится SQL Query Builder.
Stimulsoft.Database.Wpf.dll	Используется для создания и редактирования подключений к различным базам данных для WPF генератора отчетов. В составе библиотеки находится SQL Query Builder.
Stimulsoft.Design.dll	Библиотека содержит классы для работы с отчетами в Design Time в среде редактирования Visual Studio.
Stimulsoft.Editor.dll	Библиотека содержит текстовый редактор, который используется в winforms дизайнере отчетов.
Stimulsoft.Editor.Wpf.dll	Библиотека содержит текстовый редактор, который используется в wpf дизайнере отчетов.
Stimulsoft.Map.dll	Библиотека содержит ресурсы, необходимые для работы региональных карт.
Stimulsoft.Report.dll	Основная библиотека, содержит весь функционал генератора отчетов - построение, экспортирование, работа с данными.
Stimulsoft.Report.Check.dll	Библиотека содержит все необходимые методы для работы Инспектор отчетов (Report Checker).
Stimulsoft.Report.Design.dll	Библиотека содержит ресурсы и классы контрола дизайнера для WinForms.
Stimulsoft.Report.Helper.dll	Библиотека содержит ресурсы расширенной локализации продуктов – описания свойств, действий, сообщений об ошибках.

© 2003-2026 Stimulsoft

Stimulsoft.Report.Import.dll	Библиотека содержит ресурсы, необходимые для импорта отчетов.
Stimulsoft.Report.Mvc.dll	Библиотека содержит ресурсы и классы контролов вьювера и дизайнера для ASP.NET MVC.
Stimulsoft.Report.Publish.dll	Библиотека содержит ресурсы, необходимые для публикации отчетов.
Stimulsoft.Report.Web.dll	Библиотека содержит ресурсы и классы контрола вьювера для ASP.NET.
Stimulsoft.Report.WebDesign.dll	Библиотека содержит ресурсы и классы контрола HTML5 дизайнера для ASP.NET.
Stimulsoft.Report.Win.dll	Библиотека содержит ресурсы и классы контрола вьювера для WinForms.
Stimulsoft.Report.Wpf.dll	Библиотека содержит ресурсы и классы контрола вьювера для WPF.
Stimulsoft.Report.Wpf.BlackTheme.dll	Эта сборка содержит Black theme для использования во вьювере и дизайнере отчетов WPF.
Stimulsoft.Report.Wpf.Design.dll	Библиотека необходимая для интеграции вьювера WPF на инструментарий Visual Studio.
Stimulsoft.Report.Wpf.Office2003BlueTheme.dll	Эта библиотека содержит тему оформления Office 2003 Blue для использования в WPF вьювере и дизайнере отчетов.
Stimulsoft.Report.Wpf.Office2003OliveGreenTheme.dll	Эта библиотека содержит тему оформления Office 2003 Olive Green для использования в WPF вьювере и дизайнере отчетов.
Stimulsoft.Report.Wpf.Office2003SilverTheme.dll	Эта библиотека содержит тему оформления Office 2003 Silver для использования в WPF вьювере и дизайнере отчетов.
Stimulsoft.Report.Wpf.Office2007BlackTheme.dll	Эта библиотека содержит тему оформления Office 2007 Black для использования в WPF вьювере и дизайнере отчетов.
Stimulsoft.Report.Wpf.Office200	Эта библиотека содержит тему оформления Office 2007 Blue для использования в WPF вьювере и дизайнере отчетов.

© 2003-2026 Stimulsoft

7BlueTheme.dll	
Stimulsoft.Report.Wpf.Office2007SilverTheme.dll	Эта библиотека содержит тему оформления Office 2007 Silver для использования в WPF вьювере и дизайнере отчетов.
Stimulsoft.Report.Wpf.Office2010BlueTheme.dll	Эта библиотека содержит тему оформления Office 2010 Blue для использования в WPF вьювере и дизайнере отчетов.
Stimulsoft.Report.Wpf.Office2010WhiteTheme.dll	Эта библиотека содержит тему оформления Office 2010 White для использования в WPF вьювере и дизайнере отчетов.
Stimulsoft.Report.WpfDesign.dll	Библиотека содержит ресурсы и классы WPF контрола дизайнера отчетов V1.
Stimulsoft.Report.WpfDesign.Design.dll	Библиотека необходимая для интеграции дизайнера WPF на инструментарий Visual Studio.
Stimulsoft.Server.Connect.dll	Библиотека содержит классы и методы соединения с сервером.
Stimulsoft.Server.Objects.dll	Библиотека содержит описание объектов, которые можно передавать на сервер.
Stimulsoft.Wizard.Wpf.dll	Библиотека содержит ресурсы, необходимые для разработки отчетов при помощи мастеров создания отчетов.
Сторонние библиотеки:	
adodb.dll	Библиотека содержит необходимые классы для доступа к различным типам баз данных.
LibExcel.dll	Библиотека, которая необходимы для использования файлов **xls** в качестве источников данных.
Microsoft.Web.Infrastructure.dll	Библиотека, которая предоставляет возможность динамически регистрировать модули HTTP во время выполнения.
Microsoft.Web.XmlTransform.dll	Библиотека, которая предоставляет возможность преобразовать XML файлы.
Newtonsoft.Json	Библиотека, которая содержит классы преобразования

© 2003-2026 Stimulsoft

.dll	объектов в JSON объекты.
Библиотеки NuGet	Все библиотеки NuGet, которые необходимы для скачивания пакетов Stimulsoft с NuGet
Svg.dll	Необязательная библиотека, содержит функционал для работы с SVG изображениями в отчетах.
System.Web	Все необходимые библиотеки для публикации Web пакетов из дизайнера отчетов.

2.12 Файлы приложения Stimulsoft Designer.JS

Приложение **Stimulsoft Designer.JS**, после установки включает следующие файлы:

Наименование	Описание
Папка **locales**	Содержит файлы локализаций интерфейса дизайнера и вьювера отчетов.
Папка **scripts:**	Содержит скрипты Stimulsoft и стороннюю библиотеку JavaScript.
stimulsoft.dashboards.js	Скрипт для работы с панелями индикаторов (дашбордами).
stimulsoft.designer.js	Скрипт для работы с дизайнером отчетов.
stimulsoft.designer.runtime.js	Дополнительный скрипт для standalone дизайнера отчетов.
stimulsoft.reports.js	Основной скрипт, который содержит весь функционал генератора отчетов - построение, экспортирование, работа с данными.
stimulsoft.reports.maps.js	Скрипт, который содержит ресурсы, необходимые для работы региональных карт.
stimulsoft.viewer.js	Скрипт для работы с вьювером отчетов.
Папка **styles**	Содержит файлы тем оформления интерфейса дизайнера и вьювера отчетов.

© 2003-2026 Stimulsoft

Файл **Designer.exe**	Файл запуска standalone дизайнера отчетов.

Информация

Также, директория **Stimulsoft Designer.JS** содержит сторонние папки, скрипты и файлы необходимые для работы вспомогательного фрэймворка.

2.13 Файлы приложения Stimulsoft Demo

Приложение **Stimulsoft Demo**, после установки включает следующие файлы:

Наименование	Описание
Папки:	
Localization	Содержит файлы локализаций интерфейса вьювера и дизайнера.
Reports	Содержит примеры отчетов и дашбордов.
.exe файлы	
Designer.exe	Файл запуска winforms дизайнера отчетов.
Demo.exe	Файл запуска демонстрационного приложения.
Библиотеки Stimulsoft:	
Stimulsoft.Accounts.Wpf.dll	Библиотека, которая используется для авторизации пользователя в дизайнере отчетов.
Stimulsoft.Base.dll	Исходная сборка генератора отчетов.
Stimulsoft.Client.dll	Библиотека, которая содержит пользовательскую конфигурация дизайнера и вьювера отчетов.
Stimulsoft.Controls.dll	Библиотека содержит дополнительные элементы управления, которые используются в дизайнере и вьювере для настольных приложений.

© 2003-2026 Stimulsoft

Stimulsoft.Controls.Win.dll	Библиотека содержит основные элементы управления, которые используются в дизайнере и вьювере для настольных приложений.
Stimulsoft.Dashboard	Библиотека, которая содержит основные функциональные возможности дашбордов - основные задачи, классы, определения.
Stimulsoft.Dashboard.Design	Библиотека содержит ресурсы и классы контрола дизайнера дашбордов для WinForms.
Stimulsoft.Dashboard.Drawing.dll	Библиотека, которая используется для отрисовки элементов Stimulsoft Dashboards с помощью GDI технологии в winforms.
Stimulsoft.Dashboard.Drawing.Wpf.dll	Библиотека, которая используется для отрисовки элементов Stimulsoft Dashboards с помощью WPF технологии.
Stimulsoft.Dashboard.Export.dll	Библиотека содержит все необходимые классы и методы экспорта дашбордов.
Stimulsoft.Dashboard.Viewer.dll	Библиотека содержит ресурсы и классы контрола вьювера дашбордов для WinForms.
Stimulsoft.Dashboard.Viewer.Wpf.dll	Библиотека содержит ресурсы и классы контрола вьювера дашбордов для WPF.
Stimulsoft.Data.dll	Библиотека включает все классы и методы для анализа данных, их преобразования и фильтрации.
Stimulsoft.Database.dll	Используется для создания и редактирования подключений к различным базам данных. В составе библиотеки находится SQL Query Builder.
Stimulsoft.Database.Wpf.dll	Используется для создания и редактирования подключений к различным базам данных для WPF генератора отчетов. В составе библиотеки находится SQL Query Builder.
Stimulsoft.Design.dll	Библиотека содержит классы для работы с отчетами в Design Time в среде редактирования Visual Studio.
Stimulsoft.Editor.dll	Библиотека содержит текстовый редактор, который используется в winforms дизайнере отчетов.
Stimulsoft.Editor	Библиотека содержит текстовый редактор, который

© 2003-2026 Stimulsoft

.Wpf.dll	используется в wpf дизайнере отчетов.
Stimulsoft.Map.dll	Библиотека содержит ресурсы, необходимые для работы региональных карт.
Stimulsoft.Report.dll	Основная библиотека, содержит весь функционал генератора отчетов - построение, экспортирование, работа с данными.
Stimulsoft.Report.Check.dll	Библиотека содержит все необходимые методы для работы Инспектор отчетов (Report Checker).
Stimulsoft.Report.Design.dll	Библиотека содержит ресурсы и классы контрола дизайнера для WinForms.
Stimulsoft.Report.Design.WebViewer.dll	Библиотека содержит ресурсы и классы контрола web вьювера.
Stimulsoft.Report.Helper.dll	Библиотека содержит ресурсы расширенной локализации продуктов – описания свойств, действий, сообщений об ошибках.
Stimulsoft.Report.Import.dll	Библиотека содержит ресурсы, необходимые для импорта отчетов.
Stimulsoft.Report.Publish.dll	Библиотека содержит ресурсы, необходимые для публикации отчетов.
Stimulsoft.Report.Win.dll	Библиотека содержит ресурсы и классы контрола вьювера для WinForms.
Stimulsoft.Report.Wpf.dll	Библиотека содержит ресурсы и классы контрола вьювера для WPF.
Stimulsoft.Server.Connect.dll	Библиотека содержит классы и методы соединения с сервером.
Stimulsoft.Server.Objects.dll	Библиотека содержит описание объектов, которые можно передавать на сервер.
Stimulsoft.Wizard.Wpf.dll	Библиотека содержит ресурсы, необходимые для разработки отчетов при помощи мастеров создания отчетов.
Сторонние библиотеки:	
adodb.dll	Библиотека содержит необходимые классы для доступа к

© 2003-2026 Stimulsoft

	различным типам баз данных.
LibExcel.dll	Библиотека, которая необходимы для использования файлов **xls** в качестве источников данных.
Newtonsoft.Json.dll	Библиотека, которая содержит классы преобразования объектов в JSON объекты.
Библиотеки NuGet	Все библиотеки NuGet, которые необходимы для скачивания пакетов Stimulsoft с NuGet
Svg.dll	Необязательная библиотека, содержит функционал для работы с SVG изображениями в отчетах.

3 Reports and Dashboards for WebForms

ASP.NET представляет собою технологию создания веб-приложений и веб-сервисов. Компания Stimulsoft предоставляет инструменты создания и отображения отчетов и дашбордов на различных устройствах с использованием этой технологии.

Инструменты создания и редактирования отчетов:	Инструменты отображения и конвертации отчетов:	Инструменты создания и редактирования дашбордов:	Инструменты отображения и конвертации дашбордов:
HTML5 дизайнер	HTML5 вьювер	HTML5 дизайнер	HTML5 вьювер

3.1 HTML5 вьювер

YouTube

Посмотрите видеоуроки по работе с компонентом ASP.NET HTML5 Viewer. Подписывайтесь на канал Stimulsoft и узнайте первыми о новых видеоуроках. Вопросы и предложения оставляйте в комментариях к видео.

© 2003-2026 Stimulsoft

Примеры

Ознакомьтесь с примерами по работе с компонентом ASP.NET HTML5 Viewer GitHub. Все примеры представляют собой отдельные проекты, сгруппированные в один solution для Visual Studio.

Компонент **HTML5 Viewer (StiWebViewer)** предназначен для просмотра отчетов в окне браузера. При этом не требуется устанавливать у клиента .NET Framework, ActiveX компоненты или какие-либо специальные плагины. Все, что необходимо – это любой современный Web-браузер.

При помощи **HTML5 Viewer** можно просматривать, печатать, выполнять экспортирование отчетов на любом компьютере с любой установленной операционной системой. Так как вьювер использует только HTML и JavaScript технологии, он может быть запущен на устройствах, где нет поддержки Flash или Silverlight – планшеты, смартфоны. Также вьювер поддерживает Mobile и Touch интерфейсы, которые автоматически включаются при использовании мобильных устройств и мониторов с сенсорным экраном.

Компонент **HTML5 Viewer** использует технологию AJAX для выполнения всех действий (загрузки отчета, листания страниц, масштабирования, интерактивности в отчетах и др.), что позволяет избавиться от перезагрузки всей страницы, а также экономить Web-трафик и повысить скорость работы.

HTML5 Viewer поддерживает множество тем оформления, анимированный интерфейс, закладки, интерактивные отчеты, редактирование элементов отчета на странице, полноэкранный режим, поиск, и другие, необходимые для просмотра отчетов, возможности.

© 2003-2026 Stimulsoft

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для использования **HTML5 Viewer** в Web-проекте, необходимо установить NuGet пакет Stimulsoft.Reports.Web:

- Выберите пункт "Manage NuGet Packages..." в контекстном меню проекта;
- На вкладке Browse, в строке поиска, укажите Stimulsoft.Reports.Web;
- Выделите элемент, определите версию пакета и нажмите кнопку **Установить (Install)**. При обновлении пакета, следует нажать кнопку **Обновить (Update)**.

Если по какой-либо причине это невозможно, необходимо добавить в проект указанные ниже сборки:

- Stimulsoft.Base.dll
- Stimulsoft.Report.dll
- Stimulsoft.Report.Check.dll
- Stimulsoft.Report.Helper.dll
- Stimulsoft.Report.Web.dll

Для того, чтобы добавить возможности просмотра и экспорта дашбордов в Web-проекте, установить NuGet пакет Stimulsoft.Dashboards.Web (данный пакет связан с пакетом Stimulsoft.Reports.Web, при его отсутствии он будет установлен автоматически):

- Выберите пункт "Manage NuGet Packages..." в контекстном меню проекта;
- На вкладке Browse, в строке поиска, укажите Stimulsoft.Dashboards.Web;
- Выделите элемент, определите версию пакета и нажмите кнопку **Установить (Install)**. При обновлении пакета, следует нажать кнопку **Обновить (Update)**.

Если по какой-либо причине это невозможно, необходимо дополнительно добавить в проект указанные ниже сборки:

- Stimulsoft.Dashboard.dll
- Stimulsoft.Dashboard.Drawing.dll
- Stimulsoft.Dashboard.Export.dll

© 2003-2026 Stimulsoft

- Как это работает
- Динамическое сворачивание, сортировка и детализация
- Активация
- Время ожидания
- Отображение отчета и дашборда
- Редактирование построенного отчета
- Подключение данных
- Отправка отчета по Email
- Локализация вьювера
- Вызов дизайнера из вьювера
- Печать отчета
- Кэширование
- Экспорт отчета и дашборда
- Экспорт и печать из кода
- Режимы отображения
- Основные возможности
- Работам с параметрами
- События вьювера
- Работа с закладками
- Настройки вьювера

3.1.1 Как это работает

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для запуска вьювера требуется разместить на ASPX странице компонент **StiWebViewer**, задать ему необходимые свойства и, если требуется, установить необходимые обработчики событий. При запуске вьювера отчетов происходят следующие действия:

- .NET компонент генерирует HTML и JavaScript код, необходимый для отображения и работы вьювера;
- По окончании вывода компонента запускается JavaScript метод, который запрашивает на стороне сервера первую страницу отчета или весь отчет (в

© 2003-2026 Stimulsoft

зависимости от выбранного режима) и необходимые параметры отчета;

› Каждое действие во вьювере (например, листание страниц, печать либо экспорт отчета и др.) вызывает определенное действие на стороне сервера, в котором можно выполнить необходимые манипуляции с отчетом подписавшись на соответствующее событие вьювера;

› Для ускорения работы вьювер сохраняет отчет в кэше или сессии сервера, что позволяет исключить повторное построение отчета.

3.1.2 Активация

После приобретения продукта Stimulsoft вам необходимо активировать лицензию для используемых компонентов. Это можно выполнить указав лицензионный ключ или загрузив файл с лицензионным ключом. Ниже представлен пример активации компонента **StiWebViewer**.

Default.aspx.cs

```
...
public partial class _Default : Page
{
   static _Default()
   {
      //Activation with using license code
      Stimulsoft.Base.StiLicense.Key = "Your activation code...";

      //Activation with using license file
      var path = HttpContext.Current.Server.MapPath("license.key");
      Stimulsoft.Base.StiLicense.LoadFromFile(path);
   }
}
...
```

Получить лицензионный ключ или скачать файл с лицензионным ключом, вы можете в личном кабинете пользователя. Для авторизации в кабинете используйте имя пользователя и пароль, указанные при покупке подписки продукта.

© 2003-2026 Stimulsoft

3.1.3 Отображение отчета и дашборда

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Примечание

При присвоении отчета вьювер компоненту, происходит автоматическое построение отчета. Вызов метода `Report.Render()` необходим только в том случае, если требуется провести какие-либо действия с построенным отчетом перед его отображением во вьювере. Аналогично, при использовании режима компиляции, вызов метода `Report.Compile()` требуется только в том случае, если требуется провести какие-либо действия с откомпилированным отчетом перед его построением и отображением во вьювере.

Для отображения отчета необходимо добавить компонент **StiWebViewer** на ASPX страницу и присвоить ему загруженный отчет.

Default.aspx

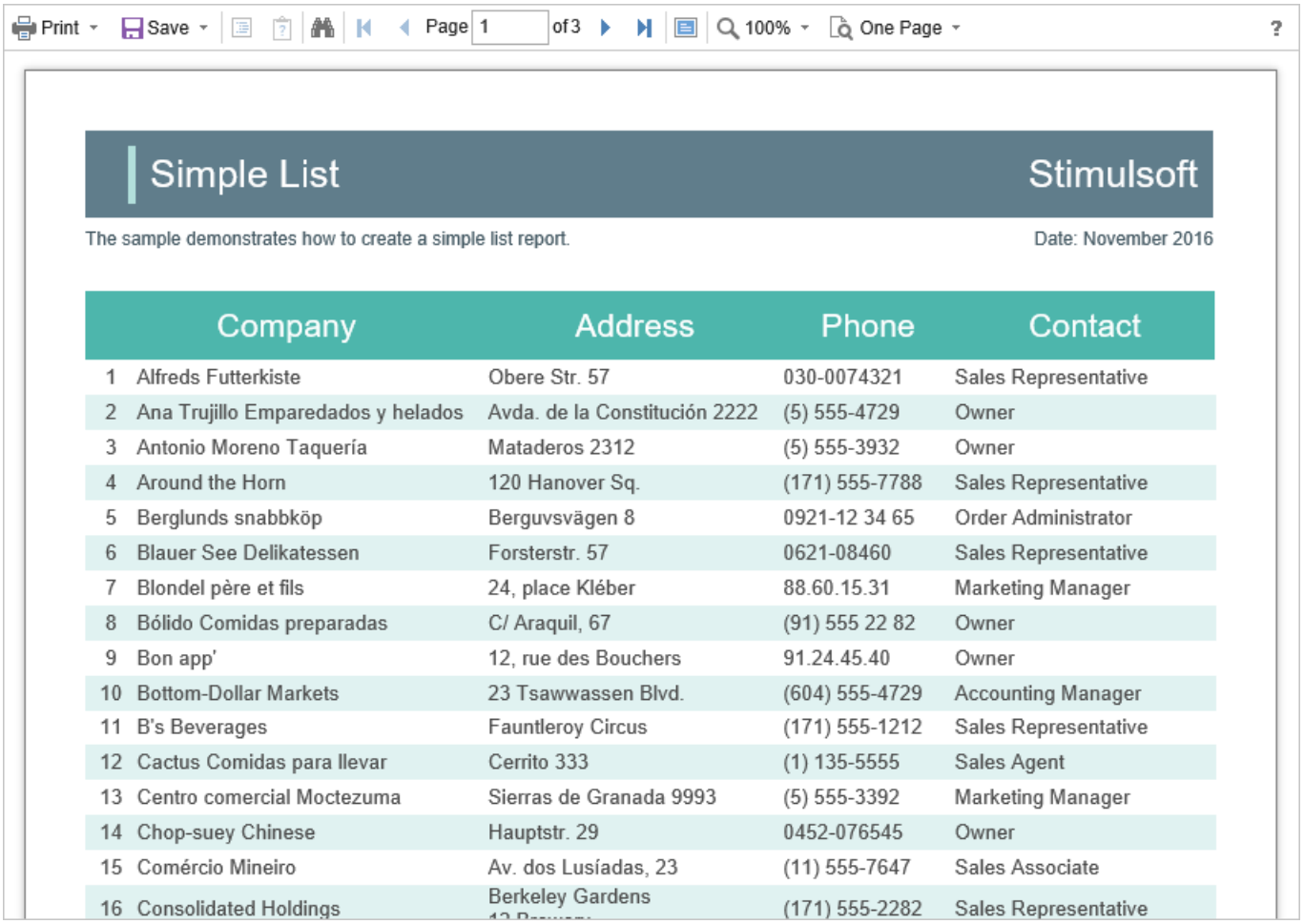
```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server">
</cc1:StiWebViewer>
...
```

© 2003-2026 Stimulsoft

Default.aspx.cs

```
...
protected void Page_Load(object sender, EventArgs e)
{
   StiReport report = new StiReport();
   report.Load(Server.MapPath("Reports/SimpleList.mrt"));
   // report.Load(Server.MapPath("Reports/Dashboard.mrt"));

   StiWebViewer1.Report = report;
}
...
```



	Company	Address	Phone	Contact
1	Alfreds Futterkiste	Obere Str. 57	030-0074321	Sales Representative
2	Ana Trujillo Emparedados y helados	Avda. de la Constitución 2222	(5) 555-4729	Owner
3	Antonio Moreno Taquería	Mataderos 2312	(5) 555-3932	Owner
4	Around the Horn	120 Hanover Sq.	(171) 555-7788	Sales Representative
5	Berglunds snabbköp	Berguvsvägen 8	0921-12 34 65	Order Administrator
6	Blauer See Delikatessen	Forsterstr. 57	0621-08460	Sales Representative
7	Blondel père et fils	24, place Kléber	88.60.15.31	Marketing Manager
8	Bólido Comidas preparadas	C/ Araquil, 67	(91) 555 22 82	Owner
9	Bon app'	12, rue des Bouchers	91.24.45.40	Owner
10	Bottom-Dollar Markets	23 Tsawwassen Blvd.	(604) 555-4729	Accounting Manager
11	B's Beverages	Fauntleroy Circus	(171) 555-1212	Sales Representative
12	Cactus Comidas para llevar	Cerrito 333	(1) 135-5555	Sales Agent
13	Centro comercial Moctezuma	Sierras de Granada 9993	(5) 555-3392	Marketing Manager
14	Chop-suey Chinese	Hauptstr. 29	0452-076545	Owner
15	Comércio Mineiro	Av. dos Lusíadas, 23	(11) 555-7647	Sales Associate
16	Consolidated Holdings	Berkeley Gardens	(171) 555-2282	Sales Representative

Так же **HTML5 Viewer** имеет специальное событие **OnGetReport**, которое можно использовать для присвоения отчета вьюверу. В этом случае загрузку отчета необходимо осуществлять в обработчике события.

© 2003-2026 Stimulsoft

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
  OnGetReport="StiWebViewer1_GetReport">
</cc1:StiWebViewer>
...
```

Default.aspx.cs

```
...
protected void StiWebViewer1_GetReport(object sender,
StiReportDataEventArgs e)
{
  StiReport report = new StiReport();
  report.Load(Server.MapPath("Reports/SimpleList.mrt"));
  // report.Load(Server.MapPath("Reports/Dashboard.mrt"));

  e.Report = report;
}
...
```

Информация

Для присвоения отчета предпочтительнее использовать указанное событие **OnGetReport**, так как в этом случае при потере по какой-либо причине объекта отчета в кэше или сессии сервера, клиентская часть вьювера инициирует это событие и просмотр отчета будет продолжен.

Если отчет не были построены перед отображением, то компонент **HTML5 Viewer** построит их автоматически. Таким образом, для отображения отчета разрешено использовать различные типы отчетов: шаблоны отчетов, построенные отчеты, отчеты в виде классов.

Default.aspx.cs

```
...
protected void StiWebViewer1_GetReport(object sender,
StiReportDataEventArgs e)
{
  StiReport report = new StiReport();
  report.LoadDocument(Server.MapPath("Reports/SimpleList.mdc"));

  e.Report = report;
}
...
```

© 2003-2026 Stimulsoft

Default.aspx.cs

```
...
protected void StiWebViewer1_GetReport(object sender,
StiReportDataEventArgs e)
{
   e.Report = new StiReportCompiledClass();
}
...
```

Так как дашборд не является статическим документов и для его работы требуются данные, то формат построенного документа MDC для него не доступен. Вместо этого предоставлена возможность использовать снимок отчета в формате MRT, который содержит в себе все необходимые для работы дашборда данные, и может быть корректно отображен во вьювере.

Default.aspx.cs

```
...
protected void StiWebViewer1_GetReport(object sender,
StiReportDataEventArgs e)
{
   StiReport report = new StiReport();
   report.Load(Server.MapPath("Reports/Snapshot.mrt"));

   e.Report = report;
}
...
```

Подключение пользовательских шрифтов

Вы можете подключить пользовательские шрифты с использованием класса **StiFontCollection**, указав файл содержащий шрифт. Для этого необходимо в конструкторе вызвать статический метод для загрузки шрифта.

Default.aspx.cs

```
...
public partial class _Default : Page
{
   static _Default()
   {
      Stimulsoft.Base.StiFontCollection.AddFontFile(Server.MapPath("fonts/
      my-font/font-name.ttf"));
   }
}
...
```

© 2003-2026 Stimulsoft

3.1.4 Подключение данных

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Данные для построения отчета могут быть подключены различными способами. Самый простой – хранить настройки подключения в самом шаблоне отчета. Так же данные можно подключить из кода, это можно сделать перед присвоением отчета вьюверу.

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server">
</cc1:StiWebViewer>
...
```

Default.aspx.cs

```
...
protected void Page_Load(object sender, EventArgs e)
{
   DataSet ds = new DataSet();
   ds.ReadXml(Server.MapPath("Reports/Demo.xml"));

   StiReport report = new StiReport();
   report.Load(Server.MapPath("Reports/TwoSimpleLists.mrt"));
   report.Dictionary.Databases.Clear();
   report.RegData("Demo", ds);

   StiWebViewer1.Report = report;
}
...
```

Для подключения данных отчета можно воспользоваться специальным событием **OnGetReportData**, которое будет вызвано перед построением отчета.

© 2003-2026 Stimulsoft

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
   OnGetReportData="StiWebViewer1_GetReportData">
</cc1:StiWebViewer>
...
```

Default.aspx.cs

```
...
protected void StiWebViewer1_GetReportData(object sender,
StiReportDataEventArgs e)
{
   DataSet dataSet = new DataSet();
   dataSet.ReadXml(Server.MapPath("Reports/Demo.xml"));
   e.Report.RegData(dataSet);
}
...
```

SQL источники данных

Параметры подключения к SQL источнику данных, как и к любому другому, можно хранить в самом шаблоне отчета. Если требуется задать параметры подключения из кода перед построением отчета (например, по причине безопасности или в зависимости от авторизованного пользователя), можно воспользоваться приведенным ниже примером.

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
   OnGetReportData="StiWebViewer1_GetReportData">
</cc1:StiWebViewer>
...
```

© 2003-2026 Stimulsoft

Default.aspx.cs

```
...
protected void StiWebViewer1_GetReportData(object sender,
StiReportDataEventArgs e)
{
   OracleConnection connection = new OracleConnection("Data
   Source=Oracle8i;Integrated Security=yes");
   connection.Open();
   OracleDataAdapter adapter = new OracleDataAdapter();
   adapter.SelectCommand = new OracleCommand("SELECT * FROM Products",
   connection);

   DataSet dataSet = new DataSet("productsDataSet");
   adapter.Fill(dataSet, "Products");

   e.Report.RegData("Products", dataSet);
}
...
```

Также, для SQL источников данных используемых в отчете, можно указать **Время ожидания запроса (Query Timeout)** в секундах. Значение данного свойства сохраняется в самом шаблоне отчёта для каждого SQL подключения отдельно.

Ниже приведен пример кода, который предоставляет возможность изменить строку соединения для MS SQL, скорректировать запрос, установить время ожидания запроса для уже созданного соединения и источников данных в отчете.

Default.aspx.cs

```
...
protected void Page_Load(object sender, EventArgs e)
{

   StiReport report = new StiReport();
   report.Load(Server.MapPath("Report.mrt"));
   ((StiSqlDatabase)
   report.Dictionary.Databases["Connection"]).ConnectionString = @"Data
   Source=server;Integrated Security=True;Initial Catalog=DataBase";
   ((StiSqlSource)
   report.Dictionary.DataSources["DataSourceName"]).SqlCommand = "select *
   from Table where Column = 100";
   ((StiSqlSource)
   report.Dictionary.DataSources["DataSourceName"]).CommandTimeout = 1000;

   StiWebViewer1.Report = report;
}
...
```

© 2003-2026 Stimulsoft

Информация

Для SQL источников данных других типов подключение создается аналогично, при этом подключается адаптер, соответствующий типу источника данных. Например, для MS SQL источника необходимо подключать SqlDataAdapter, для OLE DB необходим OleDbDataAdapter. Также, при этом следует указать строку соединения, соответствующую типу подключения.

Вы также можете использовать данные для разработки отчетов и дашбордов полученные из OData хранилищ данных. В этом случае, авторизация может быть выполнена с использованием имени, пароля пользователя или с использованием токена. Параметры авторизации указываются в строке соединения к OData хранилищу с использованием разделителя ";".

Default.aspx.cs

```
...
protected void Page_Load(object sender, EventArgs e)
{

   var report = new StiReport();

   //Authorization using a user account
   var oDataDatabase = new StiODataDatabase("OData", "OData", @"https://
   services.odata.org/V4/Northwind/
   Northwind.svc;AddressBearer=adress;UserName=UserName;Password=Password;C
   lient_Id=Your Client ID", false, null);

   //Authorization using a user token
   var oDataDatabase = new StiODataDatabase("OData", "OData", @"https://
   services.odata.org/V4/Northwind/Northwind.svc;Token=Enter your token",
   false, null);

   report.Dictionary.Databases.Add(oDataDatabase);
   oDataDatabase.Synchronize(report);

   //Query with data filter
   ((StiSqlSource)report.Dictionary.DataSources["Products"]).SqlCommand =
   "Products?$filter=ProductID eq 2";

   StiWebViewer1.Report = report;
}
...
```

© 2003-2026 Stimulsoft

В таблице ниже представлены шаблоны строк подключения для различных типов источников данных.

Источник данных	Шаблон строки соединения
MS SQL	Integrated Security=False; Data Source=myServerAddress;Initial Catalog=myDataBase; User ID=myUsername; Password=myPassword;
MySQL	Server=myServerAddress; Database=myDataBase;UserId=myUsername; Pwd=myPassword;
ODBC	Driver={SQL Server}; Server=myServerAddress;Database=myDataBase ; Uid=myUsername; Pwd=myPassword;
OLE DB	Provider=SQLOLEDB.1; Integrated Security=SSPI;Persist Security Info=False; Initial Catalog=myDataBase;Data Source=myServerAddress
Oracle	Data Source=TORCL;User Id=myUsername;Password=myPassword;
MS Access	Provider=Microsoft.Jet.OLEDB.4.0;User ID=Admin;Password=pass;Data Source=C:\ \myAccessFile.accdb;
PostgreSQL	Server=myServerAddress; Port=5432; Database=myDataBase;User Id=myUsername; Password=myPassword;
Firebird	User=SYSDBA; Password=masterkey; Database=SampleDatabase.fdb;DataSource=my ServerAddress; Port=3050; Dialect=3; Charset=NONE;Role=; Connection lifetime=15; Pooling=true; MinPoolSize=0;MaxPoolSize=50; Packet Size=8192; ServerType=0;
SQL CE	Data Source=c:\MyData.sdf; Persist Security Info=False;

© 2003-2026 Stimulsoft

SQLite	Data Source=c:\mydb.db; Version=3;
DB2	Server=myAddress:myPortNumber;Database=myDataBase;UID=myUsername;PWD=myPassword;Max Pool Size=100;Min Pool Size=10;
Infomix	Database=myDataBase;Host=192.168.10.10;Server=db_engine_tcp;Service=1492;Protocol=onsoctcp;UID=myUsername;Password=myPassword;
Sybase	Data Source=myASEserver;Port=5000;Database=myDataBase;Uid=myUsername;Pwd=myPassword;
Teradata	Data Source=myServerAddress;UserID=myUsername;Password=myPassword;
VistaDB	Data Source=D:\folder\myVistaDatabaseFile.vdb4;Open Mode=ExclusiveReadWrite;
Universal(dotConnect)	Provider=Oracle;direct=true;data source=192.168.0.1;port=1521;sid=sid;user=user;password=pass
MongoDB	mongodb://<user>:<password>@localhost/test
OData	http://services.odata.org/v3/odata/OData.svc/
Другие...	В таблице представлены наиболее часто используемые шаблоны для строки соединения. Посмотреть различные варианты строк соединения вы можете на специализированном сайте.

Данные из XML, JSON, Excel файлов

Подключение к XML и JSON источникам данных можно хранить в шаблоне отчета. Если требуется указать файлы данных из кода, можно воспользоваться приведенным ниже примером.

© 2003-2026 Stimulsoft

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
  OnGetReportData="StiWebViewer1_GetReportData">
</cc1:StiWebViewer>
...
```

Default.aspx.cs (XML file)

```
...
protected void StiWebViewer1_GetReportData(object sender,
StiReportDataEventArgs e)
{
  DataSet data = new DataSet();
  data.ReadXml(Server.MapPath("Data/Demo.xml"));

  e.Report.RegData(data);
}
...
```

Default.aspx.cs (JSON file)

```
...
protected void StiWebViewer1_GetReportData(object sender,
StiReportDataEventArgs e)
{
  DataSet data
  = StiJsonToDataSetConverterV2.GetDataSetFromFile(Server.MapPath("Data/
  Demo.json"));

  e.Report.RegData(data);
}
...
```

Информация

Во вьювере присутствует возможность получения данных из Excel файла. Для этого можно воспользоваться указанным ниже методом.

```
DataSet dataSet = StiExcelConnector.Get().GetDataSet(new
StiExcelOptions(array, this.FirstRowIsHeader));
```

3.1.5 Локализация вьювера

Компонент **HTML5 Viewer** поддерживает полную локализацию своего интерфейса. Для того чтобы локализовать интерфейс вьювера отчетов на необходимый язык, используется специальное свойство **Localization**. В качестве

© 2003-2026 Stimulsoft

значения данного свойства необходимо указать путь к XML-файлу локализации (относительный либо абсолютный).

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
   Localization="Localization/en.xml">
</cc1:StiWebViewer>
...
```

При загрузке вьювера отчетов файл локализации будет загружен автоматически.

3.1.6 Использование тем

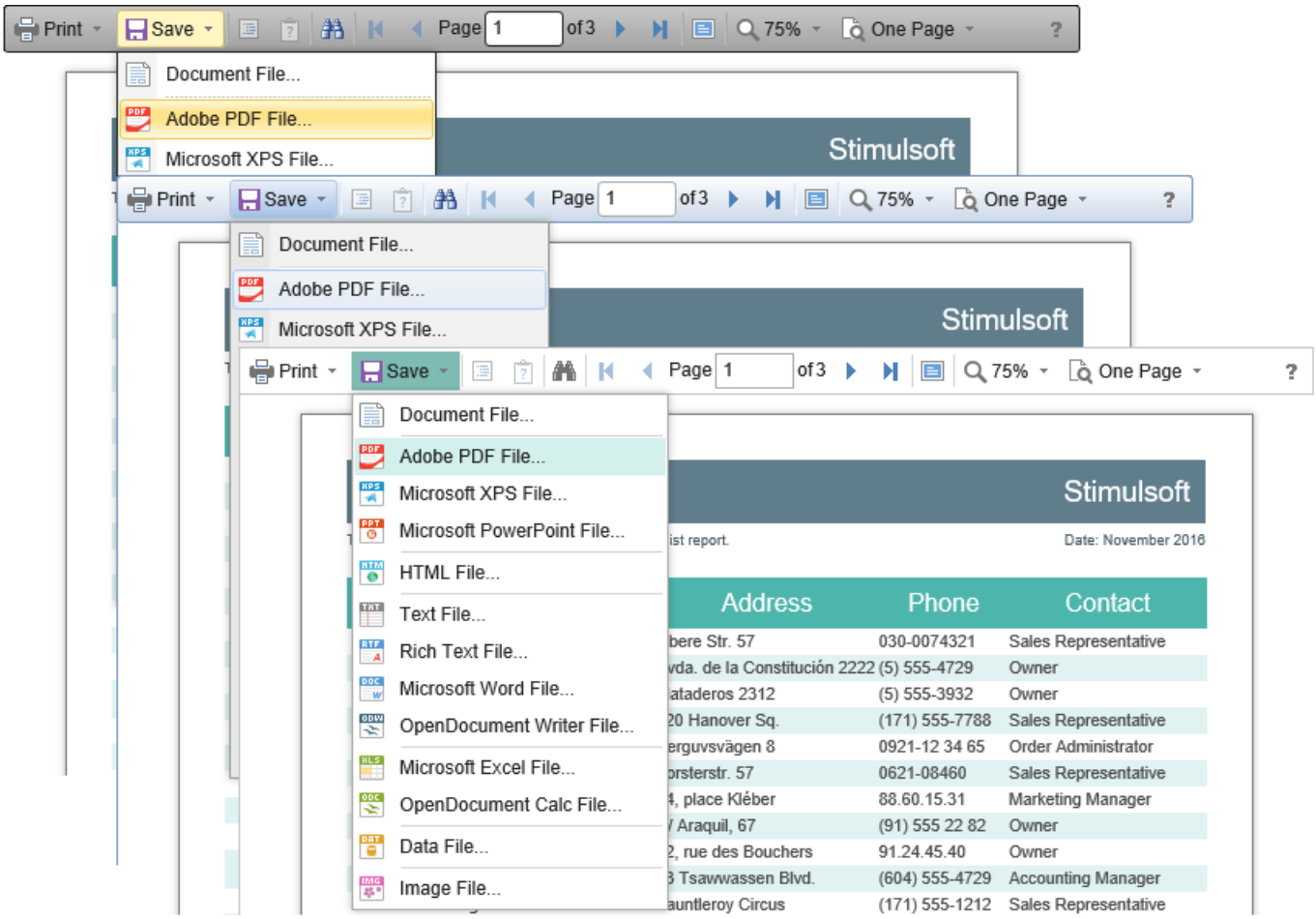
Компонент **HTML5 Viewer** имеет возможность смены тем оформления визуальных элементов управления. Для изменения темы используется свойство **Theme**.

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
   Theme="Office2022WhiteTeal">
</cc1:StiWebViewer>
...
```

На данный момент доступно **8 тем** оформления с различными цветовыми акцентами. В результате, доступно более **60** вариантов оформления. Это позволяет настроить внешний вид вьювера практически под любое оформление Web проекта.

© 2003-2026 Stimulsoft

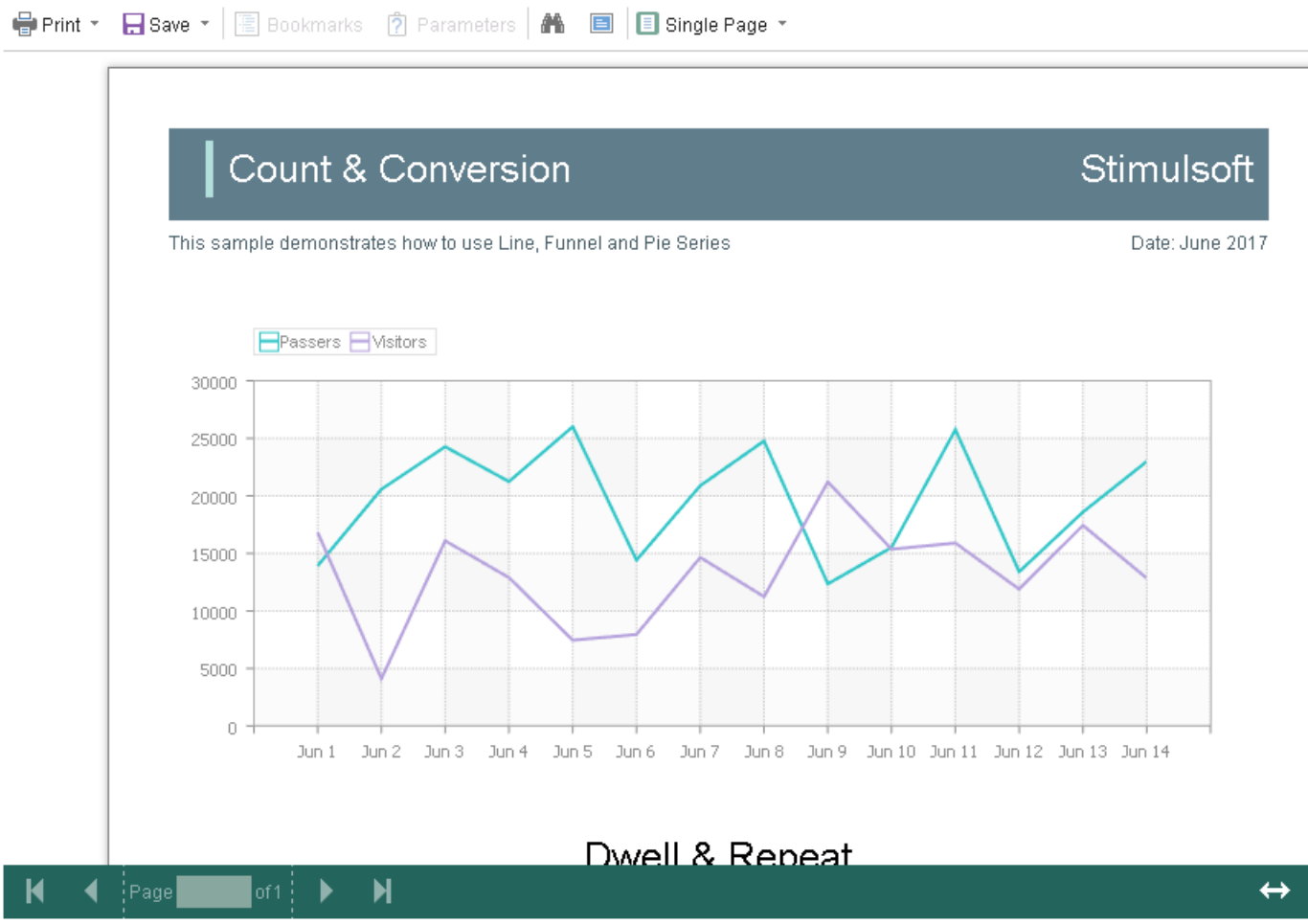


По умолчанию вьювер имеет только верхнюю панель инструментов, на которой находятся все элементы управления отчетом. При необходимости, панель инструментов можно разделить на верхнюю и нижнюю. На верхней панели будут находиться меню печати и экспорта отчета, а также кнопки работы с параметрами и закладками. Нижняя панель инструментов будет содержать элементы переключения между страницами отчета и меню управления масштабом. Для включения указанного режима предназначено свойство **ToolbarDisplayMode**, которое может иметь значение **Simple** (стандартный простой режим) и **Separated** (раздельный режим).

© 2003-2026 Stimulsoft

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
   ToolbarDisplayMode="Separated"
   ScrollbarsMode="true">
</cc1:StiWebViewer>
...
```



Дополнительно предусмотрена возможность установки параметров оформления основных элементов вьювера. Например, можно изменить шрифт и цвет надписей панели управления вьювера, установить фон вьювера, задать цвет границ страницы и др. Ниже представлен список доступных свойств, изменяющих оформление вьювера, и их значения по умолчанию.

© 2003-2026 Stimulsoft

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
   BackgroundColor="White"
   ShowPageShadow="true"
   PageBorderColor="Gray"
   ToolbarBackgroundColor="Empty"
   ToolbarBorderColor="Empty"
   ToolbarFontColor="Empty"
   ToolbarFontFamily="Arial">
</cc1:StiWebViewer>
...
```

3.1.7 Основные возможности

Информация

Обратите внимание, что в тексте документации под словом "отчет" подразумевается также и "дашборд", а под словосочетанием "шаблон отчета" - "шаблон дашборда".

К основным возможностям вьювера можно отнести следующие операции: отображение отчета, переключение между страницами отчета, изменение масштаба и режима отображения отчета. Все указанные операции выполняются в AJAX-режиме без перезагрузки страницы браузера. Для их выполнения не требуется установки никаких специальных опций либо событий.

Вьювер отчетов имеет специальное событие **OnViewerEvent**, которое будет вызвано при любом действии вьювера. В данном событии можно узнать тип действия, которое вьювер выполняет в данный момент, а также получить все параметры вьювера, переданные на сторону сервера.

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
   OnViewerEvent="StiWebViewer1_ViewerEvent">
</cc1:StiWebViewer>
...
```

© 2003-2026 Stimulsoft

Default.aspx.cs

```
...
protected void StiWebViewer1_ViewerEvent(object sender, StiViewerEventArgs
e)
{
   StiAction action = e.Action;
   StiRequestParams parameters = e.RequestParams;
}
...
```

3.1.8 Печать отчета

Информация

Обратите внимание, что возможность печати доступна только для отчетов, и недоступна для дашбордов.

В компоненте **HTML5 Viewer** предусмотрено несколько вариантов печати отчета. Каждый имеет свои особенности, достоинства и недостатки.

Печать в PDF (Print to PDF)
Печать будет осуществляться посредством экспортирования отчета в **PDF формат**. К достоинствам можно отнести большую точность расположения и печати элементов отчета по сравнению с остальными вариантами печати. Из недостатков можно упомянуть обязательное наличие установленного в браузере плагина для просмотра PDF файлов (современные браузеры имеют встроенное средство просмотра и печати PDF файлов).

Печать с просмотром (Print with Preview)
Печать отчета будет осуществлена в отдельном всплывающем окне браузера в **HTML формат**. Отчет можно предварительно просмотреть, а затем отправить на принтер или скопировать в другое место в виде текста или HTML-кода. К достоинствам можно отнести кроссбраузерность при печати, отсутствие необходимости установки специальных плагинов. Недостатком является относительно невысокая точность расположения элементов отчета, обусловленная особенностями реализации HTML-форматирования.

Печать без просмотра (Print without Preview)
Печать отчета будет осуществлена непосредственно на принтер без предварительного просмотра. После выбора этого пункта меню отображается

© 2003-2026 Stimulsoft

системный диалог печати. Так как печать в этом режиме осуществляется в **HTML формат**, то качество печати аналогично качеству печати отчета с предварительным просмотром.

Информация

При печати в **HTML формат** необходимо убедиться в соответствии параметров страницы отчета и параметров страницы принтера (размер бумаги, ориентация, поля, отступы), а также проверить настройки печати браузера, такие как отступы, колонтитулы, печать фоновых изображений, цветная печать.

События печати отчета

Для выполнения каких-либо действий перед печатью отчета предназначено специальное событие **OnPrintReport**. В данном событии можно узнать тип печати отчета, получить сам отчет, а также получить настройки экспорта отчета в случае печати в **PDF формат**.

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
    OnPrintReport="StiWebViewer1_PrintReport">
</cc1:StiWebViewer>
...
```

Default.aspx.cs

```
...
protected void StiWebViewer1_PrintReport(object sender,
StiPrintReportEventArgs e)
{
   StiPrintAction action = e.PrintAction;
   StiReport report = e.Report;
   StiExportSettings settings = e.Settings;
}
...
```

Настройка печати отчета

© 2003-2026 Stimulsoft

При выборе печати отчета на панели вьювера, отображается меню с выбором варианта печати. Компонент **HTML5 Viewer** имеет возможность принудительно установить требуемый режим печати. Для этого достаточно установить свойство **PrintDestination** в одно из указанных ниже значений.

- **Default** – при выборе печати будет отображено меню (значение свойства по умолчанию);
- **Pdf** – печать в PDF формат;
- **Direct** – печать в HTML формат непосредственно на принтер, будет отображен системный диалог печати;
- **WithPreview** – печать в HTML формат с предварительным просмотром во всплывающем окне.

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
  PrintDestination="Default">
</cc1:StiWebViewer>
...
```

Компонент **HTML5 Viewer** имеет возможность полностью отключить печать отчета, если она не требуется. Для этого необходимо установить значение **false** для свойства **ShowPrintButton**.

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
    ShowPrintButton="false">
</cc1:StiWebViewer>
...
```

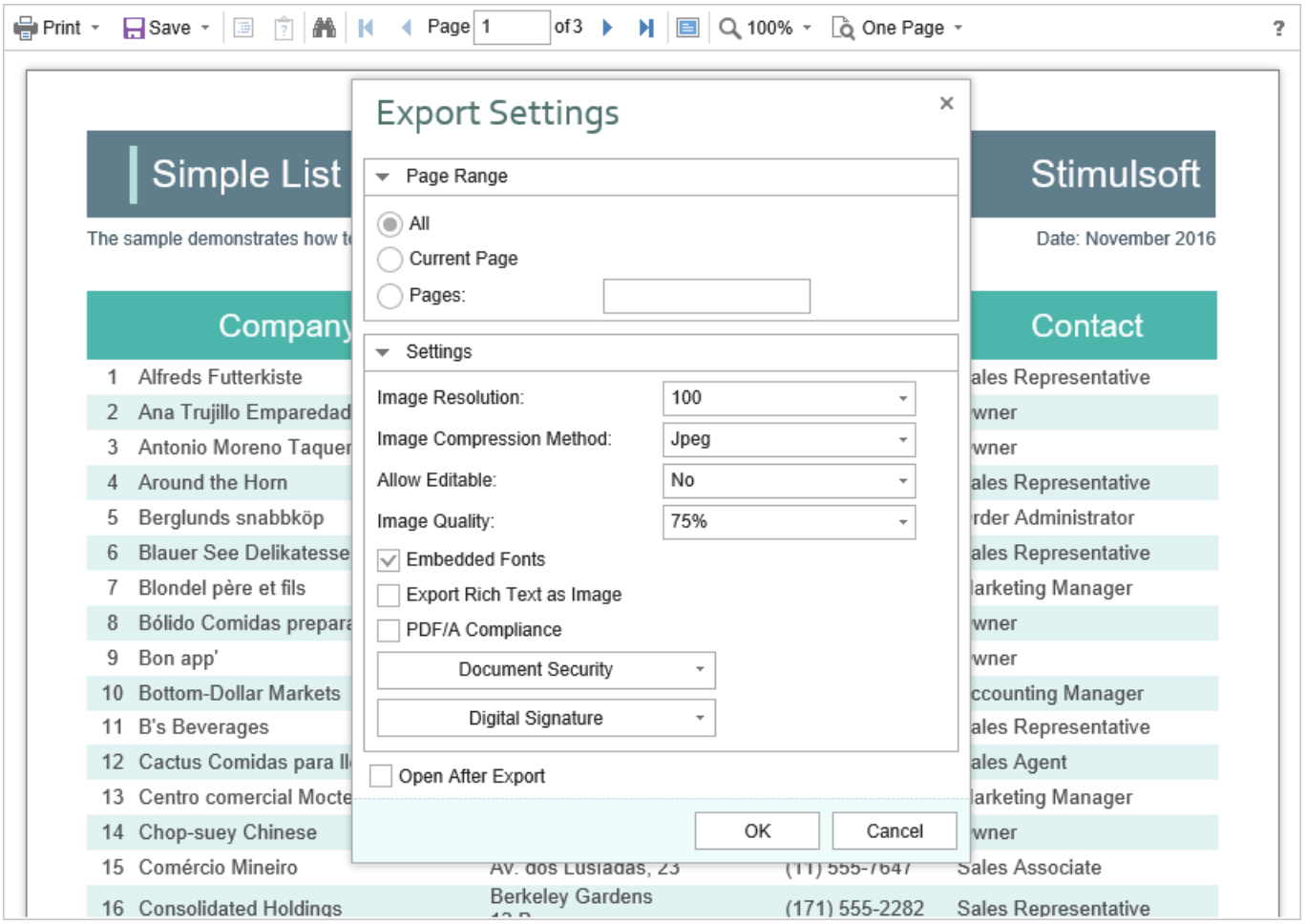
3.1.9 Экспорт отчета и дашборда

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

© 2003-2026 Stimulsoft

Компонент **HTML5 Viewer** позволяет экспортировать отображаемый отчет в три десятка различных форматов, таких как **PDF**, **HTML**, **Word**, **Excel**, **XPS**, **RTF**, изображения, текст и другие. Экспорт дашборда возможен в PDF, Excel, файлы изображений. Для работы функции экспорта не требуется дополнительных настроек вьювера.



События экспорта

Для проведения каких-либо действий перед экспортированием отчета предназначено специальное событие **OnExportReport**. В данном событии можно узнать тип экспорта отчета, получить сам отчет, а также получить настройки экспорта отчета и при необходимости изменить их.

© 2003-2026 Stimulsoft

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
  OnExportReport="StiWebViewer1_ExportReport">
</cc1:StiWebViewer>
...
```

Default.aspx.cs

```
...
protected void StiWebViewer1_ExportReport(object sender,
StiExportReportEventArgs e)
{
  StiExportFormat format = e.Format;
  StiReport report = e.Report;
  StiExportSettings settings = e.Settings;
}
...
```

Для выполнения каких-либо действий с уже экспортированным отчетом предназначено событие **OnExportReportResponse**, которое будет вызвано непосредственно перед сохранением файла. В данном событии можно получить формат экспорта, тип Web контента и имя файла для сохранения. Так же, можно получить, и при необходимости изменить, байтовый поток итогового файла экспорта.

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
  OnExportReportResponse="StiWebViewer1_ExportReportResponse">
</cc1:StiWebViewer>
...
```

Default.aspx.cs

```
...
protected void StiWebViewer1_ExportReportResponse(object sender,
StiExportReportResponseEventArgs e)
{
  StiExportFormat format = e.Format;
  string contentType = e.ContentType;
  string fileName = e.FileName;
  Stream stream = e.Stream;
}
...
```

© 2003-2026 Stimulsoft

Настройки экспорта

Каждый формат экспорта отчета компонента **HTML5 Viewer** имеет множество настроек, и каждая настройка имеет свои значения по умолчанию. Иногда требуется установить другие значения по умолчанию. Для этого предназначено специальное свойство вьювера **DefaultExportSettings**, которое представляет собой контейнер всех настроек экспортов, используемых по умолчанию.

Default.aspx.cs

```
...
protected void Page_Load(object sender, EventArgs e)
{
   StiWebViewer1.DefaultExportSettings.ExportToPdf.ImageQuality = 0.75f;
   StiWebViewer1.DefaultExportSettings.ExportToPdf.ImageFormat =
   StiImageFormat.Color;
   StiWebViewer1.DefaultExportSettings.ExportToHtml.ExportMode =
   StiHtmlExportMode.Div;
}
...
```

При вызове требуемого экспорта через меню вьювера, будет отображено диалоговое окно настроек экспорта. Значения элементов диалогового окна будут соответствовать настройкам экспортов по умолчанию. При изменении настроек в диалоговом окне и подтверждении экспорта отчета, настройки будут сохранены в cookies браузера, и при последующем вызове экспорта будут восстановлены. Вьювер позволяет отключить сохранение настроек экспорта, если требуется всегда восстанавливать настройки по умолчанию. Для этого достаточно установить значение **false** для свойства **StoreExportSettings**.

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
   StoreExportSettings="false">
</cc1:StiWebViewer>
...
```

Так же, если требуется, можно полностью убрать отображение диалоговых окон экспорта, экспортирование всегда будет осуществляться с настройками по умолчанию. Для этого достаточно установить значение **false** для свойства **ShowExportDialog**.

© 2003-2026 Stimulsoft

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
   ShowExportDialog="false">
</cc1:StiWebViewer>
...
```

Компонент **HTML5 Viewer** содержит порядка 30 различных форматов экспорта, и иногда требуется отключить не используемые форматы. Это позволяет разгрузить интерфейс и упростить использование вьювера. Для отключения не используемых форматов экспортов достаточно установить значение **false** для соответствующих свойств вьювера, представленных в списке ниже.

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
   ShowExportToDocument="true"
   ShowExportToPdf="true"
   ShowExportToXps="true"
   ShowExportToPowerPoint="true"
   ShowExportToHtml="true"
   ShowExportToHtml5="true"
   ShowExportToMht="true"
   ShowExportToText="true"
   ShowExportToRtf="true"
   ShowExportToWord="true"
   ShowExportToOpenDocumentWriter="true"
   ShowExportToExcel="true"
   ShowExportToExcelXml="true"
   ShowExportToExcelBiff="true"
   ShowExportToOpenDocumentCalc="true"
   ShowExportToCsv="true"
   ShowExportToDbf="true"
   ShowExportToXml="true"
   ShowExportToDif="true"
   ShowExportToSylk="true"
   ShowExportToImageBmp="true"
   ShowExportToImageGif="true"
   ShowExportToImageJpeg="true"
   ShowExportToImagePcx="true"
   ShowExportToImagePng="true"
   ShowExportToImageTiff="true"
   ShowExportToImageMetafile="true"
   ShowExportToImageSvg="true"
   ShowExportToImageSvgz="true">
</cc1:StiWebViewer>
...
```

© 2003-2026 Stimulsoft

Компонент **HTML5 Viewer** имеет возможность полностью отключить меню экспорта отчета, если оно не требуется. Для этого необходимо установить значение **false** для свойства **ShowSaveButton**.

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
   ShowSaveButton="false">
</cc1:StiWebViewer>
...
```

3.1.10 Режимы отображения

В компоненте **HTML5 Viewer** предусмотрено два режима отображения отчета: с полосами прокрутки и без них. По умолчанию установлен режим просмотра без полос прокрутки. Для включения режима просмотра с полосами прокрутки достаточно установить значение **true** для свойства **ScrollbarsMode**.

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
   ScrollbarsMode="false">
</cc1:StiWebViewer>
...
```

В первом режиме (без полос прокрутки), вьювер отображает страницу или отчет целиком, автоматически растягивая область просмотра. Если заданы размеры по ширине и высоте, то вьювер будет осуществлять обрезку вышедшей за границы страницы. Во втором режиме, в отличие от первого, при выходе страницы за границы размеров вьювера, обрезка осуществляться не будет. Вместо этого появятся полосы прокрутки, при помощи которых можно просмотреть страницу или отчет целиком.

Информация

В режиме просмотра отчета с полосами прокрутки необходимо задать высоту вьювера, иначе будет задана высота по умолчанию, равная **650 пикселей**.

© 2003-2026 Stimulsoft

В компоненте **HTML5 Viewer** предусмотрен режим полноэкранного отображения отчета и дашбордов. По умолчанию включен стандартный режим просмотра, вьювер имеет заданные в настройках размеры. Для включения полноэкранного режима просмотра достаточно установить значение **true** для свойства **FullScreenMode**.

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
   FullScreenMode="false">
</cc1:StiWebViewer>
...
```

Также, для включения либо отключения полноэкранного режима можно воспользоваться соответствующей кнопкой на панели управления вьювера.

В компоненте **HTML5 Viewer** доступно три режима отображения отчета: постраничное отображение, отчет целиком в виде ленты, и табличное отображение страниц отчета. Для управления режимами предназначено свойство **ViewMode**, которое соответственно принимает одно из указанных значений: **SinglePage**, **Continuous**, **MultiplePages**.

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
    ViewMode="SinglePage">
</cc1:StiWebViewer>
...
```

В компоненте **HTML5 Viewer** реализована поддержка работы как с обычным компьютером, так и с сенсорными экранами, мобильными устройствами. Для управления режимами интерфейса предназначено свойство **InterfaceType**, которое принимает одно из следующих значений:

- **Auto** – тип интерфейса вьювера будет выбран автоматически в зависимости от используемого устройства (значение по умолчанию);
- **Mouse** – принудительное использование стандартного интерфейса для управления вьювером при помощи мыши;
- **Touch** – принудительное использование Touch интерфейса для управления вьювером при помощи сенсорного экрана монитора, в этом режиме элементы

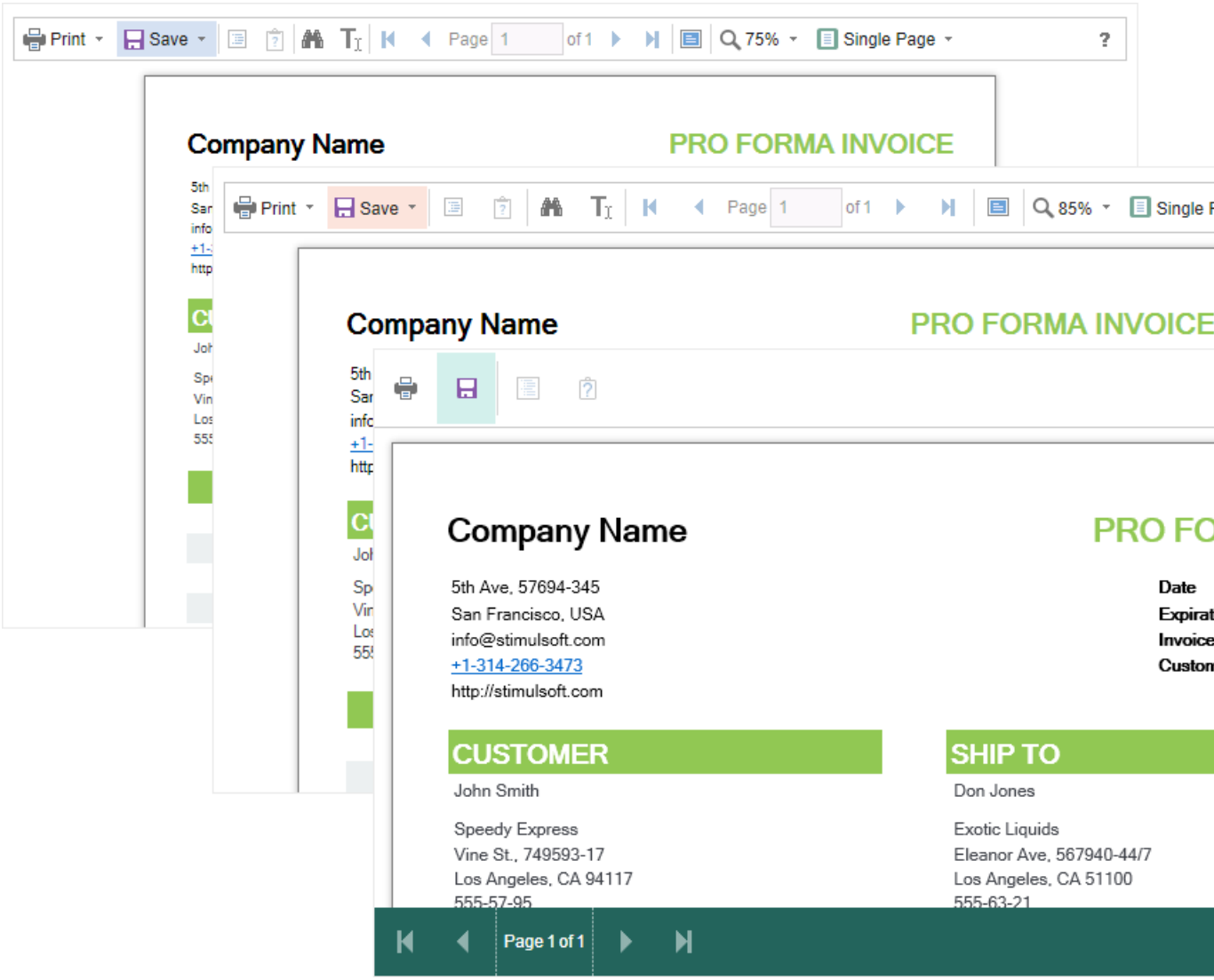
© 2003-2026 Stimulsoft

интерфейса вьювера имеют увеличенные размеры для удобства управления;

› **Mobile** – принудительное использование Mobile интерфейса для управления вьювером при помощи экрана смартфона, в этом режиме интерфейс вьювера имеет упрощенный вид и адаптирован для управления при помощи мобильного устройства.

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
   InterfaceType="Auto">
</cc1:StiWebViewer>
...
```



© 2003-2026 Stimulsoft

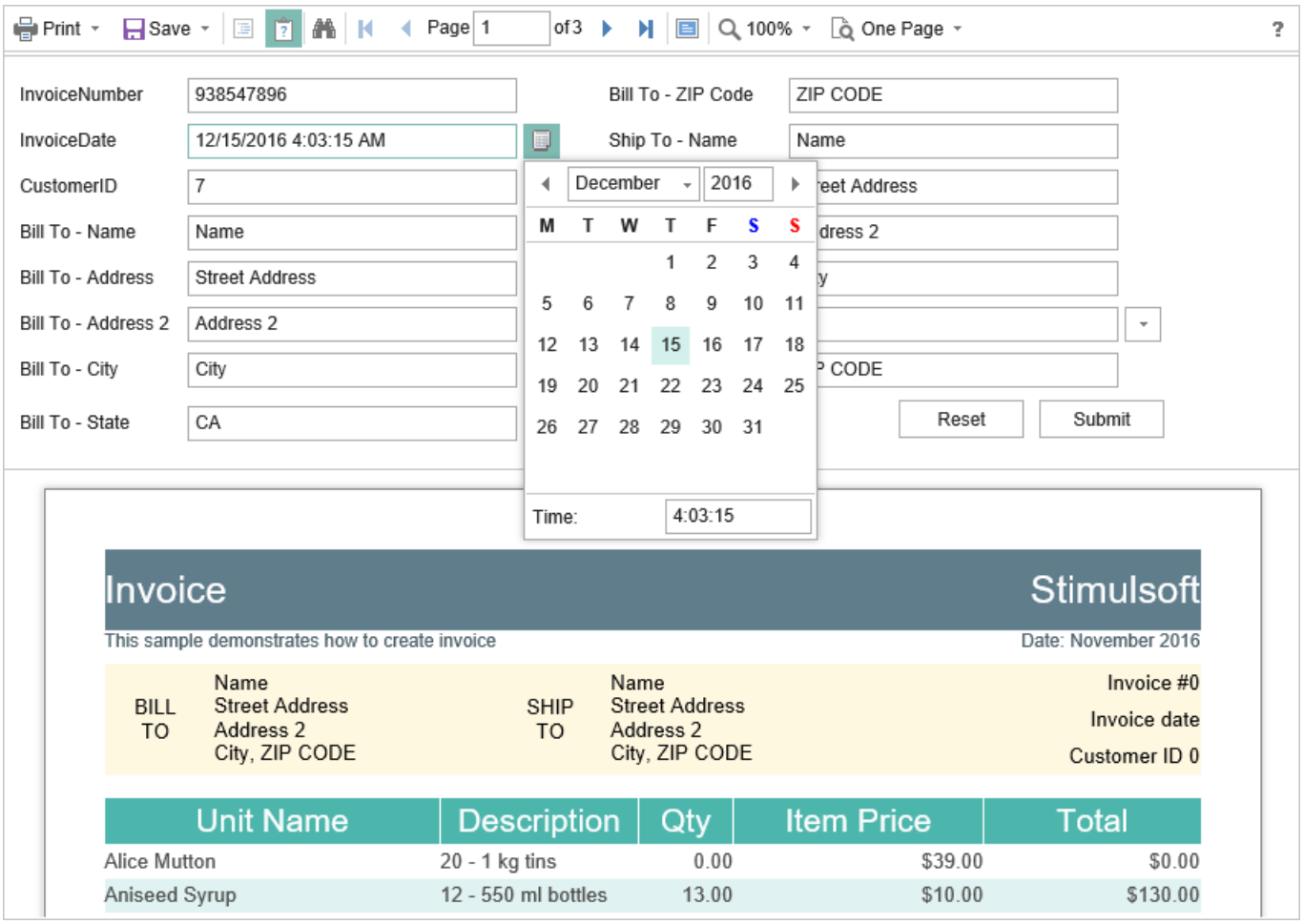
3.1.11 Работа с параметрами

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для работы с параметрами отчета в **HTML5 Viewer** реализована поддержка специальной панели параметров. Для добавления параметра на панель необходимо в шаблоне отчета определить переменную, запрашиваемую у пользователя. При просмотре отчета во вьюверe такая переменная будет автоматически добавлена на панель параметров. Поддерживаются все типы переменных отчета (обычные переменные, дата и время, границы, списки и др.).

© 2003-2026 Stimulsoft



Для выполнения каких-либо действий перед применением параметров предназначено специальное событие **OnInteraction**, которое будет вызвано при интерактивных действиях вьювера. При использовании панели параметров, тип действия будет иметь значение **Variables**.

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
  OnInteraction="StiWebViewer1_Interaction">
</cc1:StiWebViewer>
...
```

© 2003-2026 Stimulsoft

Default.aspx.cs

```
...
protected void StiWebViewer1_Interaction(object sender,
StiReportDataEventArgs e)
{
   if (e.Action == StiAction.Variables)
   {
      // The values of the variables passed from the client
      Hashtable variables = e.RequestParams.Interaction.Variables;
   }
}
...
```

Если работа с параметрами не требуется, то можно полностью отключить данную возможность. Для этого предназначено свойство **ShowParametersButton**, которому необходимо установить значение **false**.

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
   ShowParametersButton="false">
</cc1:StiWebViewer>
...
```

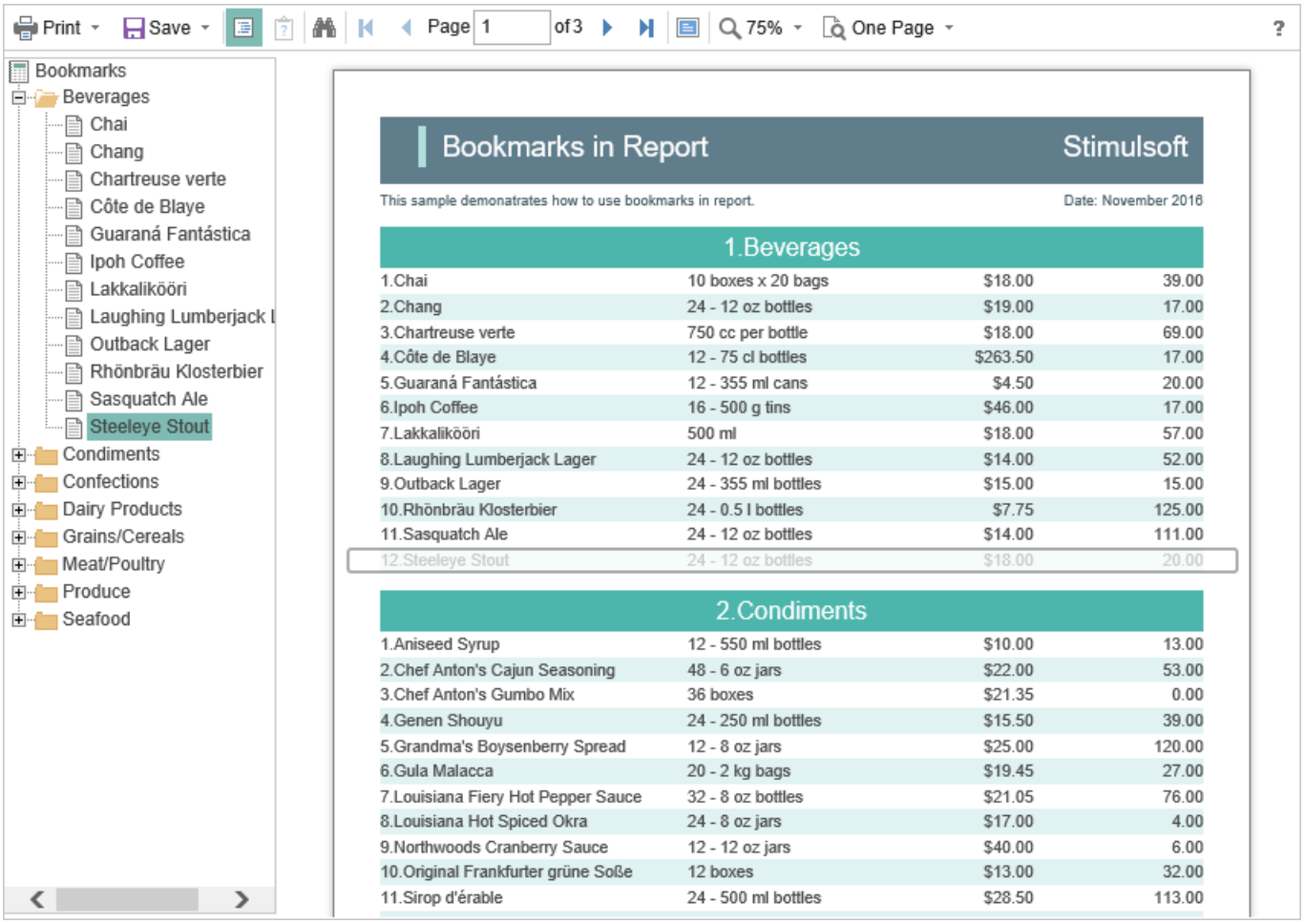
Информация

При такой конфигурации вьювера панель параметров не будет показана даже в том случае, если параметры присутствуют в отображаемом отчете.

3.1.12 Работа с закладками

В компоненте **HTML5 Viewer** реализована поддержка закладок отчета. При отображении такого отчета слева от страницы будет отображена панель с закладками. При выборе закладки отчета вьювер осуществит автоматический переход на нужную страницу, а элемент отчета с закладкой будет подсвечен.

© 2003-2026 Stimulsoft



По умолчанию ширина панели закладок равна 180 пикселей, компонент **HTML5 Viewer** позволяет изменить это значение. Для этого предназначено свойство **BookmarksTreeWidth**, значение которого указывается в пикселях.

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
    BookmarksTreeWidth="200">
</cc1:StiWebViewer>
...
```

Если работа с закладками отчета не требуется, то можно полностью отключить данную возможность. Для этого предназначено свойство **ShowBookmarksButton**, которое необходимо установить в значение **false**.

© 2003-2026 Stimulsoft

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
    ShowBookmarksButton="false">
</cc1:StiWebViewer>
...
```

Информация

В этом случае закладки отчета не будут показаны, даже если они присутствуют в отображаемом отчете. Данное свойство не оказывает влияние на печать и экспортирование отчета.

При печати отчета с закладками дерево закладок будет скрыто. Если помимо самого отчета требуется распечатать и закладки, то необходимо установить свойство **BookmarksPrint** в значение **true**.

Default.aspx

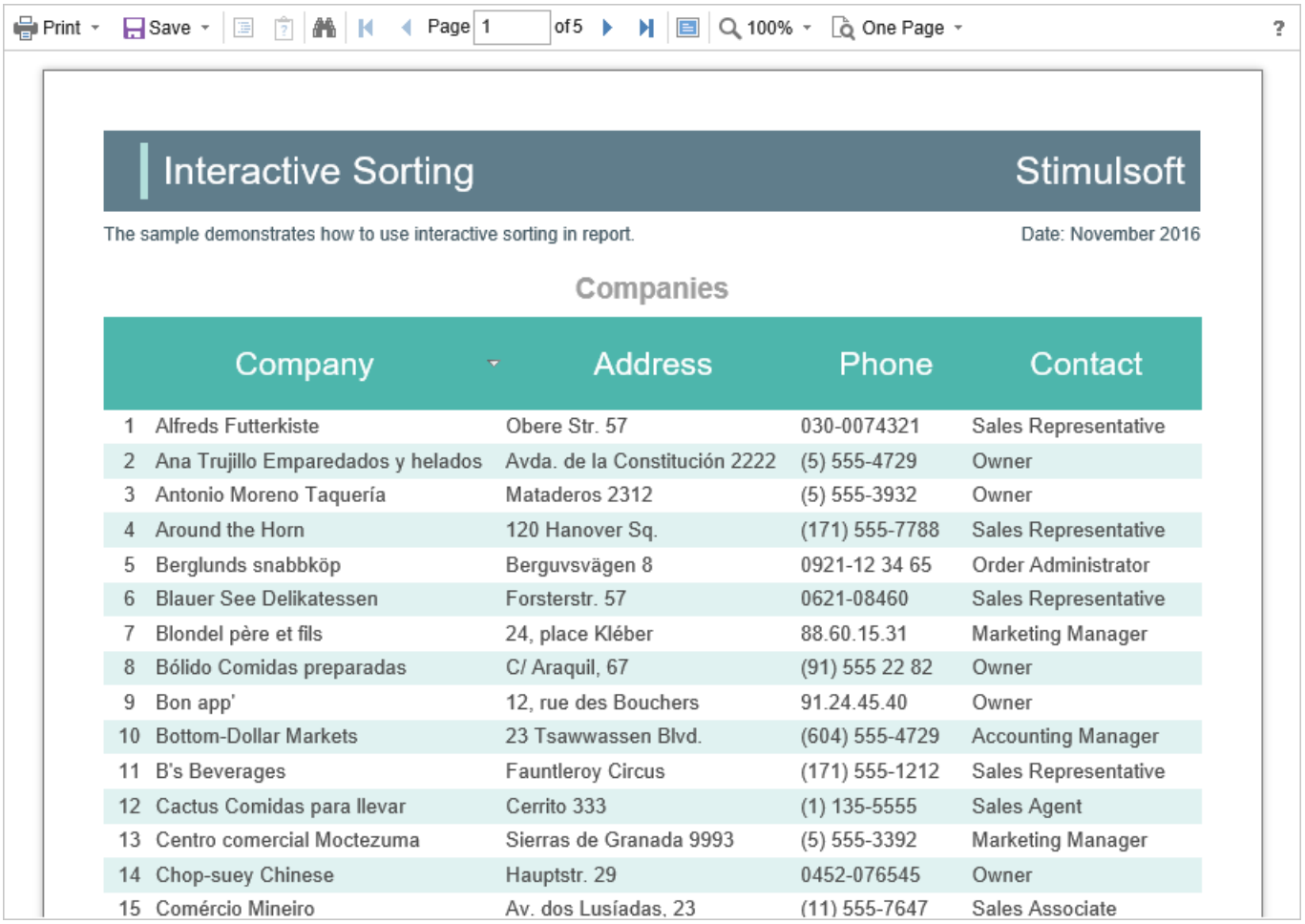
```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
    BookmarksPrint="true">
</cc1:StiWebViewer>
...
```

3.1.13 Динамическое сворачивание, сортировка и детализация

В компоненте **HTML5 Viewer** реализована поддержка динамической сортировки, сворачивания и детализации отчетов. Динамическая сортировка предоставляет возможность изменять направление сортировки в построенном отчете. Для этого следует щелкнуть по компоненту, у которого была установлена динамическая сортировка. Динамическая сортировка осуществляется в следующих направлениях: **По возрастанию (Ascending)** и **По убыванию (Descending)**. Каждый раз при щелчке по компоненту направление меняется на противоположное.

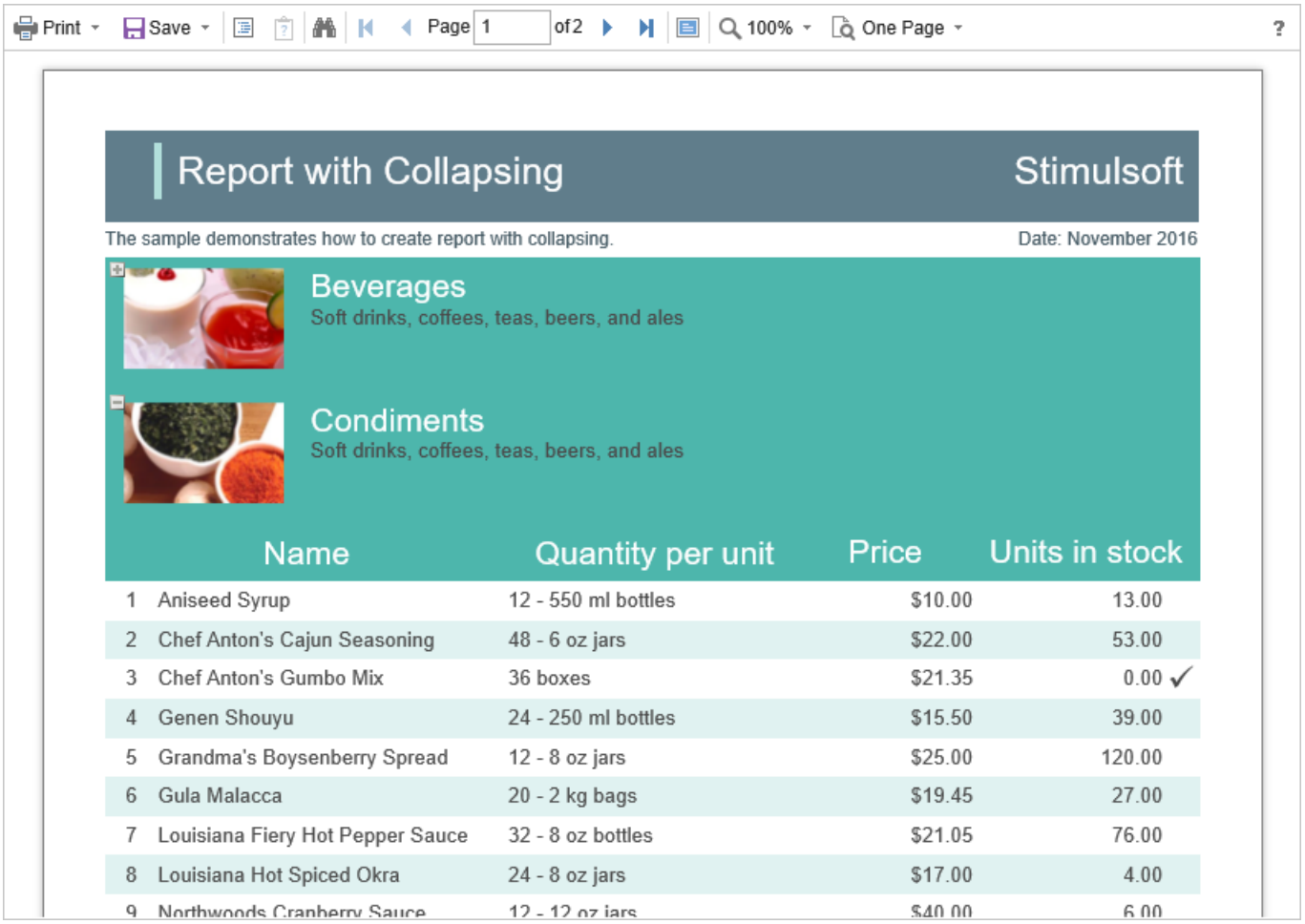
Допускается многоуровневая сортировка в отчете. Для этого необходимо удерживать клавишу **Ctrl** и последовательно нажимать на сортируемые компоненты отчета. Для сброса сортировки можно нажать на любой сортируемый компонент без удержания клавиши **Ctrl**.

© 2003-2026 Stimulsoft



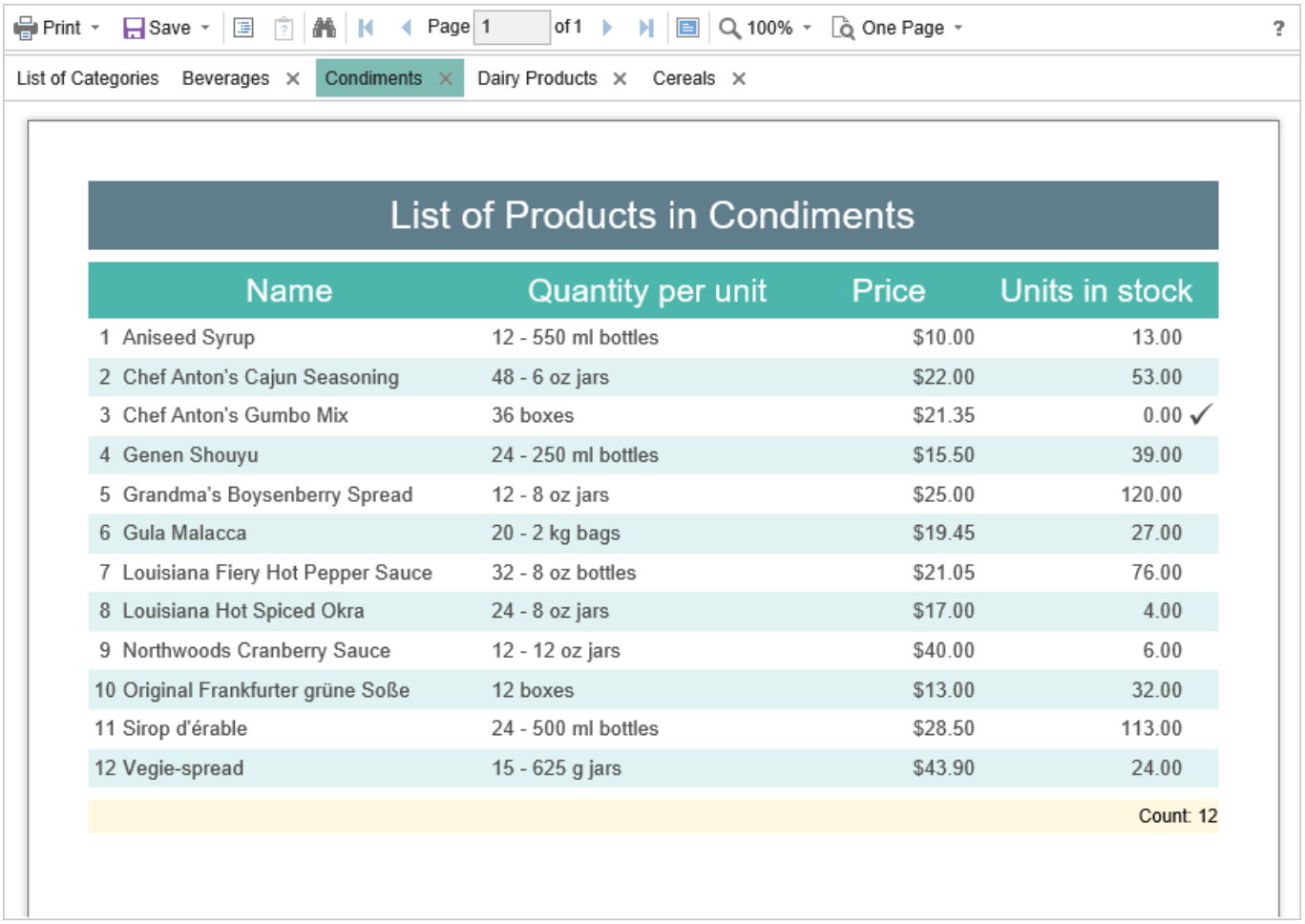
Отчет с динамическим сворачиванием представляет собой интерактивный отчет, в котором сворачивающиеся блоки могут сворачивать/разворачивать свое содержимое при нажатии на заголовок блока. Элементы отчета, которые можно свернуть/развернуть, обозначены специальной иконкой со знаком **[-]** или **[+]**.

© 2003-2026 Stimulsoft



При детализации данных под главной панелью вьювера будет отображена панель детализации с закладками детализированных отчетов. Отображенный в данный момент отчет будет подсвечен.

© 2003-2026 Stimulsoft



Для работы динамической сортировки, сворачивания и детализации отчетов не требуется дополнительных настроек вьювера. Для выполнения каких-либо действий перед сортировкой, сворачиванием или детализацией отчета предназначено специальное событие **OnInteraction**, которое будет вызвано при интерактивных действиях вьювера. Для каждого вида интерактивности вьювера предусмотрен определенный тип действия:

- **Sorting** – при использовании сортировки колонок;
- **DrillDown** – при использовании детализации отчета;
- **Collapsing** – при использовании сворачивания блоков отчета.

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
   OnInteraction="StiWebViewer1_Interaction">
</cc1:StiWebViewer>
...
```

© 2003-2026 Stimulsoft

Default.aspx.cs

```
...
protected void StiWebViewer1_Interaction(object sender,
StiReportDataEventArgs e)
{
   switch (e.Action)
   {
      case StiAction.Sorting:
        break;

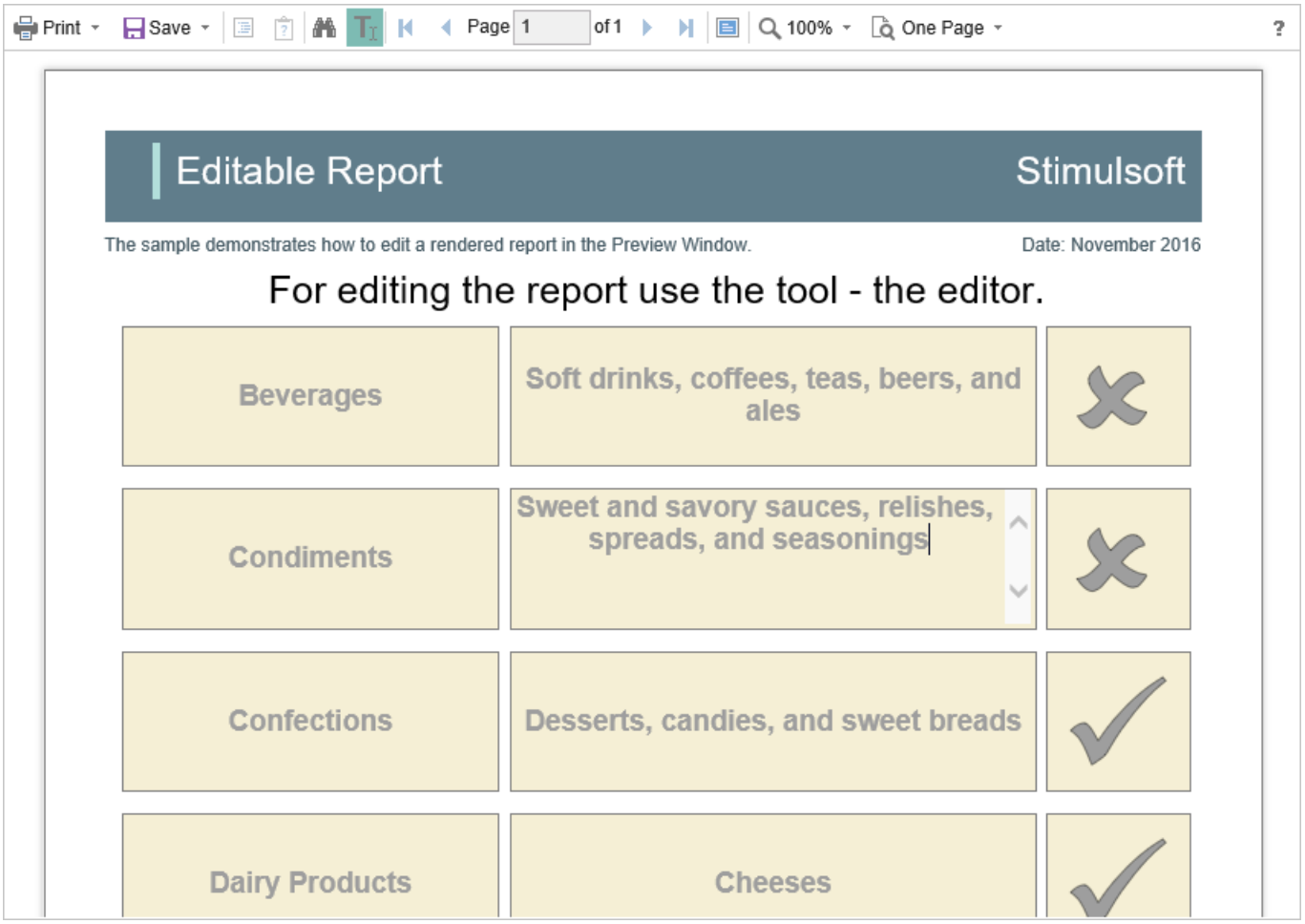
      case StiAction.DrillDown:
        break;

      case StiAction.Collapsing:
        break;
   }
}
...
```

3.1.14 Редактирование построенного отчета

Компонент **HTML5 Viewer** имеет возможность редактировать элементы построенного отчета, такие как текстовые поля и чекбоксы. Для того, чтобы редактирование стало возможным, в самом шаблоне отчета необходимо пометить нужные компоненты как редактируемые. После отображения отчета во вьювере, для начала редактирования необходимо нажать на соответствующую кнопку на панели вьювера. После завершения редактирования необходимо нажать указанную кнопку еще раз, и все внесенные изменения будут примерены к отчету.

© 2003-2026 Stimulsoft



Для работы режима редактирования отчета не требуется никаких специальных настроек вьювера.

Информация

Отредактированные значения будут применены при печати либо экспортировании отчета, при этом исходный отчет останется не тронутым. После перезагрузки вьювера все значения будут возвращены к исходным.

© 2003-2026 Stimulsoft

3.1.15 Отправка отчета по Email

Информация

Обратите внимание, что возможность Отправить отчет по Email доступна только для отчетов, и недоступна для дашбордов.

В компоненте **HTML5 Viewer** предусмотрена возможность отправки отчета по Email. Для активации данной возможности необходимо установить свойство вьювера **ShowSendEmailButton** в значение **true** и добавить обработчик события **OnEmailReport**.

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
   ShowSendEmailButton="true"
   OnEmailReport="StiWebViewer1_EmailReport">
</cc1:StiWebViewer>
...
```

© 2003-2026 Stimulsoft

Default.aspx.cs

```
...
protected void StiWebViewer1_EmailReport(object sender,
StiEmailReportEventArgs e)
{
   StiExportFormat format = e.Format;
   StiReport report = e.Report;
   StiExportSettings settings = e.Settings;
   StiEmailOptions options = e.Options;

   // Passed from the viewer, can be checked and changed
   // options.AddressTo = "";
   // options.Subject = "";
   // options.Body = "";

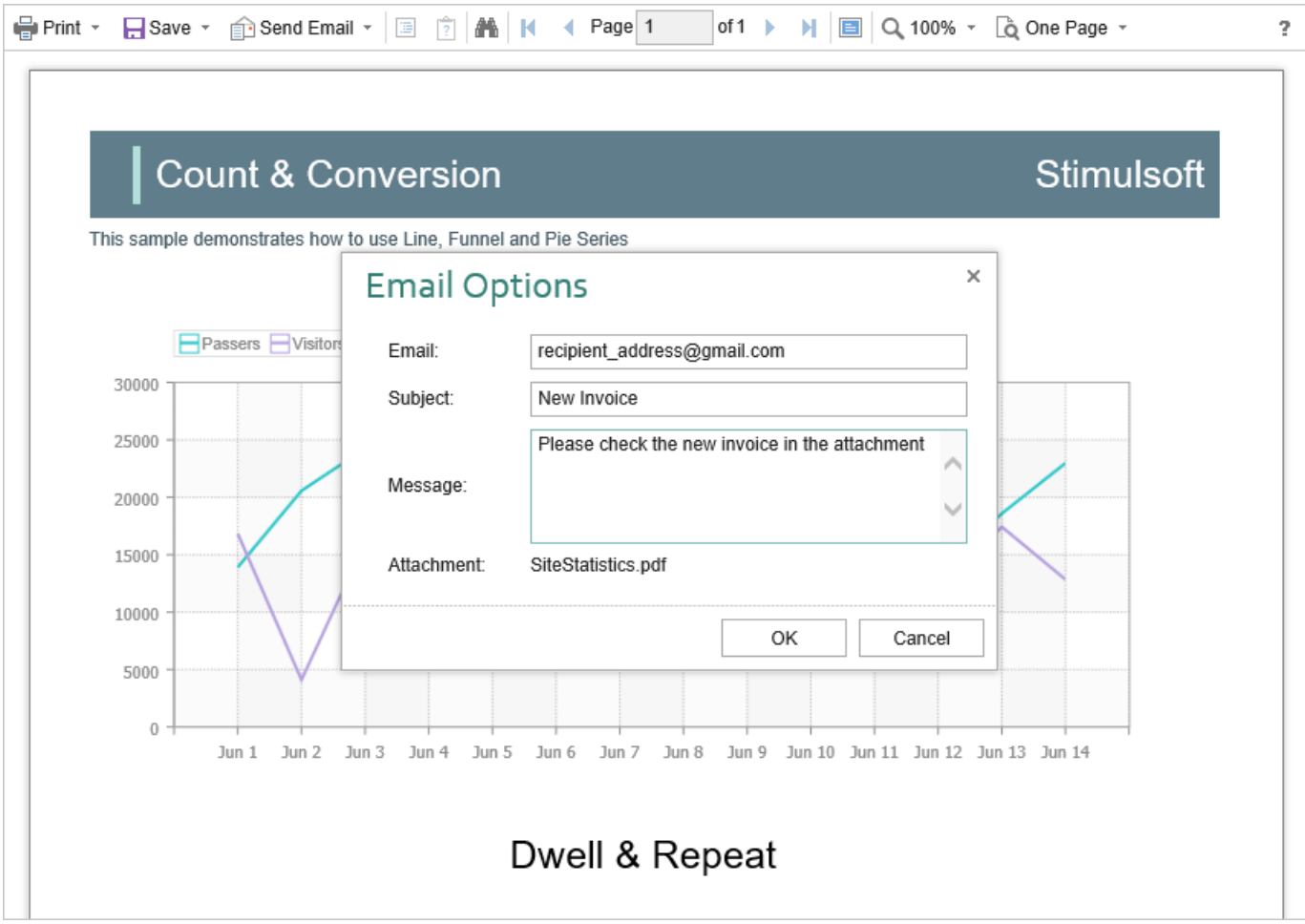
   // Should be filled here
   options.AddressFrom = "admin_address@test.com";
   options.Host = "smtp.test.com";
   options.Port = 465;
   options.UserName = "admin_address@test.com";
   options.Password = "admin_password";

   // options.CC.Add("email@test.com");
   // options.BCC.Add("email@test.com");
   // options.EnableSsl = true;
}
...
```

В событии **OnEmailReport** можно узнать тип экспорта отчета, получить сам отчет, а также получить настройки экспорта отчета и при необходимости изменить их. Так же в этом событии необходимо установить параметры отправки почты, такие как адрес отправителя, имя сервера и номер порта, имя пользователя и пароль – все эти параметры будут использованы для отправки Email.

При отправке отчета по Email отображается меню выбора формата вложения, которое соответствует меню выбора формата экспортирования отчета. После выбора формата отобразиться диалог ввода параметров отправки, таких как Email получателя, тема и текст письма.

© 2003-2026 Stimulsoft



После подтверждения отправки будет вызвано описанное выше событие **OnEmailReport**, в котором можно проверить и скорректировать данные, введенные в этой форме. Экспортированный файл отчета будет прикреплен к письму автоматически.

Компонент **HTML5 Viewer** позволяет установить значения по умолчанию для формы отправки Email. Для этого предназначены свойства **DefaultEmailAddress**, **DefaultEmailSubject** и **DefaultEmailMessage**. По умолчанию данные свойства являются пустыми.

© 2003-2026 Stimulsoft

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
   DefaultEmailAddress="recipient_address@gmail.com"
   DefaultEmailSubject="New Invoice"
   DefaultEmailMessage="Please check the new invoice in the attachment"

   ShowSendEmailButton="true"
   OnEmailReport="StiWebViewer1_EmailReport">
</cc1:StiWebViewer>
...
```

3.1.16 Вызов дизайнера из вьювера

Компонент **HTML5 Viewer** имеет возможность вызвать дизайнер отчетов. Для этого предназначена специальная кнопка **Дизайн (Design)** на панели инструментов вьювера (по умолчанию данная кнопка отключена). Для использования этой возможности необходимо установить свойство **ShowDesignButton** в значение **true**, а также определить обработчик события **OnDesignReport**.

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
   ShowDesignButton="true"
   OnDesignReport="StiWebViewer1_DesignReport">
</cc1:StiWebViewer>
...
```

Default.aspx.cs

```
...
protected void StiWebViewer1_DesignReport(object sender,
StiReportDataEventArgs e)
{
   StiReport report = e.Report;
   this.Response.Redirect("Designer.aspx?report=" + report.ReportName);
}
...
```

© 2003-2026 Stimulsoft

Информация

Сам вьювер не запускает дизайнер, он лишь вызывает указанное событие и передает в качестве аргументов просматриваемый отчет или дашборд. В событии можно выполнить перенаправление на ASPX страницу, на которой расположен дизайнер отчетов.

3.1.17 Кэширование

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Компонент **HTML5 Viewer** позволяет использовать кэш сервера для хранения построенного отчета. Если кэширование не используется, то при каждом запросе страницы необходимо загружать отчет, подключать данные и строить его заново. Если использовать кэширование, то при следующем обновлении страницы, ранее построенный отчет будет загружен из кэша.

При использовании кэширования стоит учитывать, что каждый сохраненный отчет в кэше занимает память сервера и при большом количестве запросов к отчетам это может стать критичным фактором. Поэтому необходимо выбирать из двух вариантов – или маленькие потребности к памяти, но большие к производительности, или небольшие потребности к производительности, но большие к памяти.

Управлять кэшированием можно при помощи следующих приведенных ниже свойств.

Свойство CacheMode

© 2003-2026 Stimulsoft

Данное свойство вьювера включает кэширование и устанавливает его тип. Может принимать одно из значений, указанных в перечислении **StiServerCacheMode**:

- **None** – кэширование отключено, каждое действие вьювера требует загрузки отчета и, если это шаблон отчета, последующее его построение;
- **ObjectCache** – для кэширования используется кэш сервера, в котором сохраняется объект отчета (значение по умолчанию);
- **StringCache** - для кэширования используется кэш сервера, в котором сохраняется отчет в виде упакованной строки;
- **ObjectSession** - для кэширования используется текущая сессия, в которой сохраняется объект отчета;
- **StringSession** - для кэширования используется текущая сессия, в которой сохраняется отчет в виде упакованной строки.

Свойство CacheItemPriority

Данное свойство устанавливает приоритет хранимого в кэше сервера отчета, влияет на автоматическую очистку памяти сервера в случае ее нехватки. Чем меньше приоритет, тем больше шанс удаления информации из памяти.

Свойство CacheTimeout

Данное свойство указывает величину времени в минутах, в течение которого необходимо сохранять отчет в кэше сервера. Если при использовании кэширования требуемый отчет не найден в кэше (т.е. истекло время хранения объекта), то он будет запрошен заново при помощи специального события **OnGetReport** с последующим подключением данных отчета и его построением.

Компонент **HTML5 Viewer** предоставляет возможность определить собственные методы работы с кэшированием отчета. Для этого предназначен специальный класс **StiCacheHelper**, который содержит методы получения отчета из кэша и сохранения отчета в кэш. Необходимо создать новый класс,

© 2003-2026 Stimulsoft

унаследованный от **StiCacheHelper**, и перегрузить указанные выше методы, которые соответственно имеют названия **GetReport** и **SaveReport**.

Default.aspx.cs

```
...
public partial class _Default : Page
{
  public class StiMyCacheHelper : StiCacheHelper
  {
    public override StiReport GetReport(string guid)
    {
      string path =
      Path.Combine(HttpContext.Current.Server.MapPath(string.Empty),
      "CacheFiles", guid);
      if (File.Exists(path))
      {
        StiReport report = new StiReport();
        string packedReport = File.ReadAllText(path);
        if (guid.EndsWith("template"))
        report.LoadPackedReportFromString(packedReport);
        else report.LoadPackedDocumentFromString(packedReport);

        return report;
      }
      return null;

      //return base.GetReport(guid);
    }

    public override void SaveReport(StiReport report, string guid)
    {
      string packedReport = guid.EndsWith("template") ?
      report.SavePackedReportToString() :
      report.SavePackedDocumentToString();
      string path =
      Path.Combine(HttpContext.Current.Server.MapPath(string.Empty),
      "CacheFiles", guid);
      File.WriteAllText(path, packedReport);

      //base.SaveReport(report, guid);
    }
  }

  static _Default()
  {
    StiWebViewer.CacheHelper = new StiMyCacheHelper();
  }
}
...
```

Для инициализации работы с кэшированием отчета при помощи созданного класса, достаточно задать его в качестве значения статического свойства

© 2003-2026 Stimulsoft

StiWebViewer.CacheHelper в конструкторе ASPX страницы.

Информация

При отключенном кэшировании отчета (свойство вьювера **CacheMode** имеет значение **None**), указанный класс задействован не будет.

3.1.18 Экспорт и печать из кода

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет». Кроме этого, следует понимать, что возможность печати из кода доступна только для отчетов, и недоступна для дашбордов.

Компонент **HTML5 Viewer** предоставляет возможность печатать отчет различными способами и выполнять экспортирование отчета в различные форматы. Эти действия выполняются при помощи меню вьювера. Если требуется напечатать отчет или выполнить экспорт при помощи кода, например, в событии нажатия кнопки, то можно воспользоваться специальным классом **StiReportResponse**. Данный класс содержит набор статических методов, позволяющих выполнить печать либо экспорт отчета из кода, при этом вьювер отчетов не требуется.

Default.aspx

```
...
<asp:Button ID="Button1" runat="server" onclick="Button1_Click"
Text="Print Report" />
<asp:Button ID="Button2" runat="server" onclick="Button2_Click"
Text="Export Report" />
...
```

© 2003-2026 Stimulsoft

Default.aspx.cs

```
...
private StiReport LoadSimpleList()
{
   DataSet dataSet = new DataSet();
   dataSet.ReadXml(Server.MapPath("Reports/Demo.xml"));

   StiReport report = new StiReport();
   report.Load(Server.MapPath("Reports/SimpleList.mrt"));
   report.RegData(dataSet);

   return report;
}

//For reports only
protected void Button1_Click(object sender, EventArgs e)
{
   StiReport report = LoadSimpleList();

   StiReportResponse.PrintAsPdf(report);
   //StiReportResponse.PrintAsHtml(report);
}

//For reports and dashboards
protected void Button2_Click(object sender, EventArgs e)
{
   StiReport report = LoadSimpleList();

   StiReportResponse.ResponseAsPdf(report);
   //StiReportResponse.ResponseAsExcel2007(report);
   //StiReportResponse.ResponseAsPng(report);
   //StiReportResponse.ResponseAsJson(report);
}
...
```

Класс **StiReportResponse** содержит методы печати в формате PDF и HTML, а также методы экспортирования отчета в любой из поддерживаемых форматов. В качестве аргументов методы могут принимать различные настройки экспортов, режимы отображения и варианты сохранения полученных файлов.

3.1.19 События вьювера

Компонент **HTML5 Viewer** поддерживает события, которые предоставляют возможность выполнить необходимые операции перед определенными действиями, такими как печать и экспортирование отчета, отправка отчета по Email, интерактивность, экспорт дашборда и др. Ниже приведен пример обработки событий вьювера.

© 2003-2026 Stimulsoft

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
  OnExportReport="StiWebViewer1_ExportReport">
</cc1:StiWebViewer>
...
```

Default.aspx.cs

```
...
protected void StiWebViewer1_ExportReport(object sender,
StiExportReportEventArgs e)
{
  if (e.Format == StiExportFormat.Pdf)
  {
    StiPdfExportSettings pdfSettings = e.Settings as
    StiPdfExportSettings;
    pdfSettings.ImageQuality = 50;
    pdfSettings.ImageResolution = 50;
    pdfSettings.ImageCompressionMethod =
    StiPdfImageCompressionMethod.Jpeg;
  }
}
...
```

Список событий

Наименование	Описание
OnGetReport	Событие возникает при запросе отчета для просмотра во вьювере.
OnGetReportData	Событие возникает при подключении данных отчета перед его построением.
OnPrintReport	Событие возникает при печати отчета. Не актуально при просмотре дашбордов.
OnExportReport	Событие возникает при экспортировании отчета.
OnExportReportResponse	Событие возникает после экспортирования отчета перед сохранением файла экспорта.
OnEmailReport	Событие возникает при отправке отчета по Email. Не актуально при просмотре дашбордов.
OnInteraction	Событие возникает при интерактивных действиях вьювера, таких как использование переменных отчета,

© 2003-2026 Stimulsoft

	динамическое сворачивание, детализация и сортировка отчета.
OnDesignReport	Событие возникает при нажатии на кнопку Дизайн (Design) на панели инструментов вьювера.
OnViewerEvent	Событие возникает при любом действии вьювера.

3.1.20 Время ожидания

При работе с компонентом **StiWebViewer** можно установить время ожидания (timeout) выполнения различных операций - хранение отчета в кэше, ответ сервера, выполнение запроса. Настройка времени ожидания выполняется при помощи свойств компонента и опций отчета.

Свойство CacheTimeout

Предоставляет возможность установить время в минутах, которое сервер будет хранить построенный отчет в кэше с момента последнего действия вьювера. По умолчанию установлено значение 10 минут.

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
   CacheTimeout="10">
</cc1:StiWebViewer>
...
```

Default.aspx.cs

```
...
protected void Page_Load(object sender, EventArgs e)
{
   StiWebViewer1.CacheTimeout = 10;
}
...
```

Использование кэша увеличивает скорость работы вьювера с отчётом. Более подробно можно ознакомиться в главе Кэширование.

Свойство RequestTimeout

Предоставляет возможность установить время ожидания ответа от сервера в

© 2003-2026 Stimulsoft

секундах, по истечении которого будет выдана ошибка связи с сервером. По умолчанию установлено значение 30 секунд. Для больших отчетов рекомендуется увеличить данное значение.

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
   RequestTimeout="10">
</cc1:StiWebViewer>
...
```

Default.aspx.cs

```
...
protected void Page_Load(object sender, EventArgs e)
{
   StiWebViewer1.RequestTimeout = 30;
}
...
```

Опция CommandTimeout

Предоставляет возможность установить время ожидания запроса в секундах, при использовании SQL источников данных в отчете. Значение данного свойства сохраняется в самом шаблоне отчёта для каждого SQL подключения отдельно.

Ниже приведен пример кода, который предоставляет возможность установить время ожидания запроса для уже созданного MS SQL соединения и источников данных в отчете.

Default.aspx.cs

```
...
protected void Page_Load(object sender, EventArgs e)
{

   StiReport report = new StiReport();
   report.Load(Server.MapPath("Report.mrt"));
   ((StiSqlSource)
   report.Dictionary.DataSources["DataSourceName"]).CommandTimeout = 1000;

   StiWebViewer1.Report = report;
}
...
```

© 2003-2026 Stimulsoft

3.1.21 Настройки вьювера

Настройка **HTML5 Viewer** выполняется при помощи свойств компонента, которые можно задать на ASPX странице либо при помощи C#/VB кода. Ниже приведены примеры установки свойств вьювера.

Default.aspx

```
...
<cc1:StiWebViewer ID="StiWebViewer1" runat="server"
   ShowPrintButton="false"
   ScrollbarsMode="true"
   ShowTooltips="false"
   ShowExportToDbf="false"
   ShowExportToDif="false"
   Zoom="75"
   Theme="Office2022WhiteTeal">
</cc1:StiWebViewer>
...
```

Default.aspx.cs

```
...
protected void Page_Load(object sender, EventArgs e)
{
   StiWebViewer1.ShowPrintButton = false;
   StiWebViewer1.ScrollbarsMode = true;
   StiWebViewer1.ShowTooltips = false;
   StiWebViewer1.ShowExportToDbf = false;
   StiWebViewer1.ShowExportToDif = false;
   StiWebViewer1.Zoom = 75;
   StiWebViewer1.Theme = StiTheme.Office2022WhiteTeal;
   StiWebViewer1.ReportDisplayMode = StiReportDisplayMode.Auto;
}
...
```

Обратите внимание, что все элементы дашбордов имеют собственные кнопки экспорта и полноэкранного режима просмотра. Специальных опций для управления их отображением не предусмотрено, но их можно выключить через свойства элемента. Нижеприведенный код необходимо добавить после загрузки отчета, перед его передачей вьюверу.

© 2003-2026 Stimulsoft

Default.aspx.cs

```
...
var dbsElementInteraction = (report.GetComponentByName("RegionMap1") as
Stimulsoft.Report.Dashboard.IStiElementInteraction).DashboardInteraction;
(dbsElementInteraction as
Stimulsoft.Report.Dashboard.IStiInteractionLayout).ShowFullScreenButton =
false;
(dbsElementInteraction as
Stimulsoft.Report.Dashboard.IStiInteractionLayout).ShowSaveButton = false;
...
```

Основные настройки

Наименование	Описание
Width	Задает ширину компонента в необходимых единицах измерения, которые определены в классе **Unit**. Доступна установка значения в пикселях: **Unit.Pixel()**, поинтах: **Unit.Point()** и процентах: **Unit.Percentage()**. По умолчанию установлена ширина 100%.
Height	Задает высоту компонента в необходимых единицах измерения, которые определены в классе **Unit**. Доступна установка значения в пикселях: **Unit.Pixel()**, поинтах: **Unit.Point()** и процентах: **Unit.Percentage()**. По умолчанию установлена автоматическая высота в зависимости от размера страницы отчета, либо 650 пикселей в режиме отображения вьювера с полосами прокрутки.

Работа с сервером

Наименование	Описание
RequestTimeout	Устанавливает время ожидания ответа от сервера в секундах, по истечении которого будет выдана ошибка. По умолчанию установлено значение 30 секунд. Для больших отчетов рекомендуется увеличить

© 2003-2026 Stimulsoft

	данное значение.
CacheTimeout	Устанавливает время в минутах, которое сервер будет хранить построенный отчет с момента последнего действия вьювера. По умолчанию установлено значение 10 минут.
CacheMode	Устанавливает режим кэширования отчета. Может принимать одно из указанных ниже значений перечисления **StiServerCacheMode**: › **None** – кэширование отключено, каждое действие вьювера требует загрузки отчета и, если это шаблон отчета, последующее его построение; › **ObjectCache** – в качестве хранилища используется кэш, отчет хранится как объект (значение свойства по умолчанию); › **ObjectSession** – в качестве хранилища используется сессия, отчет хранится как объект; › **StringCache** – в качестве хранилища используется кэш сервера, отчет сериализуется в упакованную строку; › **StringSession** – в качестве хранилища используется сессия, отчет сериализуется в упакованную строку.
CacheItemPriority	Устанавливает приоритет хранимого в кэше сервера отчета. Это свойство влияет на автоматическую очистку памяти сервера в случае ее нехватки. Чем меньше приоритет, тем больше шанс удаления информации из памяти.
AllowAutoUpdateCache	Устанавливает режим автоматического обновления кэша. Отчет, хранящийся в кэше либо сессии сервера, будет автоматически пересохранен через определенный промежуток времени при бездействии дизайнера (примерно каждые 3 минуты). По умолчанию свойство имеет значение **true**.

© 2003-2026 Stimulsoft

UseRelativeUrls	Устанавливает режим вьювера, в котором для AJAX запросов на сервер используются относительные ссылки. По умолчанию свойство имеет значение **true**.
PortNumber	Получает или задает значение, которое указывает номер порта для использования в URL. Значение 0 определяет автоматическое обнаружение (установлено по умолчанию). Значение -1 удаляет номер порта.
PassQueryParametersForResources	Включает передачу всех параметров URL запроса при формировании ссылок на ресурсы вьювера. При значении **false** для запроса ресурсов вьювера используются только необходимые параметры, что способствует более корректной работе кэша браузера. По умолчанию свойство имеет значение **true**.
PassQueryParametersToReport	Включает использование всех параметров URL запроса в качестве значений одноименных переменных в отчете. По умолчанию свойство имеет значение **false**.
PassFormValues	Включает передачу значений POST-формы на сторону клиента, если эти значения требуется использовать в событиях вьювера. При включенном свойстве, вспомогательный метод **GetFormValues()** будет возвращать коллекцию параметров формы. По умолчанию свойство имеет значение **false**.
ShowServerErrorPage	Включает отображение HTML-страницы с детализацией ошибки, возникшей на стороне сервера. При включенном свойстве детализация ошибки будет отображена в окне вьювера. Если свойство выключено, будет отображен только числовой код ошибки и краткий текст ошибки в диалоговом окне. По умолчанию свойство имеет значение **true**.

© 2003-2026 Stimulsoft

UseCompression	Включает сжатие запросов вьювера в GZip поток. Это позволяет уменьшить объем интернет-трафика, но немного замедляет работу вьювера. По умолчанию свойство имеет значение **false**.
UseCacheForResources	Включает кэширование на стороне сервера ресурсов компонента, таких как скрипты, стили, изображения. Позволяет ускорить загрузку компонента и уменьшить нагрузку на сервер при большом количестве клиентов. По умолчанию свойство имеет значение **true**.
UseLocalizedCache	Предоставляет возможность использовать различные кэши в зависимости от выбранной локализации. По умолчанию свойство имеет значение **false**.
AllowLoadingCustomFontsToClientSide	Позволяет передавать пользовательские шрифты на клиентскую сторону и преобразовывать их в CSS стиль для корректного отображения текста на стороне HTML с заданным шрифтом. По умолчанию свойство имеет значение **false**.

Внешний вид

Наименование	Описание
Theme	Задает тему оформления вьювера. Список доступных тем оформления находится в перечислении **StiViewerTheme**. По умолчанию установлено значение **Office2022WhiteBlue**.
IconSet	Предоставляет возможность установить набор иконок: › StiWebUIIconSet.Auto (значение по умолчанию) - устанавливает набор иконок автоматически. Для тем **Office2022** -

© 2003-2026 Stimulsoft

	используется набор иконок в стиле **Monoline**, для тем **Office2013** - используется набор иконок в стиле **Regular**. › StiWebUIIconSet.Monoline - устанавливает набор иконок в стиле **Monoline**; › StiWebUIIconSet.Regular - устанавливает набор иконок в стиле **Regular**.
CustomCss	Задает путь к CSS файлу стилей вьювера. Если данное свойство установлено, то стандартные стили выбранной темы загружены не будут. По умолчанию установлено пустое значение.
Localization	Задает путь к XML файлу локализации. Путь может быть абсолютным либо относительным. По умолчанию используется английская локализация, которая встроена во вьювер и не требует дополнительных XML файлов.
BackgroundColor	Задает цвет подложки вьювера. По умолчанию свойство имеет значение **White**.
PageBorderColor	Задает цвет границ страницы отчета. По умолчанию свойство имеет значение **Gray**.
RightToLeft	Устанавливает режим отображения **Right to Left** для элементов управления вьювера. По умолчанию свойство имеет значение **false**.
FullScreenMode	Устанавливает полноэкранный режим отображения вьювера. По умолчанию свойство имеет значение **false**.
ScrollbarsMode	Устанавливает режим отображения отчета с полосами прокрутки. По умолчанию свойство имеет значение **false**.
OpenLinksWindow	Задает целевое окно для открытия ссылок, содержащихся в отчете. По умолчанию свойство имеет значение **Blank** (новое окно).
OpenExportedReportWindow	Задает целевое окно для открытия файла экспорта из вьювера. По умолчанию свойство имеет значение **Blank** (новое окно).

© 2003-2026 Stimulsoft

DesignWindow	Задает целевое окно для открытия дизайнера отчетов. По умолчанию свойство имеет значение **Self** (текущее окно).
ShowTooltips	Включает отображение подсказок для элементов управления вьювера при наведении мыши. По умолчанию свойство имеет значение **true**.
ShowTooltipsHelp	Включает отображение ссылки на онлайн документацию в подсказках для элементов управления вьювера. По умолчанию свойство имеет значение **true**.
ShowDialogsHelp	Предоставляет возможность отображать или не отображать кнопку вызова справки в различных меню. По умолчанию, установлено значение **true**.
PageAlignment	Задает позицию страницы отчета в окне вьювера. Может принимать одно из указанных ниже значений перечисления **StiContentAlignment**: › **Left** – страница будет пристыкована к левому краю; › **Center** – страница будет центрирована (значение по умолчанию); › **Right** – страница будет пристыкована к правому краю.
ShowPageShadow	Включает отображение тени для страниц отчета. По умолчанию свойство имеет значение **true**.
BookmarksPrint	Включает печать закладок отчета на принтере (помимо самого отчета). По умолчанию свойство имеет значение **false**.
BookmarksTreeWidth	Устанавливает ширину панели закладок в пикселях. По умолчанию ширина имеет значение 180 пикселей.
ParametersPanelPosition	Задает позицию панели параметров отчета.

© 2003-2026 Stimulsoft

	Может принимать одно из указанных ниже значений перечисления **StiParametersPanelPosition**: › **Top** – панель будет пристыкована к верхнему краю (значение по умолчанию); › **Left** – панель будет пристыкована к левому краю.
ParametersPanelMaxHeight	Устанавливает максимальную высоту панели параметров в пикселях. По умолчанию максимальная высота имеет значение 300 пикселей.
ParametersPanelColumnsCount	Устанавливает количество колонок для отображения параметров отчета. По умолчанию разбиение производится на 2 колонки.
ParametersPanelSortDataItems	Устанавливает или выключает режим сортировки значений переменной. По умолчанию, опция установлена в значение **false**.
ParametersPanelDateFormat	Устанавливает формат даты и времени для переменных соответствующего типа на панели параметров. По умолчанию используется формат даты и времени, установленный браузером.
InterfaceType	Устанавливает тип используемого интерфейса вьювера. Может принимать одно из указанных ниже значений перечисления **StiInterfaceType**: › **Auto** – тип интерфейса вьювера будет выбран автоматически в зависимости от используемого устройства (значение по умолчанию); › **Mouse** – принудительное использование стандартного интерфейса для управления вьювером при помощи мыши;

© 2003-2026 Stimulsoft

	› **Touch** – принудительное использование Touch интерфейса для управления вьювером при помощи сенсорного экрана монитора, в этом режиме элементы интерфейса вьювера имеют увеличенные размеры для удобства управления; › **Mobile** – принудительное использование Mobile интерфейса для управления вьювером при помощи экрана смартфона, в этом режиме интерфейс вьювера имеет упрощенный вид и адаптирован для управления при помощи мобильного устройства.
AllowMobileMode	Включает или выключает возможность отображения отчета или дашборда в мобильном режиме. Если опция установлена в значение **false**, то мобильный режим просмотра не будет использоваться ни при каких обстоятельства. Если опция установлена в значение **true**, то мобильный режим просмотра будет использоваться запуске вьювера на мобильных устройствах. По умолчанию, опция установлена в значение **true**.
ChartRenderType	Устанавливает режим отображения диаграмм на странице отчета. Может принимать одно из указанных ниже значений перечисления **StiChartRenderType**: › **Image** – диаграммы отображаются в виде статичных изображений; › **Vector** – диаграммы отображаются в векторном режиме в виде SVG объекта; › **AnimatedVector** - диаграммы отображаются в векторном режиме в виде SVG объекта, элементы диаграммы появляются с анимацией (значение по умолчанию).

© 2003-2026 Stimulsoft

ReportDisplayMode	Устанавливает режим экспорта для отображения страниц отчета. Может принимать одно из указанных ниже значений перечисления **StiReportDisplayMode**: › **FromReport** - режим экспорта элементов отчета определяется из настроек шаблона - Div или Table; › **Table** – элементы отчета экспортируются с использованием HTML таблиц (значение по умолчанию); › **Div** – элементы отчета экспортируются с использованием DIV разметки; › **Span** - элементы отчета экспортируются с использованием SPAN разметки.
DatePickerFirstDayOfWeek	Устанавливает первый день недели для элемента выбора даты. Может принимать одно из указанных ниже значений перечисления **StiFirstDayOfWeek**: › **Auto** – автоматическое определение первого дня недели из настроек браузера (значение по умолчанию); › **Monday** – первым днем недели является понедельник; › **Sunday** – первым днем недели является воскресенье.
DatePickerIncludeCurrentDayForRanges	Предоставляет возможность включать или не включать текущий день в диапазон значений элемента **Выбор даты (Date Picker)**. По умолчанию, опция установлена в значение **false**, т.е. текущий день не включается в диапазон значений элемента.
AllowTouchZoom	Включает возможность изменения масштаба страницы отчета специальным двух пальцевым жестом (Pinch to Zoom) на сенсорном экране устройства. По умолчанию свойство имеет значение **true**.

© 2003-2026 Stimulsoft

ShowReportIsNotSpecifiedMessage	Предоставляет возможность отображать или не отображать уведомление, если отчет не задан. По умолчанию свойство имеет значение **true**.
ImagesQuality	Получает или задает качество изображения, которое будет использоваться на странице вьювера. Может принимать значения: › **StiImagesQuality.Low** - низкое качество, используется для ускорения загрузки отчета и экономии памяти; › **StiImagesQuality.Normal** - обычное качество, подходит для большинства ситуаций, установлено по умолчанию; › **StiImagesQuality.High** - высокое качество, используется для мониторов высокой четкости, но может замедлять загрузку страниц.
PrintToPdfMode	Устанавливает режим печати в PDF документ. Может принимать значения: › **StiPrintToPdfMode.Hidden** - скрытый режим печати (по умолчанию); › **StiPrintToPdfMode.Popup** - документ PDF будет отображет перед печатью во всплывающем окне.
CombineReportPages	Предоставляет возможность объединить обработанные страницы шаблона отчета в один шаблон или представить каждую страницу шаблона отдельной вкладкой во вьювере. По умолчанию, опция установлена в значение **false**, т.е. каждая страница шаблона отчета будет представлена отдельной вкладкой во вьювере.
SaveMenuImageSize	Предоставляет возможность использовать маленький размер для иконок в меню **Сохранить (Save)**. По умолчанию свойство имеет значение **true**.

© 2003-2026 Stimulsoft

AllowPropagationEvents	Предоставляет возможность разрешить распространение событий нажатия клавиш, когда вьювер отчетов не в фокусе. По умолчанию свойство имеет значение **true**.
DashboardFilterElementItemHeight	Предоставляет возможность изменить высоту флажка в элементе дашборда **Список (List Box)**.

Панель инструментов

Наименование	Описание
ShowToolbar	Включает отображение панели инструментов вьювера при просмотре отчетов или панели индикаторов (дашбордов). По умолчанию свойство имеет значение **true**.
ToolbarDisplayMode	Задает режим отображения панели инструментов вьювера при просмотре отчетов. Может принимать одно из указанных ниже значений перечисления **StiToolbarDisplayMode**: › **Simple** – простой режим отображения, все элементы управления расположены на одной панели управления (значение по умолчанию); › **Separated** – раздельный режим отображения, панель управления разделена на верхнюю и нижнюю.
ToolbarBackgroundColor	Задает цвет панели инструментов вьювера при просмотре отчетов. По умолчанию используется цветовое оформление выбранной темы.
ToolbarBorderColor	Задает цвет границ панели инструментов вьювера при просмотре отчетов. По умолчанию используется цветовое оформление выбранной темы.

© 2003-2026 Stimulsoft

ToolbarFontColor	Задает цвет текста для панели инструментов и меню вьювера при просмотре отчетов. По умолчанию используется цветовое оформление выбранной темы.
ToolbarFontFamily	Задает шрифт для панели инструментов и меню вьювера при просмотре отчетов. По умолчанию используется шрифт выбранной темы.
ToolbarAlignment	Устанавливает режим выравнивания элементов управления на панели инструментов вьювера при просмотре отчетов. Может принимать одно из указанных ниже значений перечисления **StiContentAlignment**: › **Left** – элементы будут пристыкованы к левому краю; › **Center** – элементы будут располагаться по центру панели; › **Right** – элементы будут пристыкованы к правому краю; › **Default** – выравнивание зависит от свойства RightToLeft (значение по умолчанию)
ShowButtonCaptions	Включает отображение текста кнопок панели инструментов вьювера при просмотре отчетов. По умолчанию свойство имеет значение **true**.
ShowPrintButton	Включает отображение кнопки **Печать (Print)** на панели инструментов вьювера при просмотре отчетов. По умолчанию свойство имеет значение **true**.
ShowOpenButton	Включает отображение кнопки **Открыть (Open)** на панели инструментов вьювера при просмотре отчетов или панели индикаторов (дашбордов). По умолчанию свойство имеет значение **true**.

© 2003-2026 Stimulsoft

ShowSaveButton	Включает отображение кнопки **Сохранить (Save)** на панели инструментов вьювера при просмотре отчетов или панели индикаторов (дашбордов). По умолчанию свойство имеет значение **true**.
ShowSendEmailButton	Включает отображение кнопки **Отправиль Email (Send Email)** на панели инструментов вьювера при просмотре отчетов. По умолчанию свойство имеет значение **false**. Также, необходимо добавить обработчик события OnEmailReport.
ShowFindButton	Включает отображение кнопки **Поиск (Find)** на панели инструментов вьювера при просмотре отчетов. По умолчанию свойство имеет значение **true**.
ShowBookmarksButton	Включает отображение кнопки **Закладки (Bookmarks)** на панели инструментов вьювера при просмотре отчетов. По умолчанию свойство имеет значение **true**. Если указанная кнопка скрыта, то панель закладок не будет отображена даже при наличии закладок в отчете.
ShowParametersButton	Включает отображение кнопки **Параметры (Parameters)** на панели инструментов вьювера при просмотре отчетов или панели индикаторов (дашбордов). По умолчанию свойство имеет значение **true**. Если указанная кнопка скрыта, то панель параметров не будет отображена даже при наличии параметров в отчете.
ShowResourcesButton	Включает отображение кнопки **Ресурсы (Resources)** на панели инструментов вьювера при просмотре отчетов. По умолчанию свойство имеет значение **true**. Если указанная кнопка скрыта, то панель ресурсов не будет отображена.
ShowEditorButton	Включает отображение кнопки **Редактор**

© 2003-2026 Stimulsoft

	(Editor) на панели инструментов вьювера. По умолчанию свойство имеет значение **true**.
ShowFullScreenButton	Включает отображение кнопки **Полноэкранный просмотр (Full Screen)** на панели инструментов вьювера при просмотре отчетов или панели индикаторов (дашбордов). По умолчанию свойство имеет значение **true**.
ShowFirstPageButton	Включает отображение кнопки **Первая страница (First Page)** на панели инструментов вьювера при просмотре отчетов. По умолчанию свойство имеет значение **true**.
ShowPreviousPageButton	Включает отображение кнопки **Предыдущая страница (Previous Page)** на панели инструментов вьювера при просмотре отчетов. По умолчанию свойство имеет значение **true**.
ShowCurrentPageControl	Включает отображение индикатора текущей страницы отчета при просмотре отчетов. По умолчанию свойство имеет значение **true**.
ShowNextPageButton	Включает отображение кнопки **Следующая страница (Next Page)** на панели инструментов вьювера при просмотре отчетов. По умолчанию свойство имеет значение **true**.
ShowLastPageButton	Включает отображение кнопки **Последняя страница (Last Page)** на панели инструментов вьювер при просмотре отчетов. По умолчанию свойство имеет значение **true**.
ShowZoomButton	Включает отображение кнопки выбора масштаба отчета при просмотре отчетов. По умолчанию свойство имеет значение **true**.
ShowViewModeButton	Включает отображение кнопки выбора режима отображения страниц отчета при

© 2003-2026 Stimulsoft

	просмотре отчетов. По умолчанию свойство имеет значение **true**.
ShowDesignButton	Включает отображение кнопки **Дизайн (Design)** на панели инструментов вьювера при просмотре отчетов или панели индикаторов (дашбордов). По умолчанию свойство имеет значение **false**.
ShowAboutButton	Включает отображение кнопки **О программе (About)** на панели инструментов вьювера при просмотре отчетов. По умолчанию свойство имеет значение **true**.
ShowPinToolbarButton	Включает отображение кнопки **Прикрепить панель (Pin Toolbar)** на панели инструментов вьювера при просмотре отчетов. Кнопка доступна только в мобильном режиме вьювера. По умолчанию свойство имеет значение **true**.
ShowRefreshButton	Включает или выключает отображение кнопки **Обновить (Refresh)** на панели инструментов вьювера при просмотре панелей индикаторов (дашбордов). По умолчанию свойство имеет значение **true**.
PrintDestination	Устанавливает режим печати отчета. Может принимать одно из указанных ниже значений перечисления **StiPrintDestination**: › **Default** – будет отображено меню с предоставлением выбора режима печати (значение по умолчанию); › **Pdf** – печать будет осуществлена в формате PDF; › **Direct** – печать будет осуществлена в формате HTML непосредственно на принтер, будет отображен системный диалог печати; › **PopupWindow** – печать будет осуществлена в формате HTML через всплывающее окно предварительного

© 2003-2026 Stimulsoft

	просмотра отчета.
ViewMode	Устанавливает режим отображения страниц отчета. Может принимать одно из указанных ниже значений перечисления **StiWebViewMode**: › **SinglePage** – отображается одна страница отчета, выбранная на панели инструментов вьювера (значение по умолчанию); › **Continuous** – отображаются все страницы отчета в виде ленты; › **MultiplePages** – отображаются все страницы отчета в виде таблицы.
Zoom	Задает масштаб отображения страниц отчета. По умолчанию установлен масштаб 100 процентов. Допускаются значения от 10 до 500 процентов. Также допускается устанавливать одно из следующих значений: › **StiZoomMode.PageWidth** – при запуске вьювера будет установлен масштаб, необходимый для отображения отчета по ширине страницы; › **StiZoomMode.PageHeight** – при запуске вьювера будет установлен масштаб, необходимый для отображения отчета по высоте страницы.
MenuAnimation	Включает анимацию при появлении/скрытии меню вьювера. По умолчанию свойство имеет значение **true**.
ShowMenuMode	Устанавливает режим отображения меню вьювера. Может принимать одно из указанных ниже значений перечисления **StiShowMenuMode**: › **Click** – отображение по клику мыши (значение по умолчанию); › **Hover** – отображение по наведении

© 2003-2026 Stimulsoft

	курсора мыши.
AutoHideToolbar	Включает автоматическое сворачивание панели инструментов вьювера. Свойство работает только в мобильном режиме вьювера. По умолчанию свойство имеет значение **false**.

Примеры

Ознакомтесь с примером как добавить пользовательскую кнопку на панель инструментов вьювера.

Экспорт отчета

Наименование	Описание
DefaultExportSettings	Данная группа свойств предоставляет возможность задать настройки экспортов по умолчанию для каждого типа экспорта. Эти настройки будут применены к диалоговым окнам экспортов при запуске вьювера, либо к отчету или панели индикаторов, в случае отключения отображения диалоговых окон экспортов.
StoreExportSettings	Включает режим сохранения выбранных настроек в диалоговых окнах экспорта. Настройки будут храниться в cookies браузера. По умолчанию свойство имеет значение **true**.
ShowExportDialog	Включает отображение диалогового окна параметров экспорта. Если свойство имеет значение **false**, то экспортирование будет производиться с заданными по умолчанию настройками. По умолчанию свойство имеет значение **true**.

© 2003-2026 Stimulsoft

ShowExportToDocument	Включает отображение пункта меню экспорта **Файл документа (Document File)** при просмотре отчетов, и пункт **Снимок отчета (Report Snapshot)** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToPdf	Включает отображение пункта меню экспорта **Adobe PDF файл (Adobe PDF File)** при просмотре отчетов, и пункт **Adobe PDF** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToXps	Включает отображение пункта меню экспорта **Microsoft XPS файл (Microsoft XPS File)**. По умолчанию свойство имеет значение **false**.
ShowExportToPowerPoint	Включает отображение пункта меню экспорта **Microsoft PowerPoint 2007/2010 файл (Microsoft PowerPoint 2007/2010 File)**. По умолчанию свойство имеет значение **true**.
ShowExportToHtml	Включает отображение пункта меню экспорта **HTML файл (HTML File)**. По умолчанию свойство имеет значение **true**.
ShowExportToHtml5	Включает отображение пункта меню экспорта **HTML5 файл (HTML5 File)**. По умолчанию свойство имеет значение **true**.
ShowExportToMht	Включает отображение пункта меню экспорта **MHT Web архив (MHT Web Archive)**. По умолчанию свойство имеет значение **true**.
ShowExportToText	Включает отображение пункта меню экспорта **Text файл (Text File)**. По умолчанию свойство имеет значение **true**.
ShowExportToRtf	Включает отображение пункта меню экспорта **Rich Text файл (Rich Text File)**. По умолчанию свойство имеет значение **true**.
ShowExportToWord	Включает отображение пункта меню

© 2003-2026 Stimulsoft

	экспорта **Microsoft Word 2007/2010 файл (Microsoft Word 2007/2010 File)**. По умолчанию свойство имеет значение **true**.
ShowExportToOpenDocumentWriter	Включает отображение пункта меню экспорта **OpenDocument Writer файл (OpenDocument Writer File)**. По умолчанию свойство имеет значение **true**.
ShowExportToExcelBiff	Включает отображение пункта меню экспорта **Microsoft Excel файл (Microsoft Excel File)**. По умолчанию свойство имеет значение **true**.
ShowExportToExcelXml	Включает отображение пункта меню экспорта **Microsoft Excel Xml файл (Microsoft Excel Xml File)**. По умолчанию свойство имеет значение **true**.
ShowExportToExcel	Включает отображение пункта меню экспорта **Microsoft Excel 2007/2010 файл (Microsoft Excel 2007/2010 File)** при просмотре отчетов, и пункт **Microsoft Excel** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToOpenDocumentCalc	Включает отображение пункта меню экспорта **OpenDocument Calc файл (OpenDocument Calc File)**. По умолчанию свойство имеет значение **true**.
ShowExportToCsv	Включает отображение пункта меню экспорта **CSV файл (CSV File)**. По умолчанию свойство имеет значение **true**.
ShowExportToDbf	Включает отображение пункта меню экспорта **DBF файл (DBF File)**. По умолчанию свойство имеет значение **true**.
ShowExportToXml	Включает отображение пункта меню экспорта **XML файл (XML File)**. По умолчанию свойство имеет значение **true**.
ShowExportToDif	Включает отображение пункта меню экспорта **Data Interchange формат (DIF)**

© 2003-2026 Stimulsoft

	файл (Data Interchange Format (DIF) File). По умолчанию свойство имеет значение **true**.
ShowExportToSylk	Включает отображение пункта меню экспорта **Symbolic Link (SYLK) файл (Symbolic Link (SYLK) File)**. По умолчанию свойство имеет значение **true**.
ShowExportToJson	Включает отображение пункта меню экспорта **JSON файл (JSON File)**. По умолчанию свойство имеет значение **true**.
ShowExportToImageBmp	Включает отображение пункта меню экспорта **BMP рисунок (BMP Image)** при просмотре отчетов, и пункт **BMP** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToImageGif	Включает отображение пункта меню экспорта **GIF рисунок (GIF Image)** при просмотре отчетов, и пункт **GIF** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToImageJpeg	Включает отображение пункта меню экспорта **JPEG рисунок (JPEG Image)** при просмотре отчетов, и пункт **JPEG** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToImagePcx	Включает отображение пункта меню экспорта **PCX рисунок (PCX Image)** при просмотре отчетов, и пункт **PCX** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToImagePng	Включает отображение пункта меню экспорта **PNG рисунок (PNG Image)** при просмотре отчетов, и пункт **PNG** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToImageTiff	Включает отображение пункта меню экспорта **TIFF рисунок (TIFF Image)** при

© 2003-2026 Stimulsoft

	просмотре отчетов, и пункт **TIFF** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToImageMetafile	Включает отображение пункта меню экспорта **Windows Metafile (Windows Metafile)** при просмотре отчетов, и пункт **EMF** при просмотре панелей индикаторов.. По умолчанию свойство имеет значение **true**.
ShowExportToImageSvg	Включает отображение пункта меню экспорта **Scalable Vector Graphics (SVG) файл (Scalable Vector Graphics (SVG) File)** при просмотре отчетов, и пункт **SVG** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToImageSvgz	Включает отображение пункта меню экспорта **Compressed SVG (SVGZ) файл (Compressed SVG (SVGZ) File)** при просмотре отчетов, и пункт **SVGZ** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowOpenAfterExport	Предоставляет возможность отображать или не отображать параметр **Открыть после экспорта (Open After Export)** в меню настроек экспорта. По умолчанию свойство имеет значение **true**.

Отправка отчета по Email

Наименование	Описание
ShowEmailDialog	Включает отображение диалогового окна параметров отправки отчета по Email. Если диалоговое окно отключено, то отправка по Email будет производиться с настройками, заданными на стороне сервера в событии **OnEmailReport**. По умолчанию свойство имеет значение **true**.

© 2003-2026 Stimulsoft

ShowEmailExportDialog	Включает отображение диалогового окна параметров экспорта при отправке Email. Если свойство имеет значение **false**, то экспортирование будет производиться с заданными по умолчанию настройками. По умолчанию свойство имеет значение **true**.
DefaultEmailAddress	Устанавливает Email получателя по умолчанию, т.е. адрес на который будет отправлено письмо с вложенным отчетом.
DefaultEmailSubject	Устанавливает тему (заголовок) электронного письма по умолчанию.
DefaultEmailMessage	Устанавливает сообщение (текст) электронного письма по умолчанию.

3.2 HTML5 дизайнер

YouTube

Посмотрите видеоуроки по работе с компонентом ASP.NET HTML5 Designer . Подписывайтесь на канал Stimulsoft и узнайте первыми о новых видеоуроках. Вопросы и предложения оставляйте в комментариях к видео.

Примеры

Ознакомьтесь с примерами по работе с компонентом ASP.NET HTML5 Designer GitHub. Все примеры представляют собой отдельные проекты, сгруппированные в один solution для Visual Studio.

Компонент **HTML5 Designer** (**StiWebDesigner**) предназначен для редактирования отчетов в окне браузера. При этом не требуется устанавливать у клиента .NET Framework, ActiveX компоненты или какие-либо специальные плагины. Все, что необходимо – это любой современный Web-браузер.

При помощи **HTML5 Designer** можно создавать, редактировать, сохранять,

© 2003-2026 Stimulsoft

просматривать отчеты на любом компьютере с любой установленной операционной системой. Для отчетов также предусмотрена возможность печати. Так как дизайнер использует только HTML и JavaScript технологии, он может быть запущен на устройствах, где нет поддержки Flash или Silverlight – планшеты, смартфоны. Также дизайнер поддерживает Touch интерфейс, который автоматически включается при использовании устройств с сенсорным экраном.

Компонент **HTML5 Designer** использует технологию AJAX для выполнения всех действий над редактируемым отчетом, что позволяет избавиться от перезагрузки всей страницы, а также экономить Web-трафик и повысить скорость работы.

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для использования **HTML5 Designer** в Web-проекте, необходимо установить NuGet пакет Stimulsoft.Reports.Web:

- Выберите пункт "Manage NuGet Packages..." в контекстном меню проекта;
- На вкладке Browse, в строке поиска, укажите Stimulsoft.Reports.Web;
- Выделите элемент, определите версию пакета и нажмите кнопку **Установить (Install)**. При обновлении пакета, следует нажать кнопку **Обновить (Update)**.

Если по какой-либо причине это невозможно, необходимо добавить в проект указанные ниже сборки:

- Stimulsoft.Base.dll
- Stimulsoft.Report.dll
- Stimulsoft.Report.Check.dll
- Stimulsoft.Report.Helper.dll
- Stimulsoft.Report.Web.dll
- Stimulsoft.Report.WebDesign.dll

Для того, чтобы добавить возможности создания и редактирования дашбордов

© 2003-2026 Stimulsoft

в Web-проекте, установить NuGet пакет Stimulsoft.Dashboards.Web (данный пакет связан с пакетом Stimulsoft.Reports.Web, при его отсутствии он будет установлен автоматически):

- Выберите пункт "Manage NuGet Packages..." в контекстном меню проекта;
- На вкладке Browse, в строке поиска, укажите Stimulsoft.Dashboards.Web;
- Выделите элемент, определите версию пакета и нажмите кнопку **Установить (Install)**. При обновлении пакета, следует нажать кнопку **Обновить (Update)**.

Если по какой-либо причине это невозможно, необходимо дополнительно добавить в проект указанные ниже сборки:

- Stimulsoft.Dashboard.dll
- Stimulsoft.Dashboard.Drawing.dll
- Stimulsoft.Dashboard.Export.dll

- Как это работает
- Дополнительные возможности просмотра
- Активация
- Время ожидания
- Редактирование отчета и дашборда
- Локализация дизайнера
- Создание нового отчета и нового дашборда
- Использование тем
- Сохранение отчета и дашборда
- Кэширование
- Просмотр
- События
- Настройки

3.2.1 Как это работает

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

© 2003-2026 Stimulsoft

Для запуска дизайнера требуется разместить на ASPX странице компонент **StiWebDesigner**, задать ему необходимые свойства и, если требуется, установить необходимые обработчики событий. При запуске дизайнера отчетов происходят следующие действия:

- .NET компонент генерирует HTML и JavaScript код, необходимый для отображения и работы дизайнера;
- По окончании вывода компонента запускается JavaScript метод, который запрашивает на стороне сервера шаблон отчета, и отображает его в окне редактирования;
- Различные действия, такие как предварительный просмотр отчета или дашборда, сохранение шаблона отчета, экспорт отчета, применение параметров, сортировка и детализация отчета, вызывают определенное событие на стороне сервера, в котором можно выполнить необходимые манипуляции с отчетом.

3.2.2 Активация

YouTube

Посмотрите видеоуроки по активации компонента ASP.NET HTML5 Designer . Подписывайтесь на канал Stimulsoft и узнайте первыми о новых видеоуроках. Вопросы и предложения оставляйте в комментариях к видео.

После приобретения продукта Stimulsoft вам необходимо активировать лицензию для используемых компонентов. Это можно выполнить указав лицензионный ключ или загрузив файл с лицензионным ключом. Ниже представлен пример активации компонента **StiWebDesigner**.

© 2003-2026 Stimulsoft

Default.aspx.cs

```
...
public partial class _Default : Page
{
   static _Default()
   {
      //Activation with using license code
      Stimulsoft.Base.StiLicense.Key = "Your activation code...";

      //Activation with using license file
      var path = HttpContext.Current.Server.MapPath("license.key");
      Stimulsoft.Base.StiLicense.LoadFromFile(path);
   }
}
...
```

Получить лицензионный ключ или скачать файл с лицензионным ключом, вы можете в личном кабинете пользователя. Для авторизации в кабинете используйте имя пользователя и пароль, указанные при покупке подписки продукта.

3.2.3 Редактирование отчета и дашборда

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для редактирования шаблона отчета, необходимо добавить на ASPX страницу компонент **StiWebDesigner** и присвоить ему загруженный шаблон отчета.

Default.aspx

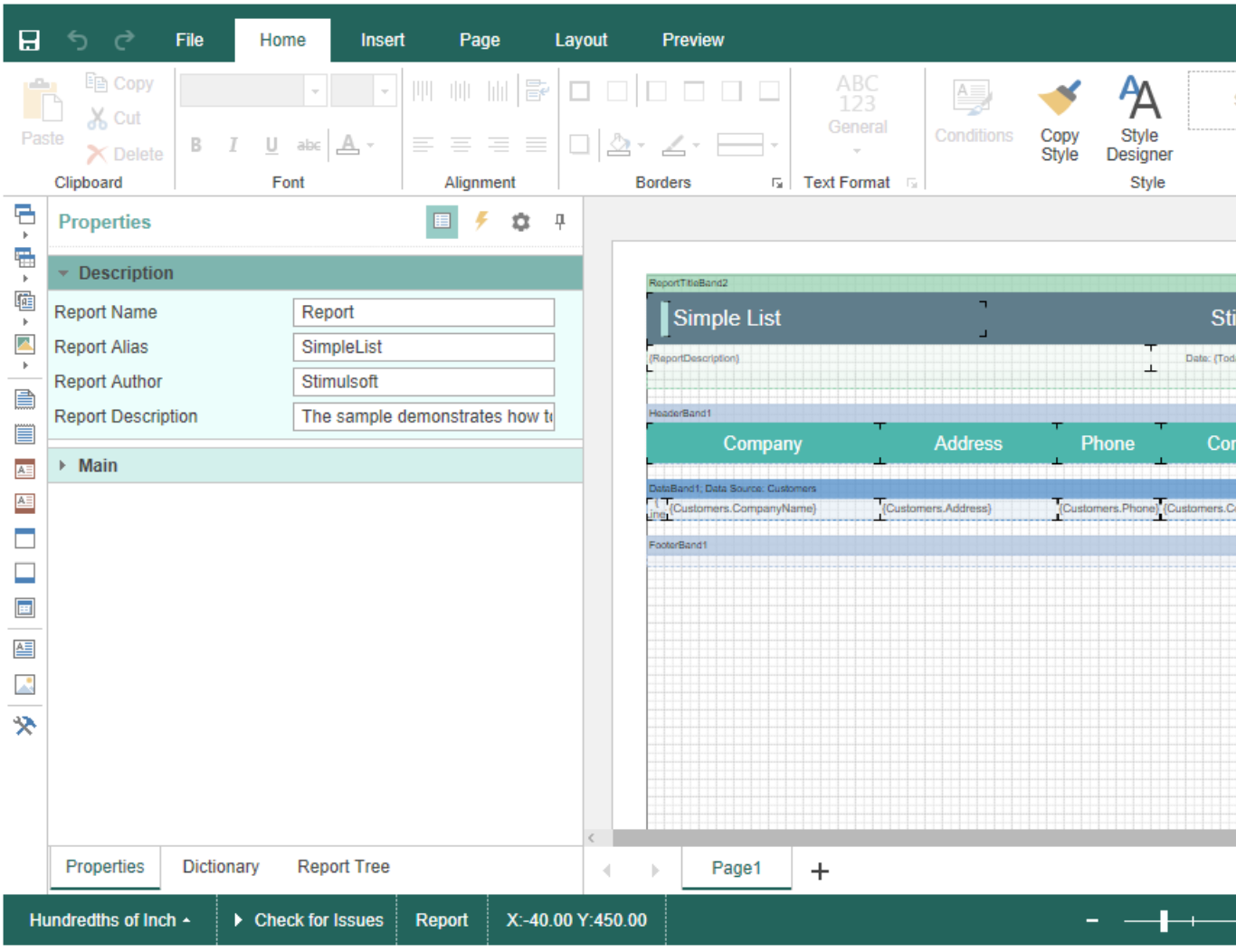
```
...
<cc1:StiWebDesigner ID="StiWebDesigner1" runat="server">
</cc1:StiWebDesigner>
...
```

© 2003-2026 Stimulsoft

Default.aspx.cs

```
...
protected void Page_Load(object sender, EventArgs e)
{
   StiReport report = new StiReport();
   report.Load(Server.MapPath("Reports/SimpleList.mrt"));
   //report.Load(Server.MapPath("Reports/Dashboard.mrt"));

   StiWebDesigner1.Report = report;
}
...
```



Так же **HTML5 Designer** имеет специальное событие **OnGetReport**, которое можно использовать для присвоения шаблона отчета. В этом случае загрузку отчета необходимо осуществлять в обработчике события.

© 2003-2026 Stimulsoft

Default.aspx

```
...
<cc1:StiWebDesigner ID="StiWebDesigner1" runat="server"
   OnGetReport="StiWebDesigner1_GetReport">
</cc1:StiWebDesigner>
...
```

Default.aspx.cs

```
...
protected void StiWebDesigner1_GetReport(object sender,
StiReportDataEventArgs e)
{
   StiReport report = new StiReport();
   report.Load(Server.MapPath("Reports/SimpleList.mrt"));

   e.Report = report;
}
...
```

Информация

Событие **OnGetReport** будет вызвано независимо от того, был ли присвоен отчет ранее или нет. Если отчет уже присвоен дизайнеру, то в аргументах события свойство **e.Report** будет содержать загруженный объект отчета. Его можно изменить либо присвоить новый отчет.

По умолчанию для редактирования отчета **HTML5 Designer** использует всю доступную область окна браузера. Для того, чтобы отобразить компонент в определенной позиции HTML страницы с заданными размерами, достаточно установить его ширину и высоту, используя свойства **Width** и **Height**.

Default.aspx

```
...
<cc1:StiWebDesigner ID="StiWebDesigner1" runat="server"
   Width="1000px" Height="800px">
</cc1:StiWebDesigner>
...
```

© 2003-2026 Stimulsoft

3.2.4 Создание нового отчета и нового дашборда

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для запуска дизайнера с новым отчетом не требуется никаких действий, при загрузке компонента будет новый отчет будет создан автоматически. При необходимости, можно создать новый объект отчета и предварительно загрузить для него данные, либо выполнить какие-либо другие необходимые действия.

Default.aspx

```
...
<cc1:StiWebDesigner ID="StiWebDesigner1" runat="server"
   OnGetReport="StiWebDesigner1_GetReport">
</cc1:StiWebDesigner>
...
```

Default.aspx.cs

```
...
protected void StiWebDesigner1_GetReport(object sender,
StiReportDataEventArgs e)
{
   e.Report = new StiReport();
   //var newDashboard = StiReport.CreateNewDashboard();
}
...
```

Также новый отчет можно создать, используя главное меню дизайнера. Для того, чтобы предварительно загрузить данные для нового отчета, либо выполнить какие-либо другие необходимые действия, предназначено событие **OnCreateReport**. Данное событие будет вызвано при создании нового пустого отчета из главного меню, либо при создании отчета с использованием мастера.

© 2003-2026 Stimulsoft

Default.aspx

```
...
<cc1:StiWebDesigner ID="StiWebDesigner1" runat="server"
  OnCreateReport="StiWebDesigner1_CreateReport">
</cc1:StiWebDesigner>
...
```

Default.aspx.cs

```
...
protected void StiWebDesigner1_CreateReport(object sender,
StiReportDataEventArgs e)
{
  StiReport report = new StiReport();
  //var newDashboard = StiReport.CreateNewDashboard();

  // Register data for the new report, if necessary
  DataSet data = new DataSet("Demo");
  data.ReadXml(Server.MapPath("Data/Demo.xml"));
  report.RegData(data);
  //newDashboard.RegData(data);
  report.Dictionary.Synchronize();
  //newDashboard.Dictionary.Synchronize();

  e.Report = report;
  //e.Report = newDashboard;
}
...
```

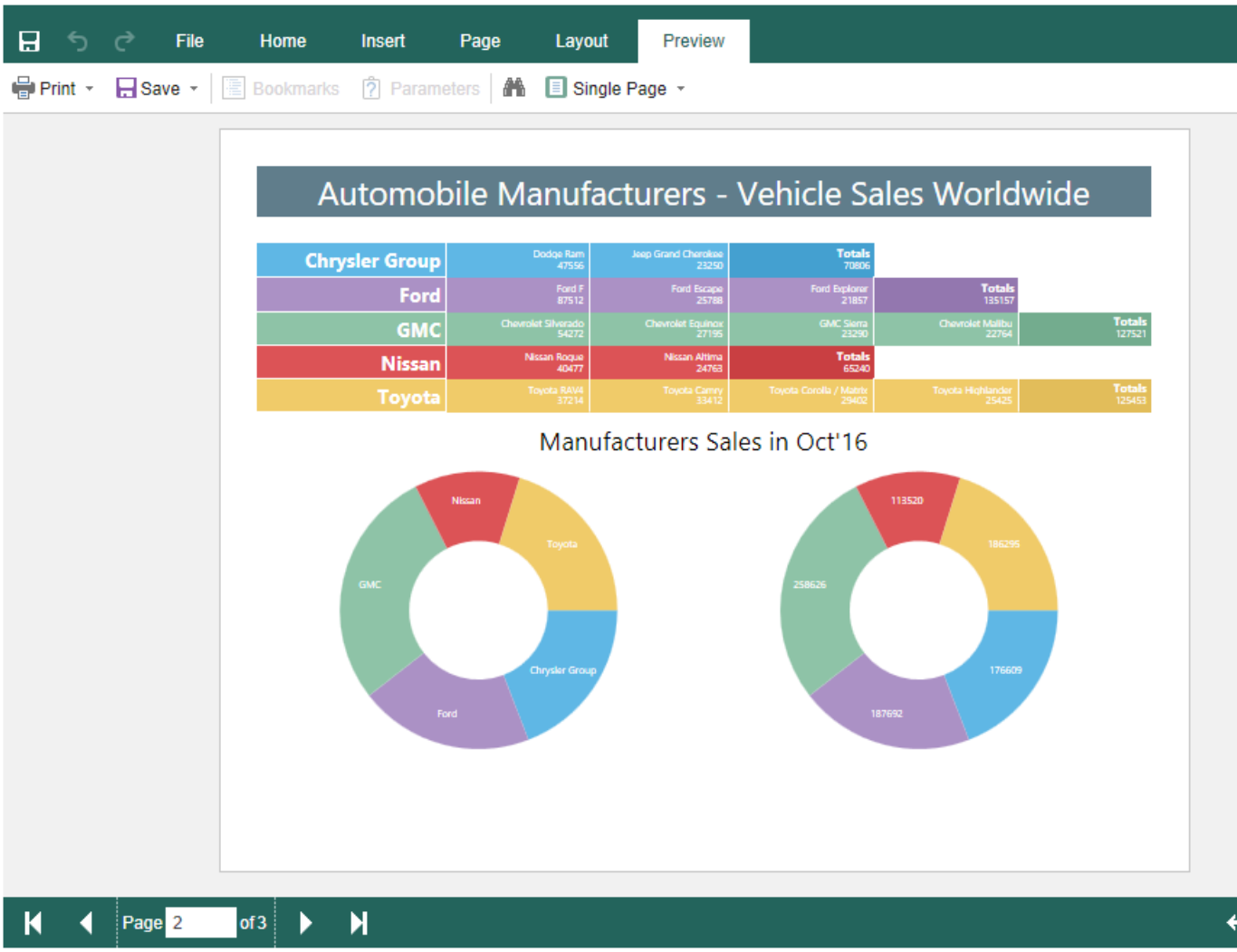
3.2.5 Просмотр

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

У компонента **HTML5 Designer** предусмотрен режим просмотра редактируемого отчета. Для этого достаточно перейти на соответствующую вкладку в окне дизайнера. Шаблон отчета будет передан на сторону сервера, построен и отображен во встроенном вьюверe.

© 2003-2026 Stimulsoft



Перед просмотром отчета предусмотрена возможность выполнить какие-либо необходимые действия, например, подключить данные для отчета. Для этого можно воспользоваться специальным событием **OnPreviewReport**, которое будет вызвано перед просмотром отчета. В аргументах события будет находиться отчет, предназначенный для просмотра. Событие **OnPreviewReport** вызывается перед подготовкой и построением отчета для просмотра, до его сохранения в кэш.

Default.aspx

```
...
<cc1:StiWebDesigner ID="StiWebDesigner1" runat="server"
   OnPreviewReport="StiWebDesigner1_PreviewReport">
</cc1:StiWebDesigner>
...
```

© 2003-2026 Stimulsoft

Default.aspx.cs

```
...
protected void StiWebDesigner1_PreviewReport(object sender,
StiReportDataEventArgs e)
{
   DataSet data = new DataSet("Demo");
   data.ReadXml(Server.MapPath("Data/Demo.xml"));
   e.Report.RegData(data);
}
...
```

При необходимости выполнить действия над отчетом непосредственно перед отображением отчета, можно воспользоваться событием **OnGetPreviewReport**, которое вызывается после запроса подготовленного отчета из кэша.

Default.aspx

```
...
<cc1:StiWebDesigner ID="StiWebDesigner1" runat="server"
   OnGetPreviewReport="StiWebDesigner1_GetPreviewReport">
</cc1:StiWebDesigner>
...
```

Default.aspx.cs

```
...
protected void StiWebDesigner1_GetPreviewReport(object sender,
StiReportDataEventArgs e)
{
   DataSet data = new DataSet("Demo");
   data.ReadXml(Server.MapPath("Data/Demo.xml"));
   e.Report.RegData(data);

   //report.IsRendered = false;
}
...
```

Информация

Так как в данном событии передается уже подготовленный для просмотра отчет, то при необходимости выполнить повторное построение, необходимо установить флаг **report.IsRendered = false**.

© 2003-2026 Stimulsoft

3.2.6 Дополнительные возможности просмотра

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Окно просмотра отчета компонента **HTML5 Designer** представляет собой полноценный интерактивный **HTML5 Viewer**, который умеет осуществлять печать отчета, экспорт отчета, поддерживает работу с параметрами отчета. Также поддерживаются интерактивные действия, такие как динамическая сортировка, детализация, сворачивание. Для использования указанных возможностей не требуется каких-либо дополнительных настроек дизайнера отчетов.

В любом из вышеперечисленных действий допускается производить манипуляции с шаблоном отчета, например, изменять его свойства и параметры, подключать новые данные для построения.

Default.aspx

```
...
<cc1:StiWebDesigner ID="StiWebDesigner1" runat="server"
   OnExportReport="StiWebDesigner1_ExportReport">
</cc1:StiWebDesigner>
...
```

Default.aspx.cs

```
...
protected void StiWebDesigner1_ExportReport(object sender,
StiReportDataEventArgs e)
{
   e.Report.ReportName = "MyReportName";
   e.Report.ReportAlias = "MyReportAlias";
}
...
```

© 2003-2026 Stimulsoft

Информация

Если какие-либо из указанных дополнительных возможностей предварительного просмотра отчета не требуются (например, экспортирование или печать отчета), можно отключить их, используя соответствующие свойства компонента **HTML5 Designer**.

3.2.7 Сохранение отчета и дашборда

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

В компоненте **HTML5 Designer** предусмотрено два варианта сохранения отчета, которые доступны в главном меню и на главной панели дизайнера: **Сохранить (Save)** и **Сохранить как (Save As)**. В свою очередь, каждый из этих вариантов сохранения имеет свои режимы и настройки.

Сохранение отчета и дашборда на стороне сервера

Для сохранения редактируемого отчета на стороне сервера необходимо воспользоваться специальным событием **OnSaveReport**, которое будет вызвано при выборе пункта **Сохранить (Save)** в главном меню, либо нажатии на кнопку **Сохранить (Save)** на главной панели дизайнера.

Default.aspx

```
...
<cc1:StiWebDesigner ID="StiWebDesigner1" runat="server"
  OnSaveReport="StiWebDesigner1_SaveReport">
</cc1:StiWebDesigner>
...
```

© 2003-2026 Stimulsoft

Default.aspx.cs

```
...
protected void StiWebDesigner1_SaveReport(object sender,
StiReportDataEventArgs e)
{
   StiReport report = e.Report;

   // Save the report template
   // ...
}
...
```

По умолчанию, после сохранения отчета дизайнер продолжает работу без отображения каких-либо сообщений. Если необходимо, после сохранения отчета предусмотрена возможность отобразить диалоговое окно с ошибкой или текстовым сообщением. Для этого предназначены специальные свойства **e.ErrorCode** и **e.ErrorString**, которые находятся в аргументах события.

Default.aspx.cs

```
...
protected void StiWebDesigner1_SaveReport(object sender,
StiReportDataEventArgs e)
{
   StiReport report = e.Report;

   // Save the report template
   // ...

   e.ErrorCode = 123;
   //e.ErrorString = "Some message after saving";
}
...
```

Если необходимо, вы можете получить доступ к оригинальному имени отчета или имени отчета из диалога сохранения.

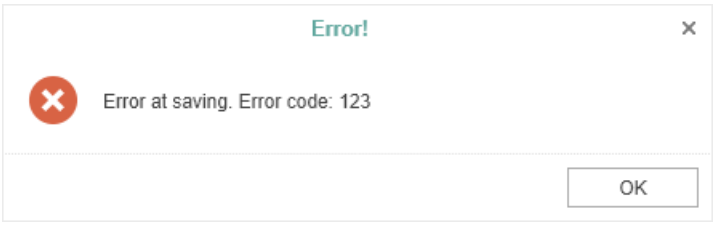
© 2003-2026 Stimulsoft

Default.aspx.cs

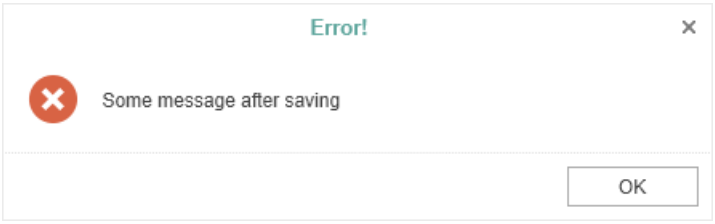
```
...
protected void StiWebDesigner1_SaveReport(object sender,
StiSaveReportEventArgs e)
{
   //Report name from the designer save dialog
   var reportName = e.FileName;

   //Original report name from properties
   var reportName = e.Report.ReportName;
}
...
```

При установке положительного целочисленного значения для свойства **e.ErrorCode**, пользователю будет отображено сообщение об ошибке сохранения отчета и код ошибки, где код ошибки – установленное целочисленное значение.



При установке строкового значения для свойства **e.ErrorString**, будет отображено диалоговое с указанным текстом. Текст может содержать как сообщение об ошибке сохранения либо предупреждение, так и любое другое сообщение.

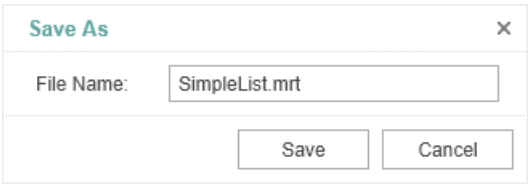


Сохранение отчета и дашборда на стороне клиента

Для сохранения редактируемого отчета на стороне клиента в виде файла не требуется дополнительных настроек дизайнера. Достаточно выбрать пункт

© 2003-2026 Stimulsoft

главного меню **Сохранить как (Save As)**, при нажатии на который отобразится диалог сохранения файла. В данном диалоге можно изменить имя файла отчета, после чего файл будет сохранен на локальном диске компьютера.



Компонент **HTML5 Designer** предоставляет возможность изменить поведение указанного варианта сохранения, для этого в дизайнере предусмотрено специальное событие **OnSaveReportAs**. При использовании указанного события сохранение отчета будет происходить на стороне сервера, работа данного события будет аналогична событию **OnSaveReport**.

Default.aspx

```
...
<cc1:StiWebDesigner ID="StiWebDesigner1" runat="server"
   OnSaveReportAs="StiWebDesigner1_SaveReportAs">
</cc1:StiWebDesigner>
...
```

Default.aspx.cs

```
...
protected void StiWebDesigner1_SaveReportAs(object sender,
StiReportDataEventArgs e)
{
   StiReport report = e.Report;

   // Save the report template
   // ...
}
...
```

Настройки сохранения

Сохранение отчета производиться в фоновом режиме, т.е. без перезагрузки страницы в окне браузера. Если необходимо каким-либо образом визуально управлять процессом сохранения отчета, следует изменить значение свойства **SaveReportMode** (либо **SaveReportAsMode**) дизайнера на одно из трех

© 2003-2026 Stimulsoft

указанных значений: **Hidden** (значение по умолчанию), **Visible** или **NewWindow**.

Default.aspx

```
...
<cc1:StiWebDesigner ID="StiWebDesigner1" runat="server"
   OnSaveReportAs="StiWebDesigner1_SaveReportAs"
   SaveReportAsMode="Visible">
</cc1:StiWebDesigner>
...
```

Если свойство **SaveReportMode** установлено в значение **Visible**, то событие сохранения отчета будет вызвано в текущем окне браузера в обычном (видимом) режиме при помощи POST-запроса. Если свойство **SaveReportMode** установлено в значение **NewWindow**, то событие сохранения отчета будет вызвано в новом окне браузера. По умолчанию данное свойство установлено в значение **Hidden**, т.е. событие сохранения отчета вызывается в фоне при помощи AJAX-запроса и никак не отображается в окне браузера. Для свойства **SaveReportAsMode** применимы те же значения и поведение.

3.2.8 Локализация дизайнера

Компонент **HTML5 Designer** поддерживает полную локализацию своего интерфейса. Для того чтобы локализовать интерфейс дизайнера отчетов на необходимый язык, используется специальное свойство **Localization**. В качестве значения данного свойства необходимо указать путь к XML-файлу локализации (относительный либо абсолютный).

Default.aspx

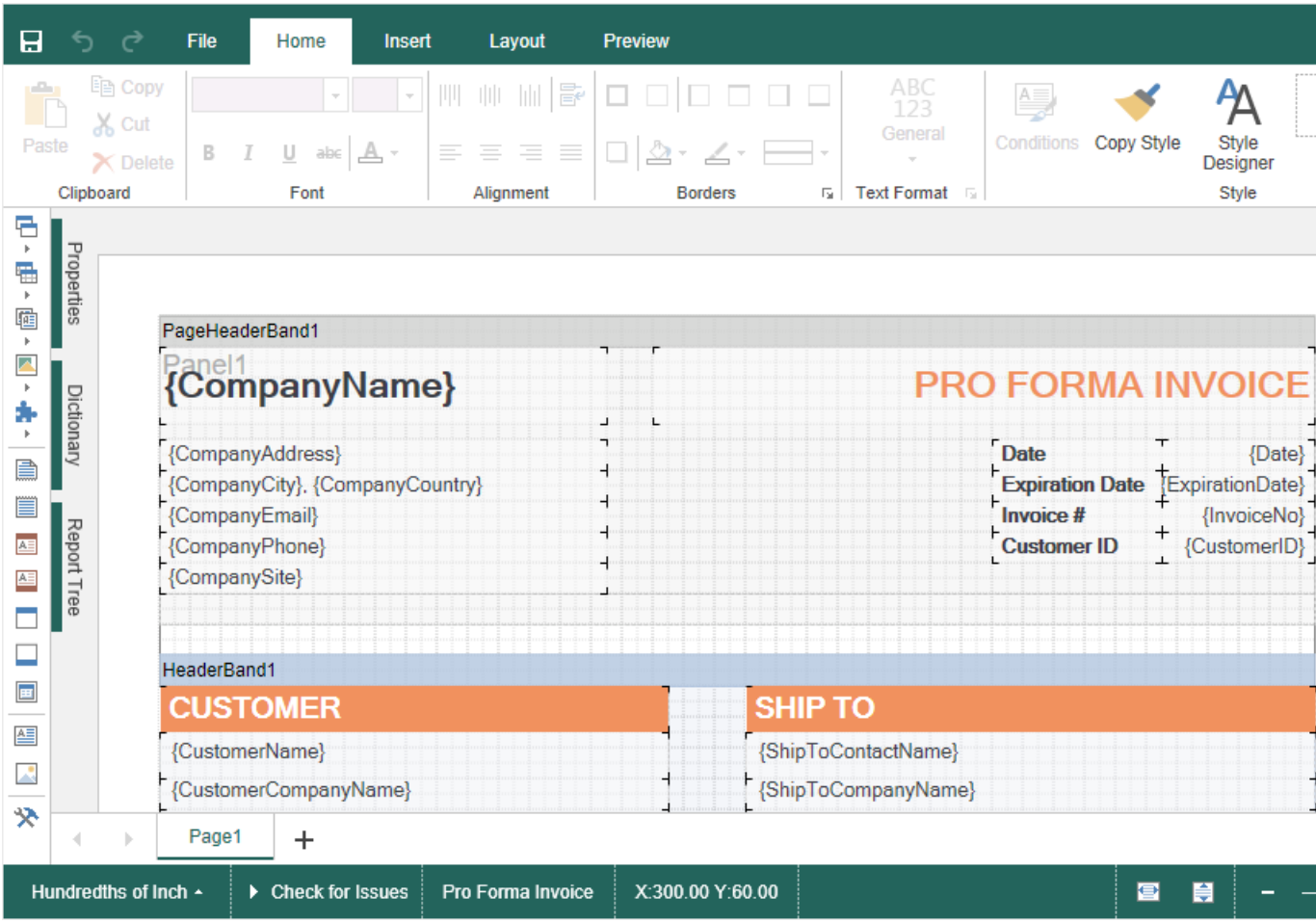
```
...
<cc1:StiWebDesigner ID="StiWebDesigner1" runat="server"
   Localization="Localization/en.xml">
</cc1:StiWebDesigner>
...
```

Интерфейс дизайнера отчетов позволяет выбрать необходимую локализацию из доступного списка. Для этого в качестве значения свойства **LocalizationDirectory** необходимо указать папку, в которой хранятся локализационные XML-файлы.

© 2003-2026 Stimulsoft

Default.aspx

```
...
<cc1:StiWebDesigner ID="StiWebDesigner1" runat="server"
   Localization="Localization/en.xml"
   LocalizationDirectory="Localization">
</cc1:StiWebDesigner>
...
```



Информация

Если задано значение для свойства **Localization**, то при запуске дизайнера отчетов всегда будет применяться указанная в этом свойстве локализация. Если значение свойства не задано, то будет автоматически загружена локализация, выбранная из списка доступных локализаций на панели дизайнера отчетов.

© 2003-2026 Stimulsoft

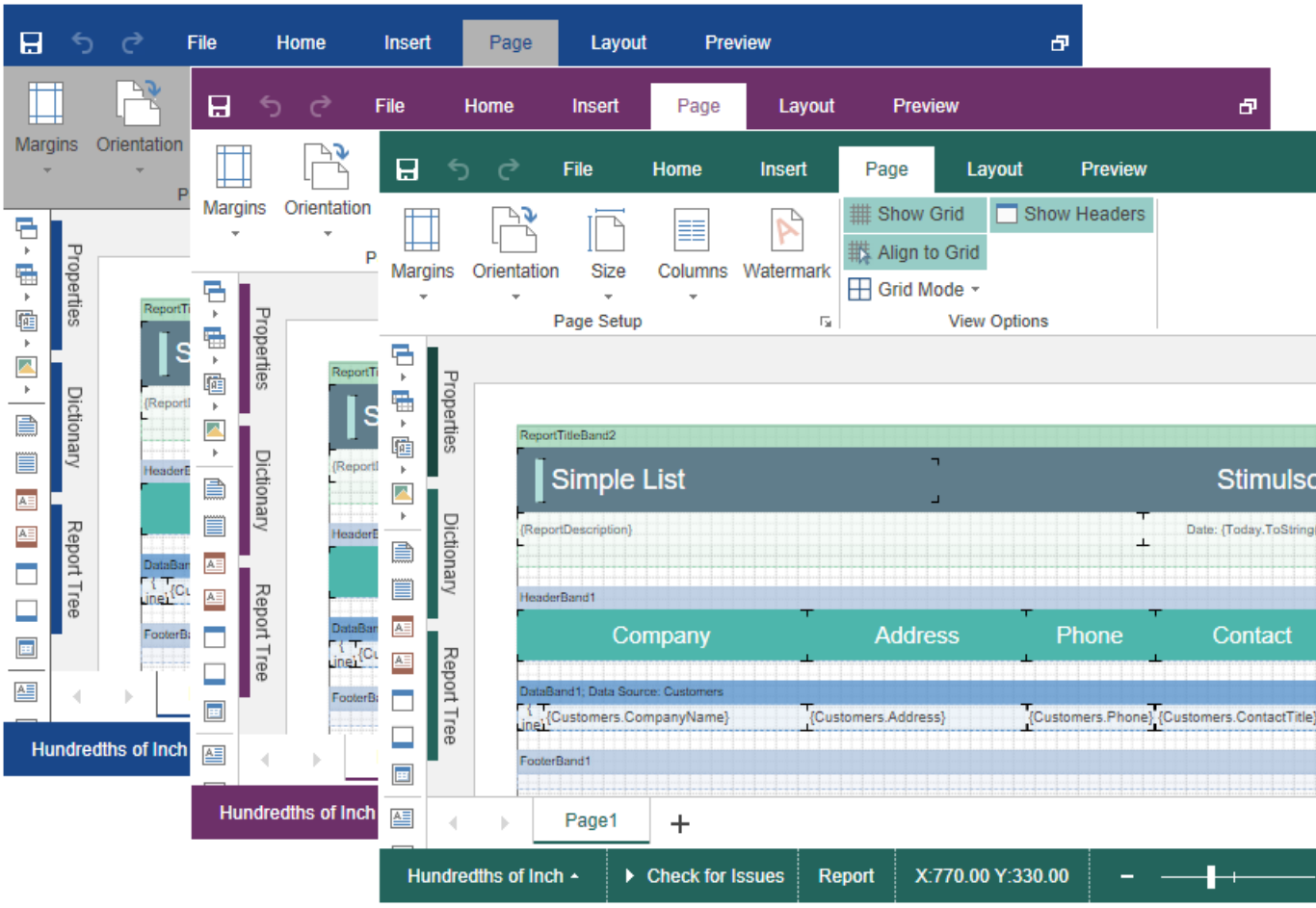
3.2.9 Использование тем

Компонент **HTML5 Designer** имеет возможность смены тем оформления визуальных элементов управления. Для изменения темы используется свойство **Theme**.

Default.aspx

```
...
<cc1:StiWebDesigner ID="StiWebDesigner1" runat="server"
   Theme="Office2022WhiteTeal">
</cc1:StiWebDesigner>
...
```

На данный момент доступно **2 различные темы** со своими цветовыми акцентами. В результате, доступно **более 50 вариантов** оформления. Это позволяет настроить внешний вид дизайнера практически под любое оформление Web проекта.



© 2003-2026 Stimulsoft

3.2.10 Кэширование

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для работы **HTM5 Designer** использует кэш сервера для хранения редактируемого шаблона отчета. Это необходимо, так как клиентская часть дизайнера содержит только визуальное представление компонентов шаблона отчета, сам объект отчета со всеми параметрами и свойствами хранится на стороне сервера.

Управлять кэшированием можно при помощи следующих приведенных ниже свойств.

Свойство CacheMode

Данное свойство дизайнера устанавливает тип кэширования. Может принимать одно из значений, указанных в перечислении **StiServerCacheMode**:

- **None** – не используется в компоненте **HTML5 Designer**;
- **ObjectCache** – для кэширования используется кэш сервера, в котором сохраняется объект отчета (значение по умолчанию);
- **StringCache** - для кэширования используется кэш сервера, в котором сохраняется отчет в виде упакованной строки;
- **ObjectSession** - для кэширования используется текущая сессия, в которой сохраняется объект отчета;
- **StringSession** - для кэширования используется текущая сессия, в которой сохраняется отчет в виде упакованной строки.

Свойство CacheItemPriority

© 2003-2026 Stimulsoft

Данное свойство устанавливает приоритет хранимого в кэше сервера отчета, влияет на автоматическую очистку памяти сервера в случае ее нехватки. Чем меньше приоритет, тем больше шанс удаления информации из памяти.

Свойство CacheTimeout

Данное свойство указывает величину времени в минутах, в течение которого необходимо сохранять отчет в кэше сервера. Если при использовании кэширования требуемый отчет не найден в кэше (т.е. истекло время хранения объекта), то он будет запрошен заново при помощи специального события **OnGetReport**, в этом случае несохраненные изменения могут быть утеряны.

Компонент **HTML5 Designer** предоставляет возможность определить собственные методы работы с кэшированием отчета. Для этого предназначен специальный класс **StiCacheHelper**, который содержит методы получения отчета из кэша и сохранения их в кэш. Необходимо создать новый класс, унаследованный от **StiCacheHelper**, и перегрузить указанные выше методы, которые соответственно имеют названия **GetObject** и **SaveObject**.

© 2003-2026 Stimulsoft

Default.aspx.cs

```
...
public partial class _Default : Page
{
   public class StiMyCacheHelper : StiCacheHelper
   {
      public override object GetObject(string guid)
      {
         string path =
         Path.Combine(HttpContext.Current.Server.MapPath(string.Empty),
         "CacheFiles", guid);
         if (File.Exists(path))
         {
            byte[] cacheData = File.ReadAllBytes(path);
            return StiCacheHelper.GetObjectFromCacheData(cacheData);
         }
         return null;

         //return base.GetObject(guid);
      }

      public override void SaveObject(object obj, string guid)
      {
         byte[] cacheData = StiCacheHelper.GetCacheDataFromObject(obj);
         string path =
         Path.Combine(HttpContext.Current.Server.MapPath(string.Empty),
         "CacheFiles", guid);
         File.WriteAllBytes(path, cacheData);

         //base.SaveObject(obj, guid);
      }

      public override void RemoveReport(string guid)
      {
         string path =
         Path.Combine(HttpContext.Current.Server.MapPath(string.Empty),
         "CacheFiles", guid);
         if (File.Exists(path))
         {
            File.Delete(path);
         }
      }
   }

   static _Default()
   {
      StiWebDesigner.CacheHelper = new StiMyCacheHelper();
   }
}
...
```

Для инициализации работы с кэшированием отчета при помощи созданного класса, достаточно задать его в качестве значения статического свойства

© 2003-2026 Stimulsoft

StiWebDesigner.CacheHelper в конструкторе ASPX страницы.

3.2.11 События

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Компонент **HTML5 Designer** поддерживает события, которые предоставляют возможность выполнить необходимые операции перед определенными действиями, такими как создание, редактирование и сохранение шаблона отчета, предварительный просмотр, печать отчета, экспортирование отчета, интерактивность и др. Ниже приведен пример обработки событий дизайнера.

Default.aspx

```
...
<cc1:StiWebDesigner ID="StiWebDesigner1" runat="server"
  OnGetReport="StiWebDesigner1_GetReport"
  OnCreateReport="StiWebDesigner1_CreateReport"
  OnSaveReport="StiWebDesigner1_SaveReport">
</cc1:StiWebDesigner>
...
```

© 2003-2026 Stimulsoft

Default.aspx.cs

```
...
protected void StiWebDesigner1_GetReport(object sender,
StiReportDataEventArgs e)
{
   StiReport report = new StiReport();
   report.Load(Server.MapPath("Reports/SimpleList.mrt"));

   e.Report = report;
}

protected void StiWebDesigner1_CreateReport(object sender,
StiReportDataEventArgs e)
{
   DataSet data = new DataSet();
   data.ReadXmlSchema(Server.MapPath("Data/Demo.xsd"));
   data.ReadXml(Server.MapPath("Data/Demo.xml"));

   e.Report.RegData(data);
   e.Report.Dictionary.Synchronize();
}

protected void StiWebDesigner1_SaveReport(object sender,
StiReportDataEventArgs e)
{
   try
   {
      e.Report.Save(Server.MapPath("Reports/" + e.Report.ReportName +
      ".mrt"));
   }
   catch (Exception ex)
   {
      e.ErrorString = ex.Message;
   }
}
...
```

Список событий

Наименование	Описание
OnGetReport	Событие возникает при запросе отчета или панели индикаторов для редактирования.
OnCreateReport	Событие возникает при создании нового отчета или новой панели индикаторов из меню дизайнера.
OnOpenReport	Событие возникает при открытии

© 2003-2026 Stimulsoft

	отчета из меню дизайнера. В аргументах события будет передан загруженный отчет или дашборд.
OnPreviewReport	Событие возникает при переходе на вкладку предварительного просмотра отчета или панели индикаторов, а также при интерактивных действиях, таких как использование переменных отчета или дашборда, динамическое сворачивание, детализация и сортировка отчета при предварительном просмотре.
OnSaveReport	Событие возникает при нажатии на кнопку Сохранить (Save) на панели либо в главном меню дизайнера.
OnSaveReportAs	Событие возникает при нажатии на кнопку Сохранить как (Save As) либо в главном меню дизайнера. Если событие не определено, отчет или панель индикаторов будет сохранен на локальный диск компьютера.
OnExportReport	Событие возникает при экспортировании отчета или панели индикаторов.
OnExportReportResponse	Событие возникает после экспортирования отчета или панели индикаторов перед сохранением файла экспорта.
OnExit	Событие возникает при нажатии на кнопку Выход (Exit) в главном меню дизайнера.

3.2.12 Время ожидания

При работе с компонентом **StiWebDesigner** можно установить время ожидания (timeout) выполнения различных операций - хранение отчета в кэше, ответ сервера, выполнение запроса. Настройка времени ожидания выполняется при

© 2003-2026 Stimulsoft

помощи свойств компонента и опций отчета.

Свойство CacheTimeout

Предоставляет возможность установить время в минутах, которое сервер будет хранить редактируемый отчет в кэше с момента последнего действия дизайнера. По умолчанию установлено значение 10 минут.

Default.aspx

```
...
<cc1:StiWebDesigner ID="StiWebDesigner1" runat="server"
   CacheTimeout="10">
</cc1:StiWebDesigner>
...
```

Default.aspx.cs

```
...
protected void Page_Load(object sender, EventArgs e)
{
   StiWebDesigner1.CacheTimeout = 10;
}
...
```

Использование кэша увеличивает скорость работы дизайнера с отчетом. Более подробно можно ознакомиться в главе Кэширование.

Свойство RequestTimeout

Предоставляет возможность установить время ожидания ответа от сервера в секундах, по истечении которого будет выдана ошибка связи с сервером. По умолчанию установлено значение 30 секунд. Для больших отчетов рекомендуется увеличить данное значение.

Default.aspx

```
...
<cc1:StiWebDesigner ID="StiWebDesigner1" runat="server"
   RequestTimeout="10">
</cc1:StiWebDesigner>
...
```

© 2003-2026 Stimulsoft

Default.aspx.cs

```
...
protected void Page_Load(object sender, EventArgs e)
{
   StiWebDesigner1.RequestTimeout = 30;
}
...
```

Опция CommandTimeout

Предоставляет возможность установить время ожидания запроса в секундах, при использовании SQL источников данных в отчете. Значение данного свойства сохраняется в самом шаблоне отчёта для каждого SQL подключения отдельно.

Ниже приведен пример кода, который предоставляет возможность установить время ожидания запроса для уже созданного MS SQL соединения и источников данных в отчете.

Default.aspx.cs

```
...
protected void Page_Load(object sender, EventArgs e)
{

   StiReport report = new StiReport();
   report.Load(Server.MapPath("Report.mrt"));
   ((StiSqlSource)
   report.Dictionary.DataSources["DataSourceName"]).CommandTimeout = 1000;

   StiWebDesigner1.Report = report;
}
...
```

3.2.13 Добавление пользовательских функций

Информация

Ознакомьтесь с примером по добавлению пользовательской функции для компонента ASP.NET HTML5 Designer GitHub.

При интеграции дизайнера отчетов в пользовательское приложение, присутствует возможность добавления custom функции в словарь данных

© 2003-2026 Stimulsoft

дизайнера отчетов. После добавления функции, её можно использовать при разработке отчетов и панелей индикаторов (дашбордов). Ниже представлен пример добавления функции расчета итога суммирования.

Default.aspx.cs

```
...
public static decimal MySum(object value)
{
   if (!ListExt.IsList(value))
   return Stimulsoft.Base.Helpers.StiValueHelper.TryToDecimal(value);

   return Stimulsoft.Data.Functions.Funcs.SkipNulls(ListExt.ToList(value))
   .TryCastToDecimal()
   .Sum();
}
...
static _Default()
{
   StiFunctions.AddFunction("MyCategory", "MySum",
   "description", typeof(_Default),
   typeof(decimal), "Calculates a sum of the specified set of values.",
   new[] { typeof(object) },
   new[] { "values" },
   new[] { "A set of values" }).UseFullPath = false;
}
...
```

3.2.14 Настройки

Настройка **HTML5 Designer** выполняется при помощи свойств компонента, которые можно задать на ASPX странице либо при помощи C#/VB кода. Ниже приведены примеры установки свойств дизайнера.

Default.aspx

```
...
<cc1:StiWebDesigner ID="StiWebDesigner1" runat="server"
    ShowClone="false"
    ShowCrossDataBand="false"
    ShowFileMenuOpen="false"
    PermissionBusinessObjects="None"
    PermissionDataConnections="View"
    DefaultUnit="Centimeters"
    Localization="Localization/de.xml"
    Theme="Office2022WhiteTeal">
</cc1:StiWebDesigner>
...
```

© 2003-2026 Stimulsoft

Default.aspx.cs

```
...
protected void Page_Load(object sender, EventArgs e)
{
   StiWebDesigner1.ShowClone = false;
   StiWebDesigner1.ShowCrossDataBand = false;
   StiWebDesigner1.ShowFileMenuOpen = false;
   StiWebDesigner1.PermissionBusinessObjects = StiDesignerPermissions.None;
   StiWebDesigner1.PermissionDataConnections = StiDesignerPermissions.View;
   StiWebDesigner1.DefaultUnit = StiReportUnitType.Centimeters;
   StiWebDesigner1.Localization = "Localization/de.xml";
   StiWebDesigner1.Theme = StiDesignerTheme.Office2022WhiteTeal;
   StiWebDesigner1.ReportDisplayMode = StiReportDisplayMode.Auto;
}
...
```

Основные настройки

Наименование	Описание
Width	Задает ширину компонента в необходимых единицах измерения, которые определены в классе **Unit**. Доступна установка значения в пикселях: **Unit.Pixel()**, поинтах: **Unit.Point()** и процентах: **Unit.Percentage()**. По умолчанию установлена ширина 100%. По умолчанию компонент разворачивается на всю область окна браузера.
Height	Задает высоту компонента в необходимых единицах измерения, которые определены в классе **Unit**. Доступна установка значения в пикселях: **Unit.Pixel()**, поинтах: **Unit.Point()** и процентах: **Unit.Percentage()**. По умолчанию установлена высота 800 пикселей. По умолчанию компонент разворачивается на всю область окна браузера.

Работа с сервером

Наименование	Описание

© 2003-2026 Stimulsoft

RequestTimeout	Устанавливает время ожидания ответа от сервера в секундах, по истечении которого будет выдана ошибка. По умолчанию установлено значение 30 секунд. Для больших отчетов рекомендуется увеличить данное значение.
CacheTimeout	Устанавливает время в минутах, которое сервер будет хранить редактируемый отчет с момента последнего действия дизайнера. По умолчанию установлено значение 10 минут.
CacheMode	Устанавливает режим кэширования отчета. Может принимать одно из указанных ниже значений перечисления **StiServerCacheMode**: › **None** – не используется в компоненте **HTML5 Designer**; › **ObjectCache** – в качестве хранилища используется кэш, отчет хранится как объект (значение свойства по умолчанию); › **ObjectSession** – в качестве хранилища используется сессия, отчет хранится как объект; › **StringCache** – в качестве хранилища используется кэш сервера, отчет сериализуется в упакованную строку; › **StringSession** – в качестве хранилища используется сессия, отчет сериализуется в упакованную строку.
CacheItemPriority	Устанавливает приоритет хранимого в кэше сервера отчета. Это свойство влияет на автоматическую очистку памяти сервера в случае ее нехватки. Чем меньше приоритет, тем больше шанс удаления информации из памяти.
AllowAutoUpdateCache	Устанавливает режим автоматического обновления кэша. Отчет, хранящийся в кэше либо сессии сервера, будет автоматически пересохранен через определенный

© 2003-2026 Stimulsoft

	промежуток времени при бездействии дизайнера (примерно каждые 3 минуты). По умолчанию свойство имеет значение **true**.
UseRelativeUrls	Устанавливает режим дизайнера, в котором для запросов на сервер используются относительные ссылки. По умолчанию свойство имеет значение **true**.
PortNumber	Получает или задает значение, которое указывает номер порта для использования в URL. Значение 0 определяет автоматическое обнаружение (установлено по умолчанию). Значение -1 удаляет номер порта.
PassQueryParametersForResources	Включает передачу всех параметров URL запроса при формировании ссылок на ресурсы дизайнера. При значении false для запроса ресурсов дизайнера используются только необходимые параметры, что способствует более корректной работе кэша браузера. По умолчанию свойство имеет значение **true**.
ShowServerErrorPage	Предоставляет возможность включить или выключить подробное отображение подробной ошибки сервера при просмотре отчета. По умолчанию свойство имеет значение **true**.
UseCompression	Включает сжатие запросов дизайнера в GZip поток. Это позволяет уменьшить объем интернет-трафика, но немного замедляет работу дизайнера. По умолчанию свойство имеет значение **false**.
UseCacheForResources	Включает кэширование на стороне сервера ресурсов компонента, таких как скрипты, стили, изображения. Позволяет ускорить загрузку компонента и уменьшить нагрузку на сервер при большом количестве клиентов. По умолчанию свойство имеет значение **true**.
AllowLoadingCustomFontsToClie	Позволяет передавать пользовательские

© 2003-2026 Stimulsoft

ntSide	шрифты на клиентскую сторону и преобразовывать их в CSS стиль для корректного отображения текста на стороне HTML с заданным шрифтом. По умолчанию свойство имеет значение **false**.

Внешний вид

Наименование	Описание
Theme	Задает тему оформления дизайнера. Список доступных тем оформления находится в перечислении **StiDesignerTheme**. По умолчанию установлено значение **Office2022WhiteBlue**.
IconSet	Предоставляет возможность установить набор иконок: › StiWebUIIconSet.Auto (значение по умолчанию) - устанавливает набор иконок автоматически. Для тем **Office2022** - используется набор иконок в стиле **Monoline**, для тем **Office2013** - используется набор иконок в стиле **Regular**. › StiWebUIIconSet.Monoline - устанавливает набор иконок в стиле **Monoline**; › StiWebUIIconSet.Regular - устанавливает набор иконок в стиле **Regular**.
CustomCss	Задает путь к CSS файлу стилей дизайнера. Если данное свойство установлено, то стандартные стили выбранной темы загружены не будут. По умолчанию установлено пустое значение.
Localization	Задает путь к XML файлу локализации. Путь может быть абсолютным либо относительным. По умолчанию используется английская локализация, которая встроена в

© 2003-2026 Stimulsoft

	дизайнер и не требует дополнительных XML файлов.
LocalizationDirectory	Задает путь к директории с XML файлами локализаций. Находящиеся в указанной папке локализационные файлы будут загружены в список выбора локализации на панели дизайнера.
DefaultUnit	Устанавливает единицы измерения размеров для отчета и всех его компонентов. По умолчанию используются сантиметры.
Zoom	Задает масштаб отображения страниц отчета. По умолчанию установлен масштаб 100 процентов. Допускаются значения от 10 до 200 процентов. Может принимать одно из указанных ниже значений перечисления **StiZoomMode**: › **PageWidth** – при запуске дизайнера будет установлен масштаб, необходимый для отображения отчета по ширине страницы; › **PageHeight** – при запуске дизайнера будет установлен масштаб, необходимый для отображения отчета по высоте страницы.
ShowAnimation	Включает анимацию элементов интерфейса дизайнера. По умолчанию свойство имеет значение **true**.
ShowTooltips	Включает отображение подсказок для элементов управления дизайнера при наведении мыши. По умолчанию свойство имеет значение **true**.
ShowTooltipsHelp	Включает отображение ссылки на онлайн документацию в подсказках для элементов управления дизайнера. По умолчанию свойство имеет значение **true**.
ShowDialogsHelp	Включает отображение ссылки на онлайн документацию на заголовках диалоговых форм дизайнера. По умолчанию свойство

© 2003-2026 Stimulsoft

	имеет значение **true**.
InterfaceType	Устанавливает тип используемого интерфейса дизайнера. Может принимать одно из указанных ниже значений перечисления **StiInterfaceType**: › **Auto** – тип интерфейса дизайнера будет выбран автоматически в зависимости от используемого устройства (значение по умолчанию); › **Mouse** – принудительное использование интерфейса для управления дизайнером при помощи мыши; › **Touch** – принудительное использование Touch интерфейса для управления дизайнером при помощи сенсорного экрана (мобильные устройства), также в этом режиме для удобства управления элементы интерфейса увеличены.
DatePickerFirstDayOfWeek	Устанавливает первый день недели для элемента выбора даты. Может принимать одно из указанных ниже значений перечисления **StiFirstDayOfWeek**: › **Monday** – первым днем недели является понедельник (значение по умолчанию); › **Sunday** – первым днем недели является воскресенье.
DatePickerIncludeCurrentDayForRanges	Предоставляет возможность включать или не включать текущий день в элементе выбора даты. По умолчанию свойство имеет значение **false**.
FormatForDateControls	Предоставляет возможность переопределить формат для элементов управления дат. По умолчанию, значение для текущей опции не задан и формат для дат определяется из локали браузера.

© 2003-2026 Stimulsoft

ShowReportTree	Включает отображение дерева компонентов отчета. По умолчанию свойство имеет значение **true**.
ChartRenderType	Устанавливает режим отображения диаграмм в отчете на вкладке **Просмотр (Preview)**. Может принимать одно из указанных ниже значений перечисления **StiChartRenderType**: › **Image** – диаграммы отображаются в виде статичных изображений; › **Vector** – диаграммы отображаются в векторном режиме в виде SVG объекта; › **AnimatedVector** - диаграммы отображаются в векторном режиме в виде SVG объекта, элементы диаграммы появляются с анимацией (значение по умолчанию).
ReportDisplayMode	Устанавливает режим экспорта для отображения страниц отчета на вкладке предварительного просмотра. Может принимать одно из указанных ниже значений перечисления **StiReportDisplayMode**: › **FromReport** - режим экспорта элементов отчета определяется из настроек шаблона - Div или Table; › **Table** – элементы отчета экспортируются с использованием HTML таблиц (значение по умолчанию); › **Div** – элементы отчета экспортируются с использованием DIV разметки; › **Span** - элементы отчета экспортируются с использованием SPAN разметки.
ParametersPanelDateFormat	Устанавливает формат даты и времени для переменных соответствующего типа на панели параметров. По умолчанию используется формат даты и времени, установленный браузером.

© 2003-2026 Stimulsoft

ParametersPanelSortDataItems	Предоставляет возможность сортировать значения переменной в направлении от наименьшего к большему (от А до Я). По умолчанию свойство имеет значение **true**.
CloseDesignerWithoutAsking	Предоставляет возможность установить режим закрытия дизайнера отчетов без подтверждения изменений в текущем отчете. По умолчанию свойство имеет значение **false**, т.е. дизайнер будет запрашивать разрешение на сохранение изменений в текущем отчете.
ShowSystemFonts	Предоставляет возможность отображать или не отображать системные шрифты в списке шрифтов дизайнера отчетов. По умолчанию свойство имеет значение **true**, т.е. системные шрифты отображаются в списке шрифтов.
ShowNewPageButton	Предоставляет возможность отображать или не отображать кнопку **Новая страница (New Page)** в дизайнере отчетов. По умолчанию свойство имеет значение **true**.
ShowNewDashboardButton	Предоставляет возможность отображать или не отображать кнопку **Новая панель индикаторов (New Dashboard)** в дизайнере отчетов. По умолчанию свойство имеет значение **true**.
WizardTypeRunningAfterLoad	Предоставляет возможность вызывать мастер создания отчетов после запуска дизайнера отчетов. Может принимать одно из указанных ниже значений перечисления **StiWizardType**: › **None** - дизайнер отчетов будет запущен без вызова мастера создания отчета; › **StandardReport** - дизайнер отчетов будет запущен с вызовом стандартного мастера создания отчета; › **MasterDetailReport** - дизайнер отчетов

© 2003-2026 Stimulsoft

	будет запущен с вызовом master-detail мастера создания отчета; › **LabelReport** - дизайнер отчетов будет запущен с вызовом мастера создания отчета с этикетками; › **InvoicesReport** - дизайнер отчетов будет запущен с вызовом мастера создания инвойса; › **OrdersReport** - дизайнер отчетов будет запущен с вызовом мастера создания ордера; › **QuotationReport** - дизайнер отчетов будет запущен с вызовом мастера создания квоты.
DefaultRibbonType	Предоставляет возможность установить используемый по умолчанию стиль панели инструментов дизайнера. Может принимать одно из указанных ниже значений перечисления: › **StiDesignerRibbonType.Classic** - Будет установлен классический стиль (значение по умолчанию); › **StiDesignerRibbonType.SingleLine** – Будет установлен компактный однострочный стиль.
AllowPropagationEvents	Предоставляет возможность разрешить распространение событий нажатия клавиш, когда дизайнер отчетов не в фокусе. По умолчанию свойство имеет значение **true**.
PropertiesPanelViewMode	Предоставляет возможность закрепить или открепить панель свойств, словарь отчета, дерево отчета. Может принимать одно из указанных ниже значений перечисления: › **Pinned** - панели закреплены (значение по умолчангию); › **Unpinned** - панели открепленны.

Поведение

Наименование	Описание

© 2003-2026 Stimulsoft

ShowSaveDialog	Включает отображение диалога ввода имени отчета при его сохранении. Имя отчета будет передано в параметрах дизайнера. По умолчанию свойство имеет значение **true**.
UndoMaxLevel	Устанавливает максимальную глубину отмены действий с отчетом (функция Undo/ Redo). Большое значение этого свойства будет потреблять больше памяти на стороне сервера для хранения параметров отмены. По умолчанию установлено значение 6.
AllowChangeWindowTitle	Разрешает использование заголовка окна браузера для отображения имени файла редактируемого отчета. По умолчанию свойство имеет значение **true**.
SaveReportMode	Задает режим сохранения отчета. Может принимать одно из трех значений перечисления **StiSaveMode**. › **Hidden** - действие сохранения отчета вызывается в фоне при помощи AJAX-запроса и никак не отображается в окне браузера (значение по умолчанию); › **Visible** - действие сохранения отчета будет вызвано в текущем окне браузера в видимом режиме при помощи POST-запроса; › **NewWindow** - действие сохранения отчета будет вызвано в новом окне (вкладке) браузера.
SaveReportAsMode	Задает режим сохранения отчета. Может принимать одно из трех значений перечисления **StiSaveMode**. › **Hidden** - действие сохранения отчета вызывается в фоне при помощи AJAX-запроса и никак не отображается в окне браузера (значение по умолчанию); › **Visible** - действие сохранения отчета будет вызвано в текущем окне браузера в видимом

© 2003-2026 Stimulsoft

	режиме при помощи POST-запроса; › **NewWindow** - действие сохранения отчета будет вызвано в новом окне (вкладке) браузера.
CheckReportBeforePreview	Предоставляет возможность проверять или не проверять отчет встроенным инструментом перед его просмотром. По умолчанию свойство имеет значение **true**.

Главное меню (FileMenu)

Наименование	Описание
Visible	Включает отображение главного меню дизайнера отчетов. По умолчанию свойство имеет значение **true**.
ShowNew	Включает отображение пункта главного меню **Создать (New)**. По умолчанию свойство имеет значение **true**.
ShowFileMenuNewReport	Включает отображение кнопки **Пустой отчет (Blank Report)** в меню **Создать**. По умолчанию свойство имеет значение **true**.
ShowFileMenuNewDashboard	Включает отображение кнопки **Пустая панель индикаторов (Blank Dashboard)** в меню **Создать**. По умолчанию свойство имеет значение **true**.
ShowOpen	Включает отображение пункта главного меню **Открыть (Open)**. По умолчанию свойство имеет значение **true**.
ShowSave	Включает отображение пункта главного меню **Сохранить (Save)**. По умолчанию свойство имеет значение **true**.
ShowSaveAs	Включает отображение пункта главного меню **Сохранить как (Save As)**. По умолчанию свойство имеет значение **true**.

© 2003-2026 Stimulsoft

ShowClose	Включает отображение пункта главного меню **Закрыть (Close)**. По умолчанию свойство имеет значение **true**.
ShowExit	Включает отображение пункта главного меню **Выход (Exit)**. По умолчанию свойство имеет значение **false**.
ShowReportSetup	Включает отображение пункта главного меню **Свойства отчета (Report Setup)**. По умолчанию свойство имеет значение **true**.
ShowOptions	Включает отображение пункта главного меню **Параметры (Options)**. По умолчанию свойство имеет значение **true**.
ShowInfo	Включает отображение пункта главного меню **Сведения (Info)**. По умолчанию свойство имеет значение **true**.
ShowAbout	Включает отображение пункта главного меню **О программе (About)**. По умолчанию свойство имеет значение **true**.
ShowHelp	Включает отображение пункта главного меню **Помощь (Help)**. По умолчанию свойство имеет значение **true**.

Словарь данных (Dictionary)

Наименование	Описание
Visible	Включает отображение словаря данных отчета. По умолчанию свойство имеет значение **true**.
UseAliases	Предоставляет возможность использовать псевдонимы в словаре данных. Может принимать одно из трех значений перечисления **StiUseAliases**: › **Auto** - определяет режим использования псевдонимов из сохраненного значения в

© 2003-2026 Stimulsoft

	файлах cookies (значение по умолчанию); › **True** - устанавливает режим использования псевдонимов в словаре данных; › **False** - выключает режим использования псевдонимов в словаре данных.
NewReportDictionary	Предоставляет возможность создавать новый словарь данных или объединить с существующим, при создании нового отчета в дизайнере. Может принимать одно из трех значений перечисления **StiNewReportDictionary**: › **Auto** - определяет режим создания или объединения словаря данных из сохраненного значения в файлах cookies(значение по умолчанию); › **DictionaryNew** - устанавливает режим создания нового словаря данных, при создании нового отчета; › **DictionaryMerge** - устанавливает режим объединения существующего словаря данных с новым, при создании нового отчета в дизайнере.
ShowDictionaryContextMenuProperties	Устанавливает видимость пункта **Свойства (Properties)** в контекстном меню словаря данных. По умолчанию свойство имеет значение **true**.
ShowDictionaryActions	Устанавливает видимость меню **Действия (Actions)** на панели инструментов словаря данных. По умолчанию свойство имеет значение **true**.
PermissionDataConnections	Устанавливает доступные действия над подключениями к данным отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionDataSources	Устанавливает доступные действия над источниками данных отчета. Может

© 2003-2026 Stimulsoft

	принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionDataColumns	Устанавливает доступные действия над колонками данных отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionBusinessObjects	Устанавливает доступные действия над бизнес-объектами отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionDataRelations	Устанавливает доступные действия над связями данных отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionVariables	Устанавливает доступные действия над переменными отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionResources	Устанавливает доступные действия над ресурсами в словаре данных отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionSqlParameters	Устанавливает доступные действия над параметрами в SQL запросах источников данных отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionDataTransformations	Устанавливает доступные действия над преобразованием данных. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionUserFunctions	Устанавливает доступные действия над пользовательскими функциями. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.

© 2003-2026 Stimulsoft

В таблице ниже приведены все доступные значения для перечисления **StiDesignerPermissions**, которые можно установить для элементов словаря отчета.

Значение	Описание
None	Запрещает любое действие над элементом словаря данных.
All	Разрешает любые действия над элементом словаря данных.
Create	Разрешает создавать определенный элемент словаря данных.
Delete	Разрешает удалять определенный элемент словаря данных.
Modify	Разрешает редактировать определенный элемент словаря данных.
View	Разрешает просматривать определенный элемент словаря данных.
ModifyView	Разрешает редактировать и просматривать определенный элемент словаря данных.

Панель инструментов (Toolbar)

Наименование	Описание
ShowToolbar	Предоставляет возможность отображать или не отображать панель инструментов в дизайнере отчетов. По умолчанию свойство имеет значение **true**.
ShowSetupToolboxButton	Включает отображение кнопки вызова окна настроек боковой панели компонентов отчета. По умолчанию свойство имеет значение **true**.
ShowInsertButton	Включает отображение вкладки **Вставить (Insert)** на панели инструментов дизайнера.

© 2003-2026 Stimulsoft

	По умолчанию свойство имеет значение **true**.
ShowLayoutButton	Включает отображение вкладки **Формат (Layout)** на панели инструментов дизайнера. По умолчанию свойство имеет значение **true**.
ShowPageButton	Включает отображение вкладки **Страница (Page)** на панели инструментов дизайнера. По умолчанию свойство имеет значение **true**.
ShowPreviewButton	Включает отображение вкладки **Предварительный просмотр (Preview)** на панели инструментов дизайнера. По умолчанию свойство имеет значение **true**.
ShowSaveButton	Включает отображение кнопки **Сохранить (Save)** на панели инструментов дизайнера. По умолчанию свойство имеет значение **true**.
ShowAboutButton	Включает отображение кнопки **О программе (About)** на панели инструментов дизайнера. По умолчанию свойство имеет значение **false**.

Панель свойств (PropertiesGrid)

Наименование	Описание
Visible	Включает отображение панели свойств компонентов отчета. По умолчанию свойство имеет значение **true**.
Width	Задает ширину панели свойств компонентов отчета. По умолчанию установлена ширина **370** точек.
LabelWidth	Задает ширину надписей панели свойств компонентов отчета. По умолчанию установлена ширина **160** точек.
PropertiesGridPosition	Предоставляет возможность определить позицию панели свойств в дизайнере отчетов. Может принимать одно из трех

© 2003-2026 Stimulsoft

	значений перечисления **StiPropertiesGridPosition**. › **Left** - панель свойств будет располагаться слева; › **Right** - панель свойств будет располагаться справа.
ShowPropertiesWhichUsedFromStyles	Устанавливает видимость свойств, которые используются из стилей в дизайнере. По умолчанию свойство имеет значение **false**.

Компоненты отчета (Components)

Наименование	Описание
ShowText	Включает отображение компонента **Текст (Text)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowTextInCells	Включает отображение компонента **Текст в ячейках (Text in Cells)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowRichText	Включает отображение компонента **Rich текст (Rich Text)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowImage	Включает отображение компонента **Картинка (Image)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowBarCode	Включает отображение компонента **Штрих-код (Bar Code)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowShape	Включает отображение компонента

© 2003-2026 Stimulsoft

	Геометрия (Shape) в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowHorizontalLinePrimitive	Включает отображение компонента **Горизонтальная линия (Horizontal Line)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowVerticalLinePrimitive	Включает отображение компонента **Вертикальная линия (Vertical Line)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowRectanglePrimitive	Включает отображение компонента **Прямоугольник (Rectangle)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowRoundedRectanglePrimitive	Включает отображение компонента **Закругленный прямоугольник (Rounded Rectangle)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowPanel	Включает отображение компонента **Панель (Panel)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowClone	Включает отображение компонента **Клон (Clone)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowCheckBox	Включает отображение компонента **Флажок (Check Box)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowSubReport	Включает отображение компонента **Вложенный отчет (Sub Report)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.

© 2003-2026 Stimulsoft

ShowZipCode	Включает отображение компонента **Почтовый индекс (Zip Code)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowTable	Включает отображение компонента **Таблица (Table)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowCrossTab	Включает отображение компонента **Кросс-таблица (Cross-Tab)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowChart	Включает отображение компонента **Диаграмма (Chart)** в меню вставки компонентов дизайнера. Действует сразу на все типы диаграмм. По умолчанию свойство имеет значение **true**.
ShowMap	Включает отображение компонента **Карта (Map)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **false**.
ShowGauge	Включает отображение компонента **Индикатор (Gauge)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **false**.
ShowSparkline	Включает отображение компонента **Спарклайн (Sparkline)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowMathFormula	Включает отображение компонента **Математическая формула (Math Formula)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowElectronicSignature	Включает отображение компонента **Электронная подпись (Electronic Signature)** в меню вставки компонентов дизайнера. По

© 2003-2026 Stimulsoft

	умолчанию свойство имеет значение **true**.
ShowPdfDigitalSignature	Включает отображение компонента **Цифровая подпись PDF (PDF Digital Signature)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.

Секции (Bands)

Наименование	Описание
ShowReportTitleBand	Включает отображение пункта **Заголовок отчета (Report Title)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowReportSummaryBand	Включает отображение пункта **Итог отчета (Report Summary)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowPageHeaderBand	Включает отображение пункта **Заголовок страницы (Page Header)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowPageFooterBand	Включает отображение пункта **Итог страницы (Page Footer)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowGroupHeaderBand	Включает отображение пункта **Заголовок группы (Group Header)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowGroupFooterBand	Включает отображение пункта **Итог группы (Group Footer)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowHeaderBand	Включает отображение пункта **Заголовок данных (Header)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.

© 2003-2026 Stimulsoft

ShowFooterBand	Включает отображение пункта **Итог данных (Footer)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowColumnHeaderBand	Включает отображение пункта **Заголовок колонки (Column Header)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowColumnFooterBand	Включает отображение пункта **Итог колонки (Column Footer)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowDataBand	Включает отображение пункта **Данные (Data)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowHierarchicalBand	Включает отображение пункта **Дерево (Hierarchical)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowChildBand	Включает отображение пункта **Подчиненный (Child)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowEmptyBand	Включает отображение пункта **Пустые данные (Empty)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowOverlayBand	Включает отображение пункта **Водяной знак (Overlay)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowTableOfContents	Включает отображение пункта **Оглавление (Table Of Contents)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.

Элементы дашборда (DashboardElements)

Наименование	Описание
ShowTableElement	Включает отображение элемента панели

© 2003-2026 Stimulsoft

	индикаторов **Таблица (Table)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowCardsElement	Включает отображение элемента панели индикаторов **Карточки (Cards)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowChartElement	Включает отображение элемента панели индикаторов **Диаграмма (Chart)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowGaugeElement	Включает отображение элемента панели индикаторов **Измерительный прибор (Gauge)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowPivotTableElement	Включает отображение элемента панели индикаторов **Сводная таблица (Pivot)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowIndicatorElement	Включает отображение элемента панели индикаторов **Индикатор (Indicator)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowProgressElement	Включает отображение элемента панели индикаторов **Прогресс (Progress)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowRegionMapElement	Включает отображение элемента панели индикаторов **Региональная карта (Region**

© 2003-2026 Stimulsoft

	Map) на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowOnlineMapElement	Включает отображение элемента панели индикаторов **Онлайн карта (Online Map)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowImageElement	Включает отображение элемента панели индикаторов **Изображение (Image)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowTextElement	Включает отображение элемента панели индикаторов **Текст (Text)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowPanelElement	Включает отображение элемента панели индикаторов **Панель (Panel)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowShapeElement	Включает отображение элемента панели индикаторов **Геометрия (Shape)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowButtonElement	Включает отображение элемента панели индикаторов **Кнопка (Button)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowListBoxElement	Включает отображение элемента панели индикаторов **Список (ListBox)** на инструментарии или вкладке **Вставить**

© 2003-2026 Stimulsoft

	(Insert) в дизайнере. По умолчанию свойство имеет значение **true**.
ShowComboBoxElement	Включает отображение элемента панели индикаторов **Выпадающий список (ComboBox)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowTreeViewElement	Включает отображение элемента панели индикаторов **Иерархический список (TreeView)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowTreeViewBoxElement	Включает отображение элемента панели индикаторов **Выпадающий иерархический список (TreeViewBox)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowDatePickerElement	Включает отображение элемента панели индикаторов **Выбор даты (Date Picker)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.

Кросс-секции (CrossBands)

Наименование	Описание
ShowCrossGroupHeaderBand	Включает отображение секции **Кросс-заголовок группы (Cross Group Header)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowCrossGroupFooterBand	Включает отображение секции **Кросс-итог группы (Cross Group Footer)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.

© 2003-2026 Stimulsoft

ShowCrossHeaderBand	Включает отображение секции **Кросс-заголовок данных (Cross Header)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowCrossFooterBand	Включает отображение секции **Кросс-итог данных (Cross Footer)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowCrossDataBand	Включает отображение секции **Кросс-данные (Cross Data)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.

Панели индикаторов(Dashboards)

Наименование	Описание
ShowNewDashboardButton	Предоставляет возможность отображать или не отображать кнопку **Новая панель индикаторов (New Dashboard)** в дизайнере отчетов. По умолчанию свойство имеет значение **true**.

Страницы(Pages)

Наименование	Описание
ShowNewPageButton	Предоставляет возможность отображать или не отображать кнопку **Новая страница (New Page)** в дизайнере отчетов. По умолчанию свойство имеет значение **true**.

При разработке отчета или дашборда в дизайнере отчетов, можно также определить настройки экспорта (ExportOptions), параметры Email (EmailOptions) и панель инструментов (PreviewToolbarOptions) на вкладке **Просмотр (Preview)**. Эти опции аналогичны опциям вьювера отчетов.

© 2003-2026 Stimulsoft

4 Reports and Dashboards for ASP.NET MVC

ASP.NET MVC представляет собою технологию создания веб-приложений и веб-сервисов по шаблону **Model-View-Controller**. Компания Stimulsoft предоставляет инструменты создания, отображения, преобразования отчетов и панелей индикаторов (дашбордов) на различных устройствах с использованием этой технологии.

Инструменты создания и редактирования отчетов:	Инструменты отображения и конвертации отчетов:	Инструменты создания и редактирования дашбордов:	Инструменты отображения и конвертации дашбордов:
HTML5 дизайнер	HTML5 вьювер	HTML5 дизайнер	HTML5 вьювер

4.1 HTML5 вьювер

YouTube

Посмотрите видеоуроки по работе с компонентом ASP.NET MVC HTML5 Viewer. Подписывайтесь на канал Stimulsoft и узнайте первыми о новых видеоуроках. Вопросы и предложения оставляйте в комментариях к видео.

Примеры

Ознакомьтесь с примерами по работе с компонентом ASP.NET MVC HTML5 Viewer GitHub. Все примеры представляют собой отдельные проекты, сгруппированные в один solution для Visual Studio.

Компонент **HTML5 Viewer (StiMvcViewer)** предназначен для просмотра отчетов в окне браузера. При этом не требуется устанавливать у клиента .NET Framework, ActiveX компоненты или какие-либо специальные плагины. Все, что

© 2003-2026 Stimulsoft

необходимо – это любой современный Web-браузер.

При помощи **HTML5 Viewer** можно просматривать, печатать, выполнять экспортирование отчетов на любом компьютере с любой установленной операционной системой. Так как вьювер использует только HTML и JavaScript технологии, он может быть запущен на устройствах, где нет поддержки Flash или Silverlight – планшеты, смартфоны. Также вьювер поддерживает Mobile и Touch интерфейсы, которые автоматически включаются при использовании мобильных устройств и мониторов с сенсорным экраном.

Компонент **HTML5 Viewer** использует технологию AJAX для выполнения всех действий (загрузки отчета, листания страниц, масштабирования, интерактивности в отчетах и др.), что позволяет избавиться от перезагрузки всей страницы, а также экономить Web-трафик и повысить скорость работы.

HTML5 Viewer поддерживает множество тем оформления, анимированный интерфейс, закладки, интерактивные отчеты, редактирование элементов отчета на странице, полноэкранный режим, поиск, и другие, необходимые для просмотра отчетов, возможности.

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для использования **HTML5 Viewer** в Web-проекте, необходимо установить NuGet пакет Stimulsoft.Reports.Web:

- Выберите пункт "Manage NuGet Packages..." в контекстном меню проекта;
- На вкладке Browse, в строке поиска, укажите Stimulsoft.Reports.Web;
- Выделите элемент, определите версию пакета и нажмите кнопку **Установить (Install)**. При обновлении пакета, следует нажать кнопку **Обновить (Update)**.

Если по какой-либо причине это невозможно, необходимо добавить в проект указанные ниже сборки:

- Stimulsoft.Base.dll

© 2003-2026 Stimulsoft

- Stimulsoft.Report.dll
- Stimulsoft.Report.Check.dll
- Stimulsoft.Report.Helper.dll
- Stimulsoft.Report.Mvc.dll
- Stimulsoft.Report.Web.dll
- Stimulsoft.Report.WebDesign.dll

Для того, чтобы добавить возможности просмотра и экспорта дашбордов в Web-проекте, установить NuGet пакет Stimulsoft.Dashboards.Web (данный пакет связан с пакетом Stimulsoft.Reports.Web, при его отсутствии он будет установлен автоматически):

- Выберите пункт "Manage NuGet Packages..." в контекстном меню проекта;
- На вкладке Browse, в строке поиска, укажите Stimulsoft.Dashboards.Web;
- Выделите элемент, определите версию пакета и нажмите кнопку **Установить (Install)**. При обновлении пакета, следует нажать кнопку **Обновить (Update)**.

Если по какой-либо причине это невозможно, необходимо дополнительно добавить в проект указанные ниже сборки:

- Stimulsoft.Dashboard.dll
- Stimulsoft.Dashboard.Drawing.dll
- Stimulsoft.Dashboard.Export.dll

- Как это работает
- Активация
- Отображение отчета и дашборда
- Подключение данных
- Локализация вьювера
- Печать отчета
- Экспорт отчета и дашборда
- Режимы отображения
- Динамическое сворачивание, сортировка и детализация
- Время ожидания
- Редактирование построенного отчета
- Отправка отчета по Email
- Вызов дизайнера из вьювера
- Кэширование
- Использование тем
- Основные возможности

© 2003-2026 Stimulsoft

- Работам с параметрами
- Работа с закладками
- Настройки вьювера
- Вспомогательные методы
- Экспорт и печать из кода

4.1.1 Как это работает

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для запуска вьювера требуется разместить на странице компонент **StiMvcViewer**, задать ему необходимые свойства, а в контроллере представления определить необходимые действия. При запуске вьювера отчетов происходят следующие действия:

- .NET MVC компонент генерирует HTML и JavaScript код, необходимый для отображения и работы вьювера;
- По окончании вывода компонента запускается JavaScript метод, который запрашивает на стороне сервера первую страницу отчета, весь отчет (в зависимости от выбранного режима) и необходимые параметры отчета;
- Каждое действие во вьювере (например, листание страниц, печать либо экспорт отчета и др.) вызывает определенное действие на стороне сервера, в котором можно выполнить необходимые манипуляции с отчетом;
- Для ускорения работы вьювер сохраняет отчет в кэше или сессии сервера, что позволяет исключить повторное построение отчета.

4.1.2 Активация

После приобретения продукта Stimulsoft вам необходимо активировать лицензию для используемых компонентов. Это можно выполнить указав лицензионный ключ или загрузив файл с лицензионным ключом. Ниже представлен пример активации компонента **StiMvcViewer**.

© 2003-2026 Stimulsoft

HomeController.cs

```
...
public class HomeController : Controller
{
   static HomeController()
   {
      //Activation with using license code
      Stimulsoft.Base.StiLicense.Key = "Your activation code...";

      //Activation with using license file
      var path = System.Web.HttpContext.Current.Server.MapPath("~/Content/
      license.key");
      Stimulsoft.Base.StiLicense.LoadFromFile(path);
   }
}
...
```

Получить лицензионный ключ или скачать файл с лицензионным ключом, вы можете в личном кабинете пользователя. Для авторизации в кабинете используйте имя пользователя и пароль, указанные при покупке подписки продукта.

4.1.3 Отображение отчета и дашборда

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Примечание

При присвоении отчета вьювер компоненту, происходит автоматическое построение отчета. Вызов метода `Report.Render()` необходим только в том случае, если требуется провести какие-либо действия с построенным отчетом перед его отображением во вьювере. Аналогично, при использовании режима компиляции, вызов метода `Report.Compile()` требуется только в том случае, если требуется провести какие-либо действия с откомпилированным отчетом перед его построением и отображением во вьювере.

© 2003-2026 Stimulsoft

Для отображения отчета необходимо добавить на страницу компонент **StiMvcViewer** и задать ему минимально необходимые свойства, а в контроллере представления определить необходимые действия.

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcViewer("MvcViewer1",
   new StiMvcViewerOptions() {
      Actions =
      {
         GetReport = "GetReport",
         ViewerEvent = "ViewerEvent"
      }
})
...
```

HomeController.cs

```
...
public ActionResult GetReport()
{
   StiReport report = new StiReport();
   report.Load(Server.MapPath("~/Content/SimpleList.mrt"));
   //report.Load(Server.MapPath("~/Content/Dashboard.mrt"));

   return StiMvcViewer.GetReportResult(report);
}

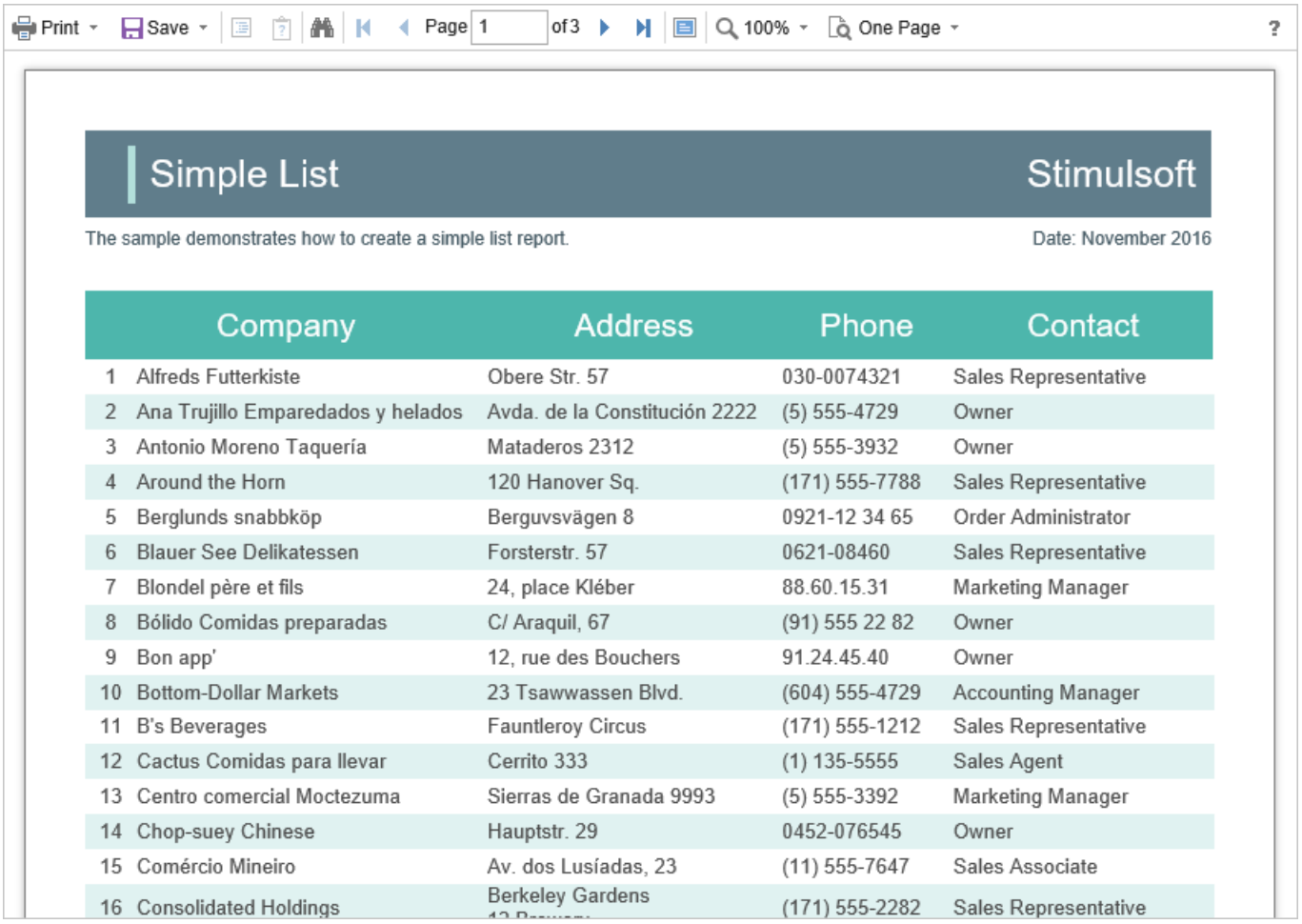
public ActionResult ViewerEvent()
{
   return StiMvcViewer.ViewerEventResult();
}
...
```

В приведенном выше примере добавлена обработка двух действий вьювера. Действие **GetReport** возвращает подготовленный отчет вьюверу, действие **ViewerEvent** обрабатывает события вьювера.

Информация

Действие **ViewerEvent** является обязательным. Без него корректная работа вьювера невозможна.

© 2003-2026 Stimulsoft



	Company	Address	Phone	Contact
1	Alfreds Futterkiste	Obere Str. 57	030-0074321	Sales Representative
2	Ana Trujillo Emparedados y helados	Avda. de la Constitución 2222	(5) 555-4729	Owner
3	Antonio Moreno Taquería	Mataderos 2312	(5) 555-3932	Owner
4	Around the Horn	120 Hanover Sq.	(171) 555-7788	Sales Representative
5	Berglunds snabbköp	Berguvsvägen 8	0921-12 34 65	Order Administrator
6	Blauer See Delikatessen	Forsterstr. 57	0621-08460	Sales Representative
7	Blondel père et fils	24, place Kléber	88.60.15.31	Marketing Manager
8	Bólido Comidas preparadas	C/ Araquil, 67	(91) 555 22 82	Owner
9	Bon app'	12, rue des Bouchers	91.24.45.40	Owner
10	Bottom-Dollar Markets	23 Tsawwassen Blvd.	(604) 555-4729	Accounting Manager
11	B's Beverages	Fauntleroy Circus	(171) 555-1212	Sales Representative
12	Cactus Comidas para llevar	Cerrito 333	(1) 135-5555	Sales Agent
13	Centro comercial Moctezuma	Sierras de Granada 9993	(5) 555-3392	Marketing Manager
14	Chop-suey Chinese	Hauptstr. 29	0452-076545	Owner
15	Comércio Mineiro	Av. dos Lusíadas, 23	(11) 555-7647	Sales Associate
16	Consolidated Holdings	Berkeley Gardens	(171) 555-2282	Sales Representative

Если отчет не был построен перед отображением, то компонент **HTML5 Viewer** построит его автоматически. Таким образом, для отображения отчета разрешено использовать шаблоны отчетов, построенные отчеты и отчеты в виде классов.

HomeController.cs

```
...
public ActionResult GetReport()
{
   StiReport report = new StiReport();
   report.LoadDocument(Server.MapPath("~/Content/SimpleList.mdc"));

   return StiMvcViewer.GetReportResult(report);
}
...
```

© 2003-2026 Stimulsoft

HomeController.cs

```
...
public ActionResult GetReport()
{
   StiReport report = new StiReportCompiledClass();

   return StiMvcViewer.GetReportResult(report);
}
...
```

Так как дашборд не является статическим документов и для его работы требуются данные, то формат построенного документа MDC для него не доступен. Вместо этого предоставлена возможность использовать снимок отчета в формате MRT, который содержит в себе все необходимые для работы дашборда данные, и может быть корректно отображен во вьювере.

HomeController.cs

```
...
public ActionResult GetReport()
{
   StiReport report = new StiReport();
   report.Load(Server.MapPath("~/Content/Report Snapshot.mrt"));

   return StiMvcViewer.GetReportResult(report);
}
...
```

Подключение пользовательских шрифтов

Вы можете подключить пользовательские шрифты с использованием класса **StiFontCollection**, указав файл содержащий шрифт. Для этого необходимо в конструкторе контроллера вызвать статический метод для загрузки шрифта.

ViewerController.cs

```
...
public class ViewerController : Controller
{
   static ViewerController()
   {
      Stimulsoft.Base.StiFontCollection.AddFontFile(Server.MapPath("~/
      fonts/my-font/font-name.ttf"));
   }
}
...
```

© 2003-2026 Stimulsoft

4.1.4 Подключение данных

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Данные для построения отчета могут быть подключены различными способами. Самый простой – хранить настройки подключения в самом шаблоне отчета. Так же данные можно подключить из кода, это можно сделать при загрузке отчета в действии **GetReport**.

HomeController.cs

```
...
public ActionResult GetReport()
{
   DataSet ds = new DataSet();
   ds.ReadXml(Server.MapPath("~/Content/Data/Demo.xml"));

   StiReport report = new StiReport();
   report.Load(Server.MapPath("~/Content/TwoSimpleLists.mrt"));
   report.Dictionary.Databases.Clear();
   report.RegData("Demo", ds);

   return StiMvcViewer.GetReportResult(report);
}
...
```

Данные для отчета можно подключать не только при его загрузке. Например, можно подключить новые данные в момент интерактивных действий во вьювере (применение параметров отчета, сортировка, детализация, сворачивание). Для этого необходимо задать действие **Interaction** для компонента **HTML5 Viewer**, а в обработчике действия подключить данные для текущего отчета. Таким же способом можно подключить данные в других действиях вьювера.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcViewer("MvcViewer1",
   new StiMvcViewerOptions() {
      Actions =
      {
         GetReport = "GetReport",
         ViewerEvent = "ViewerEvent",
         Interaction = "ViewerInteraction"
      }
})
...
```

HomeController.cs

```
...
public ActionResult ViewerInteraction()
{
   DataSet data = new DataSet();
   data.ReadXml(Server.MapPath("~/Content/Data/Demo.xml"));

   StiReport report = StiMvcViewer.GetReportObject();
   report.RegData("Demo", data);

   return StiMvcViewer.InteractionResult(report);
}
...
```

При необходимости подключить новые данные только для определенного интерактивного действия вьювера, например, только при применении параметров отчета, можно воспользоваться параметрами вьювера. Параметры вьювера представлены в виде объекта класса **StiRequestParams**, они передаются при любом запросе на сторону сервера, и содержат все необходимые сведения и состояния клиентской части вьювера. Для определения типа действия вьювера достаточно проверить свойство **Action** у параметров вьювера.

© 2003-2026 Stimulsoft

HomeController.cs

```
...
public ActionResult ViewerInteraction()
{
   StiRequestParams requestParams = StiMvcViewer.GetRequestParams();
   if (requestParams.Action == StiAction.Variables)
   {
      DataSet data = new DataSet();
      data.ReadXml(Server.MapPath("~/Content/Data/Demo.xml"));

      StiReport report = StiMvcViewer.GetReportObject();
      report.RegData("Demo", data);

      return StiMvcViewer.InteractionResult(report);
   }

   return StiMvcViewer.InteractionResult();
}
...
```

SQL источники данных

Параметры подключения к SQL источнику данных, как и к любому другому, можно хранить в самом шаблоне отчета. Если требуется задать параметры подключения из кода перед построением отчета (например, по причине безопасности или в зависимости от авторизованного пользователя), можно воспользоваться приведенным ниже примером.

HomeController.cs

```
...
public ActionResult GetReport()
{
   OracleConnection connection = new OracleConnection("Data
   Source=Oracle8i;Integrated Security=yes");
   connection.Open();
   OracleDataAdapter adapter = new OracleDataAdapter();
   adapter.SelectCommand = new OracleCommand("SELECT * FROM Products",
   connection);

   DataSet dataSet = new DataSet("productsDataSet");
   adapter.Fill(dataSet, "Products");

   StiReport report = new StiReport();
   report.Load(Server.MapPath("~/Content/SqlSampleReport.mrt"));
   report.RegData("Products", dataSet);

   return StiMvcViewer.GetReportResult(report);
}
...
```

© 2003-2026 Stimulsoft

Также, для SQL источников данных используемых в отчете, можно указать **Время ожидания запроса (Query Timeout)** в секундах. Значение данного свойства сохраняется в самом шаблоне отчёта для каждого SQL подключения отдельно.

Ниже приведен пример кода, который предоставляет возможность изменить строку соединения для MS SQL, скорректировать запрос, установить время ожидания запроса для уже созданного соединения и источников данных в отчете.

HomeController.cs

```
...
public ActionResult GetReport()
{

   StiReport report = new StiReport();
   report.Load(Server.MapPath("Report.mrt"));
   ((StiSqlDatabase)
   report.Dictionary.Databases["Connection"]).ConnectionString = @"Data
   Source=server;Integrated Security=True;Initial Catalog=DataBase";
   ((StiSqlSource)
   report.Dictionary.DataSources["DataSourceName"]).SqlCommand = "select *
   from Table where Column = 100";
   ((StiSqlSource)
   report.Dictionary.DataSources["DataSourceName"]).CommandTimeout = 1000;

   return StiMvcViewer.GetReportResult(report);
}
...
```

Информация

Для SQL источников данных других типов подключение создается аналогично, при этом подключается адаптер, соответствующий типу источника данных. Например, для MS SQL источника необходимо подключать SqlDataAdapter, для OLE DB необходим OleDbDataAdapter. Также, при этом следует указать строку соединения, соответствующую типу подключения.

Вы также можете использовать данные для разработки отчетов и дашбордов полученные из OData хранилищ данных. В этом случае, авторизация может

© 2003-2026 Stimulsoft

быть выполнена с использованием имени, пароля пользователя или с использованием токена. Параметры авторизации указываются в строке соединения к OData хранилищу с использованием разделителя ";".

HomeController.cs

```
...
public ActionResult GetReport()
{

   var report = new StiReport();

   //Authorization using a user account
   var oDataDatabase = new StiODataDatabase("OData", "OData", @"https://
   services.odata.org/V4/Northwind/
   Northwind.svc;AddressBearer=adress;UserName=UserName;Password=Password;C
   lient_Id=Your Client ID", false, null);

   //Authorization using a user token
   var oDataDatabase = new StiODataDatabase("OData", "OData", @"https://
   services.odata.org/V4/Northwind/Northwind.svc;Token=Enter your token",
   false, null);

   report.Dictionary.Databases.Add(oDataDatabase);
   oDataDatabase.Synchronize(report);

   //Query with data filter
   ((StiSqlSource)report.Dictionary.DataSources["Products"]).SqlCommand =
   "Products?$filter=ProductID eq 2";

   return StiMvcViewer.GetReportResult(report);
}
...
```

В таблице ниже представлены шаблоны строк подключения для различных типов источников данных.

Источник данных	Шаблон строки соединения
MS SQL	Integrated Security=False; Data Source=myServerAddress;Initial Catalog=myDataBase; User ID=myUsername; Password=myPassword;
MySQL	Server=myServerAddress; Database=myDataBase;UserId=myUsername; Pwd=myPassword;

© 2003-2026 Stimulsoft

ODBC	Driver={SQL Server}; Server=myServerAddress;Database=myDataBase; Uid=myUsername; Pwd=myPassword;
OLE DB	Provider=SQLOLEDB.1; Integrated Security=SSPI;Persist Security Info=False; Initial Catalog=myDataBase;Data Source=myServerAddress
Oracle	Data Source=TORCL;User Id=myUsername;Password=myPassword;
MS Access	Provider=Microsoft.Jet.OLEDB.4.0;User ID=Admin;Password=pass;Data Source=C:\\myAccessFile.accdb;
PostgreSQL	Server=myServerAddress; Port=5432; Database=myDataBase;User Id=myUsername; Password=myPassword;
Firebird	User=SYSDBA; Password=masterkey; Database=SampleDatabase.fdb;DataSource=myServerAddress; Port=3050; Dialect=3; Charset=NONE;Role=; Connection lifetime=15; Pooling=true; MinPoolSize=0;MaxPoolSize=50; Packet Size=8192; ServerType=0;
SQL CE	Data Source=c:\MyData.sdf; Persist Security Info=False;
SQLite	Data Source=c:\mydb.db; Version=3;
DB2	Server=myAddress:myPortNumber;Database=myDataBase;UID=myUsername;PWD=myPassword; Max Pool Size=100;Min Pool Size=10;
Infomix	Database=myDataBase;Host=192.168.10.10;Server=db_engine_tcp;Service=1492;Protocol=onsoctcp;UID=myUsername;Password=myPassword;
Sybase	Data Source=myASEserver;Port=5000;Database=myDataBase;Uid=myUsername;Pwd=myPassword;
Teradata	Data Source=myServerAddress;User ID=myUsername;Password=myPassword;

© 2003-2026 Stimulsoft

VistaDB	Data Source=D:\folder\myVistaDatabaseFile.vdb4;Open Mode=ExclusiveReadWrite;
Universal(dotConnect)	Provider=Oracle;direct=true;data source=192.168.0.1;port=1521;sid=sid;user=user;password=pass
MongoDB	mongodb://<user>:<password>@localhost/test
OData	http://services.odata.org/v3/odata/OData.svc/
Другие...	В таблице представлены наиболее часто используемые шаблоны для строки соединения. Посмотреть различные варианты строк соединения вы можете на специализированном сайте.

Данные из XML, JSON, Excel файлов

Подключение к XML и JSON источникам данных можно хранить в шаблоне отчета. Если требуется указать файлы данных из кода, можно воспользоваться приведенным ниже примером.

HomeController.cs

```
...
public ActionResult GetReport()
{
   DataSet data = new DataSet();
   data.ReadXml(Server.MapPath("~/Content/Data/Demo.xml"));

   StiReport report = new StiReport();
   report.Load(Server.MapPath("~/Content/SimpleList.mrt"));
   report.RegData(data);

   return StiMvcViewer.GetReportResult(report);
}
...
```

© 2003-2026 Stimulsoft

HomeController.cs

```
...
public ActionResult GetReport()
{
   DataSet data
   = StiJsonToDataSetConverterV2.GetDataSetFromFile(Server.MapPath("~/
   Content/Data/Demo.json"));

   StiReport report = new StiReport();
   report.Load(Server.MapPath("~/Content/SimpleList.mrt"));
   report.RegData(data);

   return StiMvcViewer.GetReportResult(report);
}
...
```

Информация

Во вьювере присутствует возможность получения данных из Excel файла. Для этого можно воспользоваться указанным ниже методом.

```
DataSet dataSet = StiExcelConnector.Get().GetDataSet(new
StiExcelOptions(array, this.FirstRowIsHeader));
```

4.1.5 Локализация вьювера

Компонент **HTML5 Viewer** поддерживает полную локализацию своего интерфейса. Для того чтобы локализовать интерфейс вьювера отчетов на необходимый язык, используется специальное свойство **Localization**. В качестве значения данного свойства необходимо указать путь к XML-файлу локализации (относительный либо абсолютный).

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcViewer("MvcViewer1",
   new StiMvcViewerOptions() {
      Localization = "~/Content/Localization/en.xml"
})
...
```

При загрузке вьювера отчетов файл локализации будет загружен автоматически.

© 2003-2026 Stimulsoft

4.1.6 Использование тем

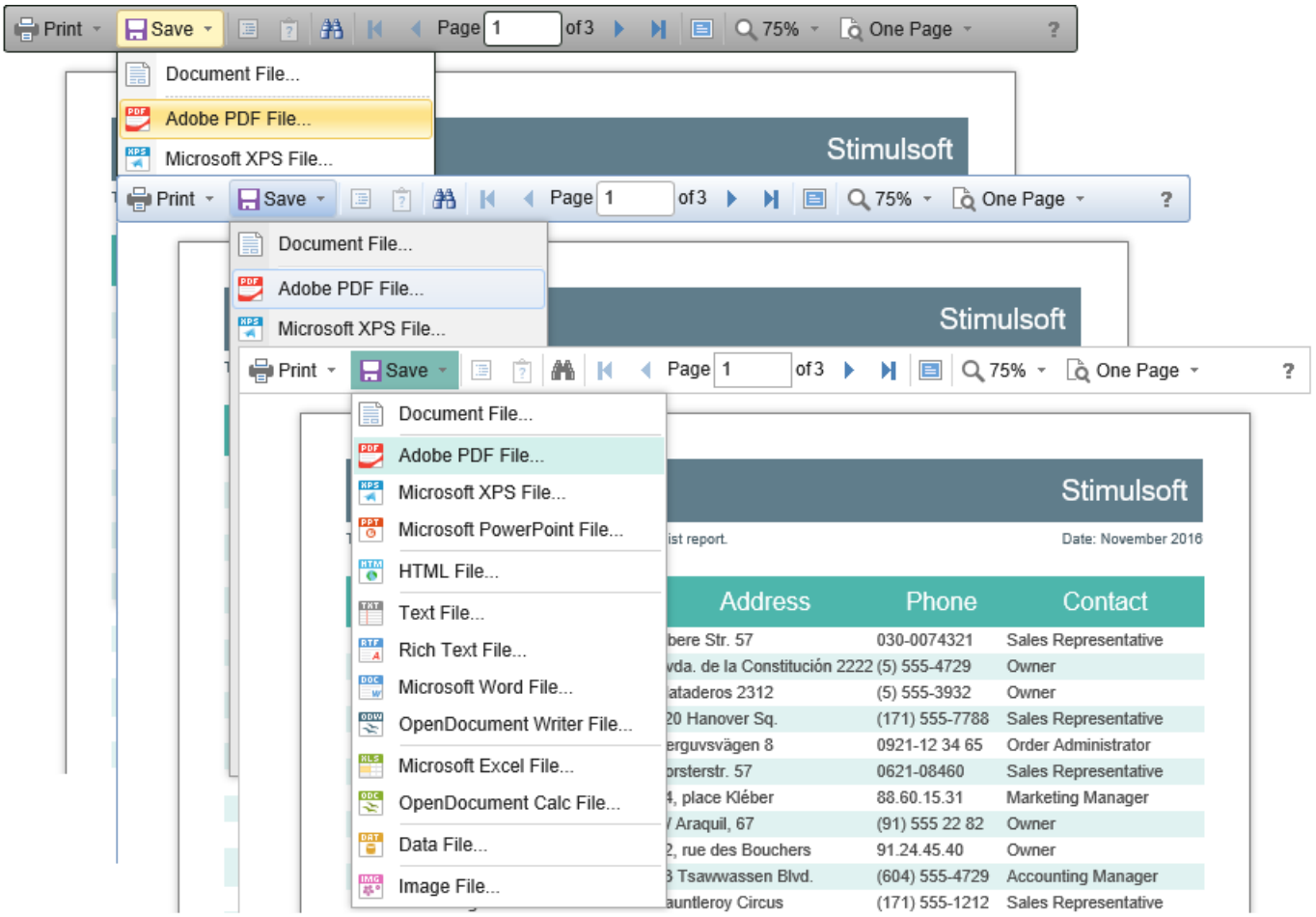
Компонент **HTML5 Viewer** имеет возможность смены тем оформления визуальных элементов управления. Для изменения темы используется свойство **Theme**, которое может принимать одно из значений перечисления **StiViewerTheme**.

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcViewer("MvcViewer1",
   new StiMvcViewerOptions() {
      Theme = StiViewerTheme.Office2022WhiteTeal
})
...
```

На данный момент доступно **8 тем** оформления с различными цветовыми акцентами. В результате, доступно более **60** вариантов оформления. Это позволяет настроить внешний вид вьювера практически под любое оформление Web проекта.

© 2003-2026 Stimulsoft

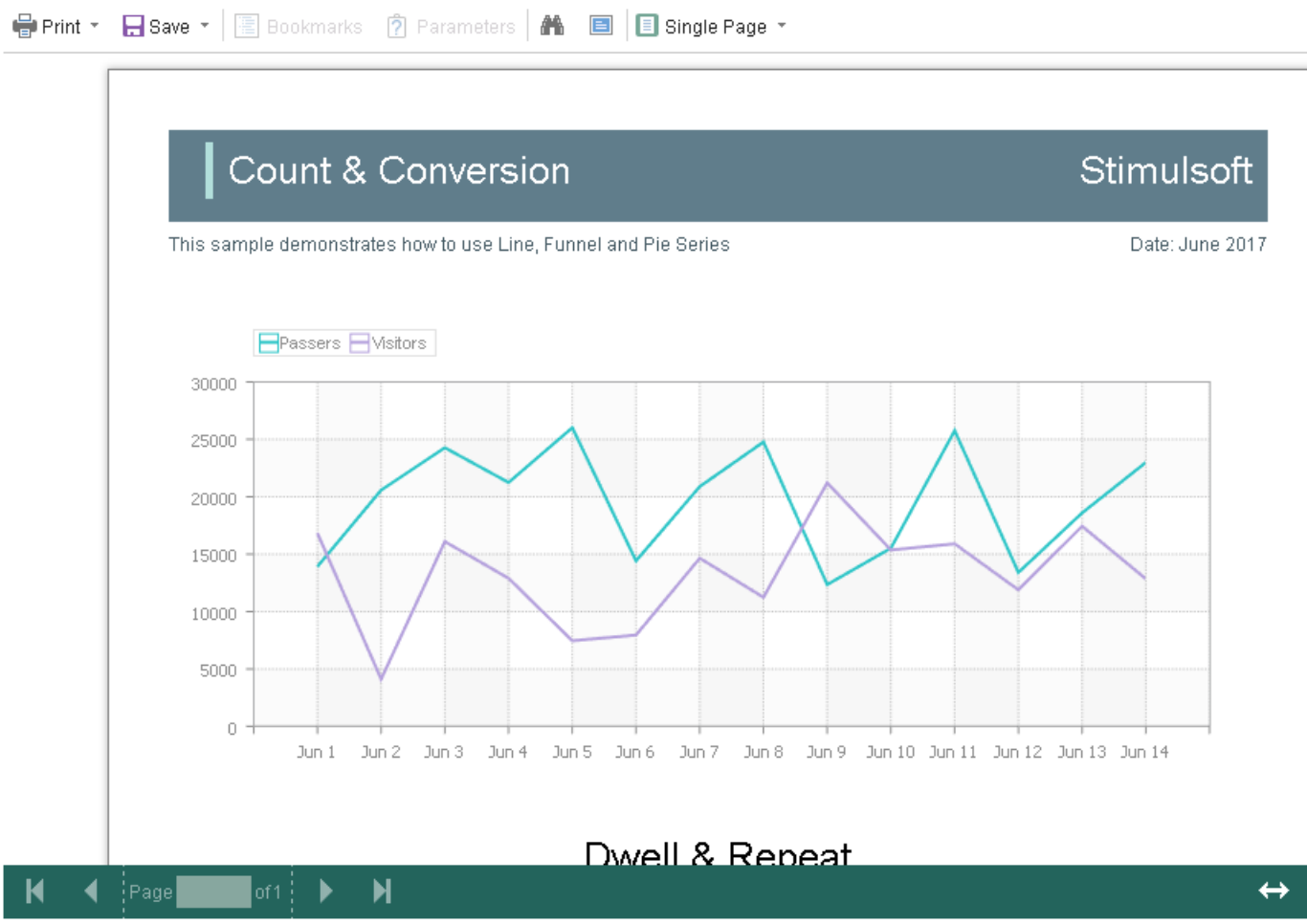


По умолчанию вьювер имеет только верхнюю панель инструментов, на которой находятся все элементы управления отчетом. При необходимости, панель инструментов можно разделить на верхнюю и нижнюю. На верхней панели будут находиться меню печати и экспорта отчета, а также кнопки работы с параметрами и закладками. Нижняя панель инструментов будет содержать элементы переключения между страницами отчета и меню управления масштабом. Для включения указанного режима предназначено свойство **DisplayMode**, которое может иметь значение **Simple** (стандартный простой режим) и **Separated** (раздельный режим).

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcViewer("MvcViewer1",
   new StiMvcViewerOptions() {
      Appearance =
      {
        ScrollbarsMode = true
      },
      Toolbar =
      {
        DisplayMode = StiToolbarDisplayMode.Separated
      }
})
...
```



Дополнительно предусмотрена возможность установки параметров оформления основных элементов вьювера. Например, можно изменить шрифт и цвет надписей панели управления вьювера, установить фон вьювера, задать

© 2003-2026 Stimulsoft

цвет границ страницы и др. Ниже представлен список доступных свойств, изменяющих оформление вьювера, и их значения по умолчанию.

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcViewer("MvcViewer1",
   new StiMvcViewerOptions() {
      Appearance =
      {
         BackgroundColor = Color.White,
         PageBorderColor = Color.Blue,
         ShowPageShadow = true
      },
      Toolbar =
      {
         BackgroundColor = Color.White,
         BorderColor = Color.Gray,
         FontColor = Color.Black,
         FontFamily = "Arial"
      }
})
...
```

4.1.7 Основные возможности

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

К основным возможностям вьювера можно отнести следующие операции: переключение между страницами отчета, изменение масштаба и режима отображения отчета, печать отчета, экспортирование отчета, применение параметров отчета, сортировку, сворачивание, детализацию, интерактивную фильтрацию. Все указанные операции выполняются в AJAX-режиме без перезагрузки страницы браузера. Для корректной работы этих операций необходимо определить специальное действие **ViewerEvent**.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcViewer("MvcViewer1",
   new StiMvcViewerOptions() {
      Actions =
      {
         ViewerEvent = "ViewerEvent"
      }
})
...
```

HomeController.cs

```
...
public ActionResult ViewerEvent()
{
   // Some code before viewer event
   // ...

   return StiMvcViewer.ViewerEventResult();
}
...
```

Информация

Данное действие являются обязательным. Без него корректная работа вьювера невозможна.

Действие **ViewerEvent** возвращает подготовленную HTML страницу отчета (или набор страниц), построенную с учетом текущего состояния вьювера. При необходимости в указанном действии можно изменить параметры текущего отчета, а также обновить их данные в случае интерактивных действий вьювера.

4.1.8 Печать отчета

Информация

Обратите внимание, что возможность печати доступна только для отчетов, и недоступна для дашбордов.

© 2003-2026 Stimulsoft

В компоненте **HTML5 Viewer** предусмотрено несколько вариантов печати отчета. Каждый имеет свои особенности, достоинства и недостатки.

Печать в PDF (Print to PDF)
Печать будет осуществляться посредством экспортирования отчета в **PDF формат**. К достоинствам можно отнести большую точность расположения и печати элементов отчета по сравнению с остальными вариантами печати. Из недостатков можно упомянуть обязательное наличие установленного в браузере плагина для просмотра PDF файлов (современные браузеры имеют встроенное средство просмотра и печати PDF файлов).

Печать с просмотром (Print with Preview)
Печать отчета будет осуществлена в отдельном всплывающем окне браузера в **HTML формат**. Отчет можно предварительно просмотреть, а затем отправить на принтер или скопировать в другое место в виде текста или HTML-кода. К достоинствам можно отнести кроссбраузерность при печати, отсутствие необходимости установки специальных плагинов. Недостатком является относительно невысокая точность расположения элементов отчета, обусловленная особенностями реализации HTML-форматирования.

Печать без просмотра (Print without Preview)
Печать отчета будет осуществлена непосредственно на принтер без предварительного просмотра. После выбора этого пункта меню отображается системный диалог печати. Так как печать в этом режиме осуществляется в HTML формат, то качество печати аналогично качеству печати отчета с предварительным просмотром.

Информация

При печати в **HTML формат** необходимо убедиться в соответствии параметров страницы отчета и параметров страницы принтера (размер бумаги, ориентация, поля, отступы), а также проверить настройки печати браузера, такие как отступы, колонтитулы, печать фоновых изображений, цветная печать.

Для работы функции печати не требуется дополнительных настроек вьювера. Если необходимо выполнить какие-либо действия перед печатью отчета, то

© 2003-2026 Stimulsoft

можно определить специальное действие **PrintReport**.

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcViewer("MvcViewer1",
   new StiMvcViewerOptions() {
      Actions =
      {
         PrintReport = "PrintReport"
      }
})
...
```

HomeController.cs

```
...
public ActionResult PrintReport()
{
   // Some code before print
   // ...

   return StiMvcViewer.PrintReportResult();
}
...
```

Настройка печати отчета

При выборе печати отчета на панели вьювера, отображается меню с выбором варианта печати. Компонент **HTML5 Viewer** имеет возможность принудительно установить требуемый режим печати. Для этого достаточно установить свойство **PrintDestination** в одно из указанных ниже значений перечисления **StiPrintDestination**.

- **Default** – при выборе печати будет отображено меню (значение свойства по умолчанию);
- **Pdf** – печать в PDF формат;
- **Direct** – печать в HTML формат непосредственно на принтер, будет отображен системный диалог печати;
- **WithPreview** – печать в HTML формат с предварительным просмотром во всплывающем окне.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcViewer("MvcViewer1",
   new StiMvcViewerOptions() {
      Toolbar =
      {
         PrintDestination = StiPrintDestination.Default
      }
})
...
```

Компонент **HTML5 Viewer** имеет возможность полностью отключить печать отчета, если она не требуется. Для этого необходимо установить значение **false** для свойства **ShowPrintButton**.

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcViewer("MvcViewer1",
   new StiMvcViewerOptions() {
      Toolbar =
      {
         ShowPrintButton = false
      }
})
...
```

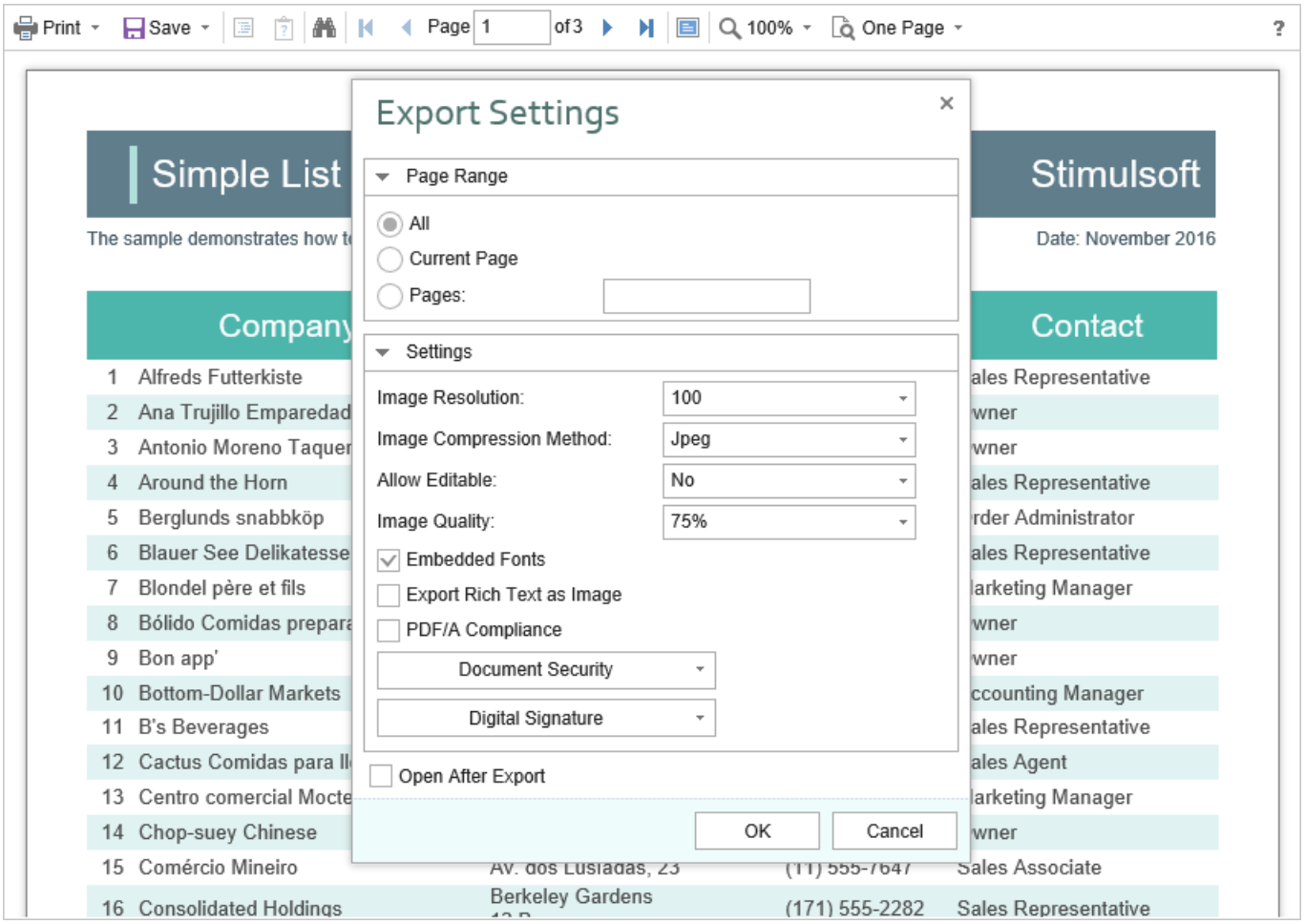
4.1.9 Экспорт отчета и дашборда

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Компонент **HTML5 Viewer** позволяет экспортировать отображаемый отчет в три десятка различных форматов, таких как **PDF**, **HTML**, **Word**, **Excel**, **XPS**, **RTF**, изображения, текст и другие. Экспорт панели индикаторов (дашборда) возможен в PDF, Excel, файлы изображений. Для работы функции экспорта не требуется дополнительных настроек вьювера.

© 2003-2026 Stimulsoft



Для работы функции экспорта не требуется дополнительных настроек вьювера. Если необходимо выполнить какие-либо действия перед экспортированием отчета, то можно определить специальное действие **ExportReport**.

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcViewer("MvcViewer1",
   new StiMvcViewerOptions() {
      Actions =
      {
         ExportReport = "ExportReport"
      }
})
...
```

© 2003-2026 Stimulsoft

HomeController.cs

```
...
public ActionResult ExportReport()
{
   // Some code before export
   // ...

   return StiMvcViewer.ExportReportResult();
}
...
```

Настройки экспорта

Каждый формат экспорта отчета компонента **HTML5 Viewer** имеет множество настроек, и каждая настройка имеет свои значения по умолчанию. Иногда требуется установить другие значения по умолчанию. Для этого предназначено специальное свойство вьювера **DefaultSettings**, которое представляет собой контейнер всех настроек экспортов, используемых по умолчанию.

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcViewer("MvcViewer1",
   new StiMvcViewerOptions() {
      Exports =
      {
         DefaultSettings =
         {
            ExportToPdf =
            {
               ImageQuality = 0.75f,
               ImageFormat = Stimulsoft.Report.Export.StiImageFormat.Color
            },
            ExportToHtml =
            {
               ExportMode = Stimulsoft.Report.Export.StiHtmlExportMode.Div,
               UseEmbeddedImages = true
            }
         }
      }
})
...
```

Если требуется, можно полностью убрать отображение диалоговых окон экспорта, экспортирование всегда будет осуществляться с настройками по умолчанию. Для этого достаточно установить значение **false** для свойства

© 2003-2026 Stimulsoft

ShowExportDialog.

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcViewer("MvcViewer1",
   new StiMvcViewerOptions() {
      Exports =
      {
         ShowExportDialog = false
      }
})
...
```

Компонент **HTML5 Viewer** содержит порядка 30 различных форматов экспорта, и иногда требуется отключить не используемые форматы. Это позволяет разгрузить интерфейс и упростить использование вьювера. Для отключения не используемых форматов экспортов достаточно установить значение **false** для соответствующих свойств вьювера, представленных в списке ниже.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcViewer("MvcViewer1",
   new StiMvcViewerOptions() {
      Exports =
      {
         ShowExportToDocument = true,
         ShowExportToPdf = true,
         ShowExportToXps = true,
         ShowExportToPowerPoint = true,
         ShowExportToHtml = true,
         ShowExportToHtml5 = true,
         ShowExportToMht = true,
         ShowExportToText = true,
         ShowExportToRtf = true,
         ShowExportToWord = true,
         ShowExportToOpenDocumentWriter = true,
         ShowExportToExcel = true,
         ShowExportToExcelXml = true,
         ShowExportToExcelBiff = true,
         ShowExportToOpenDocumentCalc = true,
         ShowExportToCsv = true,
         ShowExportToDbf = true,
         ShowExportToXml = true,
         ShowExportToDif = true,
         ShowExportToSylk = true,
         ShowExportToImageBmp = true,
         ShowExportToImageGif = true,
         ShowExportToImageJpeg = true,
         ShowExportToImagePcx = true,
         ShowExportToImagePng = true,
         ShowExportToImageTiff = true,
         ShowExportToImageMetafile = true,
         ShowExportToImageSvg = true,
         ShowExportToImageSvgz = true
      }
})
...
```

Компонент **HTML5 Viewer** имеет возможность полностью отключить меню экспорта отчета, если оно не требуется. Для этого необходимо установить значение **false** для свойства **ShowSaveButton**.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcViewer("MvcViewer1",
   new StiMvcViewerOptions() {
      Toolbar =
      {
         ShowSaveButton = false
      }
})
...
```

4.1.10 Режимы отображения

В компоненте **HTML5 Viewer** предусмотрено два режима отображения отчета: с полосами прокрутки и без них. По умолчанию установлен режим просмотра без полос прокрутки. Для включения режима просмотра с полосами прокрутки достаточно установить значение **true** для свойства **ScrollbarsMode**.

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcViewer("MvcViewer1",
   new StiMvcViewerOptions() {
      Appearance =
      {
         ScrollbarsMode = true
      }
})
...
```

В первом режиме (без полос прокрутки), вьювер отображает страницу или отчет целиком, автоматически растягивая область просмотра. Если заданы размеры по ширине и высоте, то вьювер будет осуществлять обрезку вышедшей за границы страницы. Во втором режиме, в отличие от первого, при выходе страницы за границы размеров вьювера, обрезка осуществляться не будет. Вместо этого появятся полосы прокрутки, при помощи которых можно просмотреть страницу или отчет целиком.

Информация

В режиме просмотра отчета с полосами прокрутки необходимо задать высоту вьювера, иначе будет задана высота по умолчанию, равная **650 пикселей**.

© 2003-2026 Stimulsoft

В компоненте **HTML5 Viewer** предусмотрен режим полноэкранного отображения отчета или дашборда. По умолчанию включен стандартный режим просмотра, вьювер имеет заданные в настройках размеры. Для включения полноэкранного режима просмотра достаточно установить значение **true** для свойства **FullScreenMode**.

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcViewer("MvcViewer1",
   new StiMvcViewerOptions() {
      Appearance =
      {
         FullScreenMode = true
      }
})
...
```

Также, для включения либо отключения полноэкранного режима можно воспользоваться соответствующей кнопкой на панели управления вьювера.

В компоненте **HTML5 Viewer** доступно три режима отображения отчета: постраничное отображение, отчет целиком в виде ленты, и табличное отображение страниц отчета. Для управления режимами предназначено свойство **ViewMode**, которое соответственно принимает одно из указанных значений: **SinglePage**, **Continuous**, **MultiplePages**.

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcViewer("MvcViewer1",
   new StiMvcViewerOptions() {
      Toolbar =
      {
         ViewMode = StiWebViewMode.SinglePage
      }
})
...
```

В компоненте **HTML5 Viewer** реализована поддержка работы как с обычным компьютером, так и с сенсорными экранами, мобильными устройствами. Для управления режимами интерфейса предназначено свойство **InterfaceType**, которое принимает одно из следующих значений:

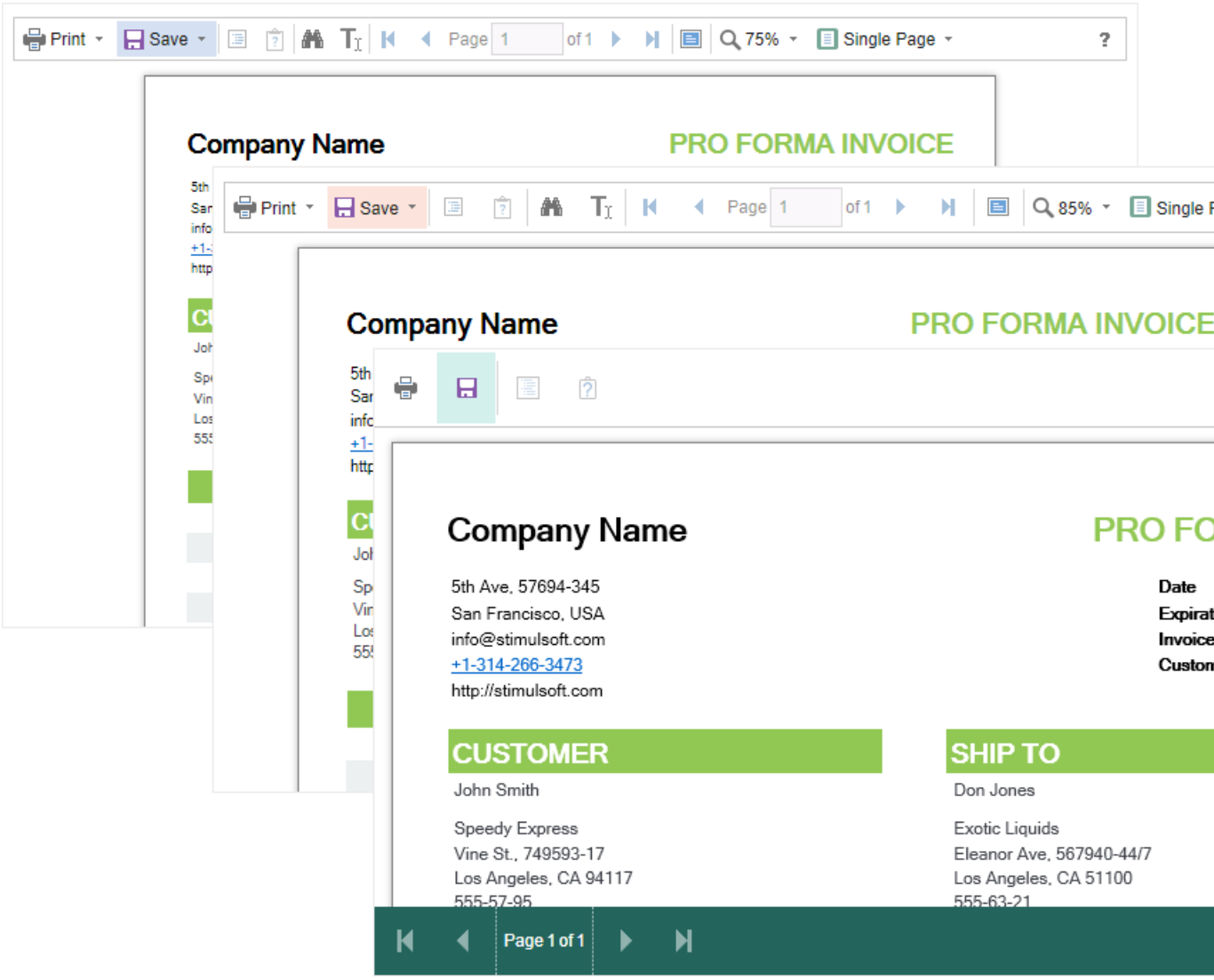
© 2003-2026 Stimulsoft

- **Auto** – тип интерфейса вьювера будет выбран автоматически в зависимости от используемого устройства (значение по умолчанию);
- **Mouse** – принудительное использование стандартного интерфейса для управления вьювером при помощи мыши;
- **Touch** – принудительное использование Touch интерфейса для управления вьювером при помощи сенсорного экрана монитора, в этом режиме элементы интерфейса вьювера имеют увеличенные размеры для удобства управления;
- **Mobile** – принудительное использование Mobile интерфейса для управления вьювером при помощи экрана смартфона, в этом режиме интерфейс вьювера имеет упрощенный вид и адаптирован для управления при помощи мобильного устройства.

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcViewer("MvcViewer1",
   new StiMvcViewerOptions() {
      Appearance =
      {
         InterfaceType = StiInterfaceType.Auto
      }
})
...
```

© 2003-2026 Stimulsoft



4.1.11 Работа с параметрами

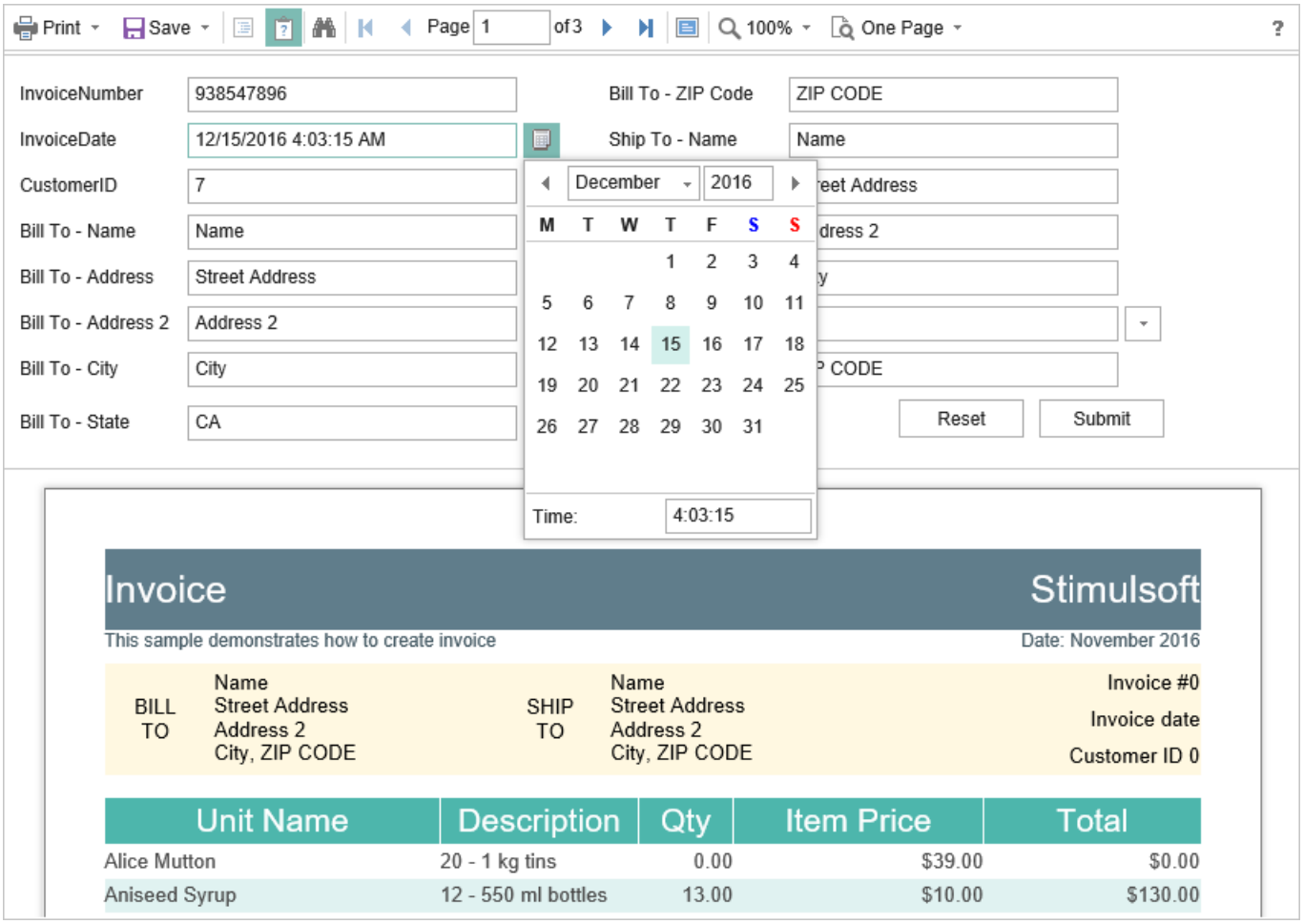
Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для работы с параметрами отчета в **HTML5 Viewer** реализована поддержка специальной панели параметров. Для добавления параметра на панель

© 2003-2026 Stimulsoft

необходимо в отчете определить переменную, запрашиваемую у пользователя. При просмотре отчета во вьювере такая переменная будет автоматически добавлена на панель параметров. Поддерживаются все типы переменных отчета (обычные переменные, дата и время, границы, списки и др.).



Для работы отчетов с параметрами не требуется дополнительных настроек вьювера. Если необходимо провести какие-либо действия перед применением параметров, то можно определить специальное действие **Interaction**.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcViewer("MvcViewer1",
   new StiMvcViewerOptions() {
      Actions =
      {
         Interaction = "ViewerInteraction"
      }
})
...
```

HomeController.cs

```
...
public ActionResult ViewerInteraction()
{
   // Some code before any interaction
   // ...

   return StiMvcViewer.InteractionResult();
}
...
```

Данное действие вызывается при любых интерактивных действиях вьювера. При необходимости выполнить какие-либо действия только при применении параметров отчета, можно воспользоваться параметрами вьювера. Параметры вьювера представлены в виде объекта класса **StiRequestParams**, они передаются при любом запросе на сторону сервера, и содержат все необходимые сведения и состояния клиентской части вьювера. Для определения типа действия вьювера достаточно проверить свойство **Action** у параметров вьювера.

HomeController.cs

```
...
public ActionResult ViewerInteraction()
{
   StiRequestParams requestParams = StiMvcViewer.GetRequestParams();
   if (requestParams.Action == StiAction.Variables)
   {
      // Some code before apply parameters
   }

   return StiMvcViewer.InteractionResult();
}
...
```

© 2003-2026 Stimulsoft

Если работа с параметрами не требуется, то можно полностью отключить данную возможность. Для этого предназначено свойство **ShowParametersButton** в разделе свойств **Toolbar**, для которого необходимо установить значение **false**.

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcViewer("MvcViewer1",
   new StiMvcViewerOptions() {
      Toolbar =
      {
         ShowParametersButton = false
      }
})
...
```

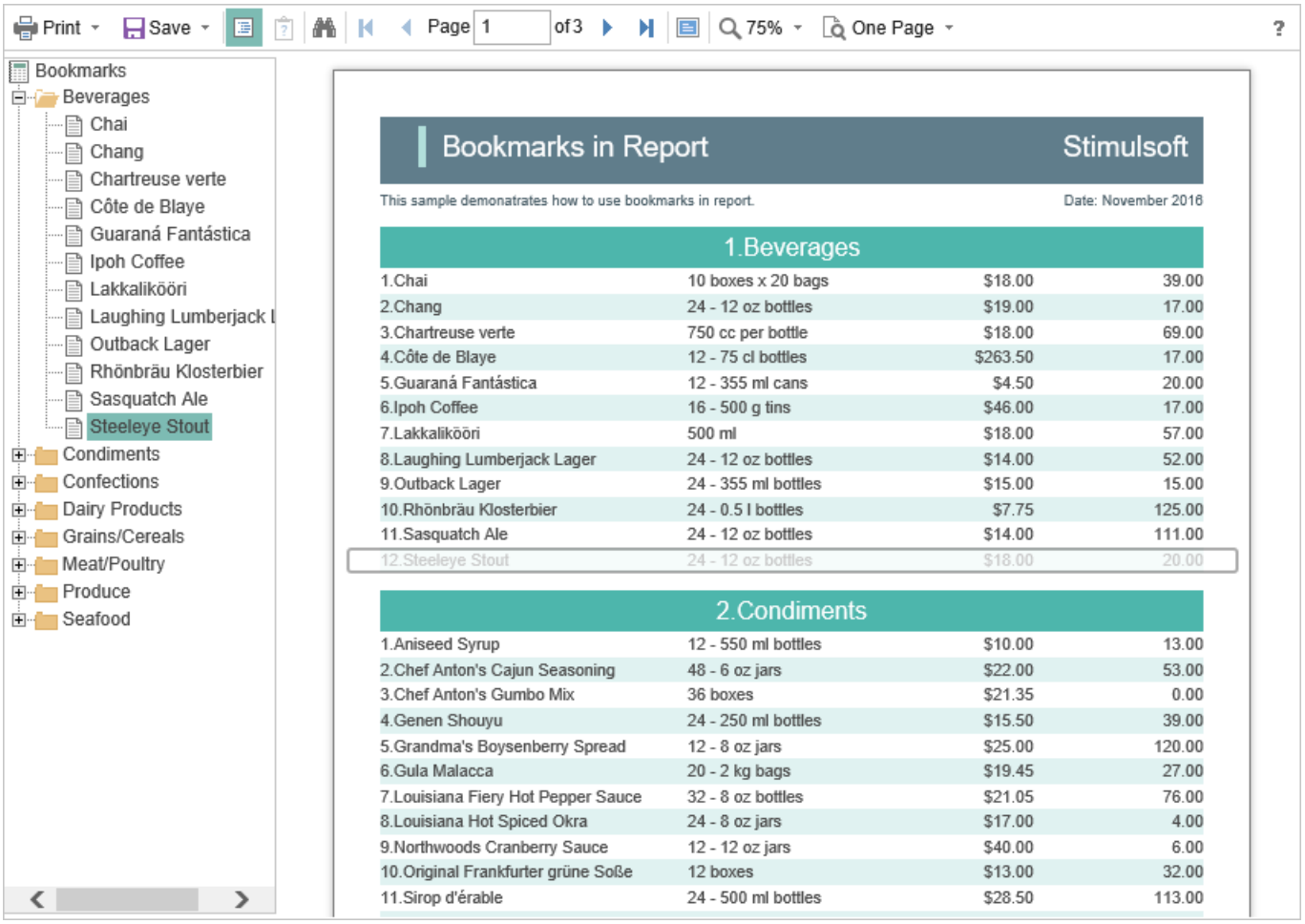
Информация

При такой конфигурации вьювера панель параметров не будет показана даже в том случае, если параметры присутствуют в отображаемом отчете.

4.1.12 Работа с закладками

В компоненте **HTML5 Viewer** реализована поддержка закладок отчета. При отображении такого отчета слева от страницы будет отображена панель с закладками. При выборе закладки отчета вьювер осуществит автоматический переход на нужную страницу, а элемент отчета с закладкой будет подсвечен.

© 2003-2026 Stimulsoft



По умолчанию ширина панели закладок равна 180 пикселей, компонент **HTML5 Viewer** позволяет изменить это значение. Для этого предназначено свойство **BookmarksTreeWidth**, значение которого указывается в пикселях.

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcViewer("MvcViewer1",
   new StiMvcViewerOptions() {
      Appearance =
      {
         BookmarksTreeWidth = 200
      }
})
...
```

Если работа с закладками отчета не требуется, то можно полностью отключить

© 2003-2026 Stimulsoft

данную возможность. Для этого предназначено свойство **ShowBookmarksButton**, которое необходимо установить в значение **false**.

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcViewer("MvcViewer1",
   new StiMvcViewerOptions() {
      Toolbar =
      {
         ShowBookmarksButton = false
      }
})
...
```

Информация

В этом случае закладки отчета не будут показаны, даже если они присутствуют в отображаемом отчете. Данное свойство не оказывает влияние на печать и экспортирование отчета.

При печати отчета с закладками дерево закладок будет скрыто. Если помимо самого отчета требуется распечатать и закладки, то необходимо установить свойство **BookmarksPrint** в значение **true**.

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcViewer("MvcViewer1",
   new StiMvcViewerOptions() {
      Appearance =
      {
         BookmarksPrint = true
      }
})
...
```

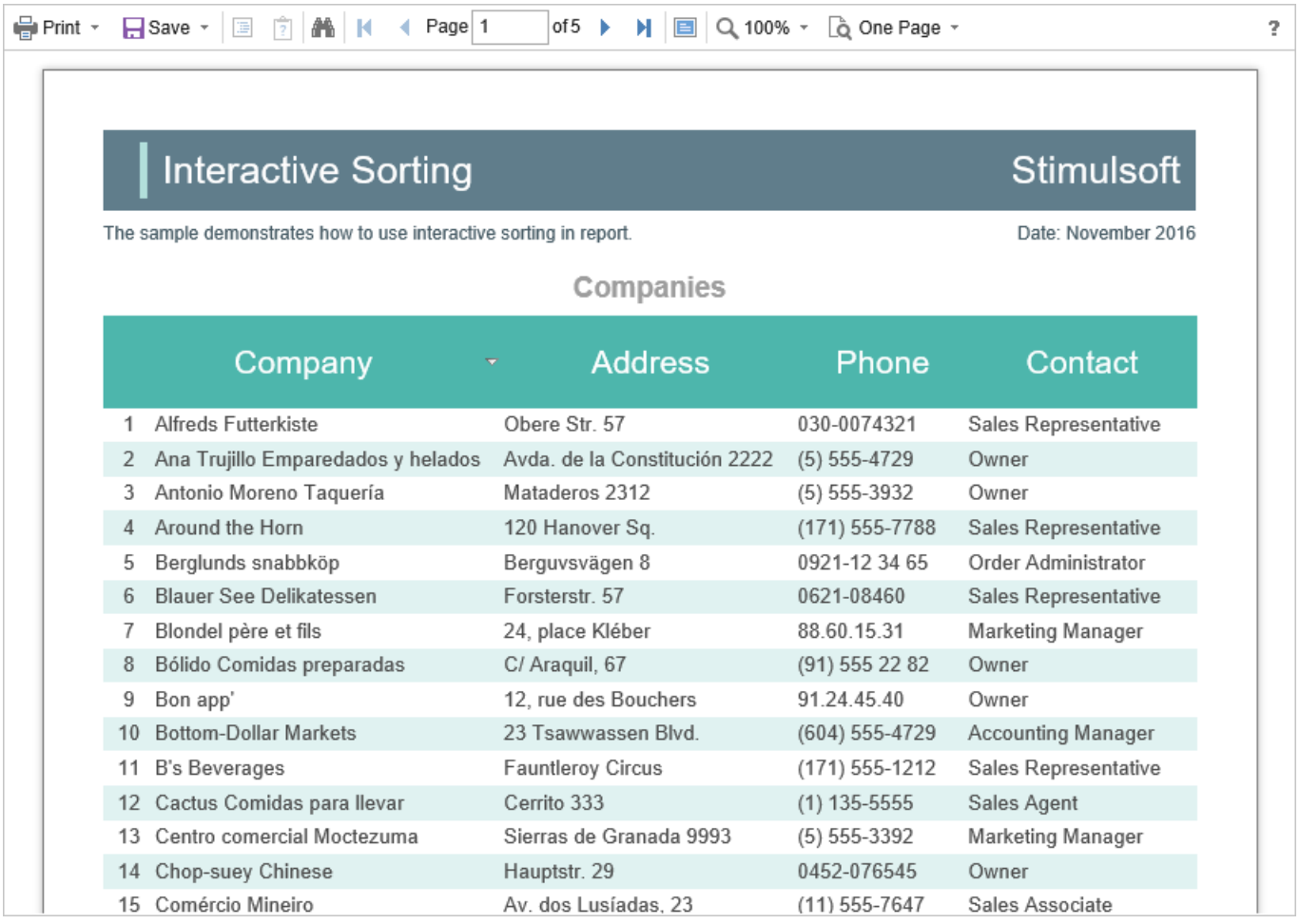
4.1.13 Динамическое сворачивание, сортировка и детализация

В компоненте **HTML5 Viewer** реализована поддержка динамической сортировки, сворачивания и детализации отчетов. Динамическая сортировка предоставляет возможность изменять направление сортировки в построенном отчете. Для этого следует щелкнуть по компоненту, у которого была установлена динамическая сортировка. Динамическая сортировка

© 2003-2026 Stimulsoft

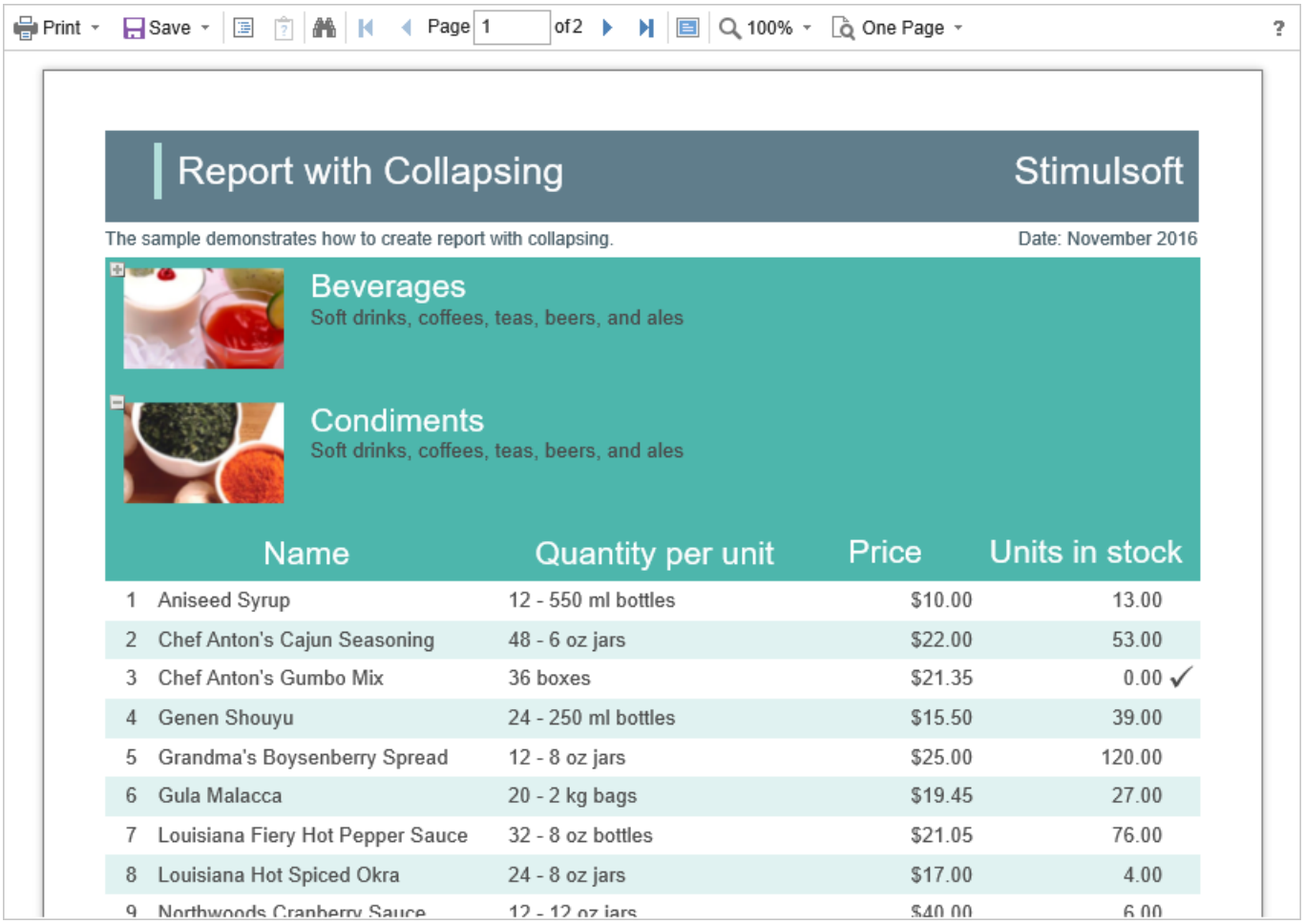
осуществляется в следующих направлениях: **По возрастанию (Ascending)** и **По убыванию (Descending)**. Каждый раз при щелчке по компоненту направление меняется на противоположное.

Допускается многоуровневая сортировка в отчете. Для этого необходимо удерживать клавишу **Ctrl** и последовательно нажимать на сортируемые компоненты отчета. Для сброса сортировки можно нажать на любой сортируемый компонент без удержания клавиши **Ctrl**.



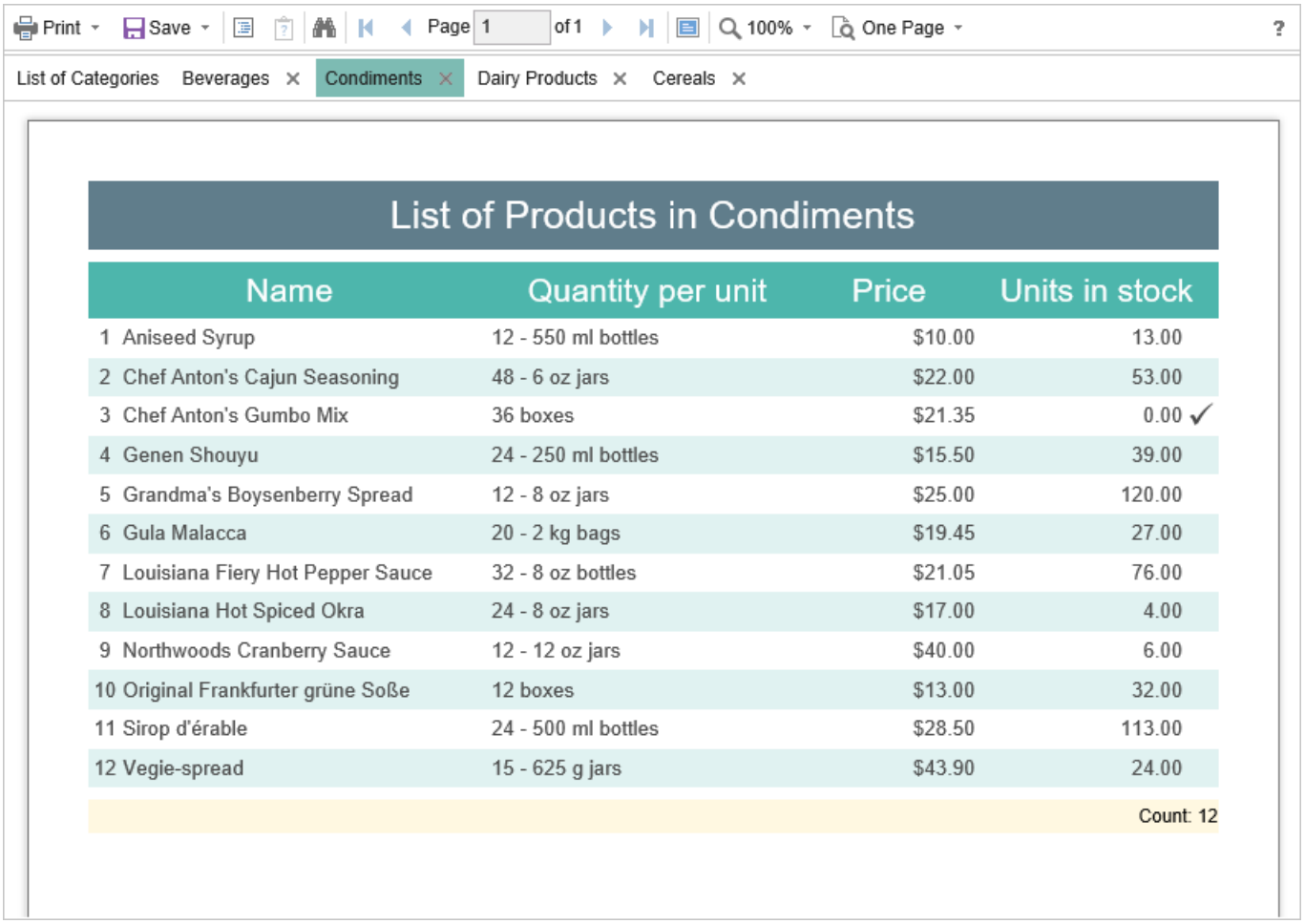
Отчет с динамическим сворачиванием представляет собой интерактивный отчет, в котором сворачивающиеся блоки могут сворачивать/разворачивать свое содержимое при нажатии на заголовок блока. Элементы отчета, которые можно свернуть/развернуть, обозначены специальной иконкой со знаком **[-]** или **[+]**.

© 2003-2026 Stimulsoft



При детализации данных под главной панелью вьювера будет отображена панель детализации с закладками детализированных отчетов. Отображенный в данный момент отчет будет подсвечен.

© 2003-2026 Stimulsoft



Для работы динамической сортировки, сворачивания и детализации отчетов не требуется дополнительных настроек вьювера. Для выполнения каких-либо действий перед сортировкой, сворачиванием или детализацией отчета предназначено специальное действие **Interaction**, которое будет вызвано при любом интерактивном действии вьювера.

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcViewer("MvcViewer1",
   new StiMvcViewerOptions() {
      Actions =
      {
         Interaction = "ViewerInteraction"
      }
})
...
```

© 2003-2026 Stimulsoft

HomeController.cs

```
...
public ActionResult ViewerInteraction()
{
   // Some code before any interaction
   // ...

   return StiMvcViewer.InteractionResult();
}
...
```

Для получения типа действия можно воспользоваться параметрами вьювера. Параметры вьювера представлены в виде объекта класса **StiRequestParams**, они передаются при любом запросе на сторону сервера, и содержат все необходимые сведения и состояния клиентской части вьювера. Для каждого вида интерактивности вьювера предусмотрен определенный тип действия:

- **Sorting** – при использовании сортировки колонок;
- **DrillDown** – при использовании детализации отчета;
- **Collapsing** – при использовании сворачивания блоков отчета.

HomeController.cs

```
...
public ActionResult ViewerInteraction()
{
   StiRequestParams requestParams = StiMvcViewer.GetRequestParams();
   switch (requestParams.Action)
   {
      case StiAction.Sorting:
         break;

      case StiAction.DrillDown:
         break;

      case StiAction.Collapsing:
         break;
   }

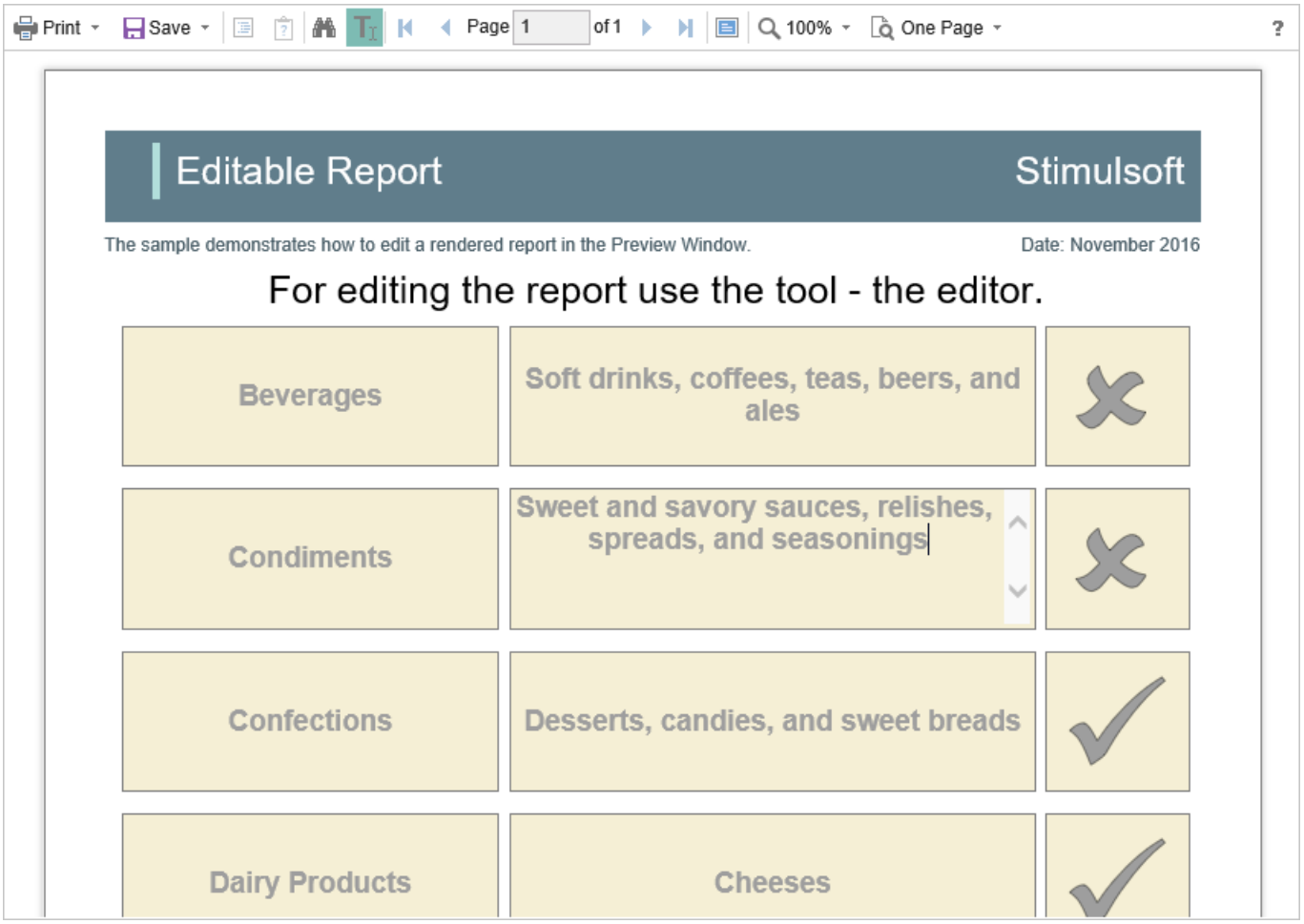
   return StiMvcViewer.InteractionResult();
}
...
```

4.1.14 Редактирование построенного отчета

Компонент **HTML5 Viewer** имеет возможность редактировать элементы построенного отчета, такие как текстовые поля и чекбоксы. Для того, чтобы редактирование стало возможным, в самом шаблоне отчета необходимо пометить нужные компоненты как редактируемые. После отображения отчета

© 2003-2026 Stimulsoft

во вьювере, для начала редактирования необходимо нажать на соответствующую кнопку на панели вьювера. После завершения редактирования необходимо нажать указанную кнопку еще раз, и все внесенные изменения будут примерены к отчету.



Для работы режима редактирования отчета не требуется никаких специальных настроек вьювера.

Информация

Отредактированные значения будут применены при печати либо экспортировании отчета, при этом исходный отчет останется не тронутым. После перезагрузки вьювера все значения будут возвращены к исходным.

© 2003-2026 Stimulsoft

4.1.15 Отправка отчета по Email

Информация

Обратите внимание, что возможность Отправить отчет по Email доступна только для отчетов, и недоступна для дашбордов.

В компоненте **HTML5 Viewer** предусмотрена возможность отправки отчета по Email. Для активации данной возможности необходимо установить свойство вьювера **ShowSendEmailButton** в значение **true** и определить действие **EmailReport**.

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcViewer("MvcViewer1",
   new StiMvcViewerOptions() {
      Actions =
      {
         EmailReport = "EmailReport"
      },
      Toolbar =
      {
         ShowSendEmailButton = true
      }
})
...
```

© 2003-2026 Stimulsoft

HomeController.cs

```
...
public ActionResult EmailReport()
{
   StiEmailOptions options = StiMvcViewer.GetEmailOptions();

   // Passed from the viewer, can be checked and changed
   // options.AddressTo = "";
   // options.Subject = "";
   // options.Body = "";

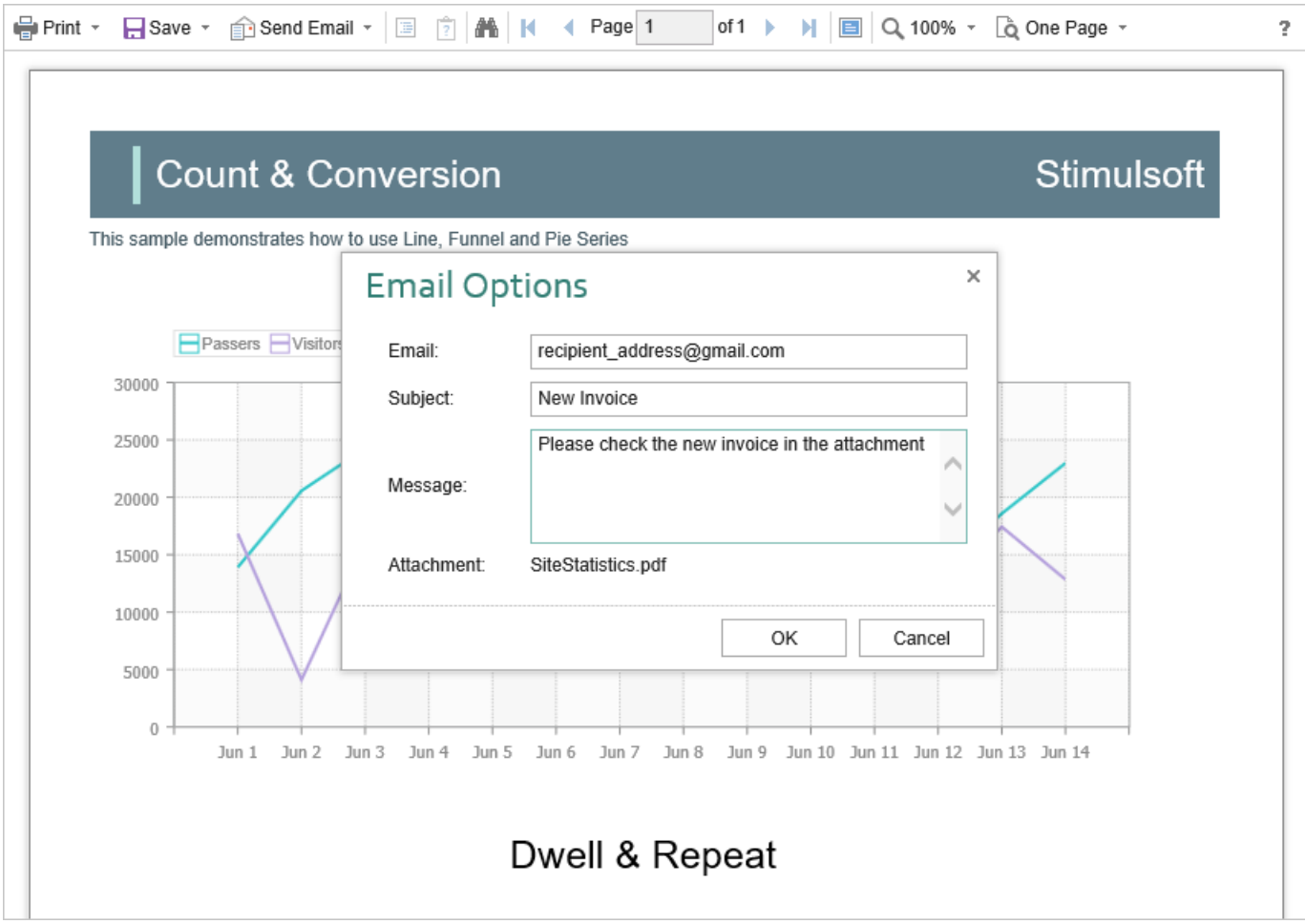
   // Should be filled here
   options.AddressFrom = "admin_address@test.com";
   options.Host = "smtp.test.com";
   options.Port = 465;
   options.UserName = "admin_address@test.com";
   options.Password = "admin_password";

   // options.CC.Add("email@test.com");
   // options.BCC.Add("email@test.com");
   // options.EnableSsl = true;

   return StiMvcViewer.EmailReportResult(options);
}
...
```

При отправке отчета по Email отображается меню выбора формата вложения, которое соответствует меню выбора формата экспортирования отчета. После выбора формата отобразиться диалог ввода параметров отправки, таких как Email получателя, тема и текст письма.

© 2003-2026 Stimulsoft



После подтверждения отправки будет вызвано описанное выше действие **EmailReport**, в котором можно проверить и скорректировать данные, введенные в этой форме. Экспортированный файл отчета будет прикреплен к письму автоматически.

Компонент **HTML5 Viewer** позволяет установить значения по умолчанию для формы отправки Email. Для этого предназначены свойства **DefaultEmailAddress**, **DefaultEmailSubject** и **DefaultEmailMessage**. По умолчанию данные свойства являются пустыми.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcViewer("MvcViewer1",
   new StiMvcViewerOptions() {
      Email =
      {
         DefaultEmailAddress = "recipient_address@gmail.com",
         DefaultEmailSubject = "New Invoice",
         DefaultEmailMessage = "Please check the new invoice in the
         attachment"
      }
})
...
```

4.1.16 Вызов дизайнера из вьювера

Компонент **HTML5 Viewer** имеет возможность вызвать дизайнер отчетов. Для этого предназначена специальная кнопка **Дизайн (Design)** на панели инструментов вьювера (по умолчанию данная кнопка отключена). Для использования этой возможности необходимо установить свойство **ShowDesignButton** в значение **true**, а также определить действие **DesignReport**.

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcViewer("MvcViewer1",
   new StiMvcViewerOptions() {
      Actions =
      {
         DesignReport = "DesignReport"
      },
      Toolbar =
      {
         ShowDesignButton = true
      }
})
...
```

HomeController.cs

```
...
public ActionResult DesignReport()
{
   StiReport report = StiMvcViewer.GetReportObject();
   ViewBag.ReportName = report.ReportName;

   return View("Designer");
}
...
```

© 2003-2026 Stimulsoft

Информация

Сам вьювер не запускает дизайнер, он лишь вызывает указанное действие, в котором можно получить все необходимые параметры. Далее в действии можно выполнить перенаправление на другой View, который содержит дизайнер отчетов.

4.1.17 Кэширование

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Компонент **HTML5 Viewer** позволяет использовать кэш сервера для хранения построенного отчета. Если кэширование не используется, то при каждом запросе страницы необходимо загружать отчет, подключать данные и строить его заново. Если использовать кэширование, то при следующем обновлении страницы, ранее построенный отчет будет загружен из кэша.

При использовании кэширования стоит учитывать, что каждый сохраненный отчет в кэше занимает память сервера и при большом количестве запросов к ним это может стать критичным фактором. Поэтому необходимо выбирать из двух вариантов – или маленькие потребности к памяти, но большие к производительности, или небольшие потребности к производительности, но большие к памяти.

Управлять кэшированием можно при помощи следующих приведенных ниже свойств.

Свойство CacheMode

© 2003-2026 Stimulsoft

Данное свойство вьювера включает кэширование и устанавливает его тип. Может принимать одно из значений, указанных в перечислении **StiServerCacheMode**:

- **None** – кэширование отключено, каждое действие вьювера требует загрузки отчета и, если это шаблон, последующее его построение;
- **ObjectCache** – для кэширования используется кэш сервера, в котором сохраняется объект отчета (значение по умолчанию);
- **StringCache** - для кэширования используется кэш сервера, в котором сохраняется отчет в виде упакованной строки;
- **ObjectSession** - для кэширования используется текущая сессия, в которой сохраняется объект отчета;
- **StringSession** - для кэширования используется текущая сессия, в которой сохраняется отчет в виде упакованной строки.

Свойство CacheItemPriority

Данное свойство устанавливает приоритет хранимого в кэше сервера отчета, влияет на автоматическую очистку памяти сервера в случае ее нехватки. Чем меньше приоритет, тем больше шанс удаления информации из памяти.

Свойство CacheTimeout

Данное свойство указывает величину времени в минутах, в течение которого необходимо сохранять отчет в кэше сервера. Если при использовании кэширования требуемый отчет не найден в кэше (т.е. истекло время хранения объекта), то он будет запрошен заново при помощи специального действия **GetReport** с последующим подключением данных отчета и его построением.

Компонент **HTML5 Viewer** предоставляет возможность определить собственные методы работы с кэшированием отчета. Для этого предназначен специальный класс **StiCacheHelper**, который содержит методы получения отчета из кэша и сохранения отчета в кэш. Необходимо создать новый класс,

© 2003-2026 Stimulsoft

унаследованный от **StiCacheHelper**, и перегрузить указанные выше методы, которые соответственно имеют названия **GetReport** и **SaveReport**.

HomeController.cs

```
...
public class ViewerController : Controller
{
   public class StiMyCacheHelper : StiCacheHelper
   {
      public override StiReport GetReport(string guid)
      {
         string path =
         System.IO.Path.Combine(System.Web.HttpContext.Current.Server.MapPat
         h("~/"), "CacheFiles", guid);
         if (System.IO.File.Exists(path))
         {
            StiReport report = new StiReport();
            string packedReport = System.IO.File.ReadAllText(path);
            if (guid.EndsWith("template"))
            report.LoadPackedReportFromString(packedReport);
            else report.LoadPackedDocumentFromString(packedReport);

            return report;
         }
         return null;

         //return base.GetReport(guid);
      }

      public override void SaveReport(StiReport report, string guid)
      {
         string packedReport = guid.EndsWith("template") ?
         report.SavePackedReportToString() :
         report.SavePackedDocumentToString();
         string path =
         System.IO.Path.Combine(System.Web.HttpContext.Current.Server.MapPat
         h("~/"), "CacheFiles", guid);
         System.IO.File.WriteAllText(path, packedReport);

         //base.SaveReport(report, guid);
      }
   }

   static ViewerController()
   {
      StiMvcViewer.CacheHelper = new StiMyCacheHelper();
   }
}
...
```

Для инициализации работы с кэшированием отчета при помощи созданного класса, достаточно задать его в качестве значения статического свойства

© 2003-2026 Stimulsoft

StiMvcViewer.CacheHelper в конструкторе контроллера.

Информация

При отключенном кэшировании отчета (свойство вьювера **CacheMode** имеет значение **None**), указанный класс задействован не будет.

4.1.18 Вспомогательные методы

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для **HTML5 Viewer** предусмотрено несколько вспомогательных методов, которые предназначены для получения объекта просматриваемого в данный момент отчета, параметров текущего состояния вьювера и других полезных данных. Эти методы можно использовать в любых действиях вьювера.

Метод GetReportObject()

Возвращает объект отчета, с которым в данный момент работает вьювер. Допускается производить с ним необходимые манипуляции - регистрировать новые наборы данных, изменять свойства отчета, присваивать параметры либо загружать в объект другой отчет. Затем отчет можно вернуть вьюверу, указав его в качестве параметра в результирующем методе действия.

© 2003-2026 Stimulsoft

HomeController.cs

```
...
public ActionResult ViewerInteraction()
{
   StiReport report = StiMvcViewer.GetReportObject();
   report.ReportName = "MyReportName";

   return StiMvcViewer.InteractionResult(report);
}
...
```

Метод GetRouteValues()

Возвращает значения маршрутов для URL, с которыми была открыта страница вьювера. Таким образом, предоставляется возможность получить исходную коллекцию параметров страницы запуска в любом действии вьювера и использовать эти значения для каких-либо проверок и условий.

HomeController.cs

```
...
public ActionResult ViewerInteraction()
{
   RouteValueDictionary routeValues = StiMvcViewer.GetRouteValues();

   return StiMvcViewer.InteractionResult();
}
...
```

Также можно получить значения параметров URL по имени параметра, указав его в качестве параметра вызываемого действия вьювера.

HomeController.cs

```
...
public ActionResult ViewerInteraction(string id)
{
   return StiMvcViewer.InteractionResult();
}
...
```

Метод GetFormValues()

© 2003-2026 Stimulsoft

Возвращает значения формы, которая инициировала (открыла POST-запросом) страницу вьювера. Таким образом предоставляется возможность получить коллекцию параметров формы в любом действии вьювера.

HomeController.cs

```
...
public ActionResult ViewerInteraction()
{
   NameValueCollection formValues = StiMvcViewer.GetFormValues();

   return StiMvcViewer.InteractionResult();
}
...
```

По умолчанию данная возможность отключена в целях оптимизации запросов клиентской стороны вьювера на сервер. Для включения достаточно установить свойство **PassFormValues** в значение **true**.

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcViewer("MvcViewer1",
   new StiMvcViewerOptions() {
      Server =
      {
         PassFormValues = true
      }
})
...
```

Метод GetRequestParams()

Возвращает все параметры текущего состояния вьювера, переданные на сторону сервера. Они могут быть полезны для определения типа действия, которое в данный момент выполняет вьювер – например, для определения типа экспорта, а также всех параметров действия.

© 2003-2026 Stimulsoft

HomeController.cs

```
...
public ActionResult ExportReport()
{
   StiRequestParams requestParams = StiMvcViewer.GetRequestParams();
   if (requestParams.ExportFormat == StiExportFormat.Pdf)
   {
      StiReport report = StiMvcViewer.GetReportObject();

      // Some action with report for the PDF export
      // ...

      return StiMvcViewer.ExportReportResult(report);
   }

   return StiMvcViewer.ExportReportResult();
}
...
```

Допускается изменять значения некоторых параметров. После внесения изменений, для корректной работы вьювера необходимо передать измененный объект параметров на вход результирующего метода.

HomeController.cs

```
...
public ActionResult ViewerInteraction()
{
   StiRequestParams requestParams = StiMvcViewer.GetRequestParams();
   if (requestParams.Action == StiAction.Variables)
   {
      requestParams.Interaction.Variables["Variable1"] = "MyValue";
      return StiMvcViewer.InteractionResult(requestParams);
   }

   return StiMvcViewer.InteractionResult();
}
...
```

Метод GetExportSettings()

Возвращает все параметры текущего экспортирования отчета. Тип объекта параметров будет соответствовать типу экспорта, выбранному в меню вьювера. Любые параметры экспорта можно изменить и передать на вход результирующего метода. В этом случае, экспортирование отчета будет произведено с переданными параметрами.

© 2003-2026 Stimulsoft

HomeController.cs

```
...
public ActionResult ExportReport()
{
   StiExportSettings settings = StiMvcViewer.GetExportSettings();
   if (settings.GetExportFormat() == StiExportFormat.Pdf)
   {
      StiPdfExportSettings pdfSettings = (StiPdfExportSettings)settings;
      pdfSettings.EmbeddedFonts = true;
      pdfSettings.AllowEditable = StiPdfAllowEditable.No;
      return StiMvcViewer.ExportReportResult(settings);
   }

   return StiMvcViewer.ExportReportResult();
}
...
```

4.1.19 Экспорт и печать из кода

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет». Кроме этого, следует понимать, что возможность печати из кода доступна только для отчетов, и недоступна для дашбордов.

Компонент **HTML5 Viewer** предоставляет возможность печатать отчет различными способами и выполнять экспортирование отчета в различные форматы. Эти действия выполняются при помощи меню вьювера. Если требуется напечатать отчет или выполнить экспорт отчета при помощи кода, например, в определенном действии контроллера, то можно воспользоваться специальным классом **StiMvcReportResponse**. Данный класс содержит набор статических методов, позволяющих выполнить печать отчета либо экспорт отчета из кода, при этом вьювер отчетов не требуется.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.ActionLink("Print Report from Code", "PrintReport")
<br />
@Html.ActionLink("Export Report from Code", "ExportReport")
...
```

HomeController.cs

```
...
private StiReport LoadSimpleList()
{
   DataSet dataSet = new DataSet();
   dataSet.ReadXml(Server.MapPath("Reports/Demo.xml"));

   StiReport report = new StiReport();
   report.Load(Server.MapPath("Reports/SimpleList.mrt"));
   report.RegData(dataSet);

   return report;
}

public ActionResult PrintReport()
{
   StiReport report = LoadSimpleList();

   return StiMvcReportResponse.PrintAsPdf(report);
   //return StiMvcReportResponse.PrintAsHtml(report);
}

public ActionResult ExportReport()
{
   StiReport report = LoadSimpleList();

   return StiMvcReportResponse.ResponseAsPdf(report);
   //return StiMvcReportResponse.ResponseAsExcel2007(report);
   //return StiMvcReportResponse.ResponseAsText(report);
   //StiMvcReportResponse.ResponseAsJson(report);
}
...
```

Класс **StiMvcReportResponse** содержит методы печати отчета в формате PDF и HTML, а также методы экспортирования отчета в любой из поддерживаемых форматов. В качестве аргументов методы могут принимать различные настройки экспортов, режимы отображения и варианты сохранения полученных файлов.

© 2003-2026 Stimulsoft

4.1.20 Время ожидания

При работе с компонентом **StiMvcViewer** можно установить время ожидания (timeout) выполнения различных операций - хранение отчета в кэше, ответ сервера, выполнение запроса. Настройка времени ожидания выполняется при помощи свойств компонента и опций отчета.

Свойство CacheTimeout

Предоставляет возможность установить время в минутах, которое сервер будет хранить построенный отчет в кэше с момента последнего действия вьювера. По умолчанию установлено значение 10 минут.

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcViewer("MvcViewer1",
   new StiMvcViewerOptions() {
      Server =
      {
         CacheTimeout = 10
      }
})
...
```

Использование кэша увеличивает скорость работы вьювера с отчётом. Более подробно можно ознакомиться в главе Кэширование.

Свойство RequestTimeout

Предоставляет возможность установить время ожидания ответа от сервера в секундах, по истечении которого будет выдана ошибка связи с сервером. По умолчанию установлено значение 30 секунд. Для больших отчетов рекомендуется увеличить данное значение.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcViewer("MvcViewer1",
   new StiMvcViewerOptions() {
      Server =
      {
         RequestTimeout = 30
      }
})
...
```

Опция CommandTimeout

Предоставляет возможность установить время ожидания запроса в секундах, при использовании SQL источников данных в отчете. Значение данного свойства сохраняется в самом шаблоне отчёта для каждого SQL подключения отдельно.

Ниже приведен пример кода, который предоставляет возможность установить время ожидания запроса для уже созданного MS SQL соединения и источников данных в отчете.

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcViewer("MvcViewer1",
   new StiMvcViewerOptions() {
      Actions =
      {
         GetReport = "GetReport",
         ViewerEvent = "ViewerEvent"
      }
})
...
```

© 2003-2026 Stimulsoft

HomeController.cs

```
...
public ActionResult GetReport()
{
   StiReport report = new StiReport();
   report.Load(Server.MapPath("Report.mrt"));
   ((StiSqlSource)
   report.Dictionary.DataSources["DataSourceName"]).CommandTimeout = 1000;

   return StiMvcViewer.GetReportResult(report);
}

public ActionResult ViewerEvent()
{
   return StiMvcViewer.ViewerEventResult();
}
...
```

4.1.21 Настройки вьювера

Настройка **HTML5 Viewer** выполняется при помощи свойств, находящихся в классе **StiMvcViewerOptions**. Все свойства разделены на группы, некоторые из групп содержат свои подгруппы для удобства использования. Ниже приведен пример установки свойств вьювера.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcViewer("MvcViewer1",
   new StiMvcViewerOptions() {
      Theme = StiViewerTheme.Office2022WhiteTeal,
      Localization = "~/Content/Localization/en.xml",
      Actions =
      {
        GetReport = "GetReport",
        ViewerEvent = "ViewerEvent"
      },
      Appearance =
      {
        ReportDisplayMode = StiReportDisplayMode.Auto,
        InterfaceType = StiInterfaceType.Auto,
        ScrollbarsMode = true,
        ShowTooltips = false
      },
      Exports =
      {
        DefaultSettings =
        {
          ExportToPdf =
          {
            CreatorString = "Company Name",
            ImageQuality = 0.75f
          }
        },
        ShowExportToDbf = false,
        ShowExportToDif = false
      }
})
...
```

Обратите внимание, что все элементы дашбордов имеют собственные кнопки экспорта и полноэкранного режима просмотра. Специальных опций для управления их отображением не предусмотрено, но их можно выключить через свойства элемента. Нижеприведенный код необходимо добавить после загрузки отчета, перед его передачей вьюверу.

Default.aspx.cs

```
...
var dbsElementInteraction = (report.GetComponentByName("RegionMap1") as
Stimulsoft.Report.Dashboard.IStiElementInteraction).DashboardInteraction;
(dbsElementInteraction as
Stimulsoft.Report.Dashboard.IStiInteractionLayout).ShowFullScreenButton =
false;
(dbsElementInteraction as
Stimulsoft.Report.Dashboard.IStiInteractionLayout).ShowSaveButton = false;
...
```

© 2003-2026 Stimulsoft

Главные настройки (без группы)

Наименование	Описание
Theme	Задает тему оформления вьювера. Список доступных тем оформления находится в перечислении **StiViewerTheme**. По умолчанию установлено значение **Office2022WhiteBlue**.
Localization	Задает путь к XML файлу локализации. Путь может быть абсолютным либо относительным. По умолчанию используется английская локализация, которая встроена во вьювер и не требует дополнительных XML файлов.
Width	Задает ширину компонента в необходимых единицах измерения, которые определены в классе **Unit**. Доступна установка значения в пикселях: **Unit.Pixel()**, поинтах: **Unit.Point()** и процентах: **Unit.Percentage()**. По умолчанию установлена ширина 100%.
Height	Задает высоту компонента в необходимых единицах измерения, которые определены в классе **Unit**. Доступна установка значения в пикселях: **Unit.Pixel()**, поинтах: **Unit.Point()** и процентах: **Unit.Percentage()**. По умолчанию установлена автоматическая высота в зависимости от размера страницы отчета, либо 650 пикселей в режиме отображения вьювера с полосами прокрутки.

Действия (Actions)

Наименование	Описание
GetReport	Задает название метода действия подготовки

© 2003-2026 Stimulsoft

	построенного отчета. Если включено кэширование отчета, то данное действие будет вызвано только один раз при первом запросе отчета, либо в случае, если требуемый отчет не найден в кэше сервера.
PrintReport	Задает название метода действия печати отчета.
ExportReport	Задает название метода действия экспорта отчета в необходимый формат.
EmailReport	Задает название метода действия отправки отчета по Email.
Interaction	Задает название метода действия работы вьювера с интерактивными операциями, такими как использование параметров, динамическая сортировка, сворачивание и детализация отчета.
DesignReport	Задает название метода действия для перехода к нужному представлению при нажатии кнопки Дизайн (Design) на панели вьювера.
ViewerEvent	Задает название метода действия основных событий вьювера и обработки действий вьювера, таких как печать и экспортирование отчета, работа с параметрами и интерактивность, если эти действия не заданы отдельно. Дополнительно данное действие используется для загрузки скриптов и стилей вьювера. Это действие является обязательным.

Работа с сервером (Server)

Наименование	Описание
Controller	Задает название контроллера обработки запросов вьювера отчетов. Если данное

© 2003-2026 Stimulsoft

	свойство не задано, то для обработки запросов будет использоваться текущий контроллер.
RouteTemplate	Задает шаблон (route template) вызова действий вьювера отчетов. Если свойство не установлено, будет использоваться шаблон, установленный в MVC проекте. Если свойство `UseRelativeUrls` установлено в true, для этого свойства не будет учитываться `BasePath`. По умолчанию это свойство не установлено.
RequestTimeout	Устанавливает время ожидания ответа от сервера в секундах, по истечении которого будет выдана ошибка. По умолчанию установлено значение 30 секунд. Для больших отчетов рекомендуется увеличить данное значение.
CacheTimeout	Устанавливает время в минутах, которое сервер будет хранить построенный отчет с момента последнего действия вьювера. По умолчанию установлено значение 10 минут.
CacheMode	Устанавливает режим кэширования отчета. Может принимать одно из указанных ниже значений перечисления **StiServerCacheMode**: › **None** – кэширование отключено, каждое действие вьювера требует загрузки отчета и, если это шаблон отчета, последующее его построение; › **ObjectCache** – в качестве хранилища используется кэш, отчет хранится как объект (значение свойства по умолчанию); › **ObjectSession** – в качестве хранилища используется сессия, отчет хранится как объект; › **StringCache** – в качестве хранилища используется кэш сервера, отчет сериализуется в упакованную строку;

© 2003-2026 Stimulsoft

	› **StringSession** – в качестве хранилища используется сессия, отчет сериализуется в упакованную строку.
CacheItemPriority	Устанавливает приоритет хранимого в кэше сервера отчета. Это свойство влияет на автоматическую очистку памяти сервера в случае ее нехватки. Чем меньше приоритет, тем больше шанс удаления информации из памяти.
AllowAutoUpdateCache	Устанавливает режим автоматического обновления кэша. Отчет, хранящийся в кэше либо сессии сервера, будет автоматически пересохранен через определенный промежуток времени при бездействии вьювера (примерно каждые 3 минуты). По умолчанию свойство имеет значение **true**.
UseRelativeUrls	Устанавливает режим вьювера, в котором для AJAX запросов на сервер используются относительные ссылки. По умолчанию свойство имеет значение **true**.
PortNumber	Получает или задает значение, которое указывает номер порта для использования в URL. Значение 0 определяет автоматическое обнаружение (установлено по умолчанию). Значение -1 удаляет номер порта.
PassQueryParametersForResources	Включает передачу всех параметров URL запроса при формировании ссылок на ресурсы вьювера. При значении **false** для запроса ресурсов вьювера используются только необходимые параметры, что способствует более корректной работе кэша браузера. По умолчанию свойство имеет значение **true**.
PassQueryParametersToReport	Включает использование всех параметров URL запроса в качестве значений одноименных переменных в отчете. По умолчанию свойство имеет значение **false**.

© 2003-2026 Stimulsoft

PassFormValues	Включает передачу значений POST-формы на сторону клиента, если эти значения требуется использовать в действиях вьювера. При включении данной возможности, вспомогательный метод **GetFormValues()** будет возвращать коллекцию параметров формы. По умолчанию свойство имеет значение **false**.
ShowServerErrorPage	Включает отображение HTML-страницы с детализацией ошибки, возникшей на стороне сервера. При включенном свойстве детализация ошибки будет отображена в окне вьювера. Если свойство выключено, будет отображен только числовой код ошибки и краткий текст ошибки в диалоговом окне. По умолчанию свойство имеет значение **true**.
UseCompression	Включает сжатие запросов вьювера в GZip поток. Это позволяет уменьшить объем интернет-трафика, но немного замедляет работу вьювера. По умолчанию свойство имеет значение **false**.
UseCacheForResources	Включает кэширование на стороне сервера ресурсов компонента, таких как скрипты, стили, изображения. Позволяет ускорить загрузку компонента и уменьшить нагрузку на сервер при большом количестве клиентов. По умолчанию свойство имеет значение **true**.
UseLocalizedCache	Предоставляет возможность использовать различные кэши в зависимости от выбранной локализации. По умолчанию свойство имеет значение **false**.
AllowLoadingCustomFontsToClientSide	Позволяет передавать пользовательские шрифты на клиентскую сторону и преобразовывать их в CSS стиль для корректного отображения текста на стороне HTML с заданным шрифтом. По умолчанию

© 2003-2026 Stimulsoft

	свойство имеет значение **false**.

Внешний вид (Appearance)

Наименование	Описание
IconSet	Предоставляет возможность установить набор иконок: › StiWebUIIconSet.Auto (значение по умолчанию) - устанавливает набор иконок автоматически. Для тем **Office2022** - используется набор иконок в стиле **Monoline**, для тем **Office2013** - используется набор иконок в стиле **Regular**. › StiWebUIIconSet.Monoline - устанавливает набор иконок в стиле **Monoline**; › StiWebUIIconSet.Regular - устанавливает набор иконок в стиле **Regular**.
CustomCss	Задает путь к CSS файлу стилей вьювера. Если данное свойство установлено, то стандартные стили выбранной темы загружены не будут. По умолчанию установлено пустое значение.
BackgroundColor	Задает цвет подложки вьювера. По умолчанию свойство имеет значение **White**.
PageBorderColor	Задает цвет границ страницы отчета. По умолчанию свойство имеет значение **Gray**.
RightToLeft	Устанавливает режим отображения **Right to Left** для элементов управления вьювера. По умолчанию свойство имеет значение **false**.
FullScreenMode	Устанавливает полноэкранный режим отображения вьювера. По умолчанию свойство имеет значение **false**.
ScrollbarsMode	Устанавливает режим отображения отчета с полосами прокрутки. По умолчанию свойство имеет значение **false**.

© 2003-2026 Stimulsoft

OpenLinksWindow	Задает целевое окно для открытия ссылок, содержащихся в отчете. По умолчанию свойство имеет значение **Blank** (новое окно).
OpenExportedReportWindow	Задает целевое окно для открытия файла экспорта из вьювера. По умолчанию свойство имеет значение **Blank** (новое окно).
DesignWindow	Задает целевое окно для открытия дизайнера отчетов. По умолчанию свойство имеет значение **Self** (текущее окно).
ShowTooltips	Включает отображение подсказок для элементов управления вьювера при наведении мыши. По умолчанию свойство имеет значение **true**.
ShowTooltipsHelp	Включает отображение ссылки на онлайн документацию в подсказках для элементов управления вьювера. По умолчанию свойство имеет значение **true**.
ShowDialogsHelp	Предоставляет возможность отображать или не отображать кнопку вызова справки в различных меню. По умолчанию, установлено значение **true**.
PageAlignment	Задает позицию страницы отчета в окне вьювера. Может принимать одно из указанных ниже значений перечисления **StiContentAlignment**: › **Left** – страница будет пристыкована к левому краю; › **Center** – страница будет центрирована (значение по умолчанию); › **Right** – страница будет пристыкована к правому краю.
ShowPageShadow	Включает отображение тени для страниц отчета. По умолчанию свойство имеет значение **true**.
BookmarksPrint	Включает печать закладок отчета на

© 2003-2026 Stimulsoft

	принтере (помимо самого отчета). По умолчанию свойство имеет значение **false**.
BookmarksTreeWidth	Устанавливает ширину панели закладок в пикселях. По умолчанию ширина имеет значение 180 пикселей.
ParametersPanelPosition	Задает позицию панели параметров отчета. Может принимать одно из указанных ниже значений перечисления **StiParametersPanelPosition**: › **Top** – панель будет пристыкована к верхнему краю (значение по умолчанию); › **Left** – панель будет пристыкована к левому краю.
ParametersPanelMaxHeight	Устанавливает максимальную высоту панели параметров в пикселях. По умолчанию максимальная высота имеет значение 300 пикселей.
ParametersPanelColumnsCount	Устанавливает количество колонок для отображения параметров отчета. По умолчанию разбиение производится на 2 колонки.
ParametersPanelSortDataItems	Устанавливает или выключает режим сортировки значений переменной. По умолчанию, опция установлена в значение **true**, т.е. значения переменной сортируются.
ParametersPanelDateFormat	Устанавливает формат даты и времени для переменных соответствующего типа на панели параметров. По умолчанию используется формат даты и времени, установленный браузером.
InterfaceType	Устанавливает тип используемого интерфейса вьювера. Может принимать одно из указанных ниже значений перечисления **StiInterfaceType**: › **Auto** – тип интерфейса вьювера будет

© 2003-2026 Stimulsoft

	выбран автоматически в зависимости от используемого устройства (значение по умолчанию); › **Mouse** – принудительное использование стандартного интерфейса для управления вьювером при помощи мыши; › **Touch** – принудительное использование Touch интерфейса для управления вьювером при помощи сенсорного экрана монитора, в этом режиме элементы интерфейса вьювера имеют увеличенные размеры для удобства управления; › **Mobile** – принудительное использование Mobile интерфейса для управления вьювером при помощи экрана смартфона, в этом режиме интерфейс вьювера имеет упрощенный вид и адаптирован для управления при помощи мобильного устройства.
AllowMobileMode	Включает или выключает возможность отображения отчета или дашборда в мобильном режиме. Если опция установлена в значение **false**, то мобильный режим просмотра не будет использоваться ни при каких обстоятельства. Если опция установлена в значение **true**, то мобильный режим просмотра будет использоваться запуске вьювера на мобильных устройствах. По умолчанию, опция установлена в значение **true**.
ChartRenderType	Устанавливает режим отображения диаграмм на странице отчета. Может принимать одно из указанных ниже значений перечисления **StiChartRenderType**: › **Image** – диаграммы отображаются в виде статичных изображений; › **Vector** – диаграммы отображаются в

© 2003-2026 Stimulsoft

	векторном режиме в виде SVG объекта; › **AnimatedVector** - диаграммы отображаются в векторном режиме в виде SVG объекта, элементы диаграммы появляются с анимацией (значение по умолчанию).
ReportDisplayMode	Устанавливает режим экспорта для отображения страниц отчета. Может принимать одно из указанных ниже значений перечисления **StiReportDisplayMode**: › **FromReport** - режим экспорта элементов отчета определяется из настроек шаблона - Div или Table; › **Table** – элементы отчета экспортируются с использованием HTML таблиц (значение по умолчанию); › **Div** – элементы отчета экспортируются с использованием DIV разметки; › **Span** - элементы отчета экспортируются с использованием SPAN разметки.
DatePickerFirstDayOfWeek	Устанавливает первый день недели для элемента выбора даты. Может принимать одно из указанных ниже значений перечисления **StiFirstDayOfWeek**: › **Auto** – автоматическое определение первого дня недели из настроек браузера (значение по умолчанию); › **Monday** – первым днем недели является понедельник; › **Sunday** – первым днем недели является воскресенье.
DatePickerIncludeCurrentDayForRanges	Предоставляет возможность включать или не включать текущий день в диапазон значений элемента **Выбор даты (Date Picker)**. По умолчанию, опция установлена в значение **false**, т.е. текущий день не включается в

© 2003-2026 Stimulsoft

	диапазон значений элемента.
AllowTouchZoom	Включает возможность изменения масштаба страницы отчета специальным двух пальцевым жестом (Pinch to Zoom) на сенсорном экране устройства. По умолчанию свойство имеет значение **true**.
ShowReportIsNotSpecifiedMessage	Предоставляет возможность отображать или не отображать уведомление, если отчет не задан. По умолчанию свойство имеет значение **true**.
PrintToPdfMode	Устанавливает режим печати в PDF документ. Может принимать значения: › **StiPrintToPdfMode.Hidden** - скрытый режим печати (по умолчанию); › **StiPrintToPdfMode.Popup** - документ PDF будет отображет перед печатью во всплывающем окне.
ImagesQuality	Получает или задает качество изображения, которое будет использоваться на странице вьювера. Может принимать значения: › **StiImagesQuality.Low** - низкое качество, используется для ускорения загрузки отчета и экономии памяти › **StiImagesQuality.Normal** - обычное качество, подходит для большинства ситуаций, установлено по умолчанию › **StiImagesQuality.High** - высокое качество, используется для мониторов высокой четкости, но может замедлять загрузку страниц
CombineReportPages	Предоставляет возможность объединить обработанные страницы шаблона отчета в один шаблон или представить каждую страницу шаблона отдельной вкладкой во вьювере. По умолчанию, опция установлена в значение **false**, т.е. каждая страница шаблона отчета будет представлена

© 2003-2026 Stimulsoft

	отдельной вкладкой во вьювере.
AllowPropagationEvents	Предоставляет возможность разрешить распространение событий нажатия клавиш, когда вьювер отчетов не в фокусе. По умолчанию свойство имеет значение **true**.
DashboardFilterElementItemHeight	Предоставляет возможность изменить высоту флажка в элементе дашборда **Список (List Box)**.

Панель инструментов (Toolbar)

Наименование	Описание
Visible	Включает отображение панели инструментов вьювера. По умолчанию свойство имеет значение **true**.
DisplayMode	Задает режим отображения панели инструментов вьювера. Может принимать одно из указанных ниже значений перечисления **StiToolbarDisplayMode**: › **Simple** – простой режим отображения, все элементы управления расположены на одной панели управления (значение по умолчанию); › **Separated** – раздельный режим отображения, панель управления разделена на верхнюю и нижнюю.
BackgroundColor	Задает цвет панели инструментов вьювера. По умолчанию используется цветовое оформление выбранной темы.
BorderColor	Задает цвет границ панели инструментов вьювера. По умолчанию используется цветовое оформление выбранной темы.
FontColor	Задает цвет текста для панели инструментов и меню вьювера. По умолчанию

© 2003-2026 Stimulsoft

	используется цветовое оформление выбранной темы.
FontFamily	Задает шрифт для панели инструментов и меню вьювера. По умолчанию используется шрифт выбранной темы.
Alignment	Устанавливает режим выравнивания элементов управления на панели инструментов вьювера. Может принимать одно из указанных ниже значений перечисления **StiContentAlignment**: › **Left** – элементы будут пристыкованы к левому краю; › **Center** – элементы будут располагаться по центру панели; › **Right** – элементы будут пристыкованы к правому краю; › **Default** – выравнивание зависит от свойства RightToLeft (значение по умолчанию)
ShowButtonCaptions	Включает отображение текста кнопок панели инструментов вьювера. По умолчанию свойство имеет значение **true**.
ShowPrintButton	Включает отображение кнопки **Печать (Print)** на панели инструментов вьювера. По умолчанию свойство имеет значение **true**.
ShowOpenButton	Включает отображение кнопки **Открыть (Open)** на панели инструментов вьювера при просмотре отчетов или панели индикаторов (дашбордов). По умолчанию свойство имеет значение **true**.
ShowSaveButton	Включает отображение кнопки **Сохранить (Save)** на панели инструментов вьювера при просмотре отчетов или панели индикаторов (дашбордов). По умолчанию свойство имеет значение **true**.
ShowSendEmailButton	Включает отображение кнопки **Отправиль**

© 2003-2026 Stimulsoft

	Email (Send Email) на панели инструментов вьювера. По умолчанию свойство имеет значение **false**. Также, необходимо определить действие EmailReport.
ShowFindButton	Включает отображение кнопки **Поиск (Find)** на панели инструментов вьювера. По умолчанию свойство имеет значение **true**.
ShowBookmarksButton	Включает отображение кнопки **Закладки (Bookmarks)** на панели инструментов вьювера. По умолчанию свойство имеет значение **true**. Если указанная кнопка скрыта, то панель закладок не будет отображена даже при наличии закладок в отчете.
ShowParametersButton	Включает отображение кнопки **Параметры (Parameters)** на панели инструментов вьювера. По умолчанию свойство имеет значение **true**. Если указанная кнопка скрыта, то панель параметров не будет отображена даже при наличии параметров в отчете.
ShowResourcesButton	Включает отображение кнопки **Ресурсы (Resources)** на панели инструментов вьювера. По умолчанию свойство имеет значение **true**. Если указанная кнопка скрыта, то панель ресурсов не будет отображена.
ShowEditorButton	Включает отображение кнопки **Редактор (Editor)** на панели инструментов вьювера при просмотре отчетов. По умолчанию свойство имеет значение **true**.
ShowFullScreenButton	Включает отображение кнопки **Полноэкранный просмотр (Full Screen)** на панели инструментов вьювера при просмотре отчетов или панели индикаторов (дашбордов). По умолчанию свойство имеет значение **true**.
ShowFirstPageButton	Включает отображение кнопки **Первая страница (First Page)** на панели инструментов вьювера. По умолчанию

© 2003-2026 Stimulsoft

	свойство имеет значение **true**.
ShowPreviousPageButton	Включает отображение кнопки **Предыдущая страница (Previous Page)** на панели инструментов вьювера. По умолчанию свойство имеет значение **true**.
ShowCurrentPageControl	Включает отображение индикатора текущей страницы отчета. По умолчанию свойство имеет значение **true**.
ShowNextPageButton	Включает отображение кнопки **Следующая страница (Next Page)** на панели инструментов вьювера. По умолчанию свойство имеет значение **true**.
ShowLastPageButton	Включает отображение кнопки **Последняя страница (Last Page)** на панели инструментов вьювер. По умолчанию свойство имеет значение **true**.
ShowZoomButton	Включает отображение кнопки выбора масштаба отчета. По умолчанию свойство имеет значение **true**.
ShowViewModeButton	Включает отображение кнопки выбора режима отображения страниц отчета. По умолчанию свойство имеет значение **true**.
ShowDesignButton	Включает отображение кнопки **Дизайн (Design)** на панели инструментов вьювера при просмотре отчетов или панели индикаторов (дашбордов). По умолчанию свойство имеет значение **false**.
ShowAboutButton	Включает отображение кнопки **О программе (About)** на панели инструментов вьювера. По умолчанию свойство имеет значение **true**.
ShowRefreshButton	Включает или выключает отображение кнопки **Обновить (Refresh)** на панели инструментов вьювера при просмотре панелей индикаторов (дашбордов). По умолчанию свойство имеет значение **true**.

© 2003-2026 Stimulsoft

ShowPinToolbarButton	Включает отображение кнопки **Прикрепить панель (Pin Toolbar)** на панели инструментов вьювера. Кнопка доступна только в мобильном режиме вьювера. По умолчанию свойство имеет значение **true**.
PrintDestination	Устанавливает режим печати отчета. Может принимать одно из указанных ниже значений перечисления **StiPrintDestination**: › **Default** – будет отображено меню с предоставлением выбора режима печати (значение по умолчанию); › **Pdf** – печать будет осуществлена в формате PDF; › **Direct** – печать будет осуществлена в формате HTML непосредственно на принтер, будет отображен системный диалог печати; › **PopupWindow** – печать будет осуществлена в формате HTML через всплывающее окно предварительного просмотра отчета.
ViewMode	Устанавливает режим отображения страниц отчета. Может принимать одно из указанных ниже значений перечисления **StiWebViewMode**: › **SinglePage** – отображается одна страница отчета, выбранная на панели инструментов вьювера (значение по умолчанию); › **Continuous** – отображаются все страницы отчета в виде ленты; › **MultiplePages** – отображаются все страницы отчета в виде таблицы.
Zoom	Задает масштаб отображения страниц отчета. По умолчанию установлен масштаб 100 процентов. Допускаются значения от 10 до 500 процентов. Также допускается устанавливать одно из следующих значений:

© 2003-2026 Stimulsoft

	› **StiZoomMode.PageWidth** – при запуске вьювера будет установлен масштаб, необходимый для отображения отчета по ширине страницы; › **StiZoomMode.PageHeight** – при запуске вьювера будет установлен масштаб, необходимый для отображения отчета по высоте страницы.
MenuAnimation	Включает анимацию при появлении/скрытии меню вьювера. По умолчанию свойство имеет значение **true**.
ShowMenuMode	Устанавливает режим отображения меню вьювера. Может принимать одно из указанных ниже значений перечисления **StiShowMenuMode**: › **Click** – отображение по клику мыши (значение по умолчанию); › **Hover** – отображение по наведении курсора мыши.
AutoHide	Включает автоматическое сворачивание панели инструментов вьювера. Свойство работает только в мобильном режиме вьювера. По умолчанию свойство имеет значение **false**.

Экспорт отчета (Export)

Наименование	Описание
DefaultSettings	Данная группа свойств предоставляет возможность задать настройки экспортов по умолчанию для каждого типа экспорта. Эти настройки будут применены к диалоговым окнам экспортов при запуске вьювера, либо к отчету или панели индикаторов, в случае

© 2003-2026 Stimulsoft

	отключения отображения диалоговых окон экспортов.
StoreExportSettings	Включает режим сохранения выбранных настроек в диалоговых окнах экспорта. Настройки будут храниться в cookies браузера. По умолчанию свойство имеет значение **true**.
ShowExportDialog	Включает отображение диалогового окна параметров экспорта. Если свойство имеет значение **false**, то экспортирование будет производиться с заданными по умолчанию настройками. По умолчанию свойство имеет значение **true**.
ShowExportToDocument	Включает отображение пункта меню экспорта **Файл документа (Document File)** при просмотре отчетов, и пункт **Снимок отчета (Report Snapshot)** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToPdf	Включает отображение пункта меню экспорта **Adobe PDF файл (Adobe PDF File)** при просмотре отчетов, и пункт **Adobe PDF** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToXps	Включает отображение пункта меню экспорта **Microsoft XPS файл (Microsoft XPS File)**. По умолчанию свойство имеет значение **false**.
ShowExportToPowerPoint	Включает отображение пункта меню экспорта **Microsoft PowerPoint 2007/2010 файл (Microsoft PowerPoint 2007/2010 File)**. По умолчанию свойство имеет значение **true**.
ShowExportToHtml	Включает отображение пункта меню экспорта **HTML файл (HTML File)**. По умолчанию свойство имеет значение **true**.
ShowExportToHtml5	Включает отображение пункта меню

© 2003-2026 Stimulsoft

	экспорта **HTML5 файл (HTML5 File)**. По умолчанию свойство имеет значение **true**.
ShowExportToMht	Включает отображение пункта меню экспорта **MHT Web архив (MHT Web Archive)**. По умолчанию свойство имеет значение **true**.
ShowExportToText	Включает отображение пункта меню экспорта **Text файл (Text File)**. По умолчанию свойство имеет значение **true**.
ShowExportToRtf	Включает отображение пункта меню экспорта **Rich Text файл (Rich Text File)**. По умолчанию свойство имеет значение **true**.
ShowExportToWord	Включает отображение пункта меню экспорта **Microsoft Word 2007/2010 файл (Microsoft Word 2007/2010 File)**. По умолчанию свойство имеет значение **true**.
ShowExportToOpenDocumentWriter	Включает отображение пункта меню экспорта **OpenDocument Writer файл (OpenDocument Writer File)**. По умолчанию свойство имеет значение **true**.
ShowExportToExcelBiff	Включает отображение пункта меню экспорта **Microsoft Excel файл (Microsoft Excel File)**. По умолчанию свойство имеет значение **true**.
ShowExportToExcelXml	Включает отображение пункта меню экспорта **Microsoft Excel Xml файл (Microsoft Excel Xml File)**. По умолчанию свойство имеет значение **true**.
ShowExportToExcel	Включает отображение пункта меню экспорта **Microsoft Excel 2007/2010 файл (Microsoft Excel 2007/2010 File)** при просмотре отчетов, и пункт **Microsoft Excel** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToOpenDocumentCalc	Включает отображение пункта меню экспорта **OpenDocument Calc файл**

© 2003-2026 Stimulsoft

	(OpenDocument Calc File). По умолчанию свойство имеет значение **true**.
ShowExportToCsv	Включает отображение пункта меню экспорта **CSV файл (CSV File)**. По умолчанию свойство имеет значение **true**.
ShowExportToDbf	Включает отображение пункта меню экспорта **DBF файл (DBF File)**. По умолчанию свойство имеет значение **true**.
ShowExportToXml	Включает отображение пункта меню экспорта **XML файл (XML File)**. По умолчанию свойство имеет значение **true**.
ShowExportToDif	Включает отображение пункта меню экспорта **Data Interchange формат (DIF) файл (Data Interchange Format (DIF) File)**. По умолчанию свойство имеет значение **true**.
ShowExportToSylk	Включает отображение пункта меню экспорта **Symbolic Link (SYLK) файл (Symbolic Link (SYLK) File)**. По умолчанию свойство имеет значение **true**.
ShowExportToJson	Включает отображение пункта меню экспорта **JSON файл (JSON File)**. По умолчанию свойство имеет значение **true**.
ShowExportToImageBmp	Включает отображение пункта меню экспорта **BMP рисунок (BMP Image)** при просмотре отчетов, и пункт **BMP** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToImageGif	Включает отображение пункта меню экспорта **GIF рисунок (GIF Image)** при просмотре отчетов, и пункт **GIF** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToImageJpeg	Включает отображение пункта меню экспорта **JPEG рисунок (JPEG Image)** при просмотре отчетов, и пункт **JPEG** при просмотре панелей индикаторов. По

© 2003-2026 Stimulsoft

	умолчанию свойство имеет значение **true**.
ShowExportToImagePcx	Включает отображение пункта меню экспорта **PCX рисунок (PCX Image)** при просмотре отчетов, и пункт **PCX** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToImagePng	Включает отображение пункта меню экспорта **PNG рисунок (PNG Image)** при просмотре отчетов, и пункт **PNG** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToImageTiff	Включает отображение пункта меню экспорта **TIFF рисунок (TIFF Image)** при просмотре отчетов, и пункт **TIFF** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToImageMetafile	Включает отображение пункта меню экспорта **EMF рисунок (EMF Image)** при просмотре отчетов, и пункт **EMF** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToImageSvg	Включает отображение пункта меню экспорта **SVG рисунок (SVG Image)** при просмотре отчетов, и пункт **SVG** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToImageSvgz	Включает отображение пункта меню экспорта **SVGZ рисунок (SVGZ Image)** при просмотре отчетов, и пункт **SVGZ** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowOpenAfterExport	Предоставляет возможность отображать или не отображать параметр **Открыть после экспорта (Open After Export)** в меню настроек экспорта. По умолчанию свойство имеет значение **true**.

© 2003-2026 Stimulsoft

Отправка отчета по Email (Email)

Наименование	Описание
ShowEmailDialog	Включает отображение диалогового окна параметров отправки отчета по Email. Если диалоговое окно отключено, то отправка по Email будет производиться с настройками, заданными на стороне сервера в действии **EmailReport**. По умолчанию свойство имеет значение **true**.
ShowExportDialog	Включает отображение диалогового окна параметров экспорта при отправке Email. Если свойство имеет значение **false**, то экспортирование будет производиться с заданными по умолчанию настройками. По умолчанию свойство имеет значение **true**.
DefaultEmailAddress	Устанавливает Email получателя по умолчанию, т.е. адрес на который будет отправлено письмо с вложенным отчетом.
DefaultEmailSubject	Устанавливает тему (заголовок) электронного письма по умолчанию.
DefaultEmailMessage	Устанавливает сообщение (текст) электронного письма по умолчанию.

4.2 HTML5 дизайнер

YouTube

Посмотрите видеоуроки по работе с компонентом ASP.NET MVC HTML5 Designer. Подписывайтесь на канал Stimulsoft и узнайте первыми о новых видеоуроках. Вопросы и предложения оставляйте в комментариях к видео.

© 2003-2026 Stimulsoft

Примеры

Ознакомьтесь с примерами по работе с компонентом ASP.NET MVC HTML5 Designer GitHub. Все примеры представляют собой отдельные проекты, сгруппированные в один solution для Visual Studio.

Компонент **HTML5 Designer** (**StiMvcDesigner**) предназначен для редактирования отчетов в окне браузера. При этом не требуется устанавливать у клиента .NET Framework, ActiveX компоненты или какие-либо специальные плагины. Все, что необходимо – это любой современный Web-браузер.

При помощи **HTML5 Designer** можно создавать, редактировать, сохранять, просматривать отчеты, а также печатать отчеты на любом компьютере с любой установленной операционной системой. Так как дизайнер использует только HTML и JavaScript технологии, он может быть запущен на устройствах, где нет поддержки Flash или Silverlight – планшеты, смартфоны. Также дизайнер поддерживает Touch интерфейс, который автоматически включается при использовании устройств с сенсорным экраном.

Компонент **HTML5 Designer** использует технологию AJAX для выполнения всех действий над редактируемым отчетом, что позволяет избавиться от перезагрузки всей страницы, а также экономить Web-трафик и повысить скорость работы.

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для использования **HTML5 Designer** в Web-проекте, необходимо установить NuGet пакет Stimulsoft.Reports.Web:

- Выберите пункт "Manage NuGet Packages..." в контекстном меню проекта;

© 2003-2026 Stimulsoft

- На вкладке Browse, в строке поиска, укажите Stimulsoft.Reports.Web;
- Выделите элемент, определите версию пакета и нажмите кнопку **Установить (Install)**. При обновлении пакета, следует нажать кнопку **Обновить (Update)**.

Если по какой-либо причине это невозможно, необходимо добавить в проект указанные ниже сборки:
- Stimulsoft.Base.dll
- Stimulsoft.Report.dll
- Stimulsoft.Report.Check.dll
- Stimulsoft.Report.Helper.dll
- Stimulsoft.Report.Mvc.dll
- Stimulsoft.Report.Web.dll
- Stimulsoft.Report.WebDesign.dll

Для того, чтобы добавить возможности создания и редактирования дашбордов в Web-проекте, установить NuGet пакет Stimulsoft.Dashboards.Web (данный пакет связан с пакетом Stimulsoft.Reports.Web, при его отсутствии он будет установлен автоматически):
- Выберите пункт "Manage NuGet Packages..." в контекстном меню проекта;
- На вкладке Browse, в строке поиска, укажите Stimulsoft.Dashboards.Web;
- Выделите элемент, определите версию пакета и нажмите кнопку **Установить (Install)**. При обновлении пакета, следует нажать кнопку **Обновить (Update)**.

Если по какой-либо причине это невозможно, необходимо дополнительно добавить в проект указанные ниже сборки:
- Stimulsoft.Dashboard.dll
- Stimulsoft.Dashboard.Drawing.dll
- Stimulsoft.Dashboard.Export.dll

- Как это работает
- Дополнительные возможности просмотра
- Активация
- Время ожидания
- Редактирование отчета и дашборда
- Локализация дизайнера
- Создание нового отчета и нового дашборда
- Использование тем

© 2003-2026 Stimulsoft

- Сохранение отчета и дашборда
- Кэширование
- Просмотр
- Вспомогательные методы
- Настройки

4.2.1 Как это работает

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для запуска дизайнера требуется разместить на странице компонент **StiMvcDesigner**, задать ему необходимые настройки, а в контроллере представления определить необходимые действия. При запуске дизайнера отчетов происходят следующие действия:

- .NET компонент генерирует HTML и JavaScript код, необходимый для отображения и работы дизайнера;
- По окончании вывода компонента запускается JavaScript метод, который запрашивает на стороне сервера шаблон отчета, и отображает его в окне редактирования;
- Различные действия, такие как предварительный просмотр, сохранение шаблона отчета, экспорт отчета, применение параметров, сортировка и детализация отчета, вызывают определенное действие на стороне сервера, в котором можно выполнить необходимые манипуляции с отчетом.

4.2.2 Активация

YouTube

Посмотрите видеоуроки по активации компонента ASP.NET MVC HTML5 Designer. Подписывайтесь на канал Stimulsoft и узнайте первыми о новых видеоуроках. Вопросы и предложения оставляйте в комментариях к видео.

© 2003-2026 Stimulsoft

После приобретения продукта Stimulsoft вам необходимо активировать лицензию для используемых компонентов. Это можно выполнить указав лицензионный ключ или загрузив файл с лицензионным ключом. Ниже представлен пример активации компонента **StiMvcDesigner**.

HomeController.cs

```
...
public class HomeController : Controller
{
   static HomeController()
   {
      //Activation with using license code
      Stimulsoft.Base.StiLicense.Key = "Your activation code...";

      //Activation with using license file
      var path = System.Web.HttpContext.Current.Server.MapPath("~/Content/
      license.key");
      Stimulsoft.Base.StiLicense.LoadFromFile(path);
   }
}
...
```

Получить лицензионный ключ или скачать файл с лицензионным ключом, вы можете в личном кабинете пользователя. Для авторизации в кабинете используйте имя пользователя и пароль, указанные при покупке подписки продукта.

4.2.3 Редактирование отчета и дашборда

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для редактирования шаблона отчета, необходимо добавить на страницу компонент **StiMvcDesigner** и задать ему минимально необходимые настройки, а в контроллере представления определить необходимые действия.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcDesigner("MvcDesigner1",
   new StiMvcDesignerOptions() {
      Actions =
      {
         GetReport = "GetReport",
         DesignerEvent = "DesignerEvent"
      }
})
...
```

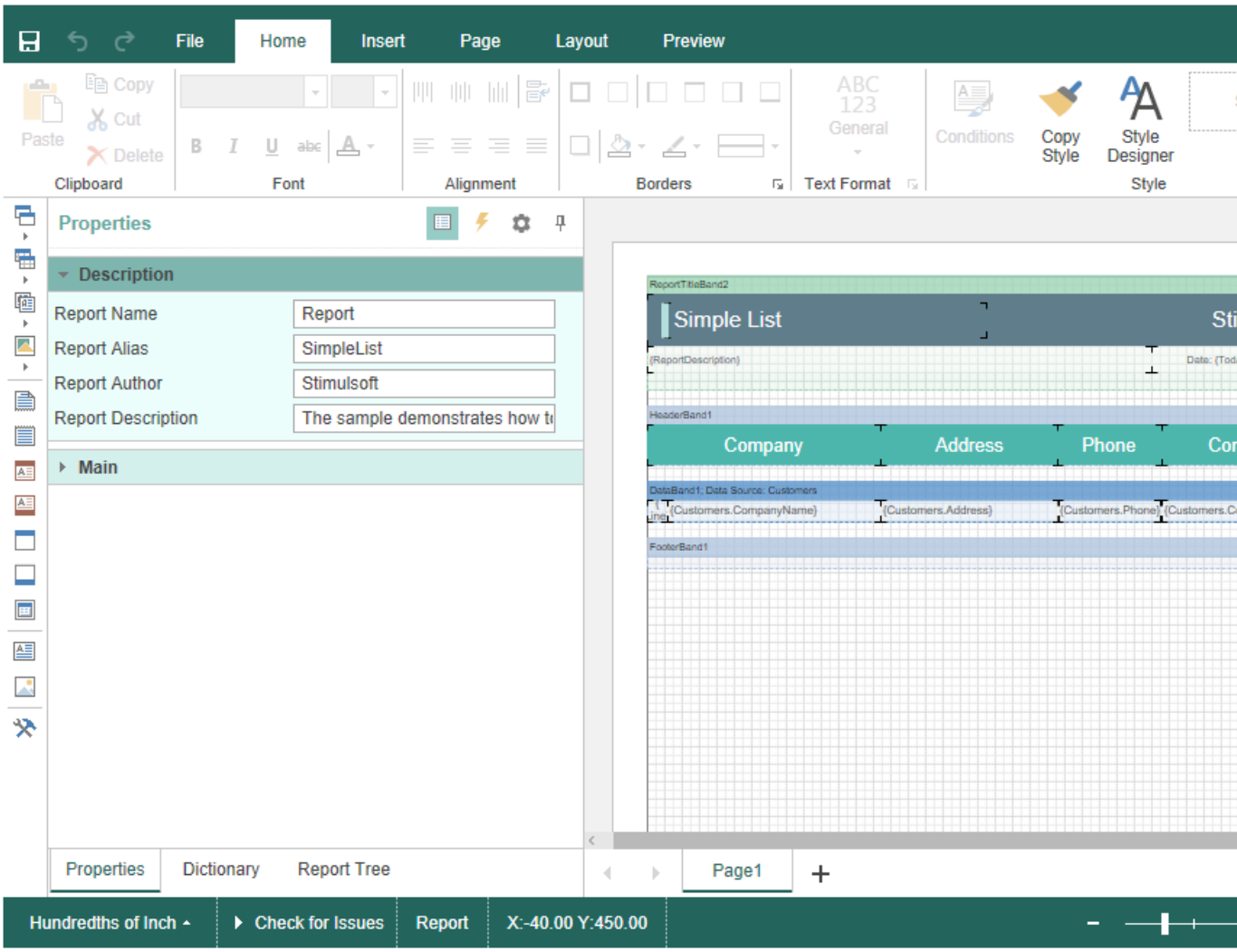
HomeController.cs

```
...
public ActionResult GetReport()
{
   StiReport report = new StiReport();
   report.Load(Server.MapPath("~/Content/SimpleList.mrt"));
   //report.Load(Server.MapPath("~/Content/Dashboard.mrt"));

   return StiMvcDesigner.GetReportResult(report);
}

public ActionResult DesignerEvent()
{
   return StiMvcDesigner.DesignerEventResult();
}
...
```

© 2003-2026 Stimulsoft



Действие **GetReport** предназначено для загрузки редактируемого шаблона отчета, оно вызывается автоматически после загрузки дизайнера отчетов. Действие **DesignerEvent** предназначено для обработки различных дополнительных действий дизайнера, таких как работа с данными, работа с компонентами, предварительный просмотр отчета и других.

Информация

Действие **DesignerEvent** является обязательным. Без него корректная работа дизайнера невозможна.

© 2003-2026 Stimulsoft

4.2.4 Создание нового отчета и нового дашборда

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для запуска дизайнера с новым отчетом достаточно создать новый объект в действии **GetReport** и вернуть его дизайнеру. При необходимости, можно предварительно загрузить данные для отчета, либо выполнить какие-либо другие необходимые действия.

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcDesigner("MvcDesigner1",
   new StiMvcDesignerOptions() {
      Actions =
      {
         GetReport = "GetReport"
      }
})
...
```

HomeController.cs

```
...
public ActionResult GetReport()
{
   StiReport report = new StiReport();
   //var newDashboard = StiReport.CreateNewDashboard();

   return StiMvcDesigner.GetReportResult(report);
   //return StiMvcDesigner.GetReportResult(newDashboard);
}
...
```

Также новый отчет можно создать, используя главное меню дизайнера. Для того, чтобы предварительно загрузить данные для нового отчета, либо выполнить какие-либо другие необходимые действия, предназначено действие **CreateReport**. Данное действие будет вызвано при создании нового пустого

© 2003-2026 Stimulsoft

отчета из главного меню, либо при создании отчета с использованием мастера.

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcDesigner("MvcDesigner1",
   new StiMvcDesignerOptions() {
      Actions =
      {
         CreateReport = "CreateReport"
      }
})
...
```

HomeController.cs

```
...
public ActionResult CreateReport()
{
   StiReport report = new StiReport();
   //var newDashboard = StiReport.CreateNewDashboard();

   // Register data for the new report, if necessary
   DataSet data = new DataSet("Demo");
   data.ReadXml(Server.MapPath("~/Content/Data/Demo.xml"));
   report.RegData(data);
   //newDashboard.RegData(data);
   report.Dictionary.Synchronize();
   //newDashboard.Dictionary.Synchronize();

   return StiMvcDesigner.GetReportResult(report);
   //return StiMvcDesigner.GetReportResult(newDashboard);
}
...
```

4.2.5 Просмотр

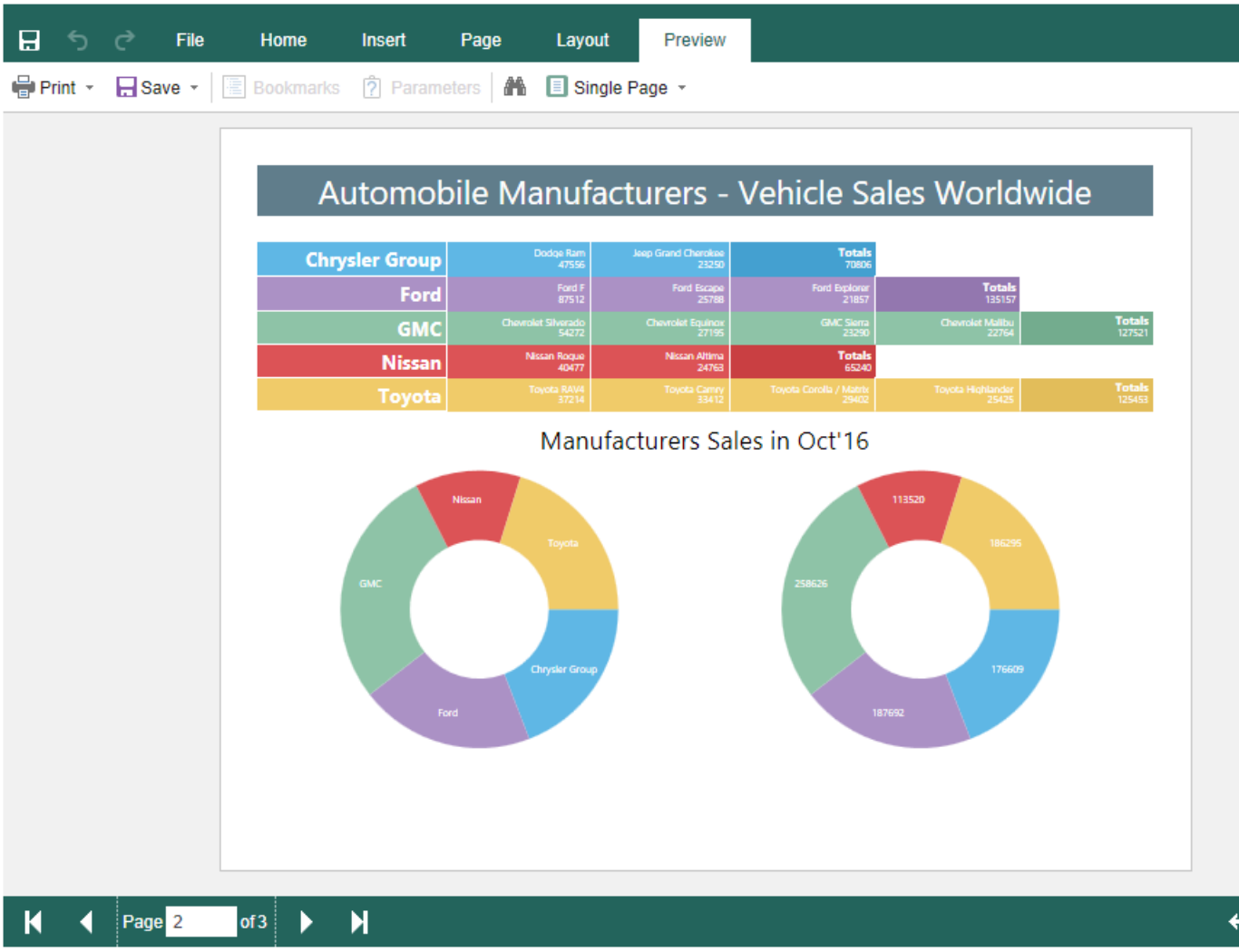
Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

У компонента **HTML5 Designer** предусмотрен режим просмотра редактируемого отчета. Для просмотра отчета, достаточно перейти на соответствующую вкладку в окне дизайнера. Шаблон будет передан на сторону

© 2003-2026 Stimulsoft

сервера, построен и отображен во встроенном вьюверере.



Перед просмотром отчета предусмотрена возможность выполнить какие-либо необходимые действия, например, подключить данные для отчета. Для этого достаточно определить специальное действие **PreviewReport**, которое будет вызвано перед просмотром отчета. Действие **PreviewReport** вызывается перед подготовкой и построением отчета для просмотра, до его сохранения в кэш.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcDesigner("MvcDesigner1",
   new StiMvcDesignerOptions() {
      Actions =
      {
         PreviewReport = "PreviewReport"
      }
})
...
```

HomeController.cs

```
...
public ActionResult PreviewReport()
{
   StiReport report = StiMvcDesigner.GetActionReportObject();
   //var Dashboard = StiMvcDesigner.GetActionReportObject();

   DataSet data = new DataSet("Demo");
   data.ReadXml(Server.MapPath("~/Content/Data/Demo.xml"));
   report.RegData(data);
   //Dashboard.RegData(data);

   return StiMvcDesigner.PreviewReportResult(report);
   //return StiMvcDesigner.PreviewReportResult(Dashboard);
}
...
```

При необходимости выполнить действия над отчетом непосредственно перед отображением отчета, можно воспользоваться специальным действием **GetPreviewReport**, которое вызывается после запроса подготовленного отчета из кэша.

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcDesigner("MvcDesigner1",
   new StiMvcDesignerOptions() {
      Actions =
      {
         GetPreviewReport = "GetPreviewReport"
      }
})
...
```

© 2003-2026 Stimulsoft

HomeController.cs

```
...
public ActionResult GetPreviewReport()
{
   StiReport report = StiMvcDesigner.GetActionReportObject();
   //var Dashboard = StiMvcDesigner.GetActionReportObject();

   DataSet data = new DataSet("Demo");
   data.ReadXml(Server.MapPath("~/Content/Data/Demo.xml"));
   report.RegData(data);
   //Dashboard.RegData(data);
   //report.IsRendered = false;

   return StiMvcDesigner.PreviewReportResult(report);
   //return StiMvcDesigner.PreviewReportResult(Dashboard);
}
...
```

Информация

Так как в данном событии передаётся уже подготовленный для просмотра отчет, то при необходимости выполнить повторное построение, необходимо установить флаг **report.IsRendered = false**.

4.2.6 Дополнительные возможности просмотра

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Окно предварительного просмотра отчета компонента **HTML5 Designer** представляет собой полноценный интерактивный **HTML5 Viewer**, который умеет осуществлять печать отчета, экспортирование отчета, поддерживает работу с их параметрами. Также поддерживаются интерактивные действия, такие как динамическая сортировка, детализация, сворачивание. Для использования указанных возможностей не требуется каких-либо дополнительных настроек дизайнера отчетов.

© 2003-2026 Stimulsoft

В любом из вышеперечисленных действий допускается производить манипуляции с шаблоном отчета, например, изменять его свойства и параметры, подключать новые данные для построения.

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcDesigner("MvcDesigner1",
   new StiMvcDesignerOptions() {
      Actions =
      {
         ExportReport = "ExportReport"
      }
})
...
```

HomeController.cs

```
...
public ActionResult ExportReport()
{
   StiReport report = StiMvcDesigner.GetActionReportObject();
   // ...

   return StiMvcDesigner.ExportReportResult(report);
}
...
```

Информация

Если какие-либо из указанных дополнительных возможностей предварительного просмотра отчета не требуются (например, экспортирование или печать отчета), можно отключить их, используя соответствующие свойства компонента **HTML5 Designer**.

© 2003-2026 Stimulsoft

4.2.7 Сохранение отчета и дашборда

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

В компоненте **HTML5 Designer** предусмотрено два варианта сохранения отчета, которые доступны в главном меню и на главной панели дизайнера: **Сохранить (Save)** и **Сохранить как (Save As)**. В свою очередь, каждый из этих вариантов сохранения имеет свои режимы и настройки.

Сохранение отчета и дашборда на стороне сервера

Для сохранения редактируемого отчета на стороне сервера необходимо определить действие **SaveReport**, которое будет вызвано при выборе пункта **Сохранить (Save)** в главном меню, либо нажатии на кнопку **Сохранить (Save)** на главной панели дизайнера.

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcDesigner("MvcDesigner1",
   new StiMvcDesignerOptions() {
      Actions =
      {
         SaveReport = "SaveReport"
      }
})
...
```

© 2003-2026 Stimulsoft

HomeController.cs

```
...
public ActionResult SaveReport()
{
   StiReport report = StiMvcDesigner.GetReportObject();

   // Save the report template
   // ...

   return StiMvcDesigner.SaveReportResult();
}
...
```

Данное действие возвращает ответ клиентской стороне дизайнера о результате сохранения отчета. Если необходимо, после сохранения отчета предусмотрена возможность отобразить диалоговое окно с ошибкой или текстовым сообщением.

HomeController.cs

```
...
public ActionResult SaveReport()
{
   StiReport report = StiMvcDesigner.GetReportObject();

   // Save the report template
   // ...

   // Completion of the report saving with message dialog box
   return StiMvcDesigner.SaveReportResult("Some message after saving");
}
...
```

Если необходимо, вы можете получить доступ к оригинальному имени отчета или имени отчета из диалога сохранения.

© 2003-2026 Stimulsoft

HomeController.cs

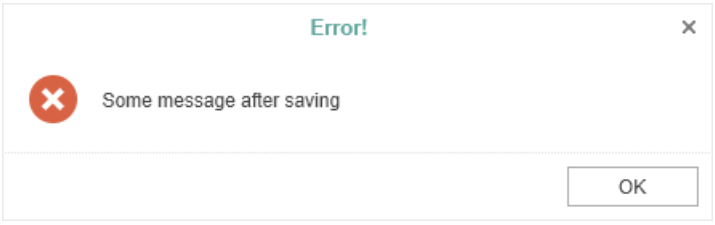
```
...
public ActionResult SaveReport()
{
   var requestParams = StiMvcDesigner.GetRequestParams();
   var report = StiMvcDesigner.GetReportObject();

   //Report name from designer save dialog
   var savingReportName = requestParams.Designer.FileName;

   //Original report name from properties
   var originalReportName = report.ReportName;

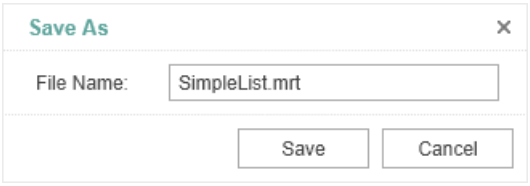
   return StiMvcDesigner.SaveReportResult();
}
...
```

В этом случае, будет отображено диалоговое с указанным текстом. Текст может содержать как сообщение об ошибке сохранения либо предупреждение, так и любое другое сообщение.



Сохранение на стороне клиента

Для сохранения редактируемого отчета на стороне клиента в виде файла не требуется дополнительных настроек дизайнера. Достаточно выбрать пункт главного меню **Сохранить как (Save As)**, при нажатии на который отобразится диалог сохранения файла. В данном диалоге можно изменить имя файла отчета, после чего файл будет сохранен на локальном диске компьютера.



© 2003-2026 Stimulsoft

Компонент **HTML5 Designer** предоставляет возможность изменить поведение указанного варианта сохранения, для этого в настройках дизайнера предусмотрено специальное действие **SaveReportAs**. При определении этого действия сохранение отчета будет происходить на стороне сервера, работа данного действия будет аналогична действию **SaveReport**.

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcDesigner("MvcDesigner1",
   new StiMvcDesignerOptions() {
      Actions =
      {
         SaveReportAs = "SaveReportAs"
      }
})
...
```

HomeController.cs

```
...
public ActionResult SaveReportAs()
{
   StiReport report = StiMvcDesigner.GetReportObject();

   // Save the report template
   // ...

   return StiMvcDesigner.SaveReportResult();
}
...
```

Настройки сохранения

Сохранение отчета производиться в фоновом режиме, т.е. без перезагрузки страницы в окне браузера. Если необходимо каким-либо образом визуально управлять процессом сохранения отчета, следует изменить значение свойства **SaveReportMode** (либо **SaveReportAsMode**) дизайнера на одно из трех указанных значений: **Hidden** (значение по умолчанию), **Visible** или **NewWindow**.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcDesigner("MvcDesigner1",
   new StiMvcDesignerOptions() {
      Actions =
      {
        SaveReportAs = "SaveReportAs"
      },
      Behavior =
      {
        SaveReportAsMode = StiSaveMode.Visible
      }
})
...
```

Если свойство **SaveReportMode** установлено в значение **Visible**, то действие сохранения отчета будет вызвано в текущем окне браузера в обычном (видимом) режиме при помощи POST-запроса. Если свойство **SaveReportMode** установлено в значение **NewWindow**, то действие сохранения отчета будет вызвано в новом окне браузера. По умолчанию данное свойство установлено в значение **Hidden**, т.е. действие сохранения отчета вызывается в фоне при помощи AJAX-запроса и никак не отображается в окне браузера. Для свойства **SaveReportAsMode** применимы те же значения и поведение.

4.2.8 Локализация дизайнера

Компонент **HTML5 Designer** поддерживает полную локализацию своего интерфейса. Для того чтобы локализовать интерфейс дизайнера отчетов на необходимый язык, используется специальное свойство **Localization**. В качестве значения данного свойства необходимо указать путь к XML-файлу локализации (относительный либо абсолютный).

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcDesigner("MvcDesigner1",
   new StiMvcDesignerOptions() {
      Localization = "~/Content/Localization/en.xml"
})
...
```

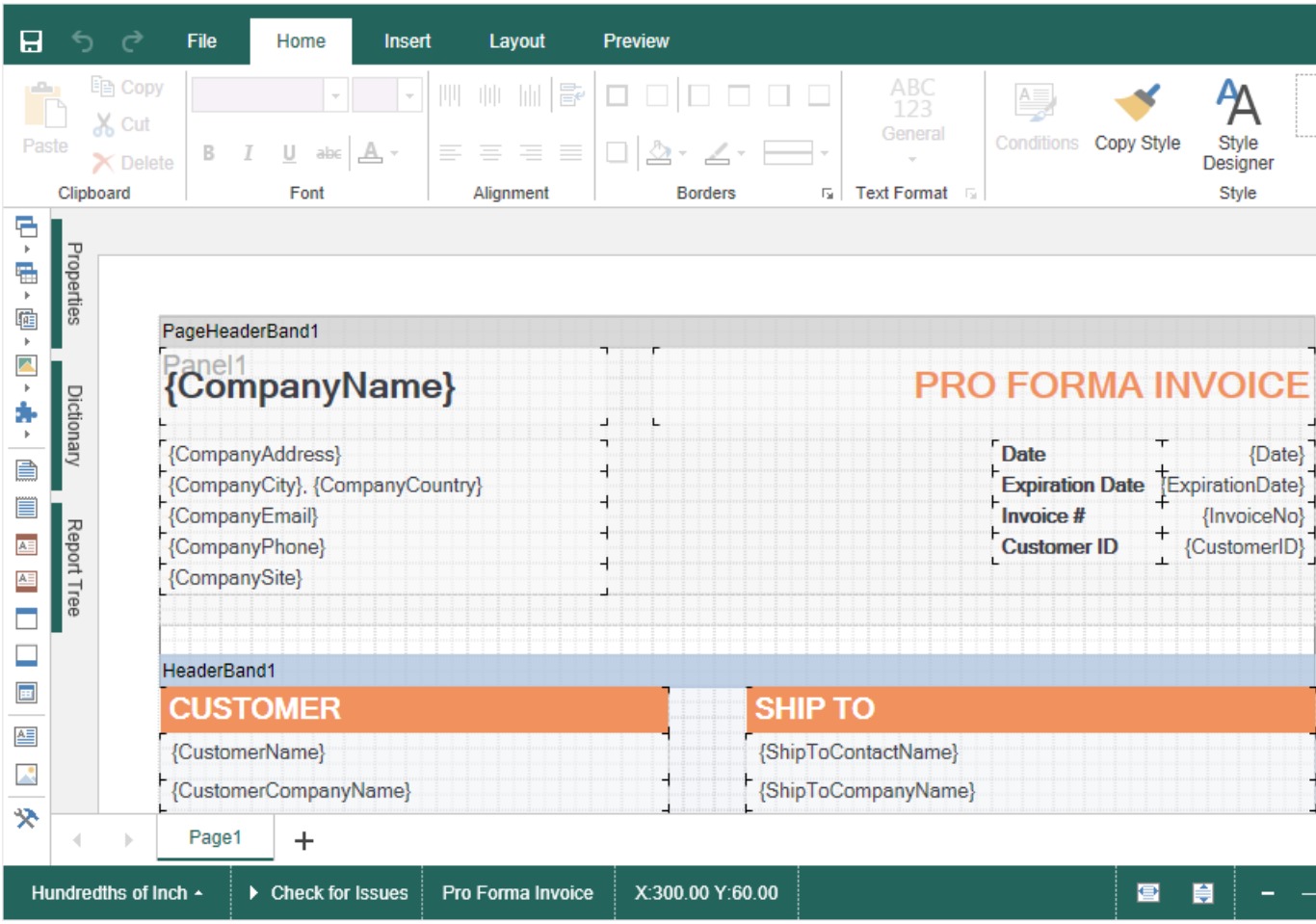
Интерфейс дизайнера отчетов позволяет выбрать необходимую локализацию из доступного списка. Для этого в качестве значения свойства **LocalizationDirectory** необходимо указать папку, в которой хранятся

© 2003-2026 Stimulsoft

локализационные XML-файлы.

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcDesigner("MvcDesigner1",
   new StiMvcDesignerOptions() {
      Localization = "~/Content/Localization/en.xml",
      LocalizationDirectory = "~/Content/Localization"
})
...
```



© 2003-2026 Stimulsoft

Информация

Если задано значение для свойства **Localization**, то при запуске дизайнера отчетов всегда будет применяться указанная в этом свойстве локализация. Если значение свойства не задано, то будет автоматически загружена локализация, выбранная из списка доступных локализаций на панели дизайнера отчетов.

4.2.9 Использование тем

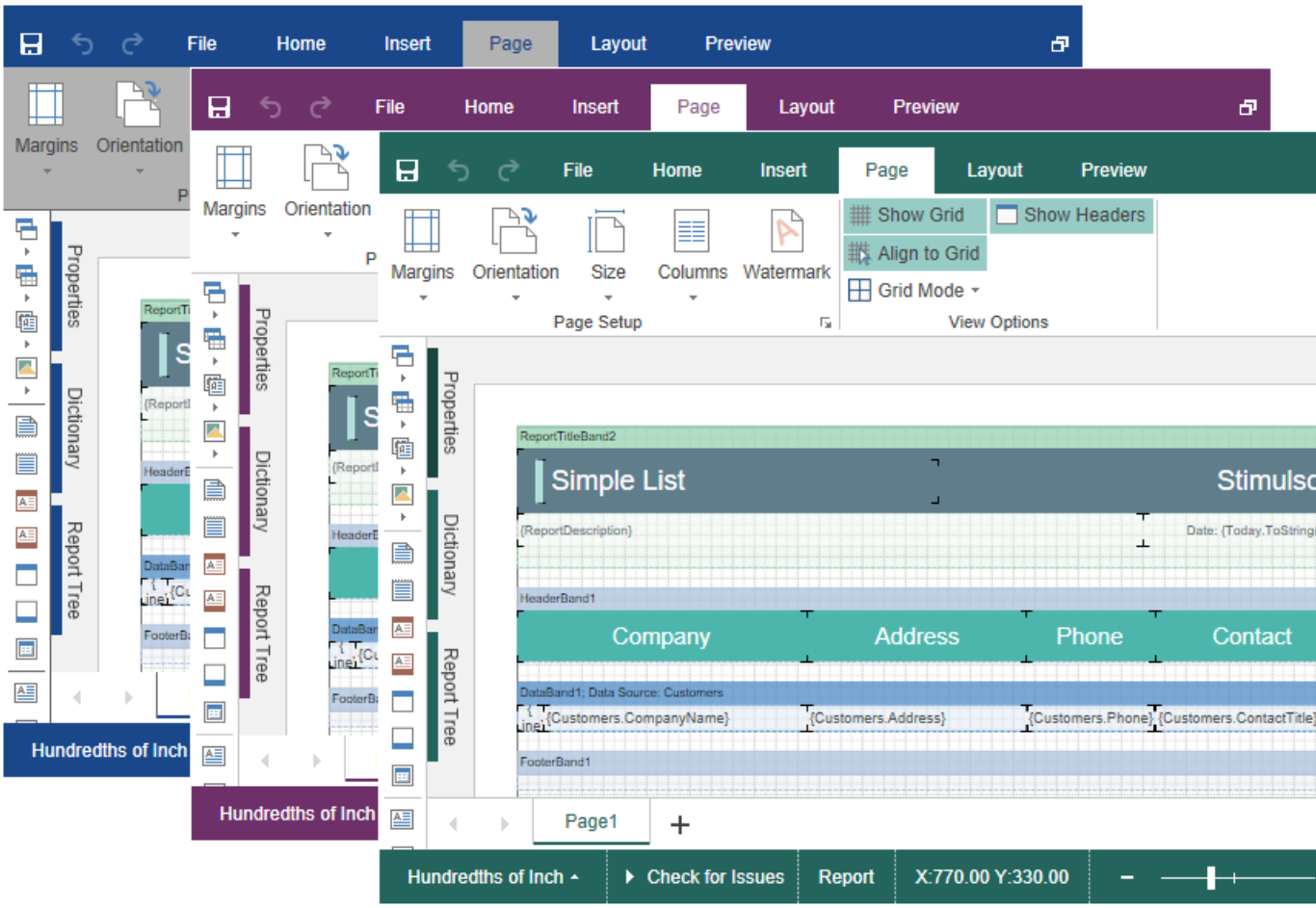
Компонент **HTML5 Designer** имеет возможность смены тем оформления визуальных элементов управления. Для изменения темы используется свойство **Theme**.

Default.aspx

```
...
@Html.Stimulsoft().StiMvcDesigner("MvcDesigner1",
   new StiMvcDesignerOptions() {
      Theme = StiDesignerTheme.Office2022WhiteTeal
})
...
```

На данный момент доступно **2 различные темы** со своими цветовыми акцентами. В результате, доступно **более 50 вариантов** оформления. Это позволяет настроить внешний вид дизайнера практически под любое оформление Web проекта.

© 2003-2026 Stimulsoft



4.2.10 Кэширование

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для работы **HTM5 Designer** использует кэш сервера для хранения редактируемого шаблона отчета. Это необходимо, так как клиентская часть дизайнера содержит только визуальное представление компонентов шаблона отчета, сам объект отчета со всеми параметрами и свойствами хранится на стороне сервера.

© 2003-2026 Stimulsoft

Управлять кэшированием можно при помощи следующих приведенных ниже свойств.

Свойство CacheMode

Данное свойство дизайнера устанавливает тип кэширования. Может принимать одно из значений, указанных в перечислении **StiServerCacheMode**:

- **None** – не используется в компоненте **HTML5 Designer**;
- **ObjectCache** – для кэширования используется кэш сервера, в котором сохраняется объект отчета (значение по умолчанию);
- **StringCache** - для кэширования используется кэш сервера, в котором сохраняется отчет в виде упакованной строки;
- **ObjectSession** - для кэширования используется текущая сессия, в которой сохраняется объект отчета;
- **StringSession** - для кэширования используется текущая сессия, в которой сохраняется отчет в виде упакованной строки.

Свойство CacheItemPriority

Данное свойство устанавливает приоритет хранимого в кэше сервера отчета, влияет на автоматическую очистку памяти сервера в случае ее нехватки. Чем меньше приоритет, тем больше шанс удаления информации из памяти.

Свойство CacheTimeout

Данное свойство указывает величину времени в минутах, в течение которого необходимо сохранять отчет в кэше сервера. Если при использовании кэширования требуемый отчет не найден в кэше (т.е. истекло время хранения объекта), то он будет запрошен заново при помощи специального действия **GetReport**, в этом случае несохраненные изменения могут быть утеряны.

Компонент **HTML5 Designer** предоставляет возможность определить

© 2003-2026 Stimulsoft

собственные методы работы с кэшированием отчета. Для этого предназначен специальный класс **StiCacheHelper**, который содержит методы получения отчета из кэша и сохранения отчета в кэш. Необходимо создать новый класс, унаследованный от **StiCacheHelper**, и перегрузить указанные выше методы, которые соответственно имеют названия **GetObject** и **SaveObject**.

© 2003-2026 Stimulsoft

HomeController.cs

```
...
public class DesignerController : Controller
{
   public class StiMyCacheHelper : StiCacheHelper
   {
      public override object GetObject(string guid)
      {
         string path =
         System.IO.Path.Combine(System.Web.HttpContext.Current.Server.MapPat
         h("~/"), "CacheFiles", guid);
         if (System.IO.File.Exists(path))
         {
            byte[] cacheData = System.IO.File.ReadAllBytes(path);
            return StiCacheHelper.GetObjectFromCacheData(cacheData);
         }
         return null;

         //return base.GetObject(guid);
      }

      public override void SaveObject(object obj, string guid)
      {
         byte[] cacheData = StiCacheHelper.GetCacheDataFromObject(obj);
         string path =
         System.IO.Path.Combine(System.Web.HttpContext.Current.Server.MapPat
         h("~/"), "CacheFiles", guid);
         System.IO.File.WriteAllBytes(path, cacheData);

         //base.SaveObject(obj, guid);
      }

      public override void RemoveReport(string guid)
      {
         string path =
         System.IO.Path.Combine(System.Web.HttpContext.Current.Server.MapPat
         h("~/"), "CacheFiles", guid);
         if (File.Exists(path))
         {
            File.Delete(path);
         }
      }
   }

   static DesignerController()
   {
      StiMvcDesigner.CacheHelper = new StiMyCacheHelper();
   }
}
...
```

Для инициализации работы с кэшированием отчета при помощи созданного класса, достаточно задать его в качестве значения статического свойства

© 2003-2026 Stimulsoft

StiMvcDesigner.CacheHelper в конструкторе контроллера.

4.2.11 Вспомогательные методы

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для **HTML5 Designer** предусмотрено несколько вспомогательных методов, которые предназначены для получения объекта редактируемого в данный момент отчета, параметров текущего состояния дизайнера и других полезных данных. Эти методы можно использовать в действиях дизайнера.

Метод GetReportObject()

Возвращает объект отчета, с которым в данный момент работает дизайнер. Допускается производить с ним необходимые манипуляции - регистрировать новые наборы данных, изменять свойства отчета, присваивать параметры либо загружать в объект другой отчет. Затем отчет можно вернуть дизайнеру, указав его в качестве параметра в результирующем методе действия.

HomeController.cs

```
...
public ActionResult ExportReport()
{
   StiReport report = StiMvcDesigner.GetReportObject();
   report.ReportName = "MyReportName";

   return StiMvcDesigner.ExportReportResult(report);
}
...
```

Метод GetActionReportObject()

Возвращает объект отчета, который будет использоваться для конкретного действия. Например, для действия **OpenReport** данный метод вернет отчет,

© 2003-2026 Stimulsoft

загруженный с локального диска компьютера. Для действия **PreviewReport** метод вернет подготовленную копию отчета для предварительного просмотра.

HomeController.cs

```
...
public ActionResult OpenReport()
{
   StiReport report = StiMvcDesigner.GetActionReportObject();

   // Register data for the opened report, if necessary
   DataSet data = new DataSet("Demo");
   data.ReadXml(Server.MapPath("~/Content/Data/Demo.xml"));
   report.RegData(data);
   report.Dictionary.Synchronize();

   return StiMvcDesigner.GetReportResult(report);
}
...
```

Метод GetRouteValues()

Возвращает значения маршрутов для URL, с которыми была открыта страница дизайнера. Таким образом, предоставляется возможность получить исходную коллекцию параметров страницы запуска дизайнера и использовать эти значения для каких-либо проверок и условий.

HomeController.cs

```
...
public ActionResult ExportReport()
{
   RouteValueDictionary routeValues = StiMvcDesigner.GetRouteValues();

   return StiMvcDesigner.ExportReportResult();
}
...
```

Также можно получить значения параметров URL по имени параметра, указав его в качестве параметра вызываемого действия дизайнера.

© 2003-2026 Stimulsoft

HomeController.cs

```
...
public ActionResult ExportReport(string id)
{
   return StiMvcDesigner.ExportReportResult();
}
...
```

Метод GetRequestParams()

Возвращает все параметры текущего состояния дизайнера, переданные на сторону сервера. Они могут быть полезны для определения типа действия, которое в данный момент выполняет дизайнер – например, для определения типа экспорта, а также всех параметров действия.

HomeController.cs

```
...
public ActionResult ExportReport()
{
   StiRequestParams requestParams = StiMvcDesigner.GetRequestParams();
   if (requestParams.ExportFormat == StiExportFormat.Pdf)
   {
      StiReport report = StiMvcDesigner.GetReportObject();

      // Some action with report for the PDF export
      // ...

      return StiMvcDesigner.ExportReportResult(report);
   }

   return StiMvcDesigner.ExportReportResult();
}
...
```

Метод GetExportSettings()

Возвращает все параметры текущего экспортирования отчета. Тип объекта параметров будет соответствовать типу экспорта, выбранному в меню предварительного просмотра отчета. Любые параметры экспорта можно изменить и передать на вход результирующего метода. В этом случае, экспортирование отчета будет произведено с переданными параметрами.

© 2003-2026 Stimulsoft

HomeController.cs

```
...
public ActionResult ExportReport()
{
   StiExportSettings settings = StiMvcDesigner.GetExportSettings();
   if (settings.GetExportFormat() == StiExportFormat.Pdf)
   {
      StiPdfExportSettings pdfSettings = (StiPdfExportSettings)settings;
      pdfSettings.EmbeddedFonts = true;
      pdfSettings.AllowEditable = StiPdfAllowEditable.No;
      return StiMvcDesigner.ExportReportResult(settings);
   }

   return StiMvcDesigner.ExportReportResult();
}
...
```

4.2.12 Время ожидания

При работе с компонентом **StiMvcDesigner** можно установить время ожидания (timeout) выполнения различных операций - хранение отчета в кэше, ответ сервера, выполнение запроса. Настройка времени ожидания выполняется при помощи свойств компонента и опций отчета.

Свойство CacheTimeout

Предоставляет возможность установить время в минутах, которое сервер будет хранить редактируемый отчет в кэше с момента последнего действия дизайнера. По умолчанию установлено значение 10 минут.

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcDesigner("MvcDesigner1",
   new StiMvcDesignerOptions() {
      Server =
      {
         CacheTimeout = 10
      }
})
...
```

Использование кэша увеличивает скорость работы дизайнера с отчетом. Более подробно можно ознакомиться в главе Кэширование.

© 2003-2026 Stimulsoft

Свойство RequestTimeout

Предоставляет возможность установить время ожидания ответа от сервера в секундах, по истечении которого будет выдана ошибка связи с сервером. По умолчанию установлено значение 30 секунд. Для больших отчетов рекомендуется увеличить данное значение.

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcDesigner("MvcDesigner1",
   new StiMvcDesignerOptions() {
      Server =
      {
         RequestTimeout = 30
      }
})
...
```

Опция CommandTimeout

Предоставляет возможность установить время ожидания запроса в секундах, при использовании SQL источников данных в отчете. Значение данного свойства сохраняется в самом шаблоне отчёта для каждого SQL подключения отдельно.

Ниже приведен пример кода, который предоставляет возможность установить время ожидания запроса для уже созданного MS SQL соединения и источников данных в отчете.

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcDesigner("MvcDesigner1",
   new StiMvcDesignerOptions() {
      Actions =
      {
         GetReport = "GetReport",
         DesignerEvent = "DesignerEvent"
      }
})
...
```

© 2003-2026 Stimulsoft

HomeController.cs

```
...
public ActionResult GetReport()
{

   StiReport report = new StiReport();
   report.Load(Server.MapPath("Report.mrt"));
   ((StiSqlSource)
   report.Dictionary.DataSources["DataSourceName"]).CommandTimeout = 1000;

   return StiMvcDesigner.GetReportResult(report);
}

public ActionResult DesignerEvent()
{
   return StiMvcDesigner.DesignerEventResult();
}
...
```

4.2.13 Добавление пользовательских функций

Информация

Ознакомьтесь с примером по добавлению пользовательской функции для компонента ASP.NET MVC HTML5 Designer GitHub.

При интеграции дизайнера отчетов в пользовательское приложение, присутствует возможность добавления custom функции в словарь данных дизайнера отчетов. После добавления функции, её можно использовать при разработке отчетов и панелей индикаторов (дашбордов). Ниже представлен пример преобразования текста в верхний регистр.

© 2003-2026 Stimulsoft

DesignerController.cs

```
...
public static string MyFunc(string value)
{
   return value.ToUpper();
}
...
static DesignerController()
{
   var ParamNames = new string[1];
   var ParamTypes = new Type[1];
   var ParamDescriptions = new string[1];

   ParamNames[0] = "value";
   ParamDescriptions[0] = "Descriptions";
   ParamTypes[0] = typeof(string);

   // How to add my function
   StiFunctions.AddFunction(
   "MyCategory",
   "MyFunc",
   "MyFunc",
   "Description",
   typeof(DesignerController),
   typeof(string),
   "Return Description",
   ParamTypes,
   ParamNames,
   ParamDescriptions);
}
...
```

4.2.14 Настройки

Настройка **HTML5 Designer** выполняется при помощи свойств, находящихся в классе **StiMvcDesignerOptions**. Все свойства разделены на группы, некоторые из групп содержат свои подгруппы свойств для удобства использования. Ниже приведен пример установки некоторых свойств дизайнера.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.Stimulsoft().StiMvcDesigner("MvcDesigner1",
   new StiMvcDesignerOptions() {
      Theme = Stimulsoft.Report.Web.StiDesignerTheme.Office2022WhiteTeal,
      Localization = "~/Content/Localization/en.xml",
      Actions =
      {
         GetReport = "GetReport",
         PreviewReport = "PreviewReport",
         SaveReport = "SaveReport",
         DesignerEvent = "DesignerEvent"
      },
      Appearance =
      {
         ReportDisplayMode = StiReportDisplayMode.Auto,
         ShowTooltipsHelp = false,
         ShowDialogsHelp = false,
         DefaultUnit = Stimulsoft.Report.StiReportUnitType.Centimeters
      },
      Dictionary =
      {
         PermissionBusinessObjects =
         Stimulsoft.Report.Web.StiDesignerPermissions.None,
         PermissionDataConnections =
         Stimulsoft.Report.Web.StiDesignerPermissions.View
      },
      Bands =
      {
         ShowChildBand = false,
         ShowEmptyBand = false,
         ShowOverlayBand = false
      }
})
...
```

Основные настройки (без группы)

Наименование	Описание
Theme	Задает тему оформления дизайнера. Список доступных тем оформления находится в перечислении **StiDesignerTheme**. По умолчанию установлено значение **Office2022WhiteBlue**.
Localization	Задает путь к XML файлу локализации. Путь может быть абсолютным либо относительным. По умолчанию используется

© 2003-2026 Stimulsoft

	английская локализация, которая встроена в дизайнер и не требует дополнительных XML файлов.
LocalizationDirectory	Задает путь к директории с XML файлами локализаций. Находящиеся в указанной папке локализационные файлы будут загружены в список выбора локализации на панели дизайнера.
Width	Задает ширину компонента в необходимых единицах измерения, которые определены в классе **Unit**. Доступна установка значения в пикселях: **Unit.Pixel()**, поинтах: **Unit.Point()** и процентах: **Unit.Percentage()**. По умолчанию компонент разворачивается на всю область окна браузера.
Height	Задает высоту компонента в необходимых единицах измерения, которые определены в классе **Unit**. Доступна установка значения в пикселях: **Unit.Pixel()**, поинтах: **Unit.Point()** и процентах: **Unit.Percentage()**. По умолчанию компонент разворачивается на всю область окна браузера.

Действия (Actions)

Наименование	Описание
GetReport	Задает название метода действия подготовки шаблона отчета при загрузке дизайнера.
OpenReport	Задает название метода действия открытия шаблона отчета из меню дизайнера.
CreateReport	Задает название метода действия подготовки шаблона отчета при создании нового отчета в дизайнере.
SaveReport	Задает название метода действия сохранения шаблона отчета на стороне сервера.

© 2003-2026 Stimulsoft

SaveReportAs	Задает название метода действия сохранения шаблона отчета на стороне сервера при использовании пункта меню **Сохранить как (Save As)**. Если действие не задано, будет использоваться встроенный метод сохранения шаблона отчета на локальный диск компьютера.
PreviewReport	Задает название метода действия подготовки построенного отчета в окне предварительного просмотра.
ExportReport	Задает название метода действия экспорта отчета в необходимый формат.
Exit	Задает название метода действия для перехода к нужному представлению при нажатии кнопки Выход (Exit) в главном меню дизайнера.
DesignerEvent	Задает название метода действия дизайнера отчетов, служащего для обработки различных дополнительных действий дизайнера, таких как работа с данными, работа с компонентами отчета, а так же для загрузки скриптов и стилей дизайнера.

Работа с сервером (Server)

Наименование	Описание
Controller	Задает название контроллера обработки запросов. Если данное свойство не задано, то для обработки запросов будет использоваться текущий контроллер.
RouteTemplate	Задает шаблон (route template) вызова действий дизайнера отчетов. Если свойство не установлено, будет использоваться шаблон, установленный в MVC проекте. Если свойство `UseRelativeUrls` установлено в true, для этого свойства не будет учитываться

© 2003-2026 Stimulsoft

	`BasePath`. По умолчанию это свойство не установлено.
RequestTimeout	Устанавливает время ожидания ответа от сервера в секундах, по истечении которого будет выдана ошибка. По умолчанию установлено значение 30 секунд. Для больших отчетов рекомендуется увеличить данное значение.
CacheTimeout	Устанавливает время в минутах, которое сервер будет хранить редактируемый отчет с момента последнего действия дизайнера. По умолчанию установлено значение 10 минут.
CacheMode	Устанавливает режим кэширования отчета. Может принимать одно из указанных ниже значений перечисления **StiServerCacheMode**: › **None** – не используется в компоненте **HTML5 Designer**; › **ObjectCache** – в качестве хранилища используется кэш, отчет хранится как объект (значение свойства по умолчанию); › **ObjectSession** – в качестве хранилища используется сессия, отчет хранится как объект; › **StringCache** – в качестве хранилища используется кэш сервера, отчет сериализуется в упакованную строку; › **StringSession** – в качестве хранилища используется сессия, отчет сериализуется в упакованную строку.
CacheItemPriority	Устанавливает приоритет хранимого в кэше сервера отчета. Это свойство влияет на автоматическую очистку памяти сервера в случае ее нехватки. Чем меньше приоритет, тем больше шанс удаления информации из памяти.
AllowAutoUpdateCache	Устанавливает режим автоматического

© 2003-2026 Stimulsoft

	обновления кэша. Отчет, хранящийся в кэше либо сессии сервера, будет автоматически пересохранен через определенный промежуток времени при бездействии дизайнера (примерно каждые 3 минуты). По умолчанию свойство имеет значение **true**.
UseRelativeUrls	Устанавливает режим дизайнера, в котором для запросов на сервер используются относительные ссылки. По умолчанию свойство имеет значение **true**.
PortNumber	Получает или задает значение, которое указывает номер порта для использования в URL. Значение 0 определяет автоматическое обнаружение (установлено по умолчанию). Значение -1 удаляет номер порта.
PassQueryParametersForResources	Включает передачу всех параметров URL запроса при формировании ссылок на ресурсы дизайнера. При значении **false** для запроса ресурсов дизайнера используются только необходимые параметры, что способствует более корректной работе кэша браузера. По умолчанию свойство имеет значение **true**.
PassFormValues	Включает передачу значений POST-формы на сторону клиента, если эти значения требуется использовать в действиях дизайнера. При включении данной возможности, вспомогательный метод **GetFormValues()** будет возвращать коллекцию параметров формы. По умолчанию свойство имеет значение **false**.
UseCompression	Включает сжатие запросов дизайнера в GZip поток. Это позволяет уменьшить объем интернет-трафика, но немного замедляет работу дизайнера. По умолчанию свойство имеет значение **false**.
UseCacheForResources	Включает кэширование на стороне сервера

© 2003-2026 Stimulsoft

	ресурсов компонента, таких как скрипты, стили, изображения. Позволяет ускорить загрузку компонента и уменьшить нагрузку на сервер при большом количестве клиентов. По умолчанию свойство имеет значение **true**.
AllowLoadingCustomFontsToClientSide	Позволяет передавать пользовательские шрифты на клиентскую сторону и преобразовывать их в CSS стиль для корректного отображения текста на стороне HTML с заданным шрифтом. По умолчанию свойство имеет значение **false**.

Внешний вид (Appearance)

Наименование	Описание
IconSet	Предоставляет возможность установить набор иконок: › StiWebUIIconSet.Auto (значение по умолчанию) - устанавливает набор иконок автоматически. Для тем **Office2022** - используется набор иконок в стиле **Monoline**, для тем **Office2013** - используется набор иконок в стиле **Regular**. › StiWebUIIconSet.Monoline - устанавливает набор иконок в стиле **Monoline**; › StiWebUIIconSet.Regular - устанавливает набор иконок в стиле **Regular**.
CustomCss	Задает путь к CSS файлу стилей дизайнера. Если данное свойство установлено, то стандартные стили выбранной темы загружены не будут. По умолчанию установлено пустое значение.
DefaultUnit	Устанавливает единицы измерения размеров для отчета и всех его компонентов. По умолчанию используются сантиметры.

© 2003-2026 Stimulsoft

Zoom	Задает масштаб отображения страниц отчета. По умолчанию установлен масштаб 100 процентов. Допускаются значения от 10 до 200 процентов. Может принимать одно из указанных ниже значений перечисления **StiZoomMode**: › **PageWidth** – при запуске дизайнера будет установлен масштаб, необходимый для отображения отчета по ширине страницы; › **PageHeight** – при запуске дизайнера будет установлен масштаб, необходимый для отображения отчета по высоте страницы.
ShowAnimation	Включает анимацию элементов интерфейса дизайнера. По умолчанию свойство имеет значение **true**.
ShowTooltips	Включает отображение подсказок для элементов управления дизайнера при наведении мыши. По умолчанию свойство имеет значение **true**.
ShowTooltipsHelp	Включает отображение ссылки на онлайн документацию в подсказках для элементов управления дизайнера. По умолчанию свойство имеет значение **true**.
ShowDialogsHelp	Включает отображение ссылки на онлайн документацию на заголовках диалоговых форм дизайнера. По умолчанию свойство имеет значение **true**.
InterfaceType	Устанавливает тип используемого интерфейса дизайнера. Может принимать одно из указанных ниже значений перечисления **StiInterfaceType**: › **Auto** – тип интерфейса дизайнера будет выбран автоматически в зависимости от используемого устройства (значение по умолчанию);

© 2003-2026 Stimulsoft

	› **Mouse** – принудительное использование интерфейса для управления дизайнером при помощи мыши; › **Touch** – принудительное использование Touch интерфейса для управления дизайнером при помощи сенсорного экрана (мобильные устройства), также в этом режиме для удобства управления элементы интерфейса увеличены.
DatePickerFirstDayOfWeek	Устанавливает первый день недели для элемента выбора даты. Может принимать одно из указанных ниже значений перечисления **StiFirstDayOfWeek**: › **Monday** – первым днем недели является понедельник (значение по умолчанию); › **Sunday** – первым днем недели является воскресенье.
DatePickerIncludeCurrentDayForRanges	Предоставляет возможность включать или не включать текущий день в элементе выбора даты. По умолчанию свойство имеет значение **false**.
FormatForDateControls	Предоставляет возможность переопределить формат для элементов управления дат. По умолчанию, значение для текущей опции не задан и формат для дат определяется из локали браузера.
ShowReportTree	Включает отображение дерева компонентов отчета. По умолчанию свойство имеет значение **true**.
ChartRenderType	Устанавливает режим отображения диаграмм в отчете на вкладке **Просмотр (Preview)**. Может принимать одно из указанных ниже значений перечисления **StiChartRenderType**: › **Image** – диаграммы отображаются в виде статичных изображений;

© 2003-2026 Stimulsoft

	› **Vector** – диаграммы отображаются в векторном режиме в виде SVG объекта; › **AnimatedVector** - диаграммы отображаются в векторном режиме в виде SVG объекта, элементы диаграммы появляются с анимацией (значение по умолчанию).
ReportDisplayMode	Устанавливает режим экспорта для отображения страниц отчета на вкладке предварительного просмотра. Может принимать одно из указанных ниже значений перечисления **StiReportDisplayMode**: › **FromReport** - режим экспорта элементов отчета определяется из настроек шаблона - Div или Table; › **Table** – элементы отчета экспортируются с использованием HTML таблиц (значение по умолчанию); › **Div** – элементы отчета экспортируются с использованием DIV разметки; › **Span** - элементы отчета экспортируются с использованием SPAN разметки.
ParametersPanelDateFormat	Устанавливает формат даты и времени для переменных соответствующего типа на панели параметров. По умолчанию используется формат даты и времени, установленный браузером.
CloseDesignerWithoutAsking	Предоставляет возможность установить режим закрытия дизайнера отчетов без подтверждения изменений в текущем отчете. По умолчанию свойство имеет значение **false**, т.е. дизайнер будет запрашивать разрешение на сохранение изменений в текущем отчете.
ShowSystemFonts	Предоставляет возможность отображать или не отображать системные шрифты в списке шрифтов дизайнера отчетов. По умолчанию

© 2003-2026 Stimulsoft

	свойство имеет значение **true**, т.е. системные шрифты отображаются в списке шрифтов.
WizardTypeRunningAfterLoad	Предоставляет возможность вызывать мастер создания отчетов после запуска дизайнера отчетов. Может принимать одно из указанных ниже значений перечисления **StiWizardType**: › **None** - дизайнер отчетов будет запущен без вызова мастера создания отчета; › **StandardReport** - дизайнер отчетов будет запущен с вызовом стандартного мастера создания отчета; › **MasterDetailReport** - дизайнер отчетов будет запущен с вызовом master-detail мастера создания отчета; › **LabelReport** - дизайнер отчетов будет запущен с вызовом мастера создания отчета с этикетками; › **InvoicesReport** - дизайнер отчетов будет запущен с вызовом мастера создания инвойса; › **OrdersReport** - дизайнер отчетов будет запущен с вызовом мастера создания ордера; › **QuotationReport** - дизайнер отчетов будет запущен с вызовом мастера создания квоты.
DefaultRibbonType	Предоставляет возможность установить используемый по умолчанию стиль панели инструментов дизайнера. Может принимать одно из указанных ниже значений перечисления: › **StiDesignerRibbonType.Classic** - Будет установлен классический стиль (значение по умолчанию); › **StiDesignerRibbonType.SingleLine** – Будет установлен компактный однострочный стиль.
AllowPropagationEvents	Предоставляет возможность разрешить распространение событий нажатия клавиш,

© 2003-2026 Stimulsoft

	когда дизайнер отчетов не в фокусе. По умолчанию свойство имеет значение **true**.
PropertiesPanelViewMode	Предоставляет возможность закрепить или открепить панель свойств, словарь отчета, дерево отчета. Может принимать одно из указанных ниже значений перечисления: › **Pinned** - панели закреплены (значение по умолчангию); › **Unpinned** - панели откреплены.

Поведение (Behavior)

Наименование	Описание
ShowSaveDialog	Включает отображение диалога ввода имени отчета при его сохранении. Имя отчета будет передано в параметрах дизайнера. По умолчанию свойство имеет значение **true**.
UndoMaxLevel	Устанавливает максимальную глубину отмены действий с отчетом (функция Undo/ Redo). Большое значение этого свойства будет потреблять больше памяти на стороне сервера для хранения параметров отмены. По умолчанию установлено значение **6**.
AllowChangeWindowTitle	Разрешает использование заголовка окна браузера для отображения имени файла редактируемого отчета. По умолчанию свойство имеет значение **true**.
SaveReportMode	Задает режим сохранения отчета. Может принимать одно из трех значений перечисления **StiSaveMode**. › **Hidden** - действие сохранения отчета вызывается в фоне при помощи AJAX-запроса и никак не отображается в окне браузера (значение по умолчанию); › **Visible** - действие сохранения отчета будет

© 2003-2026 Stimulsoft

	вызвано в текущем окне браузера в видимом режиме при помощи POST-запроса; › **NewWindow** - действие сохранения отчета будет вызвано в новом окне (вкладке) браузера.
SaveReportAsMode	Задает режим сохранения отчета. Может принимать одно из трех значений перечисления **StiSaveMode**. › **Hidden** - действие сохранения отчета вызывается в фоне при помощи AJAX-запроса и никак не отображается в окне браузера (значение по умолчанию); › **Visible** - действие сохранения отчета будет вызвано в текущем окне браузера в видимом режиме при помощи POST-запроса; › **NewWindow** - действие сохранения отчета будет вызвано в новом окне (вкладке) браузера.
CheckReportBeforePreview	Предоставляет возможность проверять или не проверять отчет встроенным инструментом перед его просмотром. По умолчанию свойство имеет значение **true**.

Главное меню (FileMenu)

Наименование	Описание
Visible	Включает отображение главного меню дизайнера отчетов. По умолчанию свойство имеет значение **true**.
ShowNew	Включает отображение пункта главного меню **Создать (New)**. По умолчанию свойство имеет значение **true**.
ShowFileMenuNewReport	Включает отображение кнопки **Пустой отчет (Blank Report)** в меню **Создать**. По умолчанию свойство имеет значение **true**.

© 2003-2026 Stimulsoft

ShowFileMenuNewDashboard	Включает отображение кнопки **Пустая панель индикаторов (Blank Dashboard)** в меню **Создать**. По умолчанию свойство имеет значение **true**.
ShowOpen	Включает отображение пункта главного меню **Открыть (Open)**. По умолчанию свойство имеет значение **true**.
ShowSave	Включает отображение пункта главного меню **Сохранить (Save)**. По умолчанию свойство имеет значение **true**.
ShowSaveAs	Включает отображение пункта главного меню **Сохранить как (Save As)**. По умолчанию свойство имеет значение **true**.
ShowClose	Включает отображение пункта главного меню **Закрыть (Close)**. По умолчанию свойство имеет значение **true**.
ShowExit	Включает отображение пункта главного меню **Выход (Exit)**. По умолчанию свойство имеет значение **false**.
ShowReportSetup	Включает отображение пункта главного меню **Свойства отчета (Report Setup)**. По умолчанию свойство имеет значение **true**.
ShowOptions	Включает отображение пункта главного меню **Параметры (Options)**. По умолчанию свойство имеет значение **true**.
ShowInfo	Включает отображение пункта главного меню **Сведения (Info)**. По умолчанию свойство имеет значение **true**.
ShowAbout	Включает отображение пункта главного меню **О программе (About)**. По умолчанию свойство имеет значение **true**.
ShowHelp	Включает отображение пункта главного меню **Помощь (Help)**. По умолчанию свойство имеет значение **true**.

© 2003-2026 Stimulsoft

Словарь данных (Dictionary)

Наименование	Описание
Visible	Включает отображение словаря данных отчета. По умолчанию свойство имеет значение **true**.
UseAliases	Предоставляет возможность использовать псевдонимы в словаре данных. Может принимать одно из трех значений перечисления **StiUseAliases**: › **Auto** - определяет режим использования псевдонимов из сохраненного значения в файлах cookies (значение по умолчанию); › **True** - устанавливает режим использования псевдонимов в словаре данных; › **False** - выключает режим использования псевдонимов в словаре данных.
NewReportDictionary	Предоставляет возможность создавать новый словарь данных или объединить с существующим, при создании нового отчета в дизайнере. Может принимать одно из трех значений перечисления **StiNewReportDictionary**: › **Auto** - определяет режим создания или объединения словаря данных из сохраненного значения в файлах cookies(значение по умолчанию); › **DictionaryNew** - устанавливает режим создания нового словаря данных, при создании нового отчета; › **DictionaryMerge** - устанавливает режим объединения существующего словаря данных с новым, при создании нового отчета в дизайнере.
ShowDictionaryContextMenuProperties	Устанавливает видимость пункта **Свойства (Properties)** в контекстном меню словаря

© 2003-2026 Stimulsoft

	данных. По умолчанию свойство имеет значение **true**.
ShowDictionaryActions	Устанавливает видимость меню **Действия (Actions)** на панели инструментов словаря данных. По умолчанию свойство имеет значение **true**.
PermissionDataConnections	Устанавливает доступные действия над подключениями к данным отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionDataSources	Устанавливает доступные действия над источниками данных отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionDataColumns	Устанавливает доступные действия над колонками данных отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionBusinessObjects	Устанавливает доступные действия над бизнес-объектами отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionDataRelations	Устанавливает доступные действия над связями данных отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionVariables	Устанавливает доступные действия над переменными отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionResources	Устанавливает доступные действия над ресурсами в словаре данных отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionSqlParameters	Устанавливает доступные действия над параметрами в SQL запросах источников

© 2003-2026 Stimulsoft

	данных отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionDataTransformations	Устанавливает доступные действия над преобразованием данных. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionUserFunctions	Устанавливает доступные действия над пользовательскими функциями. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.

В таблице ниже приведены все доступные значения для перечисления **StiDesignerPermissions**, которые можно установить для элементов словаря отчета.

Значение	**Описание**
None	Запрещает любое действие над элементом словаря данных.
All	Разрешает любые действия над элементом словаря данных.
Create	Разрешает создавать определенный элемент словаря данных.
Delete	Разрешает удалять определенный элемент словаря данных.
Modify	Разрешает редактировать определенный элемент словаря данных.
View	Разрешает просматривать определенный элемент словаря данных.
ModifyView	Разрешает редактировать и просматривать определенный элемент словаря данных.

Панель инструментов (Toolbar)

© 2003-2026 Stimulsoft

Наименование	Описание
ShowToolbar	Предоставляет возможность отображать или не отображать панель инструментов в дизайнере отчетов. По умолчанию свойство имеет значение **true**.
ShowSetupToolboxButton	Включает отображение кнопки вызова окна настроек боковой панели компонентов отчета. По умолчанию свойство имеет значение **true**.
ShowInsertButton	Включает отображение вкладки **Вставить (Insert)** на панели инструментов дизайнера. По умолчанию свойство имеет значение **true**.
ShowLayoutButton	Включает отображение вкладки **Формат (Layout)** на панели инструментов дизайнера. По умолчанию свойство имеет значение **true**.
ShowPageButton	Включает отображение вкладки **Страница (Page)** на панели инструментов дизайнера. По умолчанию свойство имеет значение **true**.
ShowPreviewButton	Включает отображение вкладки **Предварительный просмотр (Preview)** на панели инструментов дизайнера. По умолчанию свойство имеет значение **true**.
ShowSaveButton	Включает отображение кнопки **Сохранить (Save)** на панели инструментов дизайнера. По умолчанию свойство имеет значение **true**.
ShowAboutButton	Включает отображение кнопки **О программе (About)** на панели инструментов дизайнера. По умолчанию свойство имеет значение **false**.

Панель свойств (PropertiesGrid)

Наименование	Описание

© 2003-2026 Stimulsoft

Visible	Включает отображение панели свойств компонентов отчета. По умолчанию свойство имеет значение **true**.
Width	Задает ширину панели свойств компонентов отчета. По умолчанию установлена ширина **370** точек.
LabelWidth	Задает ширину надписей панели свойств компонентов отчета. По умолчанию установлена ширина **160** точек.
PropertiesGridPosition	Предоставляет возможность определить позицию панели свойств в дизайнере отчетов. Может принимать одно из трех значений перечисления **StiPropertiesGridPosition**. › **Left** - панель свойств будет располагаться слева; › **Right** - панель свойств будет располагаться справа.
ShowPropertiesWhichUsedFromStyles	Устанавливает видимость свойств, которые используются из стилей в дизайнере. По умолчанию свойство имеет значение **false**.

Компоненты отчета (Components)

Наименование	Описание
ShowText	Включает отображение компонента **Текст (Text)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowTextInCells	Включает отображение компонента **Текст в ячейках (Text in Cells)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowRichText	Включает отображение компонента **Rich**

© 2003-2026 Stimulsoft

	текст (Rich Text) в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowImage	Включает отображение компонента **Картинка (Image)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowBarCode	Включает отображение компонента **Штрих-код (Bar Code)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowShape	Включает отображение компонента **Геометрия (Shape)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowHorizontalLinePrimitive	Включает отображение компонента **Горизонтальная линия (Horizontal Line)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowVerticalLinePrimitive	Включает отображение компонента **Вертикальная линия (Vertical Line)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowRectanglePrimitive	Включает отображение компонента **Прямоугольник (Rectangle)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowRoundedRectanglePrimitive	Включает отображение компонента **Закругленный прямоугольник (Rounded Rectangle)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowPanel	Включает отображение компонента **Панель (Panel)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.

© 2003-2026 Stimulsoft

ShowClone	Включает отображение компонента **Клон (Clone)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowCheckBox	Включает отображение компонента **Флажок (Check Box)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowSubReport	Включает отображение компонента **Вложенный отчет (Sub Report)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowZipCode	Включает отображение компонента **Почтовый индекс (Zip Code)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowTable	Включает отображение компонента **Таблица (Table)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowCrossTab	Включает отображение компонента **Кросс-таблица (Cross-Tab)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowChart	Включает отображение компонента **Диаграмма (Chart)** в меню вставки компонентов дизайнера. Действует сразу на все типы диаграмм. По умолчанию свойство имеет значение **true**.
ShowMap	Включает отображение компонента **Карта (Map)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **false**.
ShowGauge	Включает отображение компонента **Индикатор (Gauge)** в меню вставки компонентов дизайнера. По умолчанию

© 2003-2026 Stimulsoft

	свойство имеет значение **false**.
ShowSparkline	Включает отображение компонента **Спарклайн (Sparkline)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowMathFormula	Включает отображение компонента **Математическая формула (Math Formula)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowElectronicSignature	Включает отображение компонента **Электронная подпись (Electronic Signature)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowPdfDigitalSignature	Включает отображение компонента **Цифровая подпись PDF (PDF Digital Signature)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.

Секции (Bands)

Наименование	Описание
ShowReportTitleBand	Включает отображение пункта **Заголовок отчета (Report Title)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowReportSummaryBand	Включает отображение пункта **Итог отчета (Report Summary)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowPageHeaderBand	Включает отображение пункта **Заголовок страницы (Page Header)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowPageFooterBand	Включает отображение пункта **Итог страницы (Page Footer)** в списке меню **Секции**. По умолчанию свойство имеет

© 2003-2026 Stimulsoft

	значение **true**.
ShowGroupHeaderBand	Включает отображение пункта **Заголовок группы (Group Header)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowGroupFooterBand	Включает отображение пункта **Итог группы (Group Footer)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowHeaderBand	Включает отображение пункта **Заголовок данных (Header)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowFooterBand	Включает отображение пункта **Итог данных (Footer)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowColumnHeaderBand	Включает отображение пункта **Заголовок колонки (Column Header)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowColumnFooterBand	Включает отображение пункта **Итог колонки (Column Footer)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowDataBand	Включает отображение пункта **Данные (Data)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowHierarchicalBand	Включает отображение пункта **Дерево (Hierarchical)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowChildBand	Включает отображение пункта **Подчиненный (Child)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowEmptyBand	Включает отображение пункта **Пустые данные (Empty)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowOverlayBand	Включает отображение пункта **Водяной знак**

© 2003-2026 Stimulsoft

	(Overlay) в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowTableOfContents	Включает отображение пункта **Оглавление (Table Of Contents)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.

Элементы дашборда (DashboardElements)

Наименование	Описание
ShowTableElement	Включает отображение элемента панели индикаторов **Таблица (Table)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowCardsElement	Включает отображение элемента панели индикаторов **Карточки (Cards)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowChartElement	Включает отображение элемента панели индикаторов **Диаграмма (Chart)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowGaugeElement	Включает отображение элемента панели индикаторов **Измерительный прибор (Gauge)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowPivotTableElement	Включает отображение элемента панели индикаторов **Сводная таблица (Pivot)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowIndicatorElement	Включает отображение элемента панели

© 2003-2026 Stimulsoft

	индикаторов **Индикатор (Indicator)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowProgressElement	Включает отображение элемента панели индикаторов **Прогресс (Progress)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowRegionMapElement	Включает отображение элемента панели индикаторов **Региональная карта (Region Map)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowOnlineMapElement	Включает отображение элемента панели индикаторов **Онлайн карта (Online Map)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowImageElement	Включает отображение элемента панели индикаторов **Изображение (Image)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowTextElement	Включает отображение элемента панели индикаторов **Текст (Text)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowPanelElement	Включает отображение элемента панели индикаторов **Панель (Panel)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowShapeElement	Включает отображение элемента панели индикаторов **Геометрия (Shape)** на

© 2003-2026 Stimulsoft

	инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowButtonElement	Включает отображение элемента панели индикаторов **Кнопка (Button)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowListBoxElement	Включает отображение элемента панели индикаторов **Список (ListBox)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowComboBoxElement	Включает отображение элемента панели индикаторов **Выпадающий список (ComboBox)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowTreeViewElement	Включает отображение элемента панели индикаторов **Иерархический список (TreeView)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowTreeViewBoxElement	Включает отображение элемента панели индикаторов **Выпадающий иерархический список (TreeViewBox)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowDatePickerElement	Включает отображение элемента панели индикаторов **Выбор даты (Date Picker)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.

Кросс-секции (CrossBands)

© 2003-2026 Stimulsoft

Наименование	Описание
ShowCrossGroupHeaderBand	Включает отображение секции **Кросс-заголовок группы (Cross Group Header)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowCrossGroupFooterBand	Включает отображение секции **Кросс-итог группы (Cross Group Footer)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowCrossHeaderBand	Включает отображение секции **Кросс-заголовок данных (Cross Header)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowCrossFooterBand	Включает отображение секции **Кросс-итог данных (Cross Footer)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowCrossDataBand	Включает отображение секции **Кросс-данные (Cross Data)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.

Панели индикаторов(Dashboards)

Наименование	Описание
ShowNewDashboardButton	Предоставляет возможность отображать или не отображать кнопку **Новая панель индикаторов (New Dashboard)** в дизайнере отчетов. По умолчанию свойство имеет значение **true**.

Страницы(Pages)

© 2003-2026 Stimulsoft

Наименование	Описание
ShowNewPageButton	Предоставляет возможность отображать или не отображать кнопку **Новая страница (New Page)** в дизайнере отчетов. По умолчанию свойство имеет значение **true**.

При разработке отчета или дашборда в дизайнере отчетов, можно также определить настройки экспорта (ExportOptions), параметры Email (EmailOptions) и панель инструментов (PreviewToolbarOptions) на вкладке **Просмотр (Preview)**. Эти опции аналогичны опциям вьювера отчетов.

5 Reports and Dashboards for ASP.NET Core MVC

.NET Core представляет собою кросс-платформенную технологию создания Web приложений для Windows, Linux и macOS. **ASP.NET Core MVC** является многофункциональной платформой для создания Web-приложений и API-интерфейсов с помощью структуры проектирования Model-View-Controller. Компания Stimulsoft предоставляет инструменты создания, отображения, преобразования отчетов и панелей индикаторов (дашбордов) с использованием этой технологии.

Инструменты создания и редактирования отчетов:	Инструменты отображения и конвертации отчетов:	Инструменты создания и редактирования дашбордов:	Инструменты отображения и конвертации дашбордов:
HTML5 дизайнер	HTML5 вьювер	HTML5 дизайнер	HTML5 вьювер

© 2003-2026 Stimulsoft

5.1 HTML5 вьювер

YouTube

Посмотрите видеоуроки по работе с компонентом ASP.NET Core MVC HTML5 Viewer. . Подписывайтесь на канал Stimulsoft и узнайте первыми о новых видеоуроках. Вопросы и предложения оставляйте в комментариях к видео.

Примеры

Ознакомьтесь с примерами по работе с компонентом ASP.NET Core MVC HTML5 Viewer на GitHub. Все примеры представляют собой отдельные проекты, сгруппированные в один solution для Visual Studio.

Компонент **HTML5 Viewer (StiNetCoreViewer)** предназначен для просмотра отчетов в окне браузера. При этом не требуется устанавливать у клиента .NET Framework, ActiveX компоненты или какие-либо специальные плагины. Все, что необходимо – это любой современный Web-браузер.

При помощи **HTML5 Viewer** можно просматривать, печатать отчеты, выполнять экспортирование отчетов на любом компьютере с любой установленной операционной системой. Так как вьювер использует только HTML и JavaScript технологии, он может быть запущен на устройствах, где нет поддержки Flash или Silverlight – планшеты, смартфоны. Также вьювер поддерживает Mobile и Touch интерфейсы, которые автоматически включаются при использовании мобильных устройств и мониторов с сенсорным экраном.

Компонент **HTML5 Viewer** использует технологию AJAX для выполнения всех действий (загрузки отчета, листания страниц, масштабирования, интерактивности в отчетах и др.), что позволяет избавиться от перезагрузки всей страницы, а также экономить Web-трафик и повысить скорость работы. Для построения отчетов используется ядро генератора отчетов, построенное с использованием .NET Core технологии. Данная технология является кроссплатформенной, что позволяет разворачивать приложение на серверах, использующих операционные системы Windows, macOS и Linux.

© 2003-2026 Stimulsoft

HTML5 Viewer поддерживает множество тем оформления, анимированный интерфейс, закладки, интерактивные отчеты, редактирование элементов отчета на странице, полноэкранный режим, поиск, и другие, необходимые для просмотра отчетов, возможности.

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для использования **HTML5 Viewer** в Web-проекте, необходимо установить NuGet пакет Stimulsoft.Reports.Web.NetCore:

- Выберите пункт "Manage NuGet Packages..." в контекстном меню проекта;
- На вкладке Browse, в строке поиска, укажите Stimulsoft.Reports.Web.NetCore;
- Выделите элемент, определите версию пакета и нажмите кнопку **Установить (Install)**. При обновлении пакета, следует нажать кнопку **Обновить (Update)**.

Для того, чтобы добавить возможности просмотра и экспорта дашбордов в Web-проекте, установить NuGet пакет Stimulsoft.Dashboards.Web.NetCore, (данный пакет связан с пакетом Stimulsoft.Reports.Web.NetCore, при его отсутствии он будет установлен автоматически):

- Выберите пункт "Manage NuGet Packages..." в контекстном меню проекта;
- На вкладке Browse, в строке поиска, укажите Stimulsoft.Dashboards.Web.NetCore;
- Выделите элемент, определите версию пакета и нажмите кнопку **Установить (Install)**. При обновлении пакета, следует нажать кнопку **Обновить (Update)**.

- Как это работает
- Интерактивные отчеты
- Активация
- Время ожидания
- Отображение отчета и дашборда
- Редактирование построенного отчета

© 2003-2026 Stimulsoft

- Подключение данных
- Локализация вьювера
- Печать отчета
- Экспорт отчета и дашборда
- Режимы отображения
- Работам с параметрами
- Настройки вьювера
- Отправка отчета по Email
- Вызов дизайнера из вьювера
- Использование тем
- Работа с закладками
- Вспомогательные методы
- Экспорт и печать из кода

5.1.1 Как это работает

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для запуска вьювера требуется разместить на странице компонент **StiNetCoreViewer**, задать ему необходимые свойства, а в контроллере представления определить необходимые действия. При запуске вьювера отчетов происходят следующие действия:

- .NET Core компонент генерирует HTML и JavaScript код, необходимый для отображения и работы вьювера;
- По окончании вывода компонента запускается JavaScript метод, который запрашивает на стороне сервера первую страницу отчета, весь отчет (в зависимости от выбранного режима) и его необходимые параметры;
- Каждое действие во вьювере (например, листание страниц, печать отчета, экспорт отчета и др.) вызывает определенное действие на стороне сервера, в котором можно выполнить необходимые манипуляции с отчетом;
- Для ускорения работы вьювер сохраняет отчет в кэше или сессии сервера, что позволяет исключить повторное построение отчета.

© 2003-2026 Stimulsoft

5.1.2 Активация

YouTube

Посмотрите видеоуроки по активации компонента ASP.NET Core HTML5 Viewer. Подписывайтесь на канал Stimulsoft и узнайте первыми о новых видеоуроках. Вопросы и предложения оставляйте в комментариях к видео.

После приобретения продукта Stimulsoft вам необходимо активировать лицензию для используемых компонентов. Это можно выполнить указав лицензионный ключ или загрузив файл с лицензионным ключом. Ниже представлен пример активации компонента **StiNetCoreViewer**.

HomeController.cs

```
...
//Activation with using license code
public class HomeController : Controller
{
   static HomeController()
   {
      Stimulsoft.Base.StiLicense.Key = "Your activation code...";
   }
}

//Activation with using license file
public class HomeController : Controller
{
   public HomeController(IHostingEnvironment hostEnvironment)
   {
      var path = Path.Combine(hostEnvironment.ContentRootPath, "Content\
      \license.key");
      Stimulsoft.Base.StiLicense.LoadFromFile(path);
   }
...
```

Получить лицензионный ключ или скачать файл с лицензионным ключом, вы можете в личном кабинете пользователя. Для авторизации в кабинете используйте имя пользователя и пароль, указанные при покупке подписки продукта.

© 2003-2026 Stimulsoft

5.1.3 Отображение отчета и дашборда

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Примечание

При присвоении отчета вьювер компоненту, происходит автоматическое построение отчета. Вызов метода `Report.Render()` необходим только в том случае, если требуется провести какие-либо действия с построенным отчетом перед его отображением во вьювере. Аналогично, при использовании режима компиляции, вызов метода `Report.Compile()` требуется только в том случае, если требуется провести какие-либо действия с откомпилированным отчетом перед его построением и отображением во вьювере.

Для отображения отчета необходимо добавить на страницу компонент **StiNetCoreViewer** и задать ему минимально необходимые свойства, а в контроллере представления определить необходимые действия.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Actions =
   {
      GetReport = "GetReport",
      ViewerEvent = "ViewerEvent"
   }
})
...
```

HomeController.cs

```
...
public IActionResult GetReport()
{
```

© 2003-2026 Stimulsoft

```
    StiReport report = new StiReport();
    report.Load(StiNetCoreHelper.MapPath(this, "Reports/SimpleList.mrt"));
    //report.Load(StiNetCoreHelper.MapPath(this, "Reports/Dashboard.mrt"));

    return StiNetCoreViewer.GetReportResult(this, report);
}

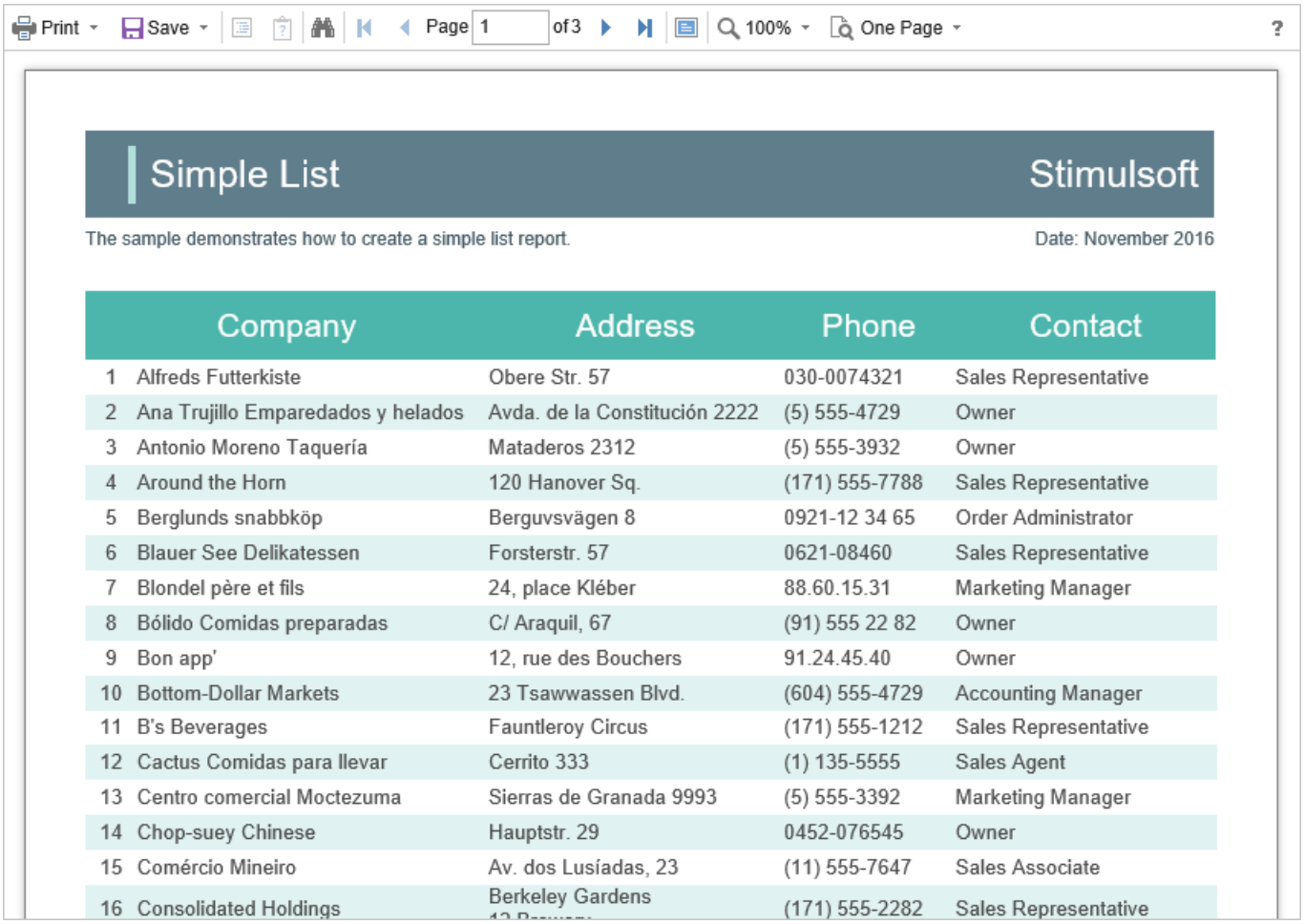
public IActionResult ViewerEvent()
{
    return StiNetCoreViewer.ViewerEventResult(this);
}
...
```

В приведенном выше примере добавлена обработка двух действий вьювера. Действие **GetReport** возвращает подготовленный отчет вьюверу, действие **ViewerEvent** обрабатывает события вьювера.

Информация

Действие **ViewerEvent** является обязательным. Без него корректная работа вьювера невозможна.

© 2003-2026 Stimulsoft



3

Если отчет не был построен перед отображением, то компонент **HTML5 Viewer** построит его автоматически. Таким образом, для отображения отчета разрешено использовать шаблоны отчетов и построенные отчеты.

HomeController.cs

```
...
public IActionResult GetReport()
{
   StiReport report = new StiReport();
   report.LoadDocument(StiNetCoreHelper.MapPath(this, "Reports/
   SimpleList.mdc"));

   return StiNetCoreViewer.GetReportResult(this, report);
}
...
```

© 2003-2026 Stimulsoft

Так как дашборд не является статическим документов и для его работы требуются данные, то формат построенного документа MDC для него не доступен. Вместо этого предоставлена возможность использовать снимок отчета в формате MRT, который содержит в себе все необходимые для работы дашборда данные, и может быть корректно отображен во вьювере.

HomeController.cs

```
...
public ActionResult GetReport()
{
   StiReport report = new StiReport();
   report.Load(StiNetCoreHelper.MapPath("~/Reports/Snapshot.mrt"));

   return StiNetCoreViewer.GetReportResult(report);
}
...
```

Подключение пользовательских шрифтов

Вы можете подключить пользовательские шрифты с использованием класса **StiFontCollection**, указав файл содержащий шрифт. Для этого необходимо в конструкторе контроллера вызвать статический метод для загрузки шрифта.

ViewerController.cs

```
...
public class ViewerController : Controller
{
   static ViewerController()
   {
      Stimulsoft.Base.StiFontCollection.AddFontFile(StiNetCoreHelper.MapPat
      h(this,"/fonts/my-font/font-name.ttf"));
   }
}
...
```

© 2003-2026 Stimulsoft

5.1.4 Подключение данных

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Данные для построения отчета могут быть подключены различными способами. Самый простой – хранить настройки подключения в самом шаблоне отчета. Так же данные можно подключить из кода, это можно сделать при загрузке отчета в действии **GetReport**.

HomeController.cs

```
...
public IActionResult GetReport()
{
   DataSet ds = new DataSet();
   ds.ReadXml(StiNetCoreHelper.MapPath(this, "Data/Demo.xml"));

   StiReport report = new StiReport();
   report.Load(StiNetCoreHelper.MapPath(this, "Reports/
   TwoSimpleLists.mrt"));
   report.Dictionary.Databases.Clear();
   report.RegData("Demo", ds);

   return StiNetCoreViewer.GetReportResult(this, report);
}
...
```

Данные для отчета можно подключать не только при загрузке отчета. Например, можно подключить новые данные в момент интерактивных действий во вьювере (применение параметров отчета, сортировка, детализация, сворачивание). Для этого необходимо задать действие **Interaction** для компонента **HTML5 Viewer**, а в обработчике действия подключить данные для текущего отчета. Таким же способом можно подключить данные в других действиях вьювера.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Actions =
   {
      GetReport = "GetReport",
      ViewerEvent = "ViewerEvent",
      Interaction = "ViewerInteraction"
   }
   })
...
```

HomeController.cs

```
...
public IActionResult ViewerInteraction()
{
   DataSet data = new DataSet();
   data.ReadXml(StiNetCoreHelper.MapPath(this, "Data/Demo.xml"));

   StiReport report = StiNetCoreViewer.GetReportObject(this);
   report.RegData("Demo", data);

   return StiNetCoreViewer.InteractionResult(this, report);
}
...
```

При необходимости подключить новые данные только для определенного интерактивного действия вьювера, например, только при применении параметров отчета, можно воспользоваться параметрами вьювера. Параметры вьювера представлены в виде объекта класса **StiRequestParams**, они передаются при любом запросе на сторону сервера, и содержат все необходимые сведения и состояния клиентской части вьювера. Для определения типа действия вьювера достаточно проверить свойство **Action** у параметров вьювера.

© 2003-2026 Stimulsoft

HomeController.cs

```
...
public IActionResult ViewerInteraction()
{
   StiRequestParams requestParams =
   StiNetCoreViewer.GetRequestParams(this);
   if (requestParams.Action == StiAction.Variables)
   {
      DataSet data = new DataSet();
      data.ReadXml(StiNetCoreHelper.MapPath(this, "Data/Demo.xml"));

      StiReport report = StiNetCoreViewer.GetReportObject(this);
      report.RegData("Demo", data);

      return StiNetCoreViewer.InteractionResult(this, report);
   }

   return StiNetCoreViewer.InteractionResult(this);
}
...
```

SQL источники данных

Параметры подключения к SQL источнику данных, как и к любому другому, можно хранить в самом шаблоне отчета. Если требуется задать параметры подключения из кода перед построением отчета (например, по причине безопасности или в зависимости от авторизованного пользователя), можно воспользоваться приведенным ниже примером.

© 2003-2026 Stimulsoft

HomeController.cs

```
...
public IActionResult GetReport()
{
   OracleConnection connection = new OracleConnection("Data
   Source=Oracle8i;Integrated Security=yes");
   connection.Open();
   OracleDataAdapter adapter = new OracleDataAdapter();
   adapter.SelectCommand = new OracleCommand("SELECT * FROM Products",
   connection);

   DataSet dataSet = new DataSet("productsDataSet");
   adapter.Fill(dataSet, "Products");

   StiReport report = new StiReport();
   report.Load(StiNetCoreHelper.MapPath(this, "Reports/
   SqlSampleReport.mrt"));
   report.RegData("Products", dataSet);

   return StiNetCoreViewer.GetReportResult(this, report);
}
...
```

Также, для SQL источников данных используемых в отчете, можно указать **Время ожидания запроса (Query Timeout)** в секундах. Значение данного свойства сохраняется в самом шаблоне отчёта для каждого SQL подключения отдельно.

Ниже приведен пример кода, который предоставляет возможность изменить строку соединения для MS SQL, скорректировать запрос, установить время ожидания запроса для уже созданного соединения и источников данных в отчете.

© 2003-2026 Stimulsoft

HomeController.cs

```
...
public IActionResult GetReport()
{

   StiReport report = new StiReport();
   report.Load(Server.MapPath("Report.mrt"));
   ((StiSqlDatabase)
   report.Dictionary.Databases["Connection"]).ConnectionString = @"Data
   Source=server;Integrated Security=True;Initial Catalog=DataBase";
   ((StiSqlSource)
   report.Dictionary.DataSources["DataSourceName"]).SqlCommand = "select *
   from Table where Column = 100";
   ((StiSqlSource)
   report.Dictionary.DataSources["DataSourceName"]).CommandTimeout = 1000;

   return StiNetCoreViewer.GetReportResult(this, report);
}
...
```

Информация

Для SQL источников данных других типов подключение создается аналогично, при этом подключается адаптер, соответствующий типу источника данных. Например, для MS SQL источника необходимо подключать SqlDataAdapter, для Oracle необходим OracleDataAdapter. Также, при этом следует указать строку соединения, соответствующую типу подключения.

Вы также можете использовать данные для разработки отчетов и дашбордов полученные из OData хранилищ данных. В этом случае, авторизация может быть выполнена с использованием имени, пароля пользователя или с использованием токена. Параметры авторизации указываются в строке соединения к OData хранилищу с использованием разделителя ";".

© 2003-2026 Stimulsoft

HomeController.cs

```
...
public IActionResult GetReport()
{

   StiReport report = new StiReport();

   //Authorization using a user account
   var oDataDatabase = new StiODataDatabase("OData", "OData", @"https://
   services.odata.org/V4/Northwind/
   Northwind.svc;AddressBearer=adress;UserName=UserName;Password=Password;C
   lient_Id=Your Client ID", false, null);

   //Authorization using a user token
   var oDataDatabase = new StiODataDatabase("OData", "OData", @"https://
   services.odata.org/V4/Northwind/Northwind.svc;Token=Enter your token",
   false, null);

   report.Dictionary.Databases.Add(oDataDatabase);
   oDataDatabase.Synchronize(report);

   //Query with data filter
   ((StiSqlSource)report.Dictionary.DataSources["Products"]).SqlCommand =
   "Products?$filter=ProductID eq 2";

   return StiNetCoreViewer.GetReportResult(this, report);
}
...
```

В таблице ниже представлены шаблоны строк подключения для различных типов источников данных.

Источник данных	Шаблон строки соединения
MS SQL	Integrated Security=False; Data Source=myServerAddress;Initial Catalog=myDataBase; User ID=myUsername; Password=myPassword;
MySQL	Server=myServerAddress; Database=myDataBase;UserId=myUsername; Pwd=myPassword;
ODBC	Driver={SQL Server}; Server=myServerAddress;Database=myDataBase ; Uid=myUsername; Pwd=myPassword;
OLE DB	Provider=SQLOLEDB.1; Integrated

© 2003-2026 Stimulsoft

	Security=SSPI;Persist Security Info=False; Initial Catalog=myDataBase;Data Source=myServerAddress
Oracle	Data Source=TORCL;User Id=myUsername;Password=myPassword;
MS Access	Provider=Microsoft.Jet.OLEDB.4.0;User ID=Admin;Password=pass;Data Source=C:\ \myAccessFile.accdb;
PostgreSQL	Server=myServerAddress; Port=5432; Database=myDataBase;User Id=myUsername; Password=myPassword;
Firebird	User=SYSDBA; Password=masterkey; Database=SampleDatabase.fdb;DataSource=my ServerAddress; Port=3050; Dialect=3; Charset=NONE;Role=; Connection lifetime=15; Pooling=true; MinPoolSize=0;MaxPoolSize=50; Packet Size=8192; ServerType=0;
SQL CE	Data Source=c:\MyData.sdf; Persist Security Info=False;
SQLite	Data Source=c:\mydb.db; Version=3;
DB2	Server=myAddress:myPortNumber;Database=m yDataBase;UID=myUsername;PWD=myPassword; Max Pool Size=100;Min Pool Size=10;
Infomix	Database=myDataBase;Host=192.168.10.10;Serv er=db_engine_tcp;Service=1492;Protocol=onsoc tcp;UID=myUsername;Password=myPassword;
Sybase	Data Source=myASEserver;Port=5000;Database=myD ataBase;Uid=myUsername;Pwd=myPassword;
Teradata	Data Source=myServerAddress;User ID=myUsername;Password=myPassword;
VistaDB	Data Source=D:\folder \myVistaDatabaseFile.vdb4;Open Mode=ExclusiveReadWrite;
Universal(dotConnect)	Provider=Oracle;direct=true;data

© 2003-2026 Stimulsoft

	source=192.168.0.1;port=1521;sid=sid;user=user;password=pass
MongoDB	mongodb://<user>:<password>@localhost/test
OData	http://services.odata.org/v3/odata/OData.svc/

Информация

В таблице представлены наиболее часто используемые шаблоны для строки соединения. Посмотреть различные варианты строк соединения вы можете на специализированном сайте.

Данные из XML, JSON, Excel файлов

Подключение к XML и JSON источникам данных можно хранить в шаблоне отчета. Если требуется указать файлы данных из кода, можно воспользоваться приведенным ниже примером.

HomeController.cs

```
...
public IActionResult GetReport()
{
   DataSet data = new DataSet();
   data.ReadXml(StiNetCoreHelper.MapPath(this, "Data/Demo.xml"));

   StiReport report = new StiReport();
   report.Load(StiNetCoreHelper.MapPath(this, "Reports/SimpleList.mrt"));
   report.RegData(data);

   return StiNetCoreViewer.GetReportResult(this, report);
}
...
```

© 2003-2026 Stimulsoft

HomeController.cs

```
...
public IActionResult GetReport()
{
   DataSet data
   = StiJsonToDataSetConverterV2.GetDataSetFromFile(StiNetCoreHelper.MapPat
   h(this, "Data/Demo.json"));

   StiReport report = new StiReport();
   report.Load(StiNetCoreHelper.MapPath(this, "Reports/SimpleList.mrt"));
   report.RegData(data);

   return StiNetCoreViewer.GetReportResult(this, report);
}
...
```

Информация

Во вьювере присутствует возможность получения данных из Excel файла. Для этого можно воспользоваться указанным ниже методом.

```
DataSet dataSet = StiExcelConnector.Get().GetDataSet(new
StiExcelOptions(array, this.FirstRowIsHeader));
```

5.1.5 Локализация

Компонент **HTML5 Viewer** поддерживает полную локализацию своего интерфейса. Для того чтобы локализовать интерфейс вьювера отчетов на необходимый язык, используется специальное свойство **Localization**. В качестве значения данного свойства необходимо указать путь к XML-файлу локализации (относительный либо абсолютный).

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Localization = "Localization/en.xml"
})
...
```

При загрузке вьювера отчетов файл локализации будет загружен автоматически.

© 2003-2026 Stimulsoft

5.1.6 Использование тем

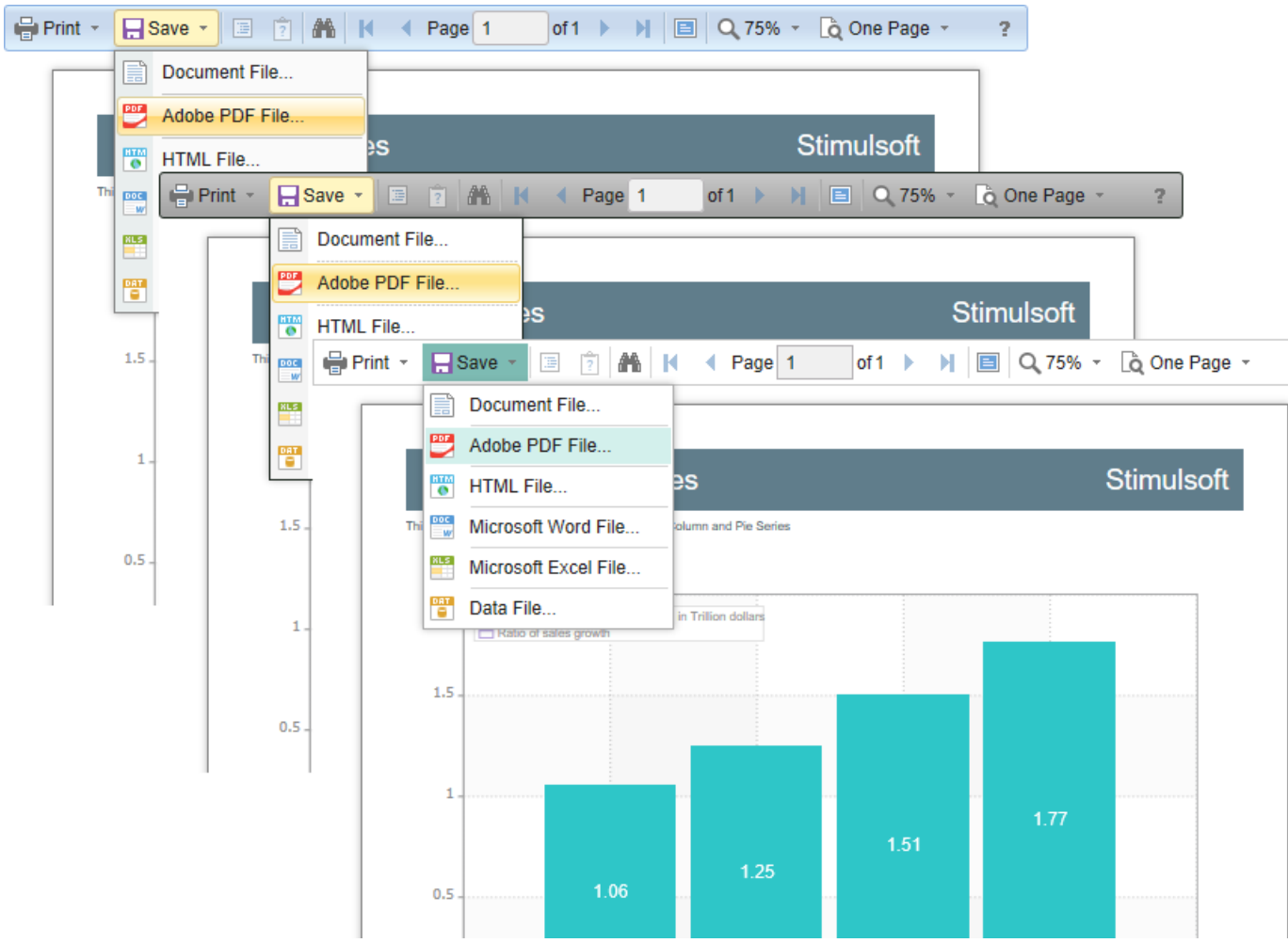
Компонент **HTML5 Viewer** имеет возможность смены тем оформления визуальных элементов управления. Для изменения темы используется свойство **Theme**, которое может принимать одно из значений перечисления **StiViewerTheme**.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Theme = StiViewerTheme.Office2022WhiteTeal
})
...
```

На данный момент доступно **8 тем** оформления с различными цветовыми акцентами. В результате, доступно более **60** вариантов оформления. Это позволяет настроить внешний вид вьювера практически под любое оформление Web проекта.

© 2003-2026 Stimulsoft



По умолчанию вьювер имеет только верхнюю панель инструментов, на которой находятся все элементы управления отчетом. При необходимости, панель инструментов можно разделить на верхнюю и нижнюю. На верхней панели будут находиться меню печати и экспорта отчета, а также кнопки работы с параметрами и закладками. Нижняя панель инструментов будет содержать элементы переключения между страницами отчета и меню управления масштабом. Для включения указанного режима предназначено свойство **DisplayMode**, которое может иметь значение **Simple** (стандартный простой режим) и **Separated** (раздельный режим).

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Appearance =
   {
      ScrollbarsMode = true
   },
   Toolbar =
   {
      DisplayMode = StiToolbarDisplayMode.Separated
   }
})
...
```

Дополнительно предусмотрена возможность установки параметров оформления основных элементов вьювера. Например, можно изменить шрифт и цвет надписей панели управления вьювера, установить фон вьювера, задать цвет границ страницы и др. Ниже представлен список доступных свойств, изменяющих оформление вьювера, и их значения по умолчанию.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Appearance =
   {
      BackgroundColor = Color.White,
      PageBorderColor = Color.Blue,
      ShowPageShadow = true
   },
   Toolbar =
   {
      BackgroundColor = Color.White,
      BorderColor = Color.Gray,
      FontColor = Color.Black,
      FontFamily = "Arial"
   }
})
...
```

© 2003-2026 Stimulsoft

5.1.7 Основные возможности

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

К основным возможностям вьювера можно отнести следующие операции: переключение между страницами отчета, изменение масштаба и режима отображения отчета, печать отчета, экспортирование отчета, применение параметров отчета, сортировку, сворачивание и детализацию. Все указанные операции выполняются в AJAX-режиме без перезагрузки страницы браузера. Для корректной работы этих операций необходимо определить специальное действие **ViewerEvent**.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Actions =
   {
      ViewerEvent = "ViewerEvent"
   }
})
...
```

HomeController.cs

```
...
public IActionResult ViewerEvent()
{
   // Some code before viewer event
   // ...

   return StiNetCoreViewer.ViewerEventResult(this);
}
...
```

© 2003-2026 Stimulsoft

Информация

Данное действие являются обязательным. Без него корректная работа вьювера невозможна.

Действие **ViewerEvent** возвращает подготовленную HTML страницу отчета (или набор страниц), построенную с учетом текущего состояния вьювера. При необходимости в указанном действии можно изменить параметры текущего отчета, а также обновить данные отчета в случае интерактивных действий вьювера.

5.1.8 Печать отчета

Информация

Обратите внимание, что возможность печати доступна только для отчетов, и недоступна для дашбордов.

В компоненте **HTML5 Viewer** предусмотрено несколько вариантов печати отчета. Каждый имеет свои особенности, достоинства и недостатки.

Печать в PDF (Print to PDF)
Печать будет осуществляться посредством экспортирования отчета в **PDF формат**. К достоинствам можно отнести большую точность расположения и печати элементов отчета по сравнению с остальными вариантами печати. Из недостатков можно упомянуть обязательное наличие установленного в браузере плагина для просмотра PDF файлов (современные браузеры имеют встроенное средство просмотра и печати PDF файлов).

Печать с просмотром (Print with Preview)
Печать отчета будет осуществлена в отдельном всплывающем окне браузера в **HTML формат**. Отчет можно предварительно просмотреть, а затем отправить на принтер или скопировать в другое место в виде текста или HTML-кода. К достоинствам можно отнести кроссбраузерность при печати, отсутствие необходимости установки специальных плагинов. Недостатком является

© 2003-2026 Stimulsoft

относительно невысокая точность расположения элементов отчета, обусловленная особенностями реализации HTML-форматирования.

Печать без просмотра (Print without Preview)
Печать отчета будет осуществлена непосредственно на принтер без предварительного просмотра. После выбора этого пункта меню отображается системный диалог печати. Так как печать в этом режиме осуществляется в HTML формат, то качество печати аналогично качеству печати отчета с предварительным просмотром.

Информация

При печати в **HTML формат** необходимо убедиться в соответствии параметров страницы отчета и параметров страницы принтера (размер бумаги, ориентация, поля, отступы), а также проверить настройки печати браузера, такие как отступы, колонтитулы, печать фоновых изображений, цветная печать.

Для работы функции печати не требуется дополнительных настроек вьювера. Если необходимо выполнить какие-либо действия перед печатью отчета, то можно определить специальное действие **PrintReport**.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Actions =
   {
      PrintReport = "PrintReport"
   }
})
...
```

© 2003-2026 Stimulsoft

HomeController.cs

```
...
public IActionResult PrintReport()
{
   // Some code before print
   // ...

   return StiNetCoreViewer.PrintReportResult(this);
}
...
```

Настройки печати

При выборе печати отчета на панели вьювера, отображается меню с выбором варианта печати. Компонент **HTML5 Viewer** имеет возможность принудительно установить требуемый режим печати. Для этого достаточно установить свойство **PrintDestination** в одно из указанных ниже значений перечисления **StiPrintDestination**.

- **Default** – при выборе печати будет отображено меню (значение свойства по умолчанию);
- **Pdf** – печать в PDF формат;
- **Direct** – печать в HTML формат непосредственно на принтер, будет отображен системный диалог печати;
- **WithPreview** – печать в HTML формат с предварительным просмотром во всплывающем окне.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Toolbar =
   {
      PrintDestination = StiPrintDestination.Default
   }
})
...
```

Компонент **HTML5 Viewer** имеет возможность полностью отключить печать отчета, если она не требуется. Для этого необходимо установить значение **false** для свойства **ShowPrintButton**.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Toolbar =
   {
      ShowPrintButton = false
   }
})
...
```

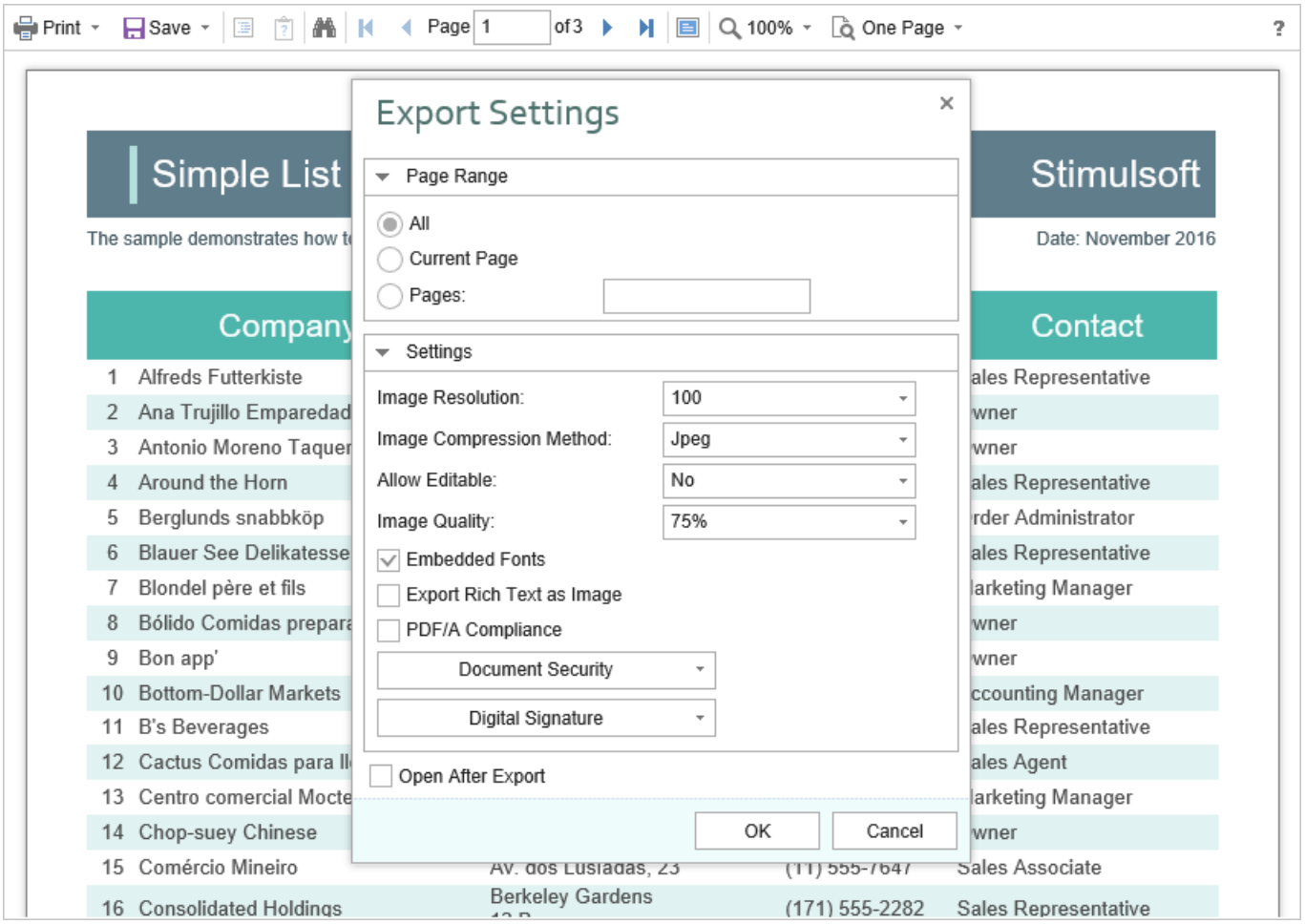
5.1.9 Экспорт отчета и дашборда

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Компонент **HTML5 Viewer** позволяет экспортировать отображаемый отчет в три десятка различных форматов, таких как **PDF**, **HTML**, **Word**, **Excel**, **XPS**, **RTF**, изображения, текст и другие. Экспорт панели индикаторов (дашборда) возможен в PDF, Excel, файлы изображений. Для работы функции экспорта не требуется дополнительных настроек вьювера.

© 2003-2026 Stimulsoft



Для работы функции экспорта не требуется дополнительных настроек вьювера. Если необходимо выполнить какие-либо действия перед экспортированием отчета, то можно определить специальное действие **ExportReport**.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Actions =
   {
      ExportReport = "ExportReport"
   }
})
...
```

© 2003-2026 Stimulsoft

HomeController.cs

```
...
public IActionResult ExportReport()
{
   // Some code before export
   // ...

   return StiNetCoreViewer.ExportReportResult(this);
}
...
```

Настройки экспорта

Каждый формат экспорта отчета компонента **HTML5 Viewer** имеет множество настроек, и каждая настройка имеет свои значения по умолчанию. Иногда требуется установить другие значения по умолчанию. Для этого предназначено специальное свойство вьювера **DefaultSettings**, которое представляет собой контейнер всех настроек экспортов, используемых по умолчанию.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Exports =
   {
      DefaultSettings =
      {
         ExportToPdf =
         {
            ImageQuality = 0.75f,
            ImageFormat = Stimulsoft.Report.Export.StiImageFormat.Color
         },
         ExportToHtml =
         {
            ExportMode = Stimulsoft.Report.Export.StiHtmlExportMode.Div,
            UseEmbeddedImages = true
         }
      }
   }
})
...
```

Если требуется, можно полностью убрать отображение диалоговых окон экспорта, экспортирование всегда будет осуществляться с настройками по умолчанию. Для этого достаточно установить значение **false** для свойства **ShowExportDialog**.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Exports =
   {
      ShowExportDialog = false
   }
})
...
```

Компонент **HTML5 Viewer** содержит порядка 30 различных форматов экспорта, и иногда требуется отключить не используемые форматы. Это позволяет разгрузить интерфейс и упростить использование вьювера. Для отключения не используемых форматов экспортов достаточно установить значение **false** для соответствующих свойств вьювера, представленных в списке ниже.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Exports =
   {
      ShowExportToDocument = true,
      ShowExportToPdf = true,
      ShowExportToXps = true,
      ShowExportToPowerPoint = true,
      ShowExportToHtml = true,
      ShowExportToHtml5 = true,
      ShowExportToMht = true,
      ShowExportToText = true,
      ShowExportToRtf = true,
      ShowExportToWord = true,
      ShowExportToOpenDocumentWriter = true,
      ShowExportToExcel = true,
      ShowExportToExcelXml = true,
      ShowExportToExcelBiff = true,
      ShowExportToOpenDocumentCalc = true,
      ShowExportToCsv = true,
      ShowExportToDbf = true,
      ShowExportToXml = true,
      ShowExportToDif = true,
      ShowExportToSylk = true,
      ShowExportToImageBmp = true,
      ShowExportToImageGif = true,
      ShowExportToImageJpeg = true,
      ShowExportToImagePcx = true,
      ShowExportToImagePng = true,
      ShowExportToImageTiff = true,
      ShowExportToImageMetafile = true,
      ShowExportToImageSvg = true,
      ShowExportToImageSvgz = true
   }
})
...
```

Компонент **HTML5 Viewer** имеет возможность полностью отключить меню экспорта отчета, если оно не требуется. Для этого необходимо установить значение **false** для свойства **ShowSaveButton**.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Toolbar =
   {
      ShowSaveButton = false
   }
})
...
```

© 2003-2026 Stimulsoft

5.1.10 Режимы отображения

В компоненте **HTML5 Viewer** предусмотрено два режима отображения отчета: с полосами прокрутки и без них. По умолчанию установлен режим просмотра без полос прокрутки. Для включения режима просмотра с полосами прокрутки достаточно установить значение **true** для свойства **ScrollbarsMode**.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Appearance =
   {
      ScrollbarsMode = true
   }
})
...
```

В первом режиме (без полос прокрутки), вьювер отображает страницу или отчет целиком, автоматически растягивая область просмотра. Если заданы размеры по ширине и высоте, то вьювер будет осуществлять обрезку вышедшей за границы страницы. Во втором режиме, в отличие от первого, при выходе страницы за границы размеров вьювера, обрезка осуществляться не будет. Вместо этого появятся полосы прокрутки, при помощи которых можно просмотреть страницу или отчет целиком.

Информация

В режиме просмотра отчета с полосами прокрутки необходимо задать высоту вьювера, иначе будет задана высота по умолчанию, равная **650 пикселей**.

В компоненте **HTML5 Viewer** предусмотрен режим полноэкранного отображения отчета или дашборда. По умолчанию включен стандартный режим просмотра, вьювер имеет заданные в настройках размеры. Для включения полноэкранного режима просмотра достаточно установить значение **true** для свойства **FullScreenMode**.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Appearance =
   {
      FullScreenMode = true
   }
})
...
```

Также, для включения либо отключения полноэкранного режима можно воспользоваться соответствующей кнопкой на панели управления вьювера.

В компоненте **HTML5 Viewer** доступно три режима отображения отчета: постраничное отображение, отчет целиком в виде ленты, и табличное отображение страниц отчета. Для управления режимами предназначено свойство **ViewMode**, которое соответственно принимает одно из указанных значений: **SinglePage**, **Continuous**, **MultiplePages**.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Toolbar =
   {
      ViewMode = StiWebViewMode.SinglePage
   }
})
...
```

В компоненте **HTML5 Viewer** реализована поддержка работы как с обычным компьютером, так и с сенсорными экранами, мобильными устройствами. Для управления режимами интерфейса предназначено свойство **InterfaceType**, которое принимает одно из следующих значений:

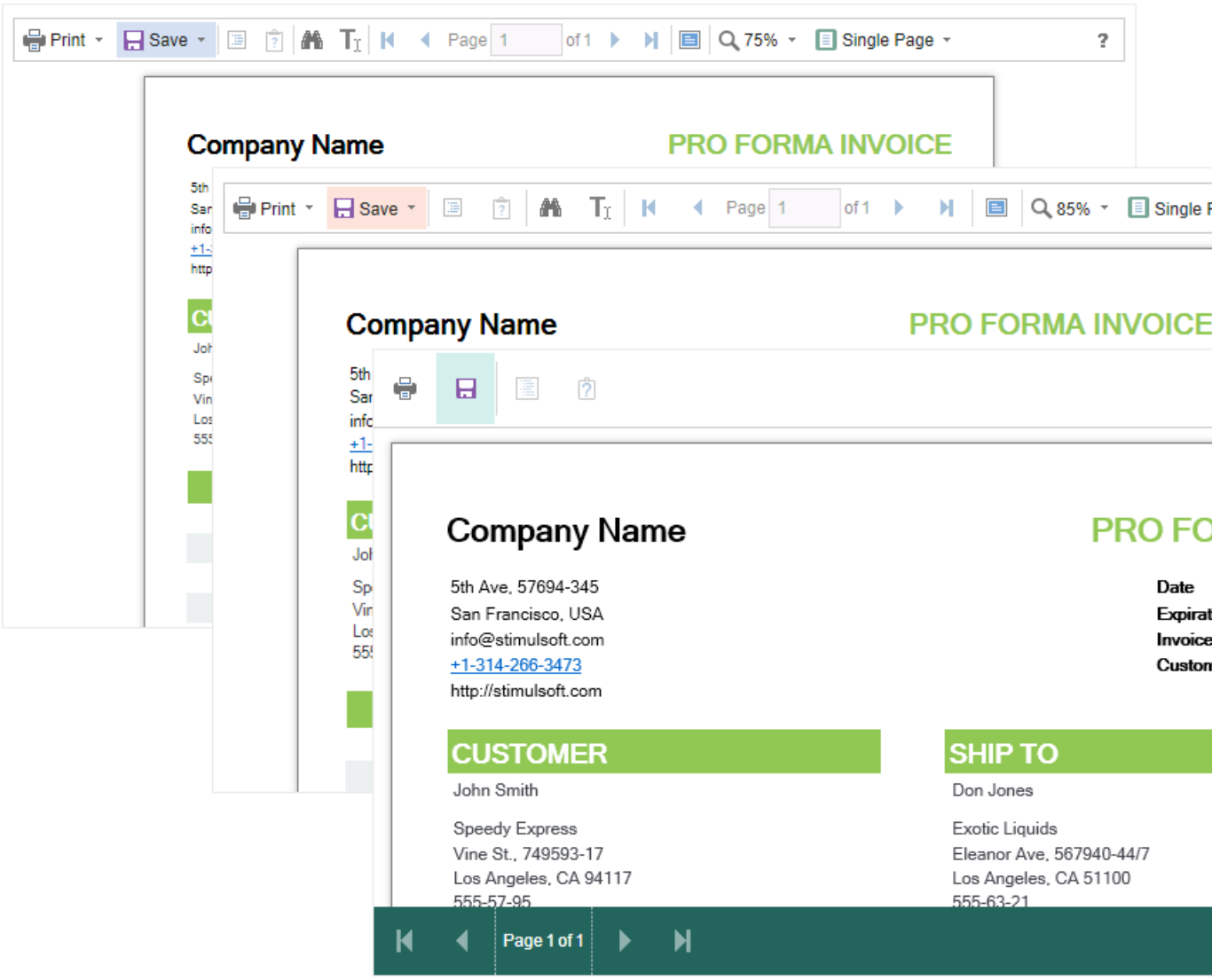
- **Auto** – тип интерфейса вьювера будет выбран автоматически в зависимости от используемого устройства (значение по умолчанию);
- **Mouse** – принудительное использование стандартного интерфейса для управления вьювером при помощи мыши;
- **Touch** – принудительное использование Touch интерфейса для управления вьювером при помощи сенсорного экрана монитора, в этом режиме элементы интерфейса вьювера имеют увеличенные размеры для удобства управления;
- **Mobile** – принудительное использование Mobile интерфейса для управления

© 2003-2026 Stimulsoft

вьювером при помощи экрана смартфона, в этом режиме интерфейс вьювера имеет упрощенный вид и адаптирован для управления при помощи мобильного устройства.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Appearance =
   {
      InterfaceType = StiInterfaceType.Auto
   }
})
...
```



© 2003-2026 Stimulsoft

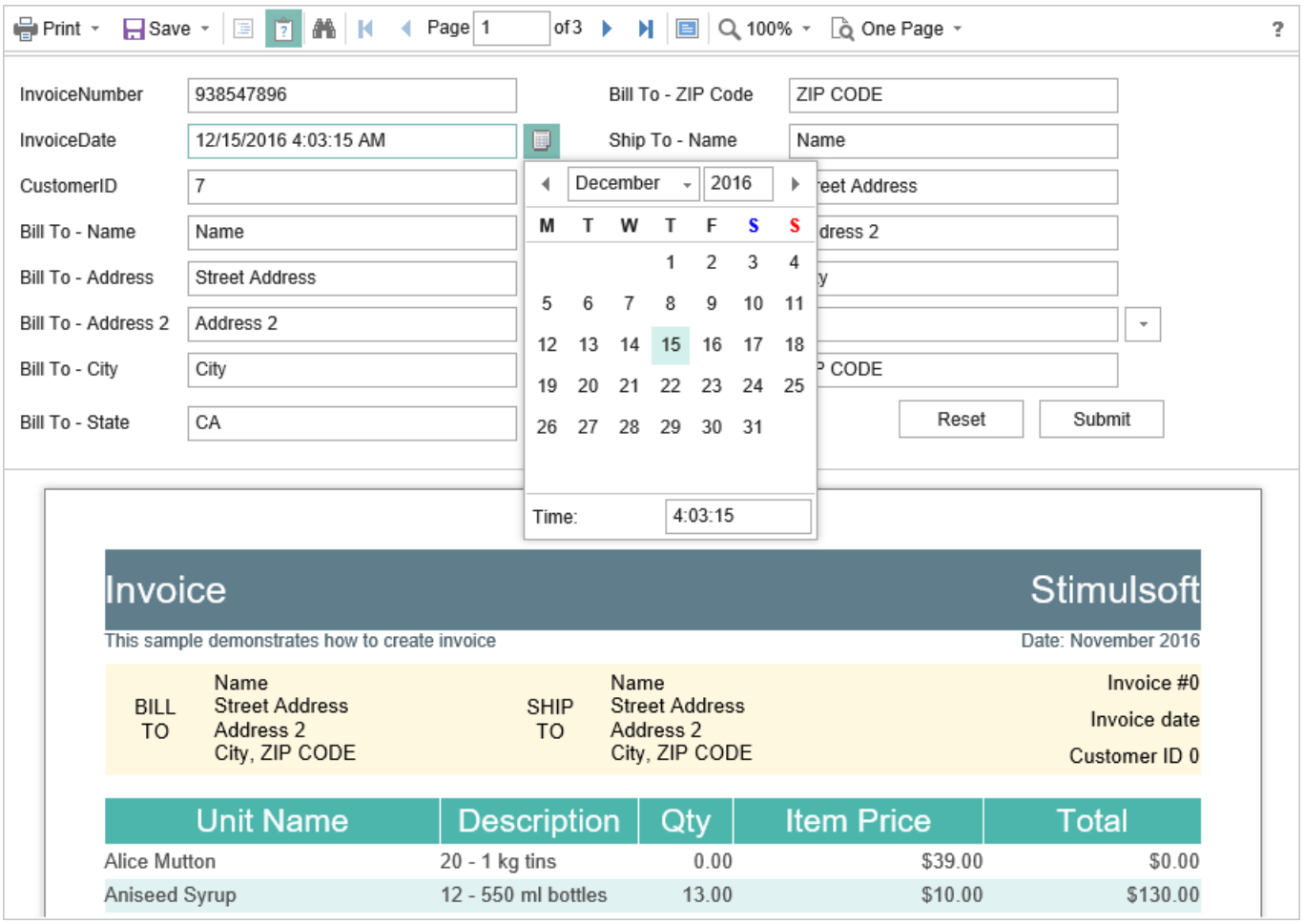
5.1.11 Работа с параметрами

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для работы с параметрами отчета в **HTML5 Viewer** реализована поддержка специальной панели параметров. Для добавления параметра на панель необходимо в отчете определить переменную, запрашиваемую у пользователя. При просмотре отчета во вьювере такая переменная будет автоматически добавлена на панель параметров. Поддерживаются все типы переменных отчета (обычные переменные, дата и время, границы, списки и др.).

© 2003-2026 Stimulsoft



Для работы отчетов с параметрами не требуется дополнительных настроек вьювера. Если необходимо провести какие-либо действия перед применением параметров, то можно определить специальное действие **Interaction**.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Actions =
   {
      Interaction = "ViewerInteraction"
   }
})
...
```

HomeController.cs

```
...
public IActionResult ViewerInteraction()
```

© 2003-2026 Stimulsoft

```
{
   // Some code before any interaction
   // ...

   return StiNetCoreViewer.InteractionResult(this);
}
...
```

Данное действие вызывается при любых интерактивных действиях вьювера. При необходимости выполнить какие-либо действия только при применении параметров отчета, можно воспользоваться параметрами вьювера. Параметры вьювера представлены в виде объекта класса **StiRequestParams**, они передаются при любом запросе на сторону сервера, и содержат все необходимые сведения и состояния клиентской части вьювера. Для определения типа действия вьювера достаточно проверить свойство **Action** у параметров вьювера.

HomeController.cs

```
...
public IActionResult ViewerInteraction()
{
   StiRequestParams requestParams =
   StiNetCoreViewer.GetRequestParams(this);
   if (requestParams.Action == StiAction.Variables)
   {
      // Some code before apply parameters
   }

   return StiNetCoreViewer.InteractionResult(this);
}
...
```

Если работа с параметрами не требуется, то можно полностью отключить данную возможность. Для этого предназначено свойство **ShowParametersButton** в разделе свойств **Toolbar**, для которого необходимо установить значение **false**.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Toolbar =
   {
      ShowParametersButton = false
   }
})
...
```

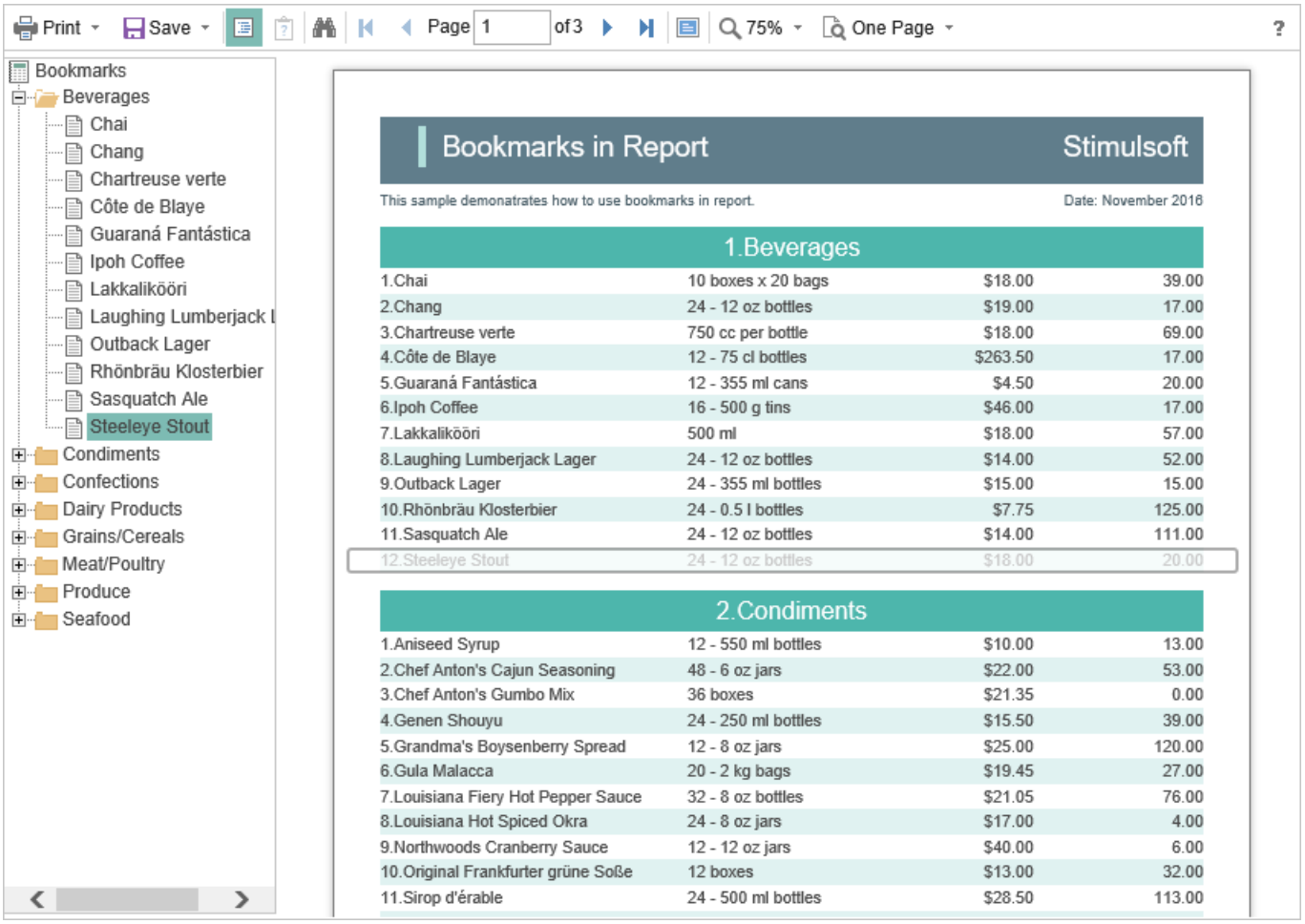
Информация

При такой конфигурации вьювера панель параметров не будет показана даже в том случае, если параметры присутствуют в отображаемом отчете.

5.1.12 Работа с закладками

В компоненте **HTML5 Viewer** реализована поддержка закладок отчета. При отображении такого отчета слева от страницы будет отображена панель с закладками. При выборе закладки отчета вьювер осуществит автоматический переход на нужную страницу, а элемент отчета с закладкой будет подсвечен.

© 2003-2026 Stimulsoft



По умолчанию ширина панели закладок равна 180 пикселей, компонент **HTML5 Viewer** позволяет изменить это значение. Для этого предназначено свойство **BookmarksTreeWidth**, значение которого указывается в пикселях.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Appearance =
   {
      BookmarksTreeWidth = 200
   }
})
...
```

Если работа с закладками отчета не требуется, то можно полностью отключить данную возможность. Для этого предназначено свойство

© 2003-2026 Stimulsoft

ShowBookmarksButton, которое необходимо установить в значение **false**.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Toolbar =
   {
      ShowBookmarksButton = false
   }
})
...
```

Информация

В этом случае закладки отчета не будут показаны, даже если они присутствуют в отображаемом отчете. Данное свойство не оказывает влияние на печать и экспортирование отчета.

При печати отчета с закладками дерево закладок будет скрыто. Если помимо самого отчета требуется распечатать и закладки, то необходимо установить свойство **BookmarksPrint** в значение **true**.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions () {
   Appearance =
   {
      BookmarksPrint = true
   }
})
...
```

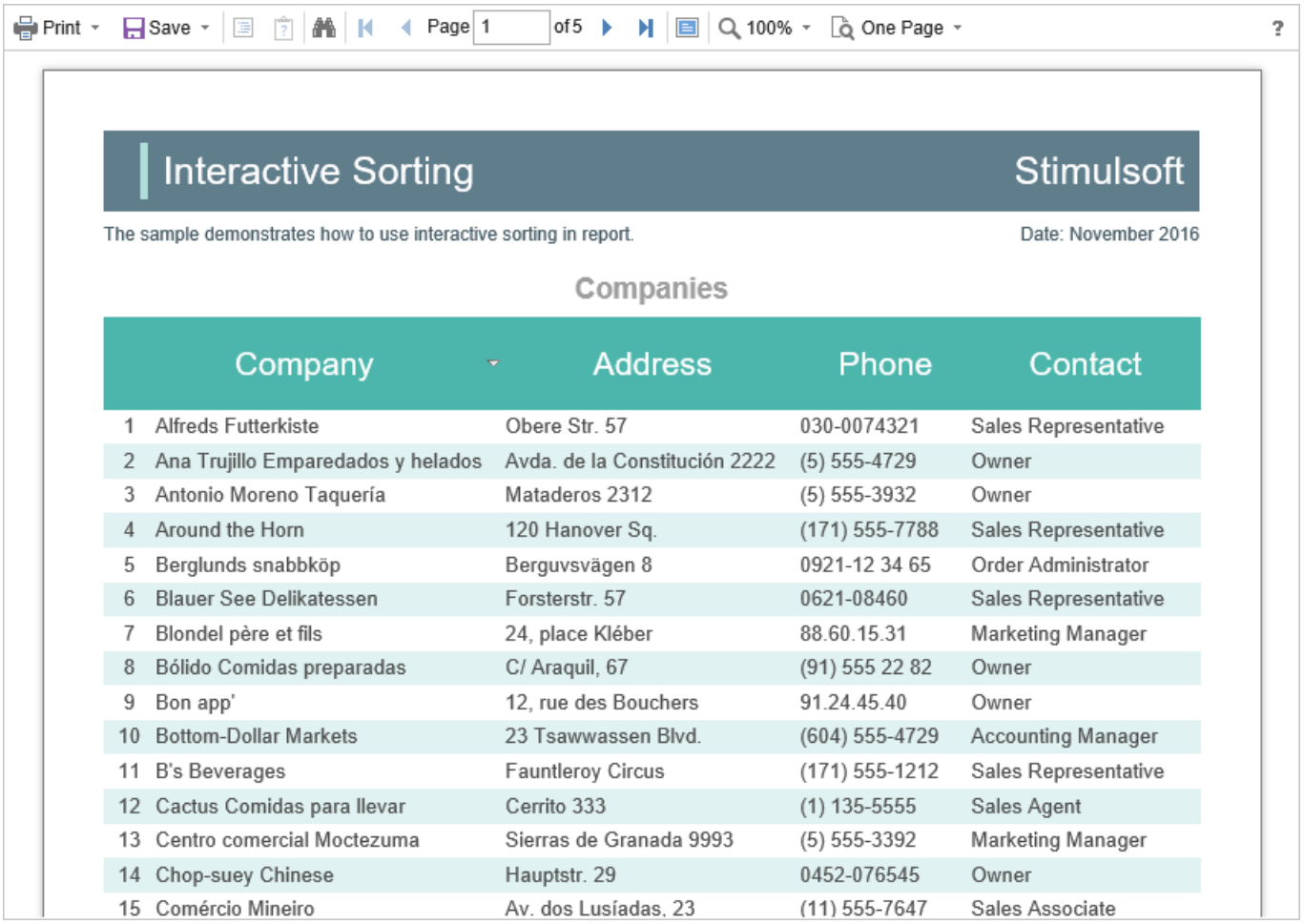
5.1.13 Динамическое сворачивание, сортировка и детализация

В компоненте **HTML5 Viewer** реализована поддержка динамической сортировки, сворачивания и детализации отчетов. Динамическая сортировка предоставляет возможность изменять направление сортировки в построенном отчете. Для этого следует щелкнуть по компоненту, у которого была установлена динамическая сортировка. Динамическая сортировка осуществляется в следующих направлениях: **По возрастанию (Ascending)** и **По убыванию (Descending)**. Каждый раз при щелчке по компоненту направление

© 2003-2026 Stimulsoft

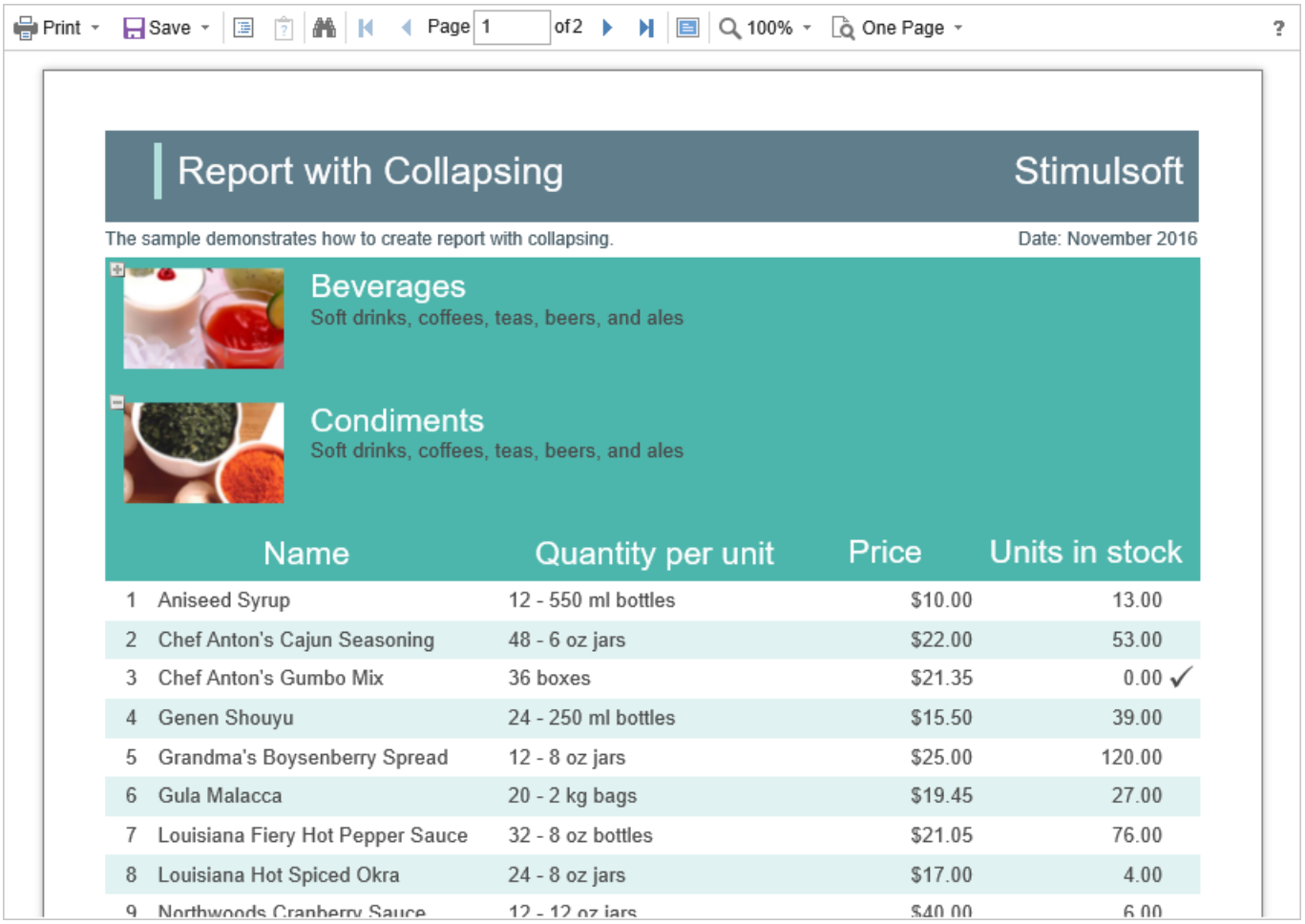
меняется на противоположное.

Допускается многоуровневая сортировка в отчете. Для этого необходимо удерживать клавишу **Ctrl** и последовательно нажимать на сортируемые компоненты отчета. Для сброса сортировки можно нажать на любой сортируемый компонент без удержания клавиши **Ctrl**.



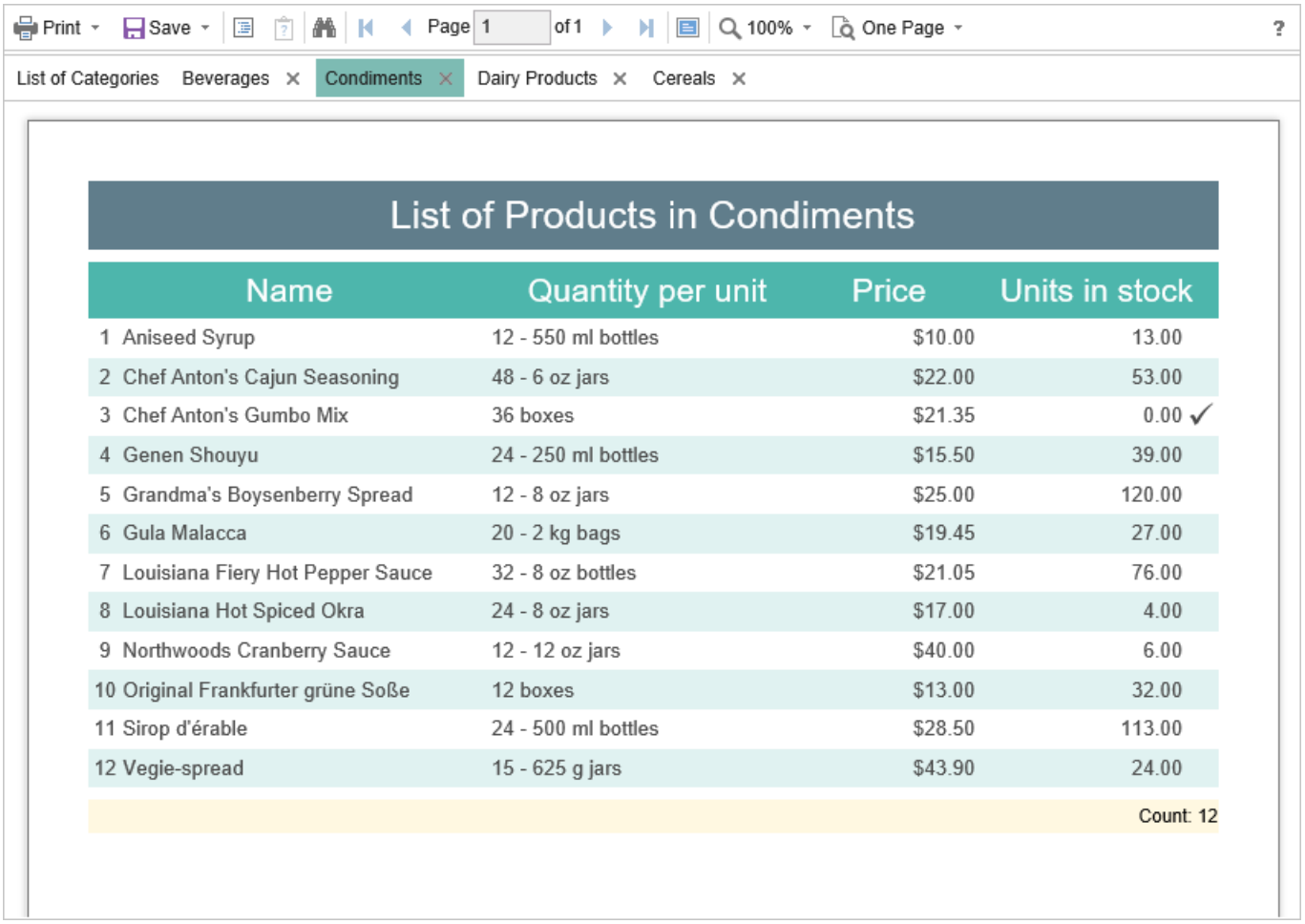
Отчет с динамическим сворачиванием представляет собой интерактивный отчет, в котором сворачивающиеся блоки могут сворачивать/разворачивать свое содержимое при нажатии на заголовок блока. Элементы отчета, которые можно свернуть/развернуть, обозначены специальной иконкой со знаком **[-]** или **[+]**.

© 2003-2026 Stimulsoft



При детализации данных под главной панелью вьювера будет отображена панель детализации с закладками детализированных отчетов. Отображенный в данный момент отчет будет подсвечен.

© 2003-2026 Stimulsoft



Для работы динамической сортировки, сворачивания и детализации отчетов не требуется дополнительных настроек вьювера. Для выполнения каких-либо действий перед сортировкой, сворачиванием или детализацией отчета предназначено специальное действие **Interaction**, которое будет вызвано при любом интерактивном действии вьювера.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Actions =
   {
      Interaction = "ViewerInteraction"
   }
})
...
```

© 2003-2026 Stimulsoft

HomeController.cs

```
...
public IActionResult ViewerInteraction()
{
   // Some code before any interaction
   // ...

   return StiNetCoreViewer.InteractionResult(this);
}
...
```

Для получения типа действия можно воспользоваться параметрами вьювера. Параметры вьювера представлены в виде объекта класса **StiRequestParams**, они передаются при любом запросе на сторону сервера, и содержат все необходимые сведения и состояния клиентской части вьювера. Для каждого вида интерактивности вьювера предусмотрен определенный тип действия:

- **Sorting** – при использовании сортировки колонок;
- **DrillDown** – при использовании детализации отчета;
- **Collapsing** – при использовании сворачивания блоков отчета.

HomeController.cs

```
...
public IActionResult ViewerInteraction()
{
   StiRequestParams requestParams =
   StiNetCoreViewer.GetRequestParams(this);
   switch (requestParams.Action)
   {
      case StiAction.Sorting:
      break;

      case StiAction.DrillDown:
      break;

      case StiAction.Collapsing:
      break;
   }

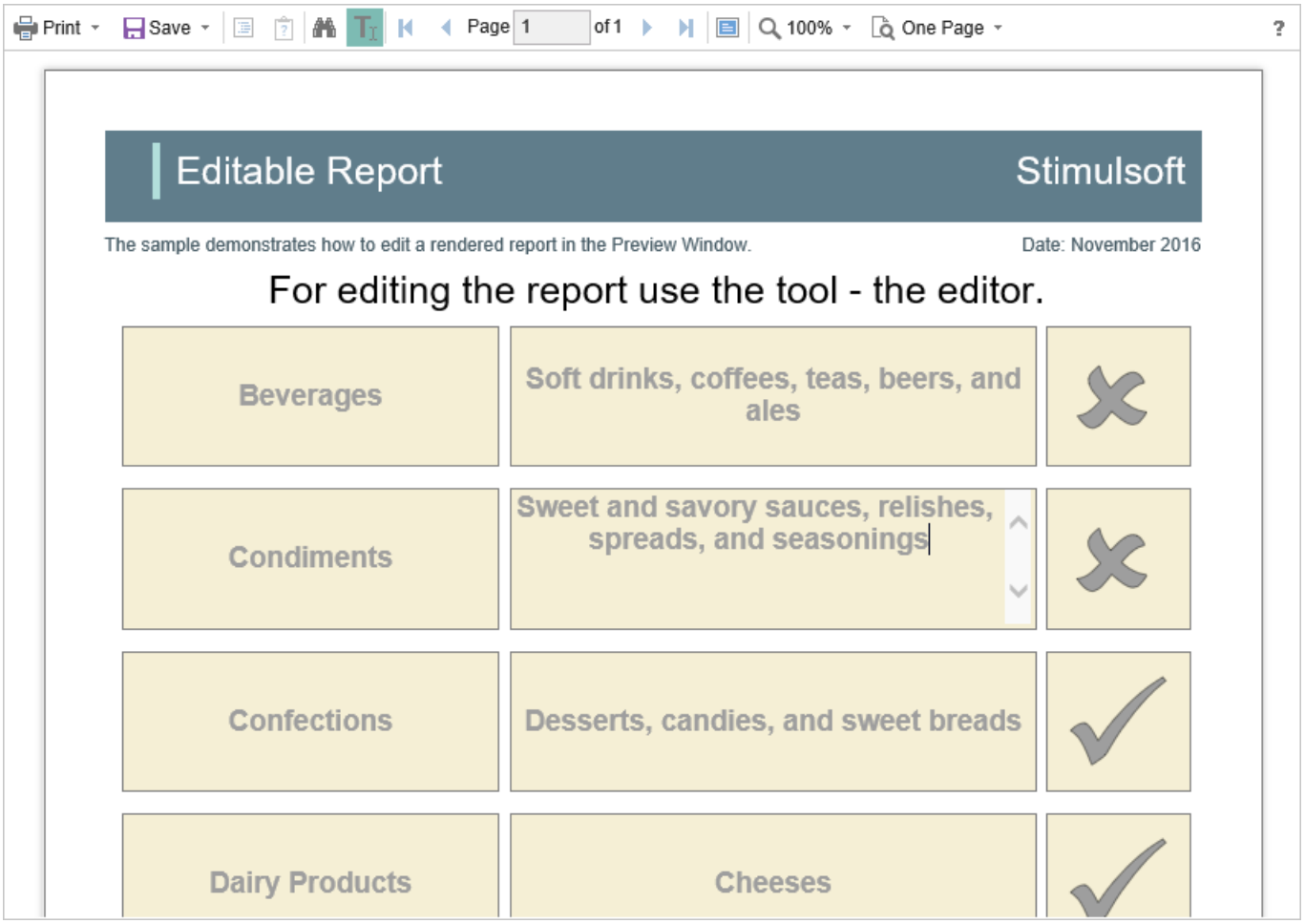
   return StiNetCoreViewer.InteractionResult(this);
}
...
```

5.1.14 Редактирование отчета

Компонент **HTML5 Viewer** имеет возможность редактировать элементы построенного отчета, такие как текстовые поля и чекбоксы. Для того, чтобы редактирование стало возможным, в самом шаблоне отчета необходимо

© 2003-2026 Stimulsoft

пометить необходимые компоненты как редактируемые. После отображения отчета во вьювере, для начала редактирования необходимо нажать на соответствующую кнопку на панели вьювера. После завершения редактирования необходимо нажать указанную кнопку еще раз, и все внесенные изменения будут примерены к отчету.



Для работы режима редактирования отчета не требуется никаких специальных настроек вьювера.

© 2003-2026 Stimulsoft

Информация

Отредактированные значения будут применены при печати либо экспортировании отчета, при этом исходный отчет останется не тронутым. После перезагрузки вьювера все значения будут возвращены к исходным.

5.1.15 Отправка отчета по Email

Информация

Обратите внимание, что возможность Отправить отчет по Email доступна только для отчетов, и недоступна для дашбордов.

В компоненте **HTML5 Viewer** предусмотрена возможность отправки отчета по Email. Для активации данной возможности необходимо установить свойство вьювера **ShowSendEmailButton** в значение **true** и определить действие **EmailReport**.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Actions =
   {
      EmailReport = "EmailReport"
   },
   Toolbar =
   {
      ShowSendEmailButton = true
   }
})
...
```

HomeController.cs

```
...
public IActionResult EmailReport()
{
   StiEmailOptions options = StiNetCoreViewer.GetEmailOptions(this);

   // Passed from the viewer, can be checked and changed
   // options.AddressTo = "";
   // options.Subject = "";
```

© 2003-2026 Stimulsoft

```
    // options.Body = "";

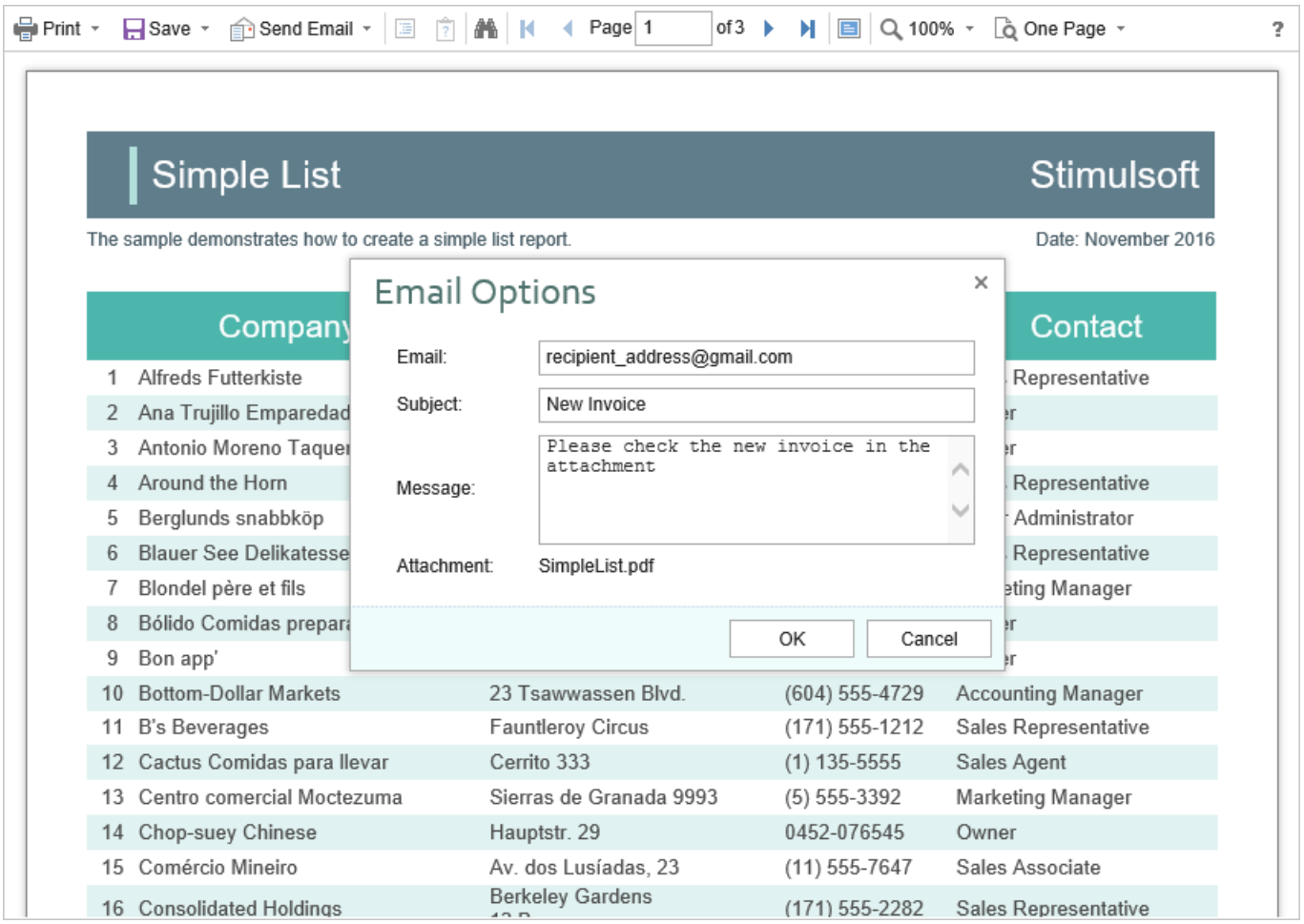
    // Should be filled here
    options.AddressFrom = "admin_address@test.com";
    options.Host = "smtp.test.com";
    options.Port = 465;
    options.UserName = "admin_address@test.com";
    options.Password = "admin_password";

    // options.CC.Add("email@test.com");
    // options.BCC.Add("email@test.com");
    // options.EnableSsl = true;

    return StiNetCoreViewer.EmailReportResult(this, options);
}
...
```

При отправке отчета по Email отображается меню выбора формата вложения, которое соответствует меню выбора формата экспортирования отчета. После выбора формата отобразиться диалог ввода параметров отправки, таких как Email получателя, тема и текст письма.

© 2003-2026 Stimulsoft



После подтверждения отправки будет вызвано описанное выше действие **EmailReport**, в котором можно проверить и скорректировать данные, введенные в этой форме. Экспортированный файл отчета будет прикреплен к письму автоматически.

Компонент **HTML5 Viewer** позволяет установить значения по умолчанию для формы отправки Email. Для этого предназначены свойства **DefaultEmailAddress**, **DefaultEmailSubject** и **DefaultEmailMessage**. По умолчанию данные свойства являются пустыми.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Email =
   {
      DefaultEmailAddress = "recipient_address@gmail.com",
      DefaultEmailSubject = "New Invoice",
      DefaultEmailMessage = "Please check the new invoice in the
      attachment"
   }
})
...
```

5.1.16 Вызов дизайнера из вьювера

Компонент **HTML5 Viewer** имеет возможность вызвать дизайнер отчетов. Для этого предназначена специальная кнопка **Дизайн (Design)** на панели инструментов вьювера (по умолчанию данная кнопка отключена). Для использования этой возможности необходимо установить свойство **ShowDesignButton** в значение **true**, а также определить действие **DesignReport**.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Actions =
   {
      DesignReport = "DesignReport"
   },
   Toolbar =
   {
      ShowDesignButton = true
   }
})
...
```

HomeController.cs

```
...
public IActionResult DesignReport()
{
   StiReport report = StiNetCoreViewer.GetReportObject(this);
   ViewBag.ReportName = report.ReportName;

   return View("Designer");
}
...
```

Информация

© 2003-2026 Stimulsoft

Сам вьювер не запускает дизайнер, он лишь вызывает указанное действие, в котором можно получить все необходимые параметры. Далее в действии можно выполнить перенаправление на другой View, который содержит дизайнер отчетов.

5.1.17 Кэширование

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Компонент **HTML5 Viewer** позволяет использовать кэш сервера для хранения построенного отчета. Если кэширование не используется, то при каждом запросе страницы необходимо загружать отчет, подключать данные и строить его заново. Если использовать кэширование, то при следующем обновлении страницы, ранее построенный отчет будет загружен из кэша.

При использовании кэширования стоит учитывать, что каждый сохраненный отчет в кэше занимает память сервера и при большом количестве запросов к отчетам это может стать критичным фактором. Поэтому необходимо выбирать из двух вариантов – или маленькие потребности к памяти, но большие к производительности, или небольшие потребности к производительности, но большие к памяти.

Для использования кэширования необходимо подключить модули для работы с сессией либо кэшем на стороне сервера. Для этого достаточно добавить в проект указанные ниже сервисы в стартовом файле проекта.

© 2003-2026 Stimulsoft

Startup.cs

```
...
public void ConfigureServices(IServiceCollection services)
{
   services.AddMemoryCache();
   services.AddSession();
   services.AddMvc();
}
...
```

Управлять кэшированием можно при помощи следующих приведенных ниже свойств.

Свойство CacheMode

Данное свойство вьювера включает кэширование и устанавливает его тип. Может принимать одно из значений, указанных в перечислении **StiServerCacheMode**:

- **None** – кэширование отключено, каждое действие вьювера требует загрузки отчета и, если это шаблон отчета, последующее его построение;
- **ObjectCache** – для кэширования используется кэш сервера, в котором сохраняется объект отчета (значение по умолчанию);
- **StringCache** - для кэширования используется кэш сервера, в котором сохраняется отчет в виде упакованной строки;
- **ObjectSession** - для кэширования используется текущая сессия, в которой сохраняется объект отчета;
- **StringSession** - для кэширования используется текущая сессия, в которой сохраняется отчет в виде упакованной строки.

Свойство CacheItemPriority

Данное свойство устанавливает приоритет хранимого в кэше сервера отчета, влияет на автоматическую очистку памяти сервера в случае ее нехватки. Чем меньше приоритет, тем больше шанс удаления информации из памяти.

Свойство CacheTimeout

© 2003-2026 Stimulsoft

Данное свойство указывает величину времени в минутах, в течение которого необходимо сохранять отчет в кэше сервера. Если при использовании кэширования требуемый отчет не найден в кэше (т.е. истекло время хранения объекта), то он будет запрошен заново при помощи специального действия **GetReport** с последующим подключением данных отчета, и их построением.

Компонент **HTML5 Viewer** предоставляет возможность определить собственные методы работы с кэшированием отчета. Для этого предназначен специальный класс **StiCacheHelper**, который содержит методы получения отчета из кэша и сохранения отчета в кэш. Необходимо создать новый класс, унаследованный от **StiCacheHelper**, и перегрузить указанные выше методы, которые соответственно имеют названия **GetReport**, **SaveReport** и **RemoveReport**.

© 2003-2026 Stimulsoft

HomeController.cs

```
...
public class ViewerController : Controller
{
   public class StiMyCacheHelper : StiCacheHelper
   {
      public override StiReport GetReport(string guid)
      {
         string path =
         System.IO.Path.Combine(this.HttpContext.Server.MapPath("CacheFiles"
         ), guid);
         if (System.IO.File.Exists(path))
         {
            StiReport report = new StiReport();
            string packedReport = System.IO.File.ReadAllText(path);
            if (guid.EndsWith("template"))
            report.LoadPackedReportFromString(packedReport);
            else report.LoadPackedDocumentFromString(packedReport);

            return report;
         }
         return null;

         //return base.GetReport(guid);
      }

      public override void SaveReport(StiReport report, string guid)
      {
         string packedReport = guid.EndsWith("template") ?
         report.SavePackedReportToString() :
         report.SavePackedDocumentToString();
         string path =
         System.IO.Path.Combine(this.HttpContext.Server.MapPath("CacheFiles"
         ), guid);
         System.IO.File.WriteAllText(path, packedReport);

         //base.SaveReport(report, guid);
      }

      public override void RemoveReport(string guid)
      {
         var path = Path.Combine(HttpContext.Server.MapPath("CacheFiles"),
         guid);
         if (File.Exists(path))
            File.Delete(path);
      }
   }

   static ViewerController()
   {
      StiNetCoreViewer.CacheHelper = new StiMyCacheHelper();
   }
}
...
```

© 2003-2026 Stimulsoft

Для инициализации работы с кэшированием отчета при помощи созданного класса, достаточно задать его в качестве значения статического свойства **StiNetCoreViewer.CacheHelper** в конструкторе контроллера.

Информация

При отключенном кэшировании отчета (свойство вьювера **CacheMode** имеет значение **None**), указанный класс задействован не будет.

5.1.18 Вспомогательные методы

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для **HTML5 Viewer** предусмотрено несколько вспомогательных методов, которые предназначены для получения объекта просматриваемого в данный момент отчета, параметров текущего состояния вьювера и других полезных данных. Эти методы можно использовать в любых действиях вьювера.

Метод GetReportObject()

© 2003-2026 Stimulsoft

Возвращает объект отчета, с которым в данный момент работает вьювер. Допускается производить с ним необходимые манипуляции - регистрировать новые наборы данных, изменять свойства отчета, присваивать параметры либо загружать в объект другой отчет. Затем отчет можно вернуть вьюверу, указав его в качестве параметра в результирующем методе действия.

HomeController.cs

```
...
public IActionResult ViewerInteraction()
{
   StiReport report = StiNetCoreViewer.GetReportObject(this);
   report.ReportName = "MyReportName";

   return StiNetCoreViewer.InteractionResult(this, report);
}
...
```

Метод GetRouteValues()

Возвращает значения маршрутов для URL, с которыми была открыта страница вьювера. Таким образом, предоставляется возможность получить исходную коллекцию параметров страницы запуска в любом действии вьювера и использовать эти значения для каких-либо проверок и условий.

HomeController.cs

```
...
public IActionResult ViewerInteraction()
{
   RouteValueDictionary routeValues =
   StiNetCoreViewer.GetRouteValues(this);

   return StiNetCoreViewer.InteractionResult(this);
}
...
```

Также можно получить значения параметров URL по имени параметра, указав его в качестве параметра вызываемого действия вьювера.

© 2003-2026 Stimulsoft

HomeController.cs

```
...
public IActionResult ViewerInteraction(string id)
{
   return StiNetCoreViewer.InteractionResult(this);
}
...
```

Метод GetFormValues()

Возвращает значения формы, которая инициировала (открыла POST-запросом) страницу вьювера. Таким образом предоставляется возможность получить коллекцию параметров формы в любом действии вьювера.

HomeController.cs

```
...
public IActionResult ViewerInteraction()
{
   NameValueCollection formValues = StiNetCoreViewer.GetFormValues(this);

   return StiNetCoreViewer.InteractionResult(this);
}
...
```

По умолчанию данная возможность отключена в целях оптимизации запросов клиентской стороны вьювера на сервер. Для включения достаточно установить свойство PassFormValues в значение true.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Server =
   {
      PassFormValues = true
   }
})
...
```

Метод GetRequestParams()

Возвращает все параметры текущего состояния вьювера, переданные на

© 2003-2026 Stimulsoft

сторону сервера. Они могут быть полезны для определения типа действия, которое в данный момент выполняет вьювер – например, для определения типа экспорта, а также всех параметров действия.

HomeController.cs

```
...
public IActionResult ExportReport()
{
   StiRequestParams requestParams =
   StiNetCoreViewer.GetRequestParams(this);
   if (requestParams.ExportFormat == StiExportFormat.Pdf)
   {
      StiReport report = StiNetCoreViewer.GetReportObject(this);

      // Some action with report for the PDF export
      // ...

      return StiNetCoreViewer.ExportReportResult(this, report);
   }

   return StiNetCoreViewer.ExportReportResult(this);
}
...
```

Допускается изменять значения некоторых параметров. После внесения изменений, для корректной работы вьювера необходимо передать измененный объект параметров на вход результирующего метода.

HomeController.cs

```
...
public IActionResult ViewerInteraction()
{
   StiRequestParams requestParams =
   StiNetCoreViewer.GetRequestParams(this);
   if (requestParams.Action == StiAction.Variables)
   {
      requestParams.Interaction.Variables["Variable1"] = "MyValue";
      return StiNetCoreViewer.InteractionResult(this, requestParams);
   }

   return StiNetCoreViewer.InteractionResult(this);
}
...
```

Метод GetExportSettings()

© 2003-2026 Stimulsoft

Возвращает все параметры текущего экспортирования отчета. Тип объекта параметров будет соответствовать типу экспорта, выбранному в меню вьювера. Любые параметры экспорта можно изменить и передать на вход результирующего метода. В этом случае, экспортирование отчета будет произведено с переданными параметрами.

HomeController.cs

```
...
public IActionResult ExportReport()
{
   StiExportSettings settings = StiNetCoreViewer.GetExportSettings(this);
   if (settings.GetExportFormat() == StiExportFormat.Pdf)
   {
      StiPdfExportSettings pdfSettings = (StiPdfExportSettings)settings;
      pdfSettings.EmbeddedFonts = true;
      pdfSettings.AllowEditable = StiPdfAllowEditable.No;
      return StiNetCoreViewer.ExportReportResult(this, settings);
   }

   return StiNetCoreViewer.ExportReportResult(this);
}
...
```

Методы MapPath() и MapWebRootPath()

Возвращает абсолютный путь соответственно к директории приложения или к директории wwwroot. Можно использовать для загрузки файлов шаблонов отчетов, файлов данных и др. Данные методы расположены в статическом классе **StiNetCoreHelper**.

HomeController.cs

```
...
public IActionResult GetReport()
{
   StiReport report = new StiReport();
   report.Load(StiNetCoreHelper.MapPath(this, "Reports/SimpleList.mrt"));

   return StiNetCoreViewer.GetReportResult(this, report);
}
...
```

© 2003-2026 Stimulsoft

5.1.19 Экспорт и печать из кода

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет». Кроме этого, следует понимать, что возможность печати из кода доступна только для отчетов, и недоступна для дашбордов.

Компонент **HTML5 Viewer** предоставляет возможность печатать отчет различными способами и выполнять экспортирование отчета в различные форматы. Эти действия выполняются при помощи меню вьювера. Если требуется напечатать отчет или выполнить экспорт при помощи кода отчета, например, в определенном действии контроллера, то можно воспользоваться специальным классом **StiNetCoreReportResponse**. Данный класс содержит набор статических методов, позволяющих выполнить печать отчета либо экспорт отчета из кода, при этом вьювер отчетов не требуется.

Index.cshtml

```
...
@Html.ActionLink("Print Report from Code", "PrintReport")
<br />
@Html.ActionLink("Export Report from Code", "ExportReport")
...
```

© 2003-2026 Stimulsoft

HomeController.cs

```
...
private StiReport LoadSimpleList()
{
   DataSet dataSet = new DataSet();
   dataSet.ReadXml(Server.MapPath("Reports/Demo.xml"));

   StiReport report = new StiReport();
   report.Load(Server.MapPath("Reports/SimpleList.mrt"));
   report.RegData(dataSet);

   return report;
}

public IActionResult PrintReport()
{
   StiReport report = LoadSimpleList();

   return StiNetCoreReportResponse.PrintAsPdf(report);
   //return StiNetCoreReportResponse.PrintAsHtml(report);
}

public IActionResult ExportReport()
{
   StiReport report = LoadSimpleList();

   return StiNetCoreReportResponse.ResponseAsPdf(report);
   //return StiNetCoreReportResponse.ResponseAsExcel2007(report);
   //return StiNetCoreReportResponse.ResponseAsText(report);
   //StiNetCoreReportResponse.ResponseAsJson(report);
}
...
```

Класс **StiNetCoreReportResponse** содержит методы печати отчета в формате PDF и HTML, а также методы экспортирования отчета в любой из поддерживаемых форматов. В качестве аргументов методы могут принимать различные настройки экспортов, режимы отображения и варианты сохранения полученных файлов.

5.1.20 Время ожидания

При работе с компонентом **StiNetCoreViewer** можно установить время ожидания (timeout) выполнения различных операций - хранение отчета в кэше, ответ сервера, выполнение запроса. Настройка времени ожидания выполняется при помощи свойств компонента и опций отчета.

Свойство CacheTimeout

© 2003-2026 Stimulsoft

Предоставляет возможность установить время в минутах, которое сервер будет хранить построенный отчет в кэше с момента последнего действия вьювера. По умолчанию установлено значение 10 минут.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Server =
   {
      CacheTimeout = 10
   }
})
...
```

Использование кэша увеличивает скорость работы вьювера с отчётом. Более подробно можно ознакомиться в главе Кэширование.

Свойство RequestTimeout

Предоставляет возможность установить время ожидания ответа от сервера в секундах, по истечении которого будет выдана ошибка связи с сервером. По умолчанию установлено значение 30 секунд. Для больших отчетов рекомендуется увеличить данное значение.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Server =
   {
      RequestTimeout = 30
   }
})
...
```

Опция CommandTimeout

Предоставляет возможность установить время ожидания запроса в секундах, при использовании SQL источников данных в отчете. Значение данного свойства сохраняется в самом шаблоне отчёта для каждого SQL подключения отдельно.

Ниже приведен пример кода, который предоставляет возможность установить

© 2003-2026 Stimulsoft

время ожидания запроса для уже созданного MS SQL соединения и источников данных в отчете.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Actions =
   {
      GetReport = "GetReport",
      ViewerEvent = "ViewerEvent"
   }
   })
...
```

HomeController.cs

```
...
public IActionResult GetReport()
{
   StiReport report = new StiReport();
   report.Load(Server.MapPath("Report.mrt"));
   ((StiSqlSource)
   report.Dictionary.DataSources["DataSourceName"]).CommandTimeout = 1000;

   return StiNetCoreViewer.GetReportResult(this, report);
}

public IActionResult ViewerEvent()
{
   return StiNetCoreViewer.ViewerEventResult(this);
}
...
```

5.1.21 Настройки вьювера

Настройка **HTML5 Viewer** выполняется при помощи свойств, находящихся в классе **StiNetCoreViewerOptions**. Все свойства разделены на группы, некоторые из групп содержат свои подгруппы для удобства использования. Ниже приведен пример установки свойств вьювера.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Theme = StiViewerTheme.Office2022WhiteTeal,
   Localization = "Localization/en.xml",
   Actions =
   {
      GetReport = "GetReport",
      ViewerEvent = "ViewerEvent"
   },
   Appearance =
   {
      ReportDisplayMode = StiReportDisplayMode.Auto,
      InterfaceType = StiInterfaceType.Auto,
      ScrollbarsMode = true,
      ShowTooltips = false
   },
   Exports =
   {
      DefaultSettings =
      {
         ExportToPdf =
         {
            CreatorString = "Company Name",
            ImageQuality = 0.75f
         }
      },
      ShowExportToDbf = false,
      ShowExportToDif = false
   }
})
...
```

Обратите внимание, что все элементы дашбордов имеют собственные кнопки экспорта и полноэкранного режима просмотра. Специальных опций для управления их отображением не предусмотрено, но их можно выключить через свойства элемента. Нижеприведенный код необходимо добавить после загрузки отчета, перед его передачей вьюверу.

HomeController.cs

```
...
var dbsElementInteraction = (report.GetComponentByName("RegionMap1") as
Stimulsoft.Report.Dashboard.IStiElementInteraction).DashboardInteraction;
(dbsElementInteraction as
Stimulsoft.Report.Dashboard.IStiInteractionLayout).ShowFullScreenButton =
false;
(dbsElementInteraction as
Stimulsoft.Report.Dashboard.IStiInteractionLayout).ShowSaveButton = false;
...
```

© 2003-2026 Stimulsoft

Главные настройки (без группы)

Наименование	Описание
Theme	Задает тему оформления вьювера. Список доступных тем оформления находится в перечислении **StiViewerTheme**. По умолчанию установлено значение **Office2022WhiteBlue**.
Localization	Задает путь к XML файлу локализации. Путь может быть абсолютным либо относительным. По умолчанию используется английская локализация, которая встроена во вьювер и не требует дополнительных XML файлов.
Width	Задает ширину компонента в необходимых единицах измерения, которые определены в классе **Unit**. Доступна установка значения в пикселях: **Unit.Pixel()**, поинтах: **Unit.Point()** и процентах: **Unit.Percentage()**. По умолчанию установлена ширина 100%.
Height	Задает высоту компонента в необходимых единицах измерения, которые определены в классе **Unit**. Доступна установка значения в пикселях: **Unit.Pixel()**, поинтах: **Unit.Point()** и процентах: **Unit.Percentage()**. По умолчанию установлена автоматическая высота в зависимости от размера страницы отчета, либо 650 пикселей в режиме отображения вьювера с полосами прокрутки.

Действия (Actions)

Наименование	Описание
GetReport	Задает название метода действия подготовки построенного отчета. Если включено

© 2003-2026 Stimulsoft

	кэширование отчета, то данное действие будет вызвано только один раз при первом запросе отчета, либо в случае, если требуемый отчет не найден в кэше сервера.
PrintReport	Задает название метода действия печати отчета.
ExportReport	Задает название метода действия экспорта отчета в необходимый формат.
EmailReport	Задает название метода действия отправки отчета по Email.
Interaction	Задает название метода действия работы вьювера с интерактивными операциями, такими как использование параметров, динамическая сортировка, сворачивание и детализация отчета.
DesignReport	Задает название метода действия для перехода к нужному представлению при нажатии кнопки Дизайн (Design) на панели вьювера.
ViewerEvent	Задает название метода действия основных событий вьювера и обработки действий вьювера, таких как печать и экспортирование отчета, работа с параметрами и интерактивность, если эти действия не заданы отдельно. Дополнительно данное действие используется для загрузки скриптов и стилей вьювера. Это действие является обязательным.

Работа с сервером (Server)

Наименование	Описание
Controller	Задает название контроллера обработки запросов вьювера отчетов. Если данное свойство не задано, то для обработки

© 2003-2026 Stimulsoft

	запросов будет использоваться текущий контроллер.
RouteTemplate	Задает шаблон (route template) вызова действий вьювера отчетов. Если свойство не установлено, будет использоваться шаблон, установленный в MVC проекте. Если свойство `UseRelativeUrls` установлено в true, для этого свойства не будет учитываться `BasePath`. По умолчанию это свойство не установлено.
AllowAutoUpdateCookies	Предоставляет возможность вьюверу автоматически обновлять файлы cookies при каждом запросе на сервер. По умолчанию cookies устанавливаются при создании вьювера, если они не указаны в отчете. По умолчанию свойство имеет значение **false**.
AllowAntiforgeryToken	Предоставляет возможность вьюверу автоматически запрашивать и отправлять токен защиты от подделки. По умолчанию свойство имеет значение **true**.
RequestTimeout	Устанавливает время ожидания ответа от сервера в секундах, по истечении которого будет выдана ошибка. По умолчанию установлено значение 30 секунд. Для больших отчетов рекомендуется увеличить данное значение.
CacheTimeout	Устанавливает время в минутах, которое сервер будет хранить построенный отчет с момента последнего действия вьювера. По умолчанию установлено значение 10 минут.
CacheMode	Устанавливает режим кэширования отчета. Может принимать одно из указанных ниже значений перечисления **StiServerCacheMode**: › **None** – кэширование отключено, каждое действие вьювера требует загрузки отчета и, если это шаблон отчета, последующее его

© 2003-2026 Stimulsoft

	построение; › **ObjectCache** – в качестве хранилища используется кэш, отчет хранится как объект (значение свойства по умолчанию); › **ObjectSession** – в качестве хранилища используется сессия, отчет хранится как объект; › **StringCache** – в качестве хранилища используется кэш сервера, отчет сериализуется в упакованную строку; › **StringSession** – в качестве хранилища используется сессия, отчет сериализуется в упакованную строку.
CacheItemPriority	Устанавливает приоритет хранимого в кэше сервера отчета. Это свойство влияет на автоматическую очистку памяти сервера в случае ее нехватки. Чем меньше приоритет, тем больше шанс удаления информации из памяти.
AllowAutoUpdateCache	Устанавливает режим автоматического обновления кэша. Отчет, хранящийся в кэше либо сессии сервера, будет автоматически пересохранен через определенный промежуток времени при бездействии вьювера (примерно каждые 3 минуты). По умолчанию свойство имеет значение **true**.
UseRelativeUrls	Устанавливает режим вьювера, в котором для AJAX запросов на сервер используются относительные ссылки. По умолчанию свойство имеет значение **true**.
PortNumber	Получает или задает значение, которое указывает номер порта для использования в URL. Значение 0 определяет автоматическое обнаружение (установлено по умолчанию). Значение -1 удаляет номер порта.
PassQueryParametersForResources	Включает передачу всех параметров URL запроса при формировании ссылок на

© 2003-2026 Stimulsoft

	ресурсы вьювера. При значении **false** для запроса ресурсов вьювера используются только необходимые параметры, что способствует более корректной работе кэша браузера. По умолчанию свойство имеет значение **true**.
PassQueryParametersToReport	Включает использование всех параметров URL запроса в качестве значений одноименных переменных в отчете. По умолчанию свойство имеет значение **false**.
PassFormValues	Включает передачу значений POST-формы на сторону клиента, если эти значения требуется использовать в действиях вьювера. При включении данной возможности, вспомогательный метод **GetFormValues()** будет возвращать коллекцию параметров формы. По умолчанию свойство имеет значение **false**.
ShowServerErrorPage	Включает отображение HTML-страницы с детализацией ошибки, возникшей на стороне сервера. При включенном свойстве детализация ошибки будет отображена в окне вьювера. Если свойство выключено, будет отображен только числовой код ошибки и краткий текст ошибки в диалоговом окне. По умолчанию свойство имеет значение **true**.
UseCompression	Включает сжатие запросов вьювера в GZip поток. Это позволяет уменьшить объем интернет-трафика, но немного замедляет работу вьювера. По умолчанию свойство имеет значение **false**.
UseCacheForResources	Включает кэширование на стороне сервера ресурсов компонента, таких как скрипты, стили, изображения. Позволяет ускорить загрузку компонента и уменьшить нагрузку на сервер при большом количестве клиентов.

© 2003-2026 Stimulsoft

	По умолчанию свойство имеет значение **true**.
UseLocalizedCache	Предоставляет возможность использовать различные кэши в зависимости от выбранной локализации. По умолчанию свойство имеет значение **false**.
AllowLoadingCustomFontsToClientSide	Позволяет передавать пользовательские шрифты на клиентскую сторону и преобразовывать их в CSS стиль для корректного отображения текста на стороне HTML с заданным шрифтом. По умолчанию свойство имеет значение **false**.

Внешний вид (Appearance)

Наименование	Описание
IconSet	Предоставляет возможность установить набор иконок: › StiWebUIIconSet.Auto (значение по умолчанию) - устанавливает набор иконок автоматически. Для тем **Office2022** - используется набор иконок в стиле **Monoline**, для тем **Office2013** - используется набор иконок в стиле **Regular**. › StiWebUIIconSet.Monoline - устанавливает набор иконок в стиле **Monoline**; › StiWebUIIconSet.Regular - устанавливает набор иконок в стиле **Regular**.
CustomCss	Задает путь к CSS файлу стилей вьювера. Если данное свойство установлено, то стандартные стили выбранной темы загружены не будут. По умолчанию установлено пустое значение.
BackgroundColor	Задает цвет подложки вьювера. По умолчанию свойство имеет значение **White**.
PageBorderColor	Задает цвет границ страницы отчета. По умолчанию свойство имеет значение **Gray**.

© 2003-2026 Stimulsoft

RightToLeft	Устанавливает режим отображения **Right to Left** для элементов управления вьювера. По умолчанию свойство имеет значение **false**.
FullScreenMode	Устанавливает полноэкранный режим отображения вьювера. По умолчанию свойство имеет значение **false**.
ScrollbarsMode	Устанавливает режим отображения отчета с полосами прокрутки. По умолчанию свойство имеет значение **false**.
OpenLinksWindow	Задает целевое окно для открытия ссылок, содержащихся в отчете. По умолчанию свойство имеет значение **Blank** (новое окно).
OpenExportedReportWindow	Задает целевое окно для открытия файла экспорта из вьювера. По умолчанию свойство имеет значение **Blank** (новое окно).
DesignWindow	Задает целевое окно для открытия дизайнера отчетов. По умолчанию свойство имеет значение **Self** (текущее окно).
ShowTooltips	Включает отображение подсказок для элементов управления вьювера при наведении мыши. По умолчанию свойство имеет значение **true**.
ShowTooltipsHelp	Включает отображение ссылки на онлайн документацию в подсказках для элементов управления вьювера. По умолчанию свойство имеет значение **true**.
ShowDialogsHelp	Предоставляет возможность отображать или не отображать кнопку вызова справки в различных меню. По умолчанию, установлено значение **true**.
PageAlignment	Задает позицию страницы отчета в окне вьювера. Может принимать одно из указанных ниже значений перечисления **StiContentAlignment**: › **Left** – страница будет пристыкована к

© 2003-2026 Stimulsoft

	левому краю; › **Center** – страница будет центрирована (значение по умолчанию); › **Right** – страница будет пристыкована к правому краю.
ShowPageShadow	Включает отображение тени для страниц отчета. По умолчанию свойство имеет значение **true**.
BookmarksPrint	Включает печать закладок отчета на принтере (помимо самого отчета). По умолчанию свойство имеет значение **false**.
BookmarksTreeWidth	Устанавливает ширину панели закладок в пикселях. По умолчанию ширина имеет значение 180 пикселей.
ParametersPanelPosition	Задает позицию панели параметров отчета. Может принимать одно из указанных ниже значений перечисления **StiParametersPanelPosition**: › **Top** – панель будет пристыкована к верхнему краю (значение по умолчанию); › **Left** – панель будет пристыкована к левому краю.
ParametersPanelMaxHeight	Устанавливает максимальную высоту панели параметров в пикселях. По умолчанию максимальная высота имеет значение 300 пикселей.
ParametersPanelColumnsCount	Устанавливает количество колонок для отображения параметров отчета. По умолчанию разбиение производится на 2 колонки.
ParametersPanelSortDataItems	Устанавливает или выключает режим сортировки значений переменной. По умолчанию, опция установлена в значение **true**, т.е. значения переменной сортируются.
ParametersPanelDateFormat	Устанавливает формат даты и времени для

© 2003-2026 Stimulsoft

	переменных соответствующего типа на панели параметров. По умолчанию используется формат даты и времени, установленный браузером.
InterfaceType	Устанавливает тип используемого интерфейса вьювера. Может принимать одно из указанных ниже значений перечисления **StiInterfaceType**: › **Auto** – тип интерфейса вьювера будет выбран автоматически в зависимости от используемого устройства (значение по умолчанию); › **Mouse** – принудительное использование стандартного интерфейса для управления вьювером при помощи мыши; › **Touch** – принудительное использование Touch интерфейса для управления вьювером при помощи сенсорного экрана монитора, в этом режиме элементы интерфейса вьювера имеют увеличенные размеры для удобства управления; › **Mobile** – принудительное использование Mobile интерфейса для управления вьювером при помощи экрана смартфона, в этом режиме интерфейс вьювера имеет упрощенный вид и адаптирован для управления при помощи мобильного устройства.
AllowMobileMode	Включает или выключает возможность отображения отчета или дашборда в мобильном режиме. Если опция установлена в значение **false**, то мобильный режим просмотра не будет использоваться ни при каких обстоятельства. Если опция установлена в значение **true**, то мобильный режим просмотра будет использоваться запуске вьювера на мобильных устройствах.

© 2003-2026 Stimulsoft

	По умолчанию, опция установлена в значение **true**.
ChartRenderType	Устанавливает режим отображения диаграмм на странице отчета. Может принимать одно из указанных ниже значений перечисления **StiChartRenderType**: › **Image** – диаграммы отображаются в виде статичных изображений; › **Vector** – диаграммы отображаются в векторном режиме в виде SVG объекта; › **AnimatedVector** - диаграммы отображаются в векторном режиме в виде SVG объекта, элементы диаграммы появляются с анимацией (значение по умолчанию).
ReportDisplayMode	Устанавливает режим экспорта для отображения страниц отчета. Может принимать одно из указанных ниже значений перечисления **StiReportDisplayMode**: › **FromReport** - режим экспорта элементов отчета определяется из настроек шаблона - Div или Table; › **Table** – элементы отчета экспортируются с использованием HTML таблиц (значение по умолчанию); › **Div** – элементы отчета экспортируются с использованием DIV разметки; › **Span** - элементы отчета экспортируются с использованием SPAN разметки.
DatePickerFirstDayOfWeek	Устанавливает первый день недели для элемента выбора даты. Может принимать одно из указанных ниже значений перечисления **StiFirstDayOfWeek**: › **Auto** – автоматическое определение первого дня недели из настроек браузера

© 2003-2026 Stimulsoft

	(значение по умолчанию); › **Monday** – первым днем недели является понедельник; › **Sunday** – первым днем недели является воскресенье.
DatePickerIncludeCurrentDayForRanges	Предоставляет возможность включать или не включать текущий день в диапазон значений элемента **Выбор даты (Date Picker)**. По умолчанию, опция установлена в значение **false**, т.е. текущий день не включается в диапазон значений элемента.
AllowTouchZoom	Включает возможность изменения масштаба страницы отчета специальным двух пальцевым жестом (Pinch to Zoom) на сенсорном экране устройства. По умолчанию свойство имеет значение **true**.
ShowReportIsNotSpecifiedMessage	Предоставляет возможность отображать или не отображать уведомление, если отчет не задан. По умолчанию свойство имеет значение **true**.
PrintToPdfMode	Устанавливает режим печати в PDF документ. Может принимать значения: › **StiPrintToPdfMode.Hidden** - скрытый режим печати (по умолчанию); › **StiPrintToPdfMode.Popup** - документ PDF будет отображет перед печатью во всплывающем окне.
ImagesQuality	Получает или задает качество изображения, которое будет использоваться на странице вьювера. Может принимать значения: › **StiImagesQuality.Low** - низкое качество, используется для ускорения загрузки отчета и экономии памяти › **StiImagesQuality.Normal** - обычное качество, подходит для большинства ситуаций, установлено по умолчанию › **StiImagesQuality.High** - высокое качество,

© 2003-2026 Stimulsoft

	используется для мониторов высокой четкости, но может замедлять загрузку страниц
CombineReportPages	Предоставляет возможность объединить обработанные страницы шаблона отчета в один шаблон или представить каждую страницу шаблона отдельной вкладкой во вьювере. По умолчанию, опция установлена в значение **false**, т.е. каждая страница шаблона отчета будет представлена отдельной вкладкой во вьювере.
AllowPropagationEvents	Предоставляет возможность разрешить распространение событий нажатия клавиш, когда вьювер отчетов не в фокусе. По умолчанию свойство имеет значение **true**.
DashboardFilterElementItemHeight	Предоставляет возможность изменить высоту флажка в элементе дашборда **Список (List Box)**.

Панель инструментов (Toolbar)

Наименование	Описание
Visible	Включает отображение панели инструментов вьювера. По умолчанию свойство имеет значение **true**.
DisplayMode	Задает режим отображения панели инструментов вьювера. Может принимать одно из указанных ниже значений перечисления **StiToolbarDisplayMode**: › **Simple** – простой режим отображения, все элементы управления расположены на одной панели управления (значение по умолчанию); › **Separated** – раздельный режим отображения, панель управления разделена

© 2003-2026 Stimulsoft

	на верхнюю и нижнюю.
BackgroundColor	Задает цвет панели инструментов вьювера. По умолчанию используется цветовое оформление выбранной темы.
BorderColor	Задает цвет границ панели инструментов вьювера. По умолчанию используется цветовое оформление выбранной темы.
FontColor	Задает цвет текста для панели инструментов и меню вьювера. По умолчанию используется цветовое оформление выбранной темы.
FontFamily	Задает шрифт для панели инструментов и меню вьювера. По умолчанию используется шрифт выбранной темы.
Alignment	Устанавливает режим выравнивания элементов управления на панели инструментов вьювера. Может принимать одно из указанных ниже значений перечисления **StiContentAlignment**: › **Left** – элементы будут пристыкованы к левому краю; › **Center** – элементы будут располагаться по центру панели; › **Right** – элементы будут пристыкованы к правому краю; › **Default** – выравнивание зависит от свойства RightToLeft (значение по умолчанию)
ShowButtonCaptions	Включает отображение текста кнопок панели инструментов вьювера. По умолчанию свойство имеет значение **true**.
ShowPrintButton	Включает отображение кнопки **Печать (Print)** на панели инструментов вьювера. По умолчанию свойство имеет значение **true**.
ShowOpenButton	Включает отображение кнопки **Открыть**

© 2003-2026 Stimulsoft

	(Open) на панели инструментов вьювера при просмотре отчетов или панели индикаторов (дашбордов). По умолчанию свойство имеет значение **true**.
ShowSaveButton	Включает отображение кнопки **Сохранить (Save)** на панели инструментов вьювера при просмотре отчетов или панели индикаторов (дашбордов). По умолчанию свойство имеет значение **true**.
ShowSendEmailButton	Включает отображение кнопки **Отправиль Email (Send Email)** на панели инструментов вьювера. По умолчанию свойство имеет значение **false**. Также, необходимо определить действие EmailReport.
ShowFindButton	Включает отображение кнопки **Поиск (Find)** на панели инструментов вьювера. По умолчанию свойство имеет значение **true**.
ShowBookmarksButton	Включает отображение кнопки **Закладки (Bookmarks)** на панели инструментов вьювера. По умолчанию свойство имеет значение **true**. Если указанная кнопка скрыта, то панель закладок не будет отображена даже при наличии закладок в отчете.
ShowParametersButton	Включает отображение кнопки **Параметры (Parameters)** на панели инструментов вьювера. По умолчанию свойство имеет значение **true**. Если указанная кнопка скрыта, то панель параметров не будет отображена даже при наличии параметров в отчете.
ShowResourcesButton	Включает отображение кнопки **Ресурсы (Resources)** на панели инструментов вьювера. По умолчанию свойство имеет значение **true**. Если указанная кнопка скрыта, то панель ресурсов не будет отображена.
ShowEditorButton	Включает отображение кнопки **Редактор (Editor)** на панели инструментов вьювера при просмотре отчетов. По умолчанию

© 2003-2026 Stimulsoft

	свойство имеет значение **true**.
ShowFullScreenButton	Включает отображение кнопки **Полноэкранный просмотр (Full Screen)** на панели инструментов вьювера при просмотре отчетов или панели индикаторов (дашбордов). По умолчанию свойство имеет значение **true**.
ShowFirstPageButton	Включает отображение кнопки **Первая страница (First Page)** на панели инструментов вьювера. По умолчанию свойство имеет значение **true**.
ShowPreviousPageButton	Включает отображение кнопки **Предыдущая страница (Previous Page)** на панели инструментов вьювера. По умолчанию свойство имеет значение **true**.
ShowCurrentPageControl	Включает отображение индикатора текущей страницы отчета. По умолчанию свойство имеет значение **true**.
ShowNextPageButton	Включает отображение кнопки **Следующая страница (Next Page)** на панели инструментов вьювера. По умолчанию свойство имеет значение **true**.
ShowLastPageButton	Включает отображение кнопки **Последняя страница (Last Page)** на панели инструментов вьювер. По умолчанию свойство имеет значение **true**.
ShowZoomButton	Включает отображение кнопки выбора масштаба отчета. По умолчанию свойство имеет значение **true**.
ShowViewModeButton	Включает отображение кнопки выбора режима отображения страниц отчета. По умолчанию свойство имеет значение **true**.
ShowDesignButton	Включает отображение кнопки **Дизайн (Design)** на панели инструментов вьюверапри просмотре отчетов или панели индикаторов (дашбордов). По умолчанию

© 2003-2026 Stimulsoft

	свойство имеет значение **false**.
ShowAboutButton	Включает отображение кнопки **О программе (About)** на панели инструментов вьювера. По умолчанию свойство имеет значение **true**.
ShowRefreshButton	Включает или выключает отображение кнопки **Обновить (Refresh)** на панели инструментов вьювера при просмотре панелей индикаторов (дашбордов). По умолчанию свойство имеет значение **true**.
ShowPinToolbarButton	Включает отображение кнопки **Прикрепить панель (Pin Toolbar)** на панели инструментов вьювера. Кнопка доступна только в мобильном режиме вьювера. По умолчанию свойство имеет значение **true**.
PrintDestination	Устанавливает режим печати отчета. Может принимать одно из указанных ниже значений перечисления **StiPrintDestination**: › **Default** – будет отображено меню с предоставлением выбора режима печати (значение по умолчанию); › **Pdf** – печать будет осуществлена в формате PDF; › **Direct** – печать будет осуществлена в формате HTML непосредственно на принтер, будет отображен системный диалог печати; › **PopupWindow** – печать будет осуществлена в формате HTML через всплывающее окно предварительного просмотра отчета.
ViewMode	Устанавливает режим отображения страниц отчета. Может принимать одно из указанных ниже значений перечисления **StiWebViewMode**: › **SinglePage** – отображается одна страница отчета, выбранная на панели инструментов

© 2003-2026 Stimulsoft

	вьювера (значение по умолчанию); › **Continuous** – отображаются все страницы отчета в виде ленты; › **MultiplePages** – отображаются все страницы отчета в виде таблицы.
Zoom	Задает масштаб отображения страниц отчета. По умолчанию установлен масштаб 100 процентов. Допускаются значения от 10 до 500 процентов. Также допускается устанавливать одно из следующих значений: › **StiZoomMode.PageWidth** – при запуске вьювера будет установлен масштаб, необходимый для отображения отчета по ширине страницы; › **StiZoomMode.PageHeight** – при запуске вьювера будет установлен масштаб, необходимый для отображения отчета по высоте страницы.
MenuAnimation	Включает анимацию при появлении/скрытии меню вьювера. По умолчанию свойство имеет значение **true**.
ShowMenuMode	Устанавливает режим отображения меню вьювера. Может принимать одно из указанных ниже значений перечисления **StiShowMenuMode**: › **Click** – отображение по клику мыши (значение по умолчанию); › **Hover** – отображение по наведении курсора мыши.
AutoHide	Включает автоматическое сворачивание панели инструментов вьювера. Свойство работает только в мобильном режиме вьювера. По умолчанию свойство имеет значение **false**.

© 2003-2026 Stimulsoft

Экспорт отчета (Export)

Наименование	Описание
DefaultSettings	Данная группа свойств предоставляет возможность задать настройки экспортов по умолчанию для каждого типа экспорта. Эти настройки будут применены к диалоговым окнам экспортов при запуске вьювера, либо к отчету в случае отключения отображения диалоговых окон экспортов.
StoreExportSettings	Включает режим сохранения выбранных настроек в диалоговых окнах экспорта. Настройки будут храниться в cookies браузера. По умолчанию свойство имеет значение **true**.
ShowExportDialog	Включает отображение диалогового окна параметров экспорта. Если свойство имеет значение **false**, то экспортирование будет производиться с заданными по умолчанию настройками. По умолчанию свойство имеет значение **true**.
ShowExportToDocument	Включает отображение пункта меню экспорта **Файл документа (Document File)**. По умолчанию свойство имеет значение **true**.
ShowExportToPdf	Включает отображение пункта меню экспорта **Adobe PDF файл (Adobe PDF File)** при просмотре отчетов, и пункт **Adobe PDF** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToXps	Включает отображение пункта меню экспорта **Microsoft XPS файл (Microsoft XPS File)**. По умолчанию свойство имеет значение **false**.
ShowExportToPowerPoint	Включает отображение пункта меню экспорта **Microsoft PowerPoint 2007/2010 файл (Microsoft PowerPoint 2007/2010 File)**. По умолчанию свойство имеет значение **true**.

© 2003-2026 Stimulsoft

ShowExportToHtml	Включает отображение пункта меню экспорта **HTML файл (HTML File)**. По умолчанию свойство имеет значение **true**.
ShowExportToHtml5	Включает отображение пункта меню экспорта **HTML5 файл (HTML5 File)**. По умолчанию свойство имеет значение **true**.
ShowExportToMht	Включает отображение пункта меню экспорта **MHT Web архив (MHT Web Archive)**. По умолчанию свойство имеет значение **true**.
ShowExportToText	Включает отображение пункта меню экспорта **Text файл (Text File)**. По умолчанию свойство имеет значение **true**.
ShowExportToRtf	Включает отображение пункта меню экспорта **Rich Text файл (Rich Text File)**. По умолчанию свойство имеет значение **true**.
ShowExportToWord	Включает отображение пункта меню экспорта **Microsoft Word 2007/2010 файл (Microsoft Word 2007/2010 File)**. По умолчанию свойство имеет значение **true**.
ShowExportToOpenDocumentWriter	Включает отображение пункта меню экспорта **OpenDocument Writer файл (OpenDocument Writer File)**. По умолчанию свойство имеет значение **true**.
ShowExportToExcelBiff	Включает отображение пункта меню экспорта **Microsoft Excel файл (Microsoft Excel File)**. По умолчанию свойство имеет значение **true**.
ShowExportToExcelXml	Включает отображение пункта меню экспорта **Microsoft Excel Xml файл (Microsoft Excel Xml File)**. По умолчанию свойство имеет значение **true**.
ShowExportToExcel	Включает отображение пункта меню экспорта **Microsoft Excel 2007/2010 файл (Microsoft Excel 2007/2010 File)** при просмотре отчетов, и пункт **Microsoft Excel**

© 2003-2026 Stimulsoft

	при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToOpenDocumentCalc	Включает отображение пункта меню экспорта **OpenDocument Calc файл (OpenDocument Calc File)**. По умолчанию свойство имеет значение **true**.
ShowExportToCsv	Включает отображение пункта меню экспорта **CSV файл (CSV File)**. По умолчанию свойство имеет значение **true**.
ShowExportToDbf	Включает отображение пункта меню экспорта **DBF файл (DBF File)**. По умолчанию свойство имеет значение **true**.
ShowExportToXml	Включает отображение пункта меню экспорта **XML файл (XML File)**. По умолчанию свойство имеет значение **true**.
ShowExportToDif	Включает отображение пункта меню экспорта **Data Interchange формат (DIF) файл (Data Interchange Format (DIF) File)**. По умолчанию свойство имеет значение **true**.
ShowExportToSylk	Включает отображение пункта меню экспорта **Symbolic Link (SYLK) файл (Symbolic Link (SYLK) File)**. По умолчанию свойство имеет значение **true**.
ShowExportToJson	Включает отображение пункта меню экспорта **JSON файл (JSON File)**. По умолчанию свойство имеет значение **true**.
ShowExportToImageBmp	Включает отображение пункта меню экспорта **BMP рисунок (BMP Image)** при просмотре отчетов, и пункт **BMP** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToImageGif	Включает отображение пункта меню экспорта **GIF рисунок (GIF Image)** при просмотре отчетов, и пункт **GIF** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.

© 2003-2026 Stimulsoft

ShowExportToImageJpeg	Включает отображение пункта меню экспорта **JPEG рисунок (JPEG Image)** при просмотре отчетов, и пункт **JPEG** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToImagePcx	Включает отображение пункта меню экспорта **PCX рисунок (PCX Image)** при просмотре отчетов, и пункт **PCX** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToImagePng	Включает отображение пункта меню экспорта **PNG рисунок (PNG Image)** при просмотре отчетов, и пункт **PNG** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToImageTiff	Включает отображение пункта меню экспорта **TIFF рисунок (TIFF Image)** при просмотре отчетов, и пункт **TIFF** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToImageSvg	Включает отображение пункта меню экспорта **Scalable Vector Graphics (SVG) файл (Scalable Vector Graphics (SVG) File)** при просмотре отчетов, и пункт **SVG** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToImageSvgz	Включает отображение пункта меню экспорта **Compressed SVG (SVGZ) файл (Compressed SVG (SVGZ) File)** при просмотре отчетов, и пункт **SVGZ** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowOpenAfterExport	Предоставляет возможность отображать или не отображать параметр **Открыть после экспорта (Open After Export)** в меню настроек экспорта. По умолчанию свойство имеет значение **true**.

© 2003-2026 Stimulsoft

Отправка отчета по Email (Email)

Наименование	Описание
ShowEmailDialog	Включает отображение диалогового окна параметров отправки отчета по Email. Если диалоговое окно отключено, то отправка по Email будет производиться с настройками, заданными на стороне сервера в действии **EmailReport**. По умолчанию свойство имеет значение **true**.
ShowExportDialog	Включает отображение диалогового окна параметров экспорта при отправке Email. Если свойство имеет значение **false**, то экспортирование будет производиться с заданными по умолчанию настройками. По умолчанию свойство имеет значение **true**.
DefaultEmailAddress	Устанавливает Email получателя по умолчанию, т.е. адрес на который будет отправлено письмо с вложенным отчетом.
DefaultEmailSubject	Устанавливает тему (заголовок) электронного письма по умолчанию.
DefaultEmailMessage	Устанавливает сообщение (текст) электронного письма по умолчанию.
DefaultEmailReplyTo	Устанавливает сообщение (текст) по умолчанию для ответа на сообщение, созданное во вьювере.

5.2 HTML5 дизайнер

YouTube

Посмотрите видеоуроки по работе с компонентом ASP.NET Core MVC HTML5 Designer. Подписывайтесь на канал Stimulsoft и узнайте первыми о новых видеоуроках. Вопросы и предложения оставляйте в комментариях к видео.

© 2003-2026 Stimulsoft

Примеры

Ознакомьтесь с примерами по работе с компонентом ASP.NET Core MVC HTML5 Designer GitHub. Все примеры представляют собой отдельные проекты, сгруппированные в один solution для Visual Studio.

Компонент **HTML5 Designer** (**StiNetCoreDesigner**) предназначен для редактирования отчетов в окне браузера. При этом не требуется устанавливать у клиента .NET Framework, ActiveX компоненты или какие-либо специальные плагины. Все, что необходимо – это любой современный Web-браузер.

При помощи **HTML5 Designer** можно создавать, редактировать, сохранять, просматривать отчеты и печатать отчеты на любом компьютере с любой установленной операционной системой. Так как дизайнер использует только HTML и JavaScript технологии, он может быть запущен на устройствах, где нет поддержки Flash или Silverlight – планшеты, смартфоны. Также дизайнер поддерживает Touch интерфейс, который автоматически включается при использовании устройств с сенсорным экраном.

Компонент **HTML5 Designer** использует технологию AJAX для выполнения всех действий над редактируемым отчетом, что позволяет избавиться от перезагрузки всей страницы, а также экономить Web-трафик и повысить скорость работы. Для построения отчетов используется ядро генератора отчетов, построенное с использованием .NET Core технологии. Данная технология является кроссплатформенной, что позволяет разворачивать приложение на серверах, использующих операционные системы Windows, macOS и Linux.

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

© 2003-2026 Stimulsoft

Для использования **HTML5 Designer** в Web-проекте, необходимо установить NuGet пакет Stimulsoft.Reports.Web.NetCore:

- Выберите пункт "Manage NuGet Packages..." в контекстном меню проекта;
- На вкладке Browse, в строке поиска, укажите Stimulsoft.Reports.Web.NetCore;
- Выделите элемент, определите версию пакета и нажмите кнопку **Установить (Install)**. При обновлении пакета, следует нажать кнопку **Обновить (Update)**.

Для того, чтобы добавить возможности создания и редактирования дашбордов в Web-проекте, установить NuGet пакет Stimulsoft.Dashboards.Web.NetCore, (данный пакет связан с пакетом Stimulsoft.Reports.Web.NetCore, при его отсутствии он будет установлен автоматически):

- Выберите пункт "Manage NuGet Packages..." в контекстном меню проекта;
- На вкладке Browse, в строке поиска, укажите Stimulsoft.Dashboards.Web.NetCore;
- Выделите элемент, определите версию пакета и нажмите кнопку **Установить (Install)**. При обновлении пакета, следует нажать кнопку **Обновить (Update)**.

- Как это работает
- Активация
- Редактирование отчета и дашборда
- Создание нового отчета и нового дашборда
- Просмотр
- Сохранение отчета и дашборда
- Вспомогательные методы
- Дополнительные возможности просмотра
- Время ожидания
- Локализация дизайнера отчетов
- Использование тем
- Кэширование
- Настройки дизайнера отчетов

© 2003-2026 Stimulsoft

5.2.1 Как это работает

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для запуска дизайнера требуется разместить на странице компонент **StiNetCoreDesigner**, задать ему необходимые настройки, а в контроллере представления определить необходимые действия. При запуске дизайнера отчетов происходят следующие действия:

- .NET Core компонент генерирует HTML и JavaScript код, необходимый для отображения и работы дизайнера;
- По окончании вывода компонента запускается JavaScript метод, который запрашивает на стороне сервера шаблон отчета, и отображает его в окне редактирования;
- Различные действия, такие как предварительный просмотр отчета, сохранение шаблона отчета, экспорт отчета, применение параметров, сортировка и детализация отчета, вызывают определенное действие на стороне сервера, в котором можно выполнить необходимые манипуляции с отчетом.

5.2.2 Активация

YouTube

Посмотрите видеоуроки по активации компонента ASP.NET Core HTML5 Designer. Подписывайтесь на канал Stimulsoft и узнайте первыми о новых видеоуроках. Вопросы и предложения оставляйте в комментариях к видео.

После приобретения продукта Stimulsoft вам необходимо активировать лицензию для используемых компонентов. Это можно выполнить указав лицензионный ключ или загрузив файл с лицензионным ключом. Ниже представлен пример активации компонента **StiNetCoreDesigner**.

© 2003-2026 Stimulsoft

HomeController.cs

```
...
//Activation with using license code
public class HomeController : Controller
{
   static HomeController()
   {
      Stimulsoft.Base.StiLicense.Key = "Your activation code...";
   }
}

//Activation with using license file
public class HomeController : Controller
{
   public HomeController(IHostingEnvironment hostEnvironment)
   {
      var path = Path.Combine(hostEnvironment.ContentRootPath, "Content\
      \license.key");
      Stimulsoft.Base.StiLicense.LoadFromFile(path);
   }
...
```

Получить лицензионный ключ или скачать файл с лицензионным ключом, вы можете в личном кабинете пользователя. Для авторизации в кабинете используйте имя пользователя и пароль, указанные при покупке подписки продукта.

5.2.3 Редактирование отчета и дашборда

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для редактирования шаблона отчета необходимо добавить на страницу компонент **StiNetCoreDesigner** и задать ему минимально необходимые настройки, а в контроллере представления определить необходимые действия.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.StiNetCoreDesigner(new StiNetCoreDesignerOptions() {
   Actions =
   {
      GetReport = "GetReport",
      DesignerEvent = "DesignerEvent"
   }
})
...
```

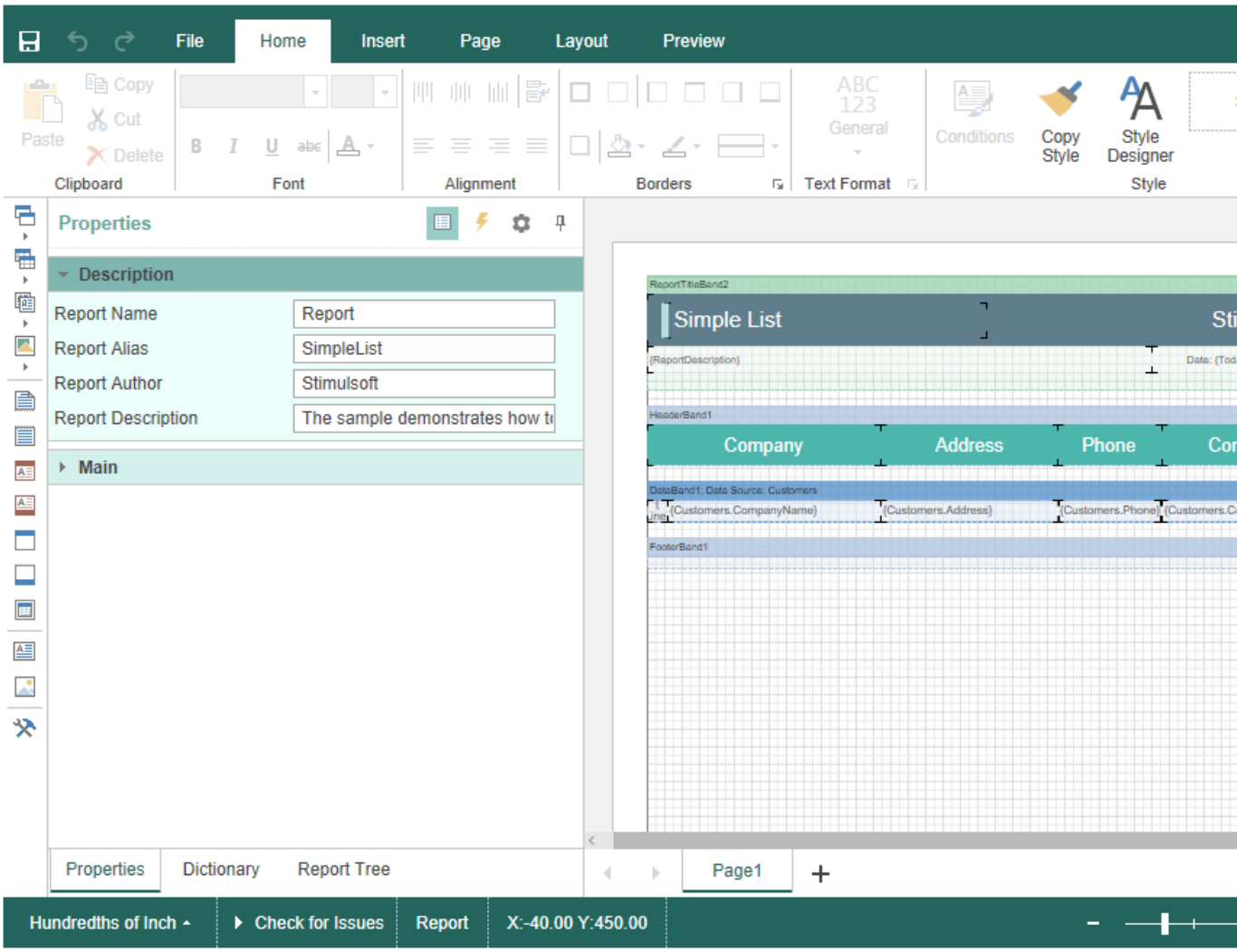
HomeController.cs

```
...
public IActionResult GetReport()
{
   StiReport report = new StiReport();
   report.Load(StiNetCoreHelper.MapPath(this, "Reports/SimpleList.mrt"));
   //report.Load(StiNetCoreHelper.MapPath(this, "Reports/Dashboard.mrt"));

   return StiNetCoreDesigner.GetReportResult(this, report);
}

public IActionResult DesignerEvent()
{
   return StiNetCoreDesigner.DesignerEventResult(this);
}
...
```

© 2003-2026 Stimulsoft



Действие **GetReport** предназначено для загрузки редактируемого шаблона отчета, оно вызывается автоматически после загрузки дизайнера отчетов. Действие **DesignerEvent** предназначено для обработки различных дополнительных действий дизайнера, таких как работа с данными, работа с компонентами, предварительный просмотр отчета и других.

Информация

Действие **DesignerEvent** является обязательным. Без него корректная работа дизайнера невозможна.

© 2003-2026 Stimulsoft

5.2.4 Создание нового отчета и нового дашборда

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для запуска дизайнера с новым (пустым) отчетом достаточно создать новый отчет в действии **GetReport** и вернуть его дизайнеру. При необходимости, можно предварительно загрузить данные для отчета, либо выполнить какие-либо другие необходимые действия.

Index.cshtml

```
...
@Html.StiNetCoreDesigner(new StiNetCoreDesignerOptions() {
   Actions =
   {
      GetReport = "GetReport"
   }
})
...
```

HomeController.cs

```
...
public IActionResult GetReport()
{
   StiReport report = new StiReport();
   //var newDashboard = StiReport.CreateNewDashboard();

   return StiNetCoreDesigner.GetReportResult(this, report);
   //return StiNetCoreDesigner.GetReportResult(this, newDashboard);
}
...
```

Также новый отчет можно создать, используя главное меню дизайнера. Для того, чтобы предварительно загрузить данные для нового отчета, либо выполнить какие-либо другие необходимые действия, предназначено действие **CreateReport**. Данное действие будет вызвано при создании нового пустого отчета из главного меню, либо при создании отчета с использованием мастера.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.StiNetCoreDesigner(new StiNetCoreDesignerOptions() {
   Actions =
   {
      CreateReport = "CreateReport"
   }
})
...
```

HomeController.cs

```
...
public IActionResult CreateReport()
{
   StiReport report = new StiReport();
   //var newDashboard = StiReport.CreateNewDashboard();

   // Register data for the new report, if necessary
   DataSet data = new DataSet("Demo");
   data.ReadXml(StiNetCoreHelper.MapPath(this, "Data/Demo.xml"));
   report.RegData(data);
   //newDashboard.RegData(data);
   report.Dictionary.Synchronize();
   //newDashboard.Dictionary.Synchronize();

   return StiNetCoreDesigner.GetReportResult(this, report);
   //return StiNetCoreDesigner.GetReportResult(this, newDashboard);
}
...
```

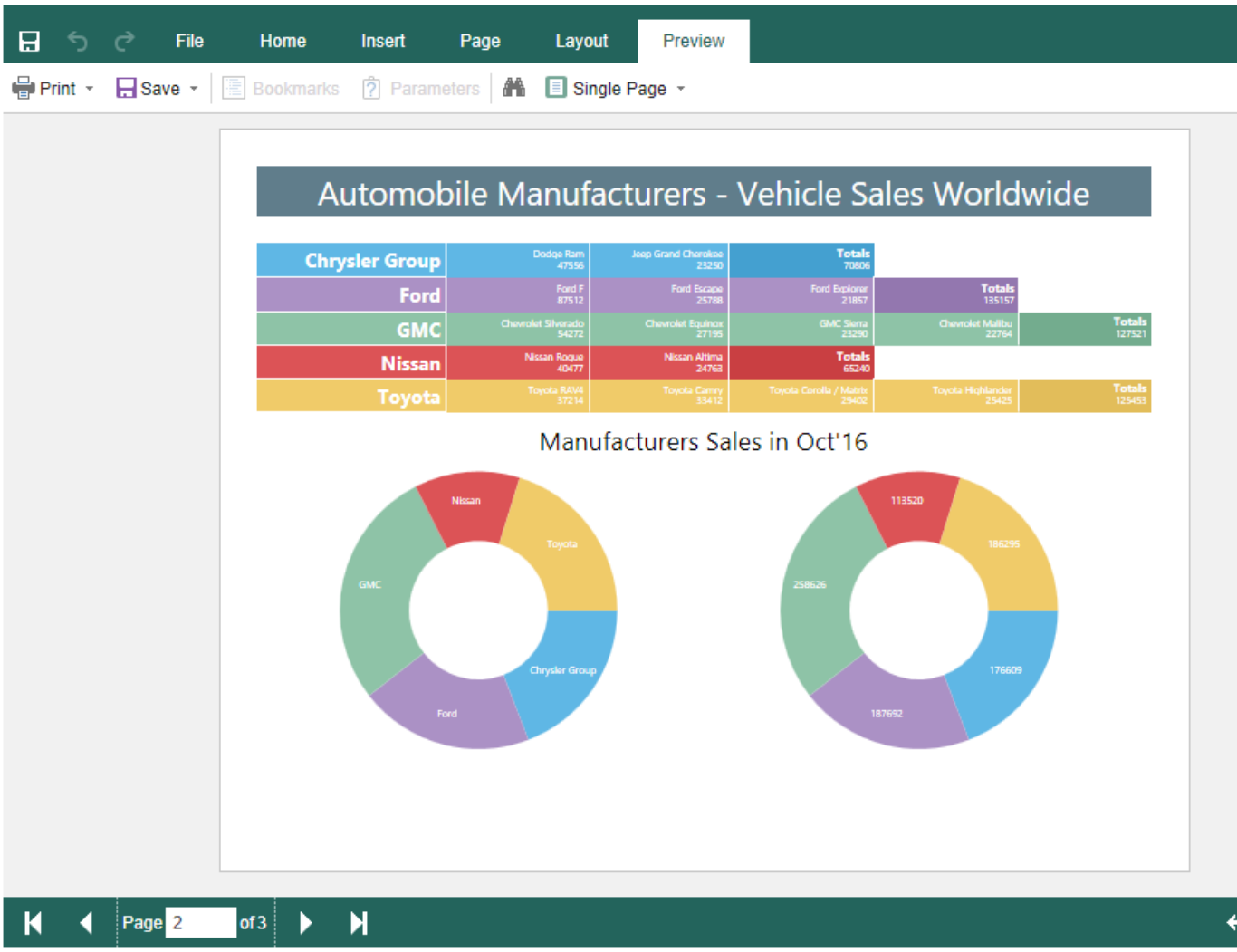
5.2.5 Просмотр

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

У компонента **HTML5 Designer** предусмотрен режим просмотра редактируемого отчета. Для просмотра достаточно перейти на соответствующую вкладку в окне дизайнера. Шаблон отчета будет передан на сторону сервера, построен и отображен во встроенном вьювере.

© 2003-2026 Stimulsoft



Перед просмотром отчета предусмотрена возможность выполнить какие-либо необходимые действия, например, подключить данные для отчета. Для этого достаточно определить специальное действие **PreviewReport**, которое будет вызвано перед просмотром отчета. Действие **PreviewReport** вызывается перед подготовкой и построением отчета для просмотра, до его сохранения в кэш.

Index.cshtml

```
...
@Html.StiNetCoreDesigner(new StiNetCoreDesignerOptions() {
   Actions =
   {
      PreviewReport = "PreviewReport"
   }
})
...
```

© 2003-2026 Stimulsoft

HomeController.cs

```
...
public IActionResult PreviewReport()
{
   StiReport report = StiNetCoreDesigner.GetActionReportObject(this);

   DataSet data = new DataSet("Demo");
   data.ReadXml(StiNetCoreHelper.MapPath(this, "Data/Demo.xml"));
   report.RegData(data);

   return StiNetCoreDesigner.PreviewReportResult(this, report);
}
...
```

При необходимости выполнить действия над отчетом непосредственно перед отображением отчета, можно воспользоваться специальным действием **GetPreviewReport**, которое вызывается после запроса подготовленного отчета из кэша.

Index.cshtml

```
...
@Html.StiNetCoreDesigner(new StiNetCoreDesignerOptions() {
   Actions =
   {
      GetPreviewReport = "GetPreviewReport"
   }
})
...
```

HomeController.cs

```
...
public IActionResult GetPreviewReport()
{
   StiReport report = StiNetCoreDesigner.GetActionReportObject(this);

   DataSet data = new DataSet("Demo");
   data.ReadXml(StiNetCoreHelper.MapPath(this, "Data/Demo.xml"));
   report.RegData(data);
   //report.IsRendered = false;

   return StiNetCoreDesigner.PreviewReportResult(this, report);
}
...
```

© 2003-2026 Stimulsoft

Информация

Так как в данном событии передаётся уже подготовленный для просмотра отчет, то при необходимости выполнить повторное построение, необходимо установить флаг **report.IsRendered = false**.

5.2.6 Дополнительные возможности просмотра

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Окно предварительного просмотра отчета компонента **HTML5 Designer** представляет собой полноценный интерактивный **HTML5 Viewer**, который умеет осуществлять печать отчета и экспортирование отчета, поддерживает работу с параметрами отчета. Также поддерживаются интерактивные действия, такие как динамическая сортировка, детализация, сворачивание. Для использования указанных возможностей не требуется каких-либо дополнительных настроек дизайнера отчетов.

В любом из вышеперечисленных действий допускается производить манипуляции с шаблоном отчета, например, изменять их свойства и параметры, подключать новые данные для построения.

Index.cshtml

```
...
@Html.StiNetCoreDesigner(new StiNetCoreDesignerOptions() {
   Actions =
   {
      ExportReport = "ExportReport"
   }
})
...
```

© 2003-2026 Stimulsoft

HomeController.cs

```
...
public IActionResult ExportReport()
{
   StiReport report = StiNetCoreDesigner.GetActionReportObject(this);
   // ...

   return StiNetCoreDesigner.ExportReportResult(this, report);
}
...
```

Информация

Если какие-либо из указанных дополнительных возможностей предварительного просмотра отчета не требуются (например, экспортирование или печать отчета), можно отключить их, используя соответствующие свойства компонента **HTML5 Designer**.

5.2.7 Сохранение отчета и дашборда

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

В компоненте **HTML5 Designer** предусмотрено два варианта сохранения отчета, которые доступны в главном меню и на главной панели дизайнера: **Сохранить (Save)** и **Сохранить как (Save As)**. В свою очередь, каждый из этих вариантов сохранения имеет свои режимы и настройки.

Сохранение отчета или дашборда на стороне сервера

Для сохранения редактируемого отчета на стороне сервера необходимо определить действие **SaveReport**, которое будет вызвано при выборе пункта **Сохранить (Save)** в главном меню, либо нажатии на кнопку **Сохранить (Save)** на главной панели дизайнера.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.StiNetCoreDesigner(new StiNetCoreDesignerOptions() {
   Actions =
   {
      SaveReport = "SaveReport"
   }
})
...
```

HomeController.cs

```
...
public IActionResult SaveReport()
{
   StiReport report = StiNetCoreDesigner.GetReportObject(this);

   // Save the report template
   // ...

   return StiNetCoreDesigner.SaveReportResult(this);
}
...
```

Данное действие возвращает ответ клиентской стороне дизайнера о результате сохранения отчета. Если необходимо, после сохранения отчета предусмотрена возможность отобразить диалоговое окно с ошибкой или текстовым сообщением.

HomeController.cs

```
...
public IActionResult SaveReport()
{
   StiReport report = StiNetCoreDesigner.GetReportObject(this);

   // Save the report template
   // ...

   // Completion of the report saving with message dialog box
   return StiNetCoreDesigner.SaveReportResult(this, "Some message after
   saving");
   //return Content("{"infoMessage":"Some info message after saving"}");
   //return Content("{\"warningMessage\":\"Some info message after saving
   \"}");
}
...
```

© 2003-2026 Stimulsoft

Если необходимо, вы можете получить доступ к оригинальному имени отчета или имени отчета из диалога сохранения.

HomeController.cs

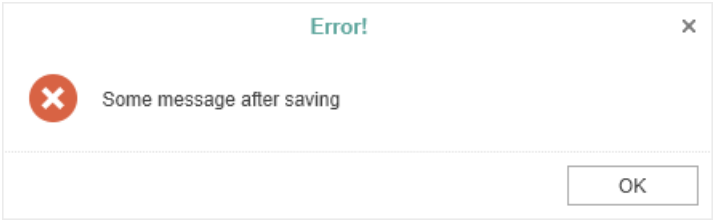
```
...
public IActionResult SaveReport()
{
   StiRequestParams requestParams = StiNetCoreDesigner.GetRequestParams();
   StiReport report = StiNetCoreDesigner.GetReportObject();

   //Report name from designer save dialog
   string savingReportName = requestParams.Designer.FileName;

   //Original report name from properties
   string originalReportName = report.ReportName;

   return StiNetCoreDesigner.SaveReportResult(this, "Some message after
   saving");
}
...
```

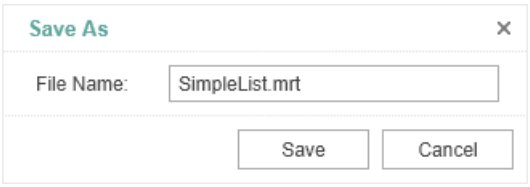
В этом случае, будет отображено диалоговое с указанным текстом. Текст может содержать как сообщение об ошибке сохранения либо предупреждение, так и любое другое сообщение.



Сохранение отчета или дашборда на стороне клиента

Для сохранения редактируемого отчета на стороне клиента в виде файла не требуется дополнительных настроек дизайнера. Достаточно выбрать пункт главного меню **Сохранить как (Save As)**, при нажатии на который отобразится диалог сохранения файла. В данном диалоге можно изменить имя файла отчета, после чего файл будет сохранен на локальном диске компьютера.

© 2003-2026 Stimulsoft



Компонент **HTML5 Designer** предоставляет возможность изменить поведение указанного варианта сохранения, для этого в настройках дизайнера предусмотрено специальное действие **SaveReportAs**. При определении этого действия сохранение отчета будет происходить на стороне сервера, работа данного действия будет аналогична действию **SaveReport**.

Index.cshtml

```
...
@Html.StiNetCoreDesigner(new StiNetCoreDesignerOptions() {
   Actions =
   {
      SaveReportAs = "SaveReportAs"
   }
})
...
```

HomeController.cs

```
...
public IActionResult SaveReportAs()
{
   StiReport report = StiNetCoreDesigner.GetReportObject(this);

   // Save the report template
   // ...

   return StiNetCoreDesigner.SaveReportResult(this);
}
...
```

Чтобы получить наименование отчета из диалогового окна сохранения, используйте следующий код.

© 2003-2026 Stimulsoft

HomeController.cs

```
public IActionResult SaveReport()
{
   StiReport report = StiNetCoreDesigner.GetReportObject(this);
   var requestParams = StiNetCoreDesigner.GetRequestParams(this);
   var reportName = requestParams.Designer.FileName;

   return StiNetCoreDesigner.SaveReportResult(this);
}
```

Настройки сохранения

Сохранение отчета производиться в фоновом режиме, т.е. без перезагрузки страницы в окне браузера. Если необходимо каким-либо образом визуально управлять процессом сохранения отчета, следует изменить значение свойства **SaveReportMode** (либо **SaveReportAsMode**) дизайнера на одно из трех указанных значений: **Hidden** (значение по умолчанию), **Visible** или **NewWindow**.

Index.cshtml

```
...
@Html.StiNetCoreDesigner(new StiNetCoreDesignerOptions() {
   Actions =
   {
      SaveReportAs = "SaveReportAs"
   },
   Behavior =
   {
      SaveReportAsMode = StiSaveMode.Visible
   }
})
...
```

Если свойство **SaveReportMode** установлено в значение **Visible**, то действие сохранения отчета будет вызвано в текущем окне браузера в обычное (видимом) режиме при помощи POST-запроса. Если свойство **SaveReportMode** установлено в значение **NewWindow**, то действие сохранения отчета будет вызвано в новом окне браузера. По умолчанию данное свойство установлено в значение **Hidden**, т.е. действие сохранения отчета вызывается в фоне при помощи AJAX-запроса и никак не отображается в окне браузера. Для свойства **SaveReportAsMode** применимы те же значения и поведение.

© 2003-2026 Stimulsoft

5.2.8 Локализация дизайнера

Компонент **HTML5 Designer** поддерживает полную локализацию своего интерфейса. Для того чтобы локализовать интерфейс дизайнера отчетов на необходимый язык, используется специальное свойство **Localization**. В качестве значения данного свойства необходимо указать путь к XML-файлу локализации (относительный либо абсолютный).

Index.cshtml

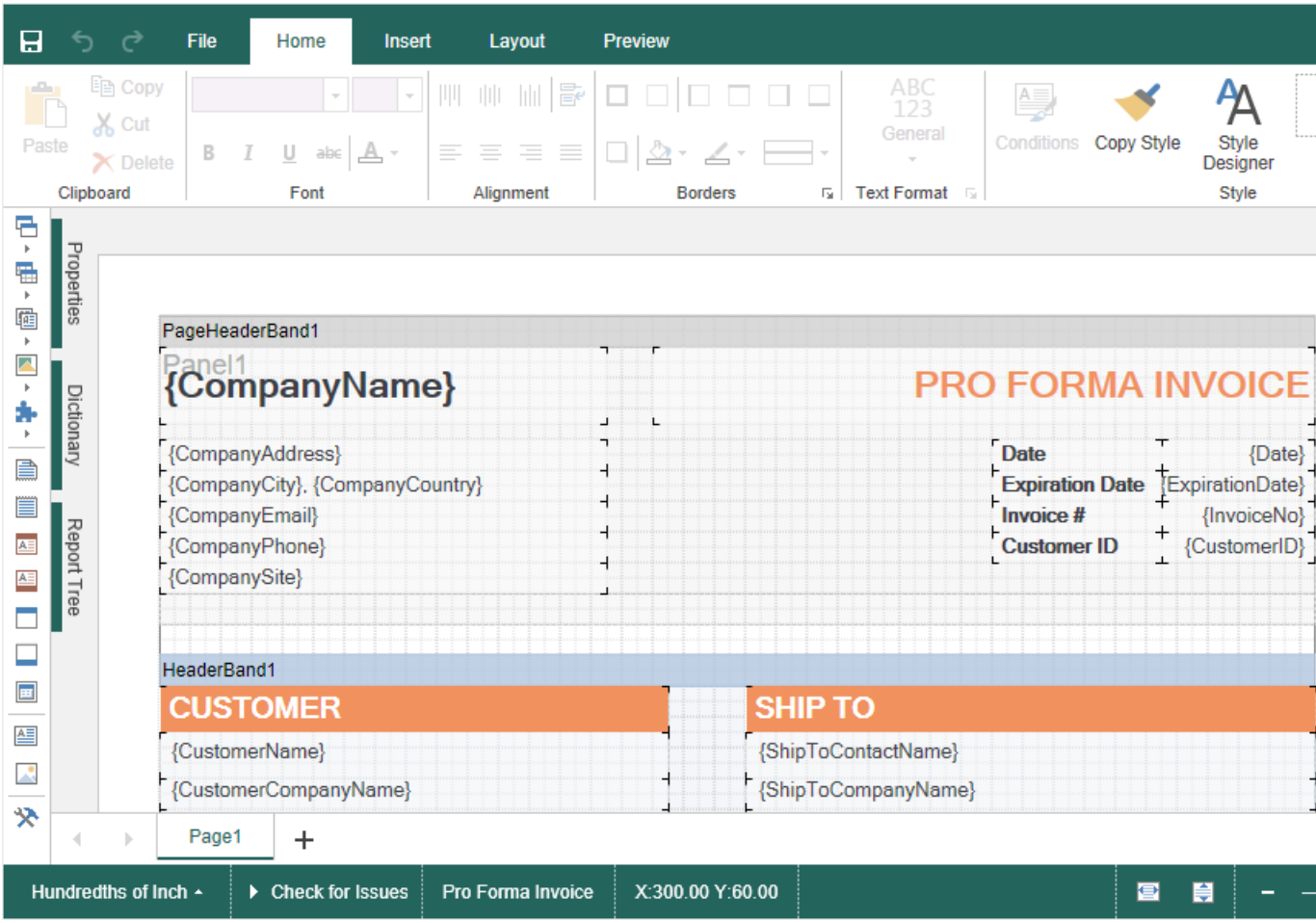
```
...
@Html.StiNetCoreDesigner(new StiNetCoreDesignerOptions() {
   Localization = "Localization/en.xml"
})
...
```

Интерфейс дизайнера отчетов позволяет выбрать необходимую локализацию из доступного списка. Для этого в качестве значения свойства **LocalizationDirectory** необходимо указать папку, в которой хранятся локализационные XML-файлы.

Index.cshtml

```
...
@Html.StiNetCoreDesigner(new StiNetCoreDesignerOptions() {
   Localization = "Localization/en.xml",
   LocalizationDirectory = "Localization"
})
...
```

© 2003-2026 Stimulsoft



Информация

Если задано значение для свойства **Localization**, то при запуске дизайнера отчетов всегда будет применяться указанная в этом свойстве локализация. Если значение свойства не задано, то будет автоматически загружена локализация, выбранная из списка доступных локализаций на панели дизайнера отчетов.

5.2.9 Использование тем

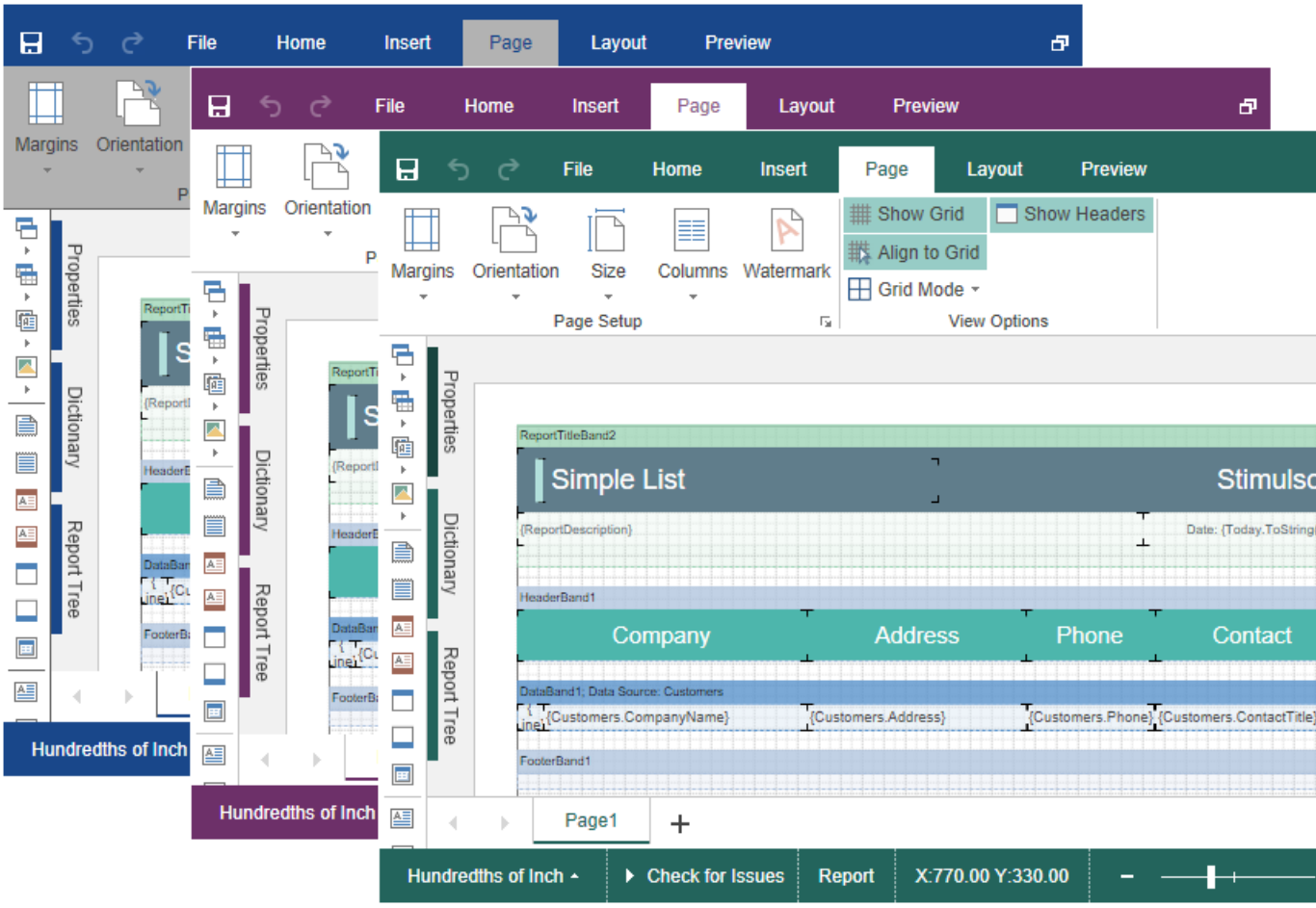
Компонент **HTML5 Designer** имеет возможность смены тем оформления визуальных элементов управления. Для изменения темы используется свойство **Theme**.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.StiNetCoreDesigner(new StiNetCoreDesignerOptions() {
   Theme = StiDesignerTheme.Office2022WhiteTeal
})
...
```

На данный момент доступно **2 различные темы** со своими цветовыми акцентами. В результате, доступно **более 50 вариантов** оформления. Это позволяет настроить внешний вид дизайнера практически под любое оформление Web проекта.



© 2003-2026 Stimulsoft

5.2.10 Кэширование

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для работы **HTM5 Designer** использует кэш сервера для хранения редактируемого шаблона отчета. Это необходимо, так как клиентская часть дизайнера содержит только визуальное представление компонентов шаблона отчета, сам объект отчета со всеми параметрами и свойствами хранится на стороне сервера.

Для использования кэширования необходимо подключить модули для работы с сессией либо кэшем на стороне сервера. Для этого достаточно добавить в проект указанные ниже сервисы в стартовом файле проекта.

Startup.cs

```
...
public void ConfigureServices(IServiceCollection services)
{
   services.AddMemoryCache();
   services.AddSession();
   services.AddMvc();
}
...
```

Управлять кэшированием можно при помощи следующих приведенных ниже свойств.

Свойство CacheMode

© 2003-2026 Stimulsoft

Данное свойство дизайнера устанавливает тип кэширования. Может принимать одно из значений, указанных в перечислении **StiServerCacheMode**:

- **None** – не используется в компоненте **HTML5 Designer**;
- **ObjectCache** – для кэширования используется кэш сервера, в котором сохраняется объект отчета (значение по умолчанию);
- **StringCache** - для кэширования используется кэш сервера, в котором сохраняется отчет в виде упакованной строки;
- **ObjectSession** - для кэширования используется текущая сессия, в которой сохраняется объект отчета;
- **StringSession** - для кэширования используется текущая сессия, в которой сохраняется отчет в виде упакованной строки.

Свойство CacheItemPriority

Данное свойство устанавливает приоритет хранимого в кэше сервера отчета, влияет на автоматическую очистку памяти сервера в случае ее нехватки. Чем меньше приоритет, тем больше шанс удаления информации из памяти.

Свойство CacheTimeout

Данное свойство указывает величину времени в минутах, в течение которого необходимо сохранять отчет в кэше сервера. Если при использовании кэширования требуемый отчет не найден в кэше (т.е. истекло время хранения объекта), то он будет запрошен заново при помощи специального действия **GetReport**, в этом случае несохраненные изменения могут быть утеряны.

Компонент **HTML5 Designer** предоставляет возможность определить собственные методы работы с кэшированием отчета. Для этого предназначен специальный класс **StiCacheHelper**, который содержит методы получения отчета из кэша и сохранения отчета в кэш. Необходимо создать новый класс, унаследованный от **StiCacheHelper**, и перегрузить указанные выше методы,

© 2003-2026 Stimulsoft

которые соответственно имеют названия **GetObject**, **SaveObject** и **RemoveObject**.

HomeController.cs

```
...
public class DesignerController : Controller
{
   public class StiMyCacheHelper : StiCacheHelper
   {
      public override object GetObject(string guid)
      {
         string path =
         System.IO.Path.Combine(this.HttpContext.Server.MapPath("CacheFiles"
         ), guid);
         if (System.IO.File.Exists(path))
         {
            byte[] cacheData = System.IO.File.ReadAllBytes(path);
            return StiCacheHelper.GetObjectFromCacheData(cacheData);
         }
         return null;

         //return base.GetObject(guid);
      }

      public override void SaveObject(object obj, string guid)
      {
         byte[] cacheData = StiCacheHelper.GetCacheDataFromObject(obj);
         string path =
         System.IO.Path.Combine(this.HttpContext.Server.MapPath("CacheFiles"
         ), guid);
         System.IO.File.WriteAllBytes(path, cacheData);

         //base.SaveObject(obj, guid);
      }

      public override void RemoveObject(string guid)
      {
         string path =
         System.IO.Path.Combine(this.HttpContext.Server.MapPath("CacheFiles"
         ), guid);
         if (File.Exists(path))
         {
            File.Delete(path);
         }
      }
   }

   static DesignerController()
   {
      StiNetCoreDesigner.CacheHelper = new StiMyCacheHelper();
   }
}
...
```

© 2003-2026 Stimulsoft

Для инициализации работы с кэшированием отчета при помощи созданного класса, достаточно задать его в качестве значения статического свойства **StiNetCoreDesigner.CacheHelper** в конструкторе контроллера.

5.2.11 Вспомогательные методы

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для **HTML5 Designer** предусмотрено несколько вспомогательных методов, которые предназначены для получения объекта редактируемого в данный момент отчета, параметров текущего состояния дизайнера и других полезных данных. Эти методы можно использовать в действиях дизайнера.

Метод GetReportObject()

Возвращает объект отчета, с которым в данный момент работает дизайнер. Допускается производить с ним необходимые манипуляции - регистрировать новые наборы данных, изменять свойства отчета, присваивать параметры либо загружать в объект другой отчет. Затем отчет можно вернуть дизайнеру, указав его в качестве параметра в результирующем методе действия.

HomeController.cs

```
...
public IActionResult ExportReport()
{
   StiReport report = StiNetCoreDesigner.GetReportObject(this);
   report.ReportName = "MyReportName";

   return StiNetCoreDesigner.ExportReportResult(this, report);
}
...
```

© 2003-2026 Stimulsoft

Метод GetActionReportObject()

Возвращает объект отчета, который будет использоваться для конкретного действия. Например, для действия **OpenReport** данный метод вернет отчет, загруженный с локального диска компьютера. Для действия **PreviewReport** метод вернет подготовленную копию отчета для предварительного просмотра.

HomeController.cs

```
...
public IActionResult OpenReport()
{
   StiReport report = StiNetCoreDesigner.GetActionReportObject(this);

   // Register data for the opened report, if necessary
   DataSet data = new DataSet("Demo");
   data.ReadXml(StiNetCoreHelper.MapPath(this, "Data/Demo.xml"));
   report.RegData(data);
   report.Dictionary.Synchronize();

   return StiNetCoreDesigner.GetReportResult(this, report);
}
...
```

Метод GetRouteValues()

Возвращает значения маршрутов для URL, с которыми была открыта страница дизайнера. Таким образом, предоставляется возможность получить исходную коллекцию параметров страницы запуска дизайнера и использовать эти значения для каких-либо проверок и условий.

HomeController.cs

```
...
public IActionResult ExportReport()
{
   RouteValueDictionary routeValues =
   StiNetCoreDesigner.GetRouteValues(this);

   return StiNetCoreDesigner.ExportReportResult(this);
}
...
```

Также можно получить значения параметров URL по имени параметра, указав его в качестве параметра вызываемого действия дизайнера.

© 2003-2026 Stimulsoft

HomeController.cs

```
...
public IActionResult ExportReport(string id)
{
   return StiNetCoreDesigner.ExportReportResult(this);
}
...
```

Метод GetFormValues()

Возвращает значения формы, которая инициировала (открыла POST-запросом) страницу дизайнера. Таким образом предоставляется возможность получить коллекцию параметров формы в любом действии дизайнера.

Index.cshtml.cs

```
...
public IActionResult DesignerInteraction()
{
   NameValueCollection formValues = StiNetCoreDesigner.GetFormValues(this);

   return StiNetCoreDesigner.InteractionResult(this);
}
...
```

По умолчанию данная возможность отключена в целях оптимизации запросов клиентской стороны дизайнера на сервер. Для включения достаточно установить свойство PassFormValues в значение true.

Index.cshtml

```
...
@Html.StiNetCoreDesigner(new StiNetCoreDesignerOptions() {
   Server =
   {
      PassFormValues = true
   }
})
...
```

Метод GetRequestParams()

© 2003-2026 Stimulsoft

Возвращает все параметры текущего состояния дизайнера, переданные на сторону сервера. Они могут быть полезны для определения типа действия, которое в данный момент выполняет дизайнер – например, для определения типа экспорта, а также всех параметров действия.

HomeController.cs

```
...
public IActionResult ExportReport()
{
   StiRequestParams requestParams =
   StiNetCoreDesigner.GetRequestParams(this);
   if (requestParams.ExportFormat == StiExportFormat.Pdf)
   {
      StiReport report = StiNetCoreDesigner.GetReportObject(this);

      // Some action with report for the PDF export
      // ...

      return StiNetCoreDesigner.ExportReportResult(this, report);
   }

   return StiNetCoreDesigner.ExportReportResult(this);
}
...
```

Метод GetExportSettings()

Возвращает все параметры текущего экспортирования отчета. Тип объекта параметров будет соответствовать типу экспорта, выбранному в меню предварительного просмотра отчета. Любые параметры экспорта можно изменить и передать на вход результирующего метода. В этом случае, экспортирование отчета будет произведено с переданными параметрами.

© 2003-2026 Stimulsoft

HomeController.cs

```
...
public IActionResult ExportReport()
{
   StiExportSettings settings = StiNetCoreDesigner.GetExportSettings(this);
   if (settings.GetExportFormat() == StiExportFormat.Pdf)
   {
      StiPdfExportSettings pdfSettings = (StiPdfExportSettings)settings;
      pdfSettings.EmbeddedFonts = true;
      pdfSettings.AllowEditable = StiPdfAllowEditable.No;
      return StiNetCoreDesigner.ExportReportResult(this, settings);
   }

   return StiNetCoreDesigner.ExportReportResult(this);
}
...
```

Методы MapPath() и MapWebRootPath()

Возвращает абсолютный путь соответственно к директории приложения или к директории wwwroot. Можно использовать для загрузки файлов шаблонов отчетов, файлов данных и др. Данные методы расположены в статическом классе **StiNetCoreHelper**.

HomeController.cs

```
...
public IActionResult GetReport()
{
   StiReport report = new StiReport();
   report.Load(StiNetCoreHelper.MapPath(this, "Reports/SimpleList.mrt"));

   return StiNetCoreDesigner.GetReportResult(this, report);
}
...
```

5.2.12 Время ожидания

При работе с компонентом **StiNetCoreDesigner** можно установить время ожидания (timeout) выполнения различных операций - хранение отчета в кэше, ответ сервера, выполнение запроса. Настройка времени ожидания выполняется при помощи свойств компонента и опций отчета.

Свойство CacheTimeout

Предоставляет возможность установить время в минутах, которое сервер будет хранить редактируемый отчет в кэше с момента последнего действия

© 2003-2026 Stimulsoft

дизайнера. По умолчанию установлено значение 10 минут.

Index.cshtml

```
...
@Html.StiNetCoreDesigner(new StiNetCoreDesignerOptions() {
   Server =
   {
      CacheTimeout = 10
   }
})
...
```

Использование кэша увеличивает скорость работы дизайнера с отчетом. Более подробно можно ознакомиться в главе Кэширование.

Свойство RequestTimeout

Предоставляет возможность установить время ожидания ответа от сервера в секундах, по истечении которого будет выдана ошибка связи с сервером. По умолчанию установлено значение 30 секунд. Для больших отчетов рекомендуется увеличить данное значение.

Index.cshtml

```
...
@Html.StiNetCoreDesigner(new StiNetCoreDesignerOptions() {
   Server =
   {
      RequestTimeout = 30
   }
})
...
```

Опция CommandTimeout

Предоставляет возможность установить время ожидания запроса в секундах, при использовании SQL источников данных в отчете. Значение данного свойства сохраняется в самом шаблоне отчёта для каждого SQL подключения отдельно.

Ниже приведен пример кода, который предоставляет возможность установить время ожидания запроса для уже созданного MS SQL соединения и источников данных в отчете.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.StiNetCoreDesigner(new StiNetCoreDesignerOptions() {
   Actions =
   {
      GetReport = "GetReport",
      DesignerEvent = "DesignerEvent"
   }
})
...
```

HomeController.cs

```
...
public IActionResult GetReport()
{
   StiReport report = new StiReport();
   report.Load(Server.MapPath("Report.mrt"));
   ((StiSqlSource)
   report.Dictionary.DataSources["DataSourceName"]).CommandTimeout = 1000;

   return StiNetCoreDesigner.GetReportResult(this, report);
}

public IActionResult DesignerEvent()
{
   return StiNetCoreDesigner.DesignerEventResult(this);
}
...
```

5.2.13 Добавление пользовательских функций

Информация

Ознакомьтесь с примером по добавлению пользовательской функции для компонента ASP.NET Core HTML5 Designer GitHub.

При интеграции дизайнера отчетов в пользовательское приложение, присутствует возможность добавления custom функции в словарь данных дизайнера отчетов. После добавления функции, её можно использовать при разработке отчетов и панелей индикаторов (дашбордов). Ниже представлен пример добавления функции расчета итога суммирования.

© 2003-2026 Stimulsoft

DesignerController.cs

```
...
public static decimal MySum(object value)
{
   if (!ListExt.IsList(value))
   return Stimulsoft.Base.Helpers.StiValueHelper.TryToDecimal(value);

   return Stimulsoft.Data.Functions.Funcs.SkipNulls(ListExt.ToList(value))
   .TryCastToDecimal()
   .Sum();
}
...
static DesignerController()
{
   StiFunctions.AddFunction("MyCategory", "MySum",
   "description", typeof(DesignerController),
   typeof(decimal), "Calculates a sum of the specified set of values.",
   new[] { typeof(object) },
   new[] { "values" },
   new[] { "A set of values" }).UseFullPath = false;
}
...
```

5.2.14 Настройки

Настройка **HTML5 Designer** выполняется при помощи свойств, находящихся в классе **StiNetCoreDesignerOptions**. Все свойства разделены на группы, некоторые из групп содержат свои подгруппы свойств для удобства использования. Ниже приведен пример установки некоторых свойств дизайнера.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.StiNetCoreDesigner(new StiNetCoreDesignerOptions() {
   Theme = Stimulsoft.Report.Web.StiDesignerTheme.Office2022WhiteTeal,
   Localization = "Localization/en.xml",
   Actions =
   {
      GetReport = "GetReport",
      PreviewReport = "PreviewReport",
      SaveReport = "SaveReport",
      DesignerEvent = "DesignerEvent"
   },
   Appearance =
   {
      ReportDisplayMode = StiReportDisplayMode.Auto,
      ShowTooltipsHelp = false,
      ShowDialogsHelp = false,
      DefaultUnit = Stimulsoft.Report.StiReportUnitType.Centimeters
   },
   Dictionary =
   {
      PermissionBusinessObjects =
      Stimulsoft.Report.Web.StiDesignerPermissions.None,
      PermissionDataConnections =
      Stimulsoft.Report.Web.StiDesignerPermissions.View
   },
   Bands =
   {
      ShowChildBand = false,
      ShowEmptyBand = false,
      ShowOverlayBand = false
   }
})
...
```

Основные настройки (без группы)

Наименование	Описание
Theme	Задает тему оформления дизайнера. Список доступных тем оформления находится в перечислении **StiDesignerTheme**. По умолчанию установлено значение **Office2022WhiteBlue**.
Localization	Задает путь к XML файлу локализации. Путь может быть абсолютным либо относительным. По умолчанию используется английская локализация, которая встроена в

© 2003-2026 Stimulsoft

	дизайнер и не требует дополнительных XML файлов.
LocalizationDirectory	Задает путь к директории с XML файлами локализаций. Находящиеся в указанной папке локализационные файлы будут загружены в список выбора локализации на панели дизайнера.
Width	Задает ширину компонента в необходимых единицах измерения, которые определены в классе **Unit**. Доступна установка значения в пикселях: **Unit.Pixel()**, поинтах: **Unit.Point()** и процентах: **Unit.Percentage()**. По умолчанию компонент разворачивается на всю область окна браузера.
Height	Задает высоту компонента в необходимых единицах измерения, которые определены в классе **Unit**. Доступна установка значения в пикселях: **Unit.Pixel()**, поинтах: **Unit.Point()** и процентах: **Unit.Percentage()**. По умолчанию компонент разворачивается на всю область окна браузера.

Действия (Actions)

Наименование	Описание
GetReport	Задает название метода действия подготовки шаблона отчета при загрузке дизайнера.
OpenReport	Задает название метода действия открытия шаблона отчета из меню дизайнера.
CreateReport	Задает название метода действия подготовки шаблона отчета при создании нового отчета в дизайнере.
SaveReport	Задает название метода действия сохранения шаблона отчета на стороне сервера.
SaveReportAs	Задает название метода действия сохранения

© 2003-2026 Stimulsoft

	шаблона отчета на стороне сервера при использовании пункта меню **Сохранить как (Save As)**. Если действие не задано, будет использоваться встроенный метод сохранения шаблона отчета на локальный диск компьютера.
PreviewReport	Задает название метода действия подготовки построенного отчета в окне просмотра.
GetPreviewReport	Задает название метода действия непосредственно перед отображением отчета в окне просмотра.
ExportReport	Задает название метода действия экспорта отчета в необходимый формат.
Exit	Задает название метода действия для перехода к нужному представлению при нажатии кнопки Выход (Exit) в главном меню дизайнера.
DesignerEvent	Задает название метода действия дизайнера отчетов, служащего для обработки различных дополнительных действий дизайнера, таких как работа с данными, работа с компонентами отчета, а так же для загрузки скриптов и стилей дизайнера.

Работа с сервером (Server)

Наименование	Описание
Controller	Задает название контроллера обработки запросов. Если данное свойство не задано, то для обработки запросов будет использоваться текущий контроллер.
RouteTemplate	Задает шаблон (route template) вызова действий дизайнера отчетов. Если свойство не установлено, будет использоваться шаблон, установленный в MVC проекте. Если

© 2003-2026 Stimulsoft

	свойство `UseRelativeUrls` установлено в true, для этого свойства не будет учитываться `BasePath`. По умолчанию это свойство не установлено.
ShowServerErrorPage	Включает отображение HTML-страницы с детализацией ошибки, возникшей на стороне сервера. При включенном свойстве детализация ошибки будет отображена в окне предварительного просмотра. Если свойство выключено, будет отображен только числовой код ошибки и краткий текст ошибки в диалоговом окне. По умолчанию свойство имеет значение true.
AllowAutoUpdateCookies	Предоставляет возможность дизайнеру автоматически обновлять файлы cookies при каждом запросе на сервер. По умолчанию cookies устанавливаются при создании дизайнера, если они не указаны в отчете. По умолчанию свойство имеет значение **false**.
AllowAntiforgeryToken	Предоставляет возможность дизайнеру автоматически запрашивать и отправлять токен защиты от подделки. По умолчанию свойство имеет значение **true**.
RequestTimeout	Устанавливает время ожидания ответа от сервера в секундах, по истечении которого будет выдана ошибка. По умолчанию установлено значение 30 секунд. Для больших отчетов рекомендуется увеличить данное значение.
CacheTimeout	Устанавливает время в минутах, которое сервер будет хранить редактируемый отчет с момента последнего действия дизайнера. По умолчанию установлено значение 10 минут.
CacheMode	Устанавливает режим кэширования отчета. Может принимать одно из указанных ниже значений перечисления **StiServerCacheMode**:

© 2003-2026 Stimulsoft

	› **None** – не используется в компоненте **HTML5 Designer**; › **ObjectCache** – в качестве хранилища используется кэш, отчет хранится как объект (значение свойства по умолчанию); › **ObjectSession** – в качестве хранилища используется сессия, отчет хранится как объект; › **StringCache** – в качестве хранилища используется кэш сервера, отчет сериализуется в упакованную строку; › **StringSession** – в качестве хранилища используется сессия, отчет сериализуется в упакованную строку.
CacheItemPriority	Устанавливает приоритет хранимого в кэше сервера отчета. Это свойство влияет на автоматическую очистку памяти сервера в случае ее нехватки. Чем меньше приоритет, тем больше шанс удаления информации из памяти.
AllowAutoUpdateCache	Устанавливает режим автоматического обновления кэша. Отчет, хранящийся в кэше либо сессии сервера, будет автоматически пересохранен через определенный промежуток времени при бездействии дизайнера (примерно каждые 3 минуты). По умолчанию свойство имеет значение **true**.
UseRelativeUrls	Устанавливает режим дизайнера, в котором для запросов на сервер используются относительные ссылки. По умолчанию свойство имеет значение **true**.
PortNumber	Получает или задает значение, которое указывает номер порта для использования в URL. Значение 0 определяет автоматическое обнаружение (установлено по умолчанию). Значение -1 удаляет номер порта.
PassQueryParametersForResource	Включает передачу всех параметров URL

© 2003-2026 Stimulsoft

s	запроса при формировании ссылок на ресурсы дизайнера. При значении **false** для запроса ресурсов дизайнера используются только необходимые параметры, что способствует более корректной работе кэша браузера. По умолчанию свойство имеет значение **true**.
PassFormValues	Включает передачу значений POST-формы на сторону клиента, если эти значения требуется использовать в действиях дизайнера. При включении данной возможности, вспомогательный метод **GetFormValues()** будет возвращать коллекцию параметров формы. По умолчанию свойство имеет значение **false**.
UseCompression	Включает сжатие запросов дизайнера в GZip поток. Это позволяет уменьшить объем интернет-трафика, но немного замедляет работу дизайнера. По умолчанию свойство имеет значение **false**.
UseCacheForResources	Включает кэширование на стороне сервера ресурсов компонента, таких как скрипты, стили, изображения. Позволяет ускорить загрузку компонента и уменьшить нагрузку на сервер при большом количестве клиентов. По умолчанию свойство имеет значение **true**.
AllowLoadingCustomFontsToClientSide	Позволяет передавать пользовательские шрифты на клиентскую сторону и преобразовывать их в CSS стиль для корректного отображения текста на стороне HTML с заданным шрифтом. По умолчанию свойство имеет значение **false**.

Внешний вид (Appearance)

Наименование	Описание

© 2003-2026 Stimulsoft

IconSet	Предоставляет возможность установить набор иконок: › StiWebUIIconSet.Auto (значение по умолчанию) - устанавливает набор иконок автоматически. Для тем **Office2022** - используется набор иконок в стиле **Monoline**, для тем **Office2013** - используется набор иконок в стиле **Regular**. › StiWebUIIconSet.Monoline - устанавливает набор иконок в стиле **Monoline**; › StiWebUIIconSet.Regular - устанавливает набор иконок в стиле **Regular**.
CustomCss	Задает путь к CSS файлу стилей дизайнера. Если данное свойство установлено, то стандартные стили выбранной темы загружены не будут. По умолчанию установлено пустое значение.
DefaultUnit	Устанавливает единицы измерения размеров для отчета и всех его компонентов. По умолчанию используются сантиметры.
Zoom	Задает масштаб отображения страниц отчета. По умолчанию установлен масштаб 100 процентов. Допускаются значения от 10 до 200 процентов. Может принимать одно из указанных ниже значений перечисления **StiZoomMode**: › **PageWidth** – при запуске дизайнера будет установлен масштаб, необходимый для отображения отчета по ширине страницы; › **PageHeight** – при запуске дизайнера будет установлен масштаб, необходимый для отображения отчета по высоте страницы.
ShowAnimation	Включает анимацию элементов интерфейса дизайнера. По умолчанию свойство имеет значение **true**.

© 2003-2026 Stimulsoft

ShowOpenDialog	Устанавливает режим открытия с диалогом или через событие open. По умолчанию свойство имеет значение **true**, т.е. диалог открытия отображается.
ShowTooltips	Включает отображение подсказок для элементов управления дизайнера при наведении мыши. По умолчанию свойство имеет значение **true**.
ShowTooltipsHelp	Включает отображение ссылки на онлайн документацию в подсказках для элементов управления дизайнера. По умолчанию свойство имеет значение **true**.
ShowDialogsHelp	Включает отображение ссылки на онлайн документацию на заголовках диалоговых форм дизайнера. По умолчанию свойство имеет значение **true**.
InterfaceType	Устанавливает тип используемого интерфейса дизайнера. Может принимать одно из указанных ниже значений перечисления **StiInterfaceType**: › **Auto** – тип интерфейса дизайнера будет выбран автоматически в зависимости от используемого устройства (значение по умолчанию); › **Mouse** – принудительное использование интерфейса для управления дизайнером при помощи мыши; › **Touch** – принудительное использование Touch интерфейса для управления дизайнером при помощи сенсорного экрана (мобильные устройства), также в этом режиме для удобства управления элементы интерфейса увеличены.
DatePickerFirstDayOfWeek	Устанавливает первый день недели для элемента выбора даты. Может принимать одно из указанных ниже значений

© 2003-2026 Stimulsoft

	перечисления **StiFirstDayOfWeek**: › **Auto** - автоматическое определение первого дня недели из настроек браузера (значение по умолчанию); › **Monday** – первым днем недели является понедельник; › **Sunday** – первым днем недели является воскресенье.
DatePickerIncludeCurrentDayForRanges	Предоставляет возможность включать или не включать текущий день в элементе выбора даты. По умолчанию свойство имеет значение **false**.
FormatForDateControls	Предоставляет возможность переопределить формат для элементов управления дат. По умолчанию, значение для текущей опции не задан и формат для дат определяется из локали браузера.
ShowReportTree	Включает отображение дерева компонентов отчета. По умолчанию свойство имеет значение **true**.
ChartRenderType	Устанавливает режим отображения диаграмм в отчете на вкладке **Просмотр (Preview)**. Может принимать одно из указанных ниже значений перечисления **StiChartRenderType**: › **Image** – диаграммы отображаются в виде статичных изображений; › **Vector** – диаграммы отображаются в векторном режиме в виде SVG объекта; › **AnimatedVector** - диаграммы отображаются в векторном режиме в виде SVG объекта, элементы диаграммы появляются с анимацией (значение по умолчанию).
ReportDisplayMode	Устанавливает режим экспорта для отображения страниц отчета на вкладке

© 2003-2026 Stimulsoft

	предварительного просмотра. Может принимать одно из указанных ниже значений перечисления **StiReportDisplayMode**: › **FromReport** - режим экспорта элементов отчета определяется из настроек шаблона - Div или Table; › **Table** – элементы отчета экспортируются с использованием HTML таблиц (значение по умолчанию); › **Div** – элементы отчета экспортируются с использованием DIV разметки; › **Span** - элементы отчета экспортируются с использованием SPAN разметки.
ParametersPanelDateFormat	Устанавливает формат даты и времени для переменных соответствующего типа на панели параметров. По умолчанию используется формат даты и времени, установленный браузером.
CloseDesignerWithoutAsking	Предоставляет возможность установить режим закрытия дизайнера отчетов без подтверждения изменений в текущем отчете. По умолчанию свойство имеет значение **false**, т.е. дизайнер будет запрашивать разрешение на сохранение изменений в текущем отчете.
ShowSystemFonts	Предоставляет возможность отображать или не отображать системные шрифты в списке шрифтов дизайнера отчетов. По умолчанию свойство имеет значение **true**, т.е. системные шрифты отображаются в списке шрифтов.
WizardTypeRunningAfterLoad	Предоставляет возможность вызывать мастер создания отчетов после запуска дизайнера отчетов. Может принимать одно из указанных ниже значений перечисления **StiWizardType**: › **None** - дизайнер отчетов будет запущен

© 2003-2026 Stimulsoft

	без вызова мастера создания отчета; › **StandardReport** - дизайнер отчетов будет запущен с вызовом стандартного мастера создания отчета; › **MasterDetailReport** - дизайнер отчетов будет запущен с вызовом master-detail мастера создания отчета; › **LabelReport** - дизайнер отчетов будет запущен с вызовом мастера создания отчета с этикетками; › **InvoicesReport** - дизайнер отчетов будет запущен с вызовом мастера создания инвойса; › **OrdersReport** - дизайнер отчетов будет запущен с вызовом мастера создания ордера; › **QuotationReport** - дизайнер отчетов будет запущен с вызовом мастера создания квоты.
AllowWordWrapTextEditors	Предоставляет возможность установить режим переноса в текстовых редакторах. По умолчанию свойство имеет значение **true**.
DefaultRibbonType	Предоставляет возможность установить используемый по умолчанию стиль панели инструментов дизайнера. Может принимать одно из указанных ниже значений перечисления: › **StiDesignerRibbonType.Classic** - Будет установлен классический стиль (значение по умолчанию); › **StiDesignerRibbonType.SingleLine** – Будет установлен компактный однострочный стиль.
AllowPropagationEvents	Предоставляет возможность разрешить распространение событий нажатия клавиш, когда дизайнер отчетов не в фокусе. По умолчанию свойство имеет значение **true**.
PropertiesPanelViewMode	Предоставляет возможность закрепить или открепить панель свойств, словарь отчета, дерево отчета. Может принимать одно из указанных ниже значений перечисления:

© 2003-2026 Stimulsoft

	› **Pinned** - панели закреплены (значение по умолчангию); › **Unpinned** - панели откреплены.

Поведение (Behavior)

Наименование	Описание
ShowSaveDialog	Включает отображение диалога ввода имени отчета при его сохранении. Имя отчета будет передано в параметрах дизайнера. По умолчанию свойство имеет значение **true**.
UndoMaxLevel	Устанавливает максимальную глубину отмены действий с отчетом (функция Undo/ Redo). Большое значение этого свойства будет потреблять больше памяти на стороне сервера для хранения параметров отмены. По умолчанию установлено значение **6**.
AllowChangeWindowTitle	Разрешает использование заголовка окна браузера для отображения имени файла редактируемого отчета. По умолчанию свойство имеет значение **true**.
SaveReportMode	Задает режим сохранения отчета. Может принимать одно из трех значений перечисления **StiSaveMode**. › **Hidden** - действие сохранения отчета вызывается в фоне при помощи AJAX-запроса и никак не отображается в окне браузера (значение по умолчанию); › **Visible** - действие сохранения отчета будет вызвано в текущем окне браузера в видимом режиме при помощи POST-запроса; › **NewWindow** - действие сохранения отчета будет вызвано в новом окне (вкладке) браузера.
SaveReportAsMode	Задает режим сохранения отчета. Может

© 2003-2026 Stimulsoft

	принимать одно из трех значений перечисления **StiSaveMode**. › **Hidden** - действие сохранения отчета вызывается в фоне при помощи AJAX-запроса и никак не отображается в окне браузера (значение по умолчанию); › **Visible** - действие сохранения отчета будет вызвано в текущем окне браузера в видимом режиме при помощи POST-запроса; › **NewWindow** - действие сохранения отчета будет вызвано в новом окне (вкладке) браузера.
CheckReportBeforePreview	Предоставляет возможность проверять или не проверять отчет встроенным инструментом перед его просмотром. По умолчанию свойство имеет значение **true**.

Главное меню (FileMenu)

Наименование	Описание
Visible	Включает отображение главного меню дизайнера отчетов. По умолчанию свойство имеет значение **true**.
ShowNew	Включает отображение пункта главного меню **Создать (New)**. По умолчанию свойство имеет значение **true**.
ShowFileMenuNewReport	Включает отображение кнопки **Пустой отчет (Blank Report)** в меню **Создать**. По умолчанию свойство имеет значение **true**.
ShowFileMenuNewDashboard	Включает отображение кнопки **Пустая панель индикаторов (Blank Dashboard)** в меню **Создать**. По умолчанию свойство имеет значение **true**.
ShowOpen	Включает отображение пункта главного меню **Открыть (Open)**. По умолчанию

© 2003-2026 Stimulsoft

	свойство имеет значение **true**.
ShowSave	Включает отображение пункта главного меню **Сохранить (Save)**. По умолчанию свойство имеет значение **true**.
ShowSaveAs	Включает отображение пункта главного меню **Сохранить как (Save As)**. По умолчанию свойство имеет значение **true**.
ShowClose	Включает отображение пункта главного меню **Закрыть (Close)**. По умолчанию свойство имеет значение **true**.
ShowExit	Включает отображение пункта главного меню **Выход (Exit)**. По умолчанию свойство имеет значение **false**.
ShowReportSetup	Включает отображение пункта главного меню **Свойства отчета (Report Setup)**. По умолчанию свойство имеет значение **true**.
ShowOptions	Включает отображение пункта главного меню **Параметры (Options)**. По умолчанию свойство имеет значение **true**.
ShowInfo	Включает отображение пункта главного меню **Сведения (Info)**. По умолчанию свойство имеет значение **true**.
ShowAbout	Включает отображение пункта главного меню **О программе (About)**. По умолчанию свойство имеет значение **true**.
ShowHelp	Включает отображение пункта главного меню **Помощь (Help)**. По умолчанию свойство имеет значение **true**.

Словарь данных (Dictionary)

Наименование	Описание
Visible	Включает отображение словаря данных отчета. По умолчанию свойство имеет

© 2003-2026 Stimulsoft

	значение **true**.
UseAliases	Предоставляет возможность использовать псевдонимы в словаре данных. Может принимать одно из трех значений перечисления **StiUseAliases**: › **Auto** - определяет режим использования псевдонимов из сохраненного значения в файлах cookies (значение по умолчанию); › **True** - устанавливает режим использования псевдонимов в словаре данных; › **False** - выключает режим использования псевдонимов в словаре данных.
NewReportDictionary	Предоставляет возможность создавать новый словарь данных или объединить с существующим, при создании нового отчета в дизайнере. Может принимать одно из трех значений перечисления **StiNewReportDictionary**: › **Auto** - определяет режим создания или объединения словаря данных из сохраненного значения в файлах cookies(значение по умолчанию); › **DictionaryNew** - устанавливает режим создания нового словаря данных, при создании нового отчета; › **DictionaryMerge** - устанавливает режим объединения существующего словаря данных с новым, при создании нового отчета в дизайнере.
ShowDictionaryContextMenuProperties	Устанавливает видимость пункта **Свойства (Properties)** в контекстном меню словаря данных. По умолчанию свойство имеет значение **true**.
ShowDictionaryActions	Устанавливает видимость меню **Действия (Actions)** на панели инструментов словаря данных. По умолчанию свойство имеет

© 2003-2026 Stimulsoft

	значение **true**.
PermissionDataConnections	Устанавливает доступные действия над подключениями к данным отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionDataSources	Устанавливает доступные действия над источниками данных отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionDataColumns	Устанавливает доступные действия над колонками данных отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionBusinessObjects	Устанавливает доступные действия над бизнес-объектами отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionDataRelations	Устанавливает доступные действия над связями данных отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionVariables	Устанавливает доступные действия над переменными отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionResources	Устанавливает доступные действия над ресурсами в словаре данных отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionSqlParameters	Устанавливает доступные действия над параметрами в SQL запросах источников данных отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionDataTransformations	Устанавливает доступные действия над преобразованием данных. Может принимать

© 2003-2026 Stimulsoft

	одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionUserFunctions	Устанавливает доступные действия над пользовательскими функциями. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.

В таблице ниже приведены все доступные значения для перечисления **StiDesignerPermissions**, которые можно установить для элементов словаря отчета.

Значение	**Описание**
None	Запрещает любое действие над элементом словаря данных.
All	Разрешает любые действия над элементом словаря данных.
Create	Разрешает создавать определенный элемент словаря данных.
Delete	Разрешает удалять определенный элемент словаря данных.
Modify	Разрешает редактировать определенный элемент словаря данных.
View	Разрешает просматривать определенный элемент словаря данных.
ModifyView	Разрешает редактировать и просматривать определенный элемент словаря данных.

Панель инструментов (Toolbar)

Наименование	**Описание**
ShowToolbar	Предоставляет возможность отображать или не отображать панель инструментов в дизайнере отчетов. По умолчанию свойство

© 2003-2026 Stimulsoft

	имеет значение **true**.
ShowSetupToolboxButton	Включает отображение кнопки вызова окна настроек боковой панели компонентов отчета. По умолчанию свойство имеет значение **true**.
ShowInsertButton	Включает отображение вкладки **Вставить (Insert)** на панели инструментов дизайнера. По умолчанию свойство имеет значение **true**.
ShowLayoutButton	Включает отображение вкладки **Формат (Layout)** на панели инструментов дизайнера. По умолчанию свойство имеет значение **true**.
ShowPageButton	Включает отображение вкладки **Страница (Page)** на панели инструментов дизайнера. По умолчанию свойство имеет значение **true**.
ShowPreviewButton	Включает отображение вкладки **Предварительный просмотр (Preview)** на панели инструментов дизайнера. По умолчанию свойство имеет значение **true**.
ShowSaveButton	Включает отображение кнопки **Сохранить (Save)** на панели инструментов дизайнера. По умолчанию свойство имеет значение **true**.
ShowAboutButton	Включает отображение кнопки **О программе (About)** на панели инструментов дизайнера. По умолчанию свойство имеет значение **false**.

Панель свойств (PropertiesGrid)

Наименование	Описание
Visible	Включает отображение панели свойств компонентов отчета. По умолчанию свойство имеет значение **true**.
Width	Задает ширину панели свойств компонентов отчета. По умолчанию установлена ширина

© 2003-2026 Stimulsoft

	370 точек.
LabelWidth	Задает ширину надписей панели свойств компонентов отчета. По умолчанию установлена ширина **160** точек.
PropertiesGridPosition	Предоставляет возможность определить позицию панели свойств в дизайнере отчетов. Может принимать одно из трех значений перечисления **StiPropertiesGridPosition**. › **Left** - панель свойств будет располагаться слева; › **Right** - панель свойств будет располагаться справа.
ShowPropertiesWhichUsedFromStyles	Устанавливает видимость свойств, которые используются из стилей в дизайнере. По умолчанию свойство имеет значение **false**.

Компоненты отчета (Components)

Наименование	Описание
ShowText	Включает отображение компонента **Текст (Text)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowTextInCells	Включает отображение компонента **Текст в ячейках (Text in Cells)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowRichText	Включает отображение компонента **Rich текст (Rich Text)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowImage	Включает отображение компонента **Картинка (Image)** в меню вставки

© 2003-2026 Stimulsoft

	компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowBarCode	Включает отображение компонента **Штрих-код (Bar Code)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowShape	Включает отображение компонента **Геометрия (Shape)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowHorizontalLinePrimitive	Включает отображение компонента **Горизонтальная линия (Horizontal Line)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowVerticalLinePrimitive	Включает отображение компонента **Вертикальная линия (Vertical Line)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowRectanglePrimitive	Включает отображение компонента **Прямоугольник (Rectangle)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowRoundedRectanglePrimitive	Включает отображение компонента **Закругленный прямоугольник (Rounded Rectangle)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowPanel	Включает отображение компонента **Панель (Panel)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowClone	Включает отображение компонента **Клон (Clone)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowCheckBox	Включает отображение компонента **Флажок**

© 2003-2026 Stimulsoft

	(Check Box) в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowSubReport	Включает отображение компонента **Вложенный отчет (Sub Report)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowZipCode	Включает отображение компонента **Почтовый индекс (Zip Code)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowTable	Включает отображение компонента **Таблица (Table)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowCrossTab	Включает отображение компонента **Кросс-таблица (Cross-Tab)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowChart	Включает отображение компонента **Диаграмма (Chart)** в меню вставки компонентов дизайнера. Действует сразу на все типы диаграмм. По умолчанию свойство имеет значение **true**.
ShowMap	Включает отображение компонента **Карта (Map)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **false**.
ShowGauge	Включает отображение компонента **Индикатор (Gauge)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **false**.
ShowSparkline	Включает отображение компонента **Спарклайн (Sparkline)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.

© 2003-2026 Stimulsoft

ShowMathFormula	Включает отображение компонента **Математическая формула (Math Formula)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowElectronicSignature	Включает отображение компонента **Электронная подпись (Electronic Signature)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowPdfDigitalSignature	Включает отображение компонента **Цифровая подпись PDF (PDF Digital Signature)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.

Секции (Bands)

Наименование	Описание
ShowReportTitleBand	Включает отображение секции **Заголовок отчета (Report Title)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowReportSummaryBand	Включает отображение секции **Итог отчета (Report Summary)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowPageHeaderBand	Включает отображение секции **Заголовок страницы (Page Header)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowPageFooterBand	Включает отображение секции **Итог страницы (Page Footer)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowGroupHeaderBand	Включает отображение секции **Заголовок группы (Group Header)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.

© 2003-2026 Stimulsoft

ShowGroupFooterBand	Включает отображение секции **Итог группы (Group Footer)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowHeaderBand	Включает отображение секции **Заголовок данных (Header)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowFooterBand	Включает отображение секции **Итог данных (Footer)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowColumnHeaderBand	Включает отображение секции **Заголовок колонки (Column Header)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowColumnFooterBand	Включает отображение секции **Итог колонки (Column Footer)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowDataBand	Включает отображение секции **Данные (Data)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowHierarchicalBand	Включает отображение секции **Дерево (Hierarchical)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowChildBand	Включает отображение секции **Подчиненный (Child)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowEmptyBand	Включает отображение секции **Пустые данные (Empty)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowOverlayBand	Включает отображение секции **Водяной знак (Overlay)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowTableOfContents	Включает отображение пункт **Оглавление (Table Of Contents)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.

© 2003-2026 Stimulsoft

Элементы дашборда (DashboardElements)

Наименование	Описание
ShowTableElement	Включает отображение элемента панели индикаторов **Таблица (Table)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowCardsElement	Включает отображение элемента панели индикаторов **Карточки (Cards)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowChartElement	Включает отображение элемента панели индикаторов **Диаграмма (Chart)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowGaugeElement	Включает отображение элемента панели индикаторов **Измерительный прибор (Gauge)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowPivotTableElement	Включает отображение элемента панели индикаторов **Сводная таблица (Pivot)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowIndicatorElement	Включает отображение элемента панели индикаторов **Индикатор (Indicator)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowProgressElement	Включает отображение элемента панели

© 2003-2026 Stimulsoft

	индикаторов **Прогресс (Progress)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowRegionMapElement	Включает отображение элемента панели индикаторов **Региональная карта (Region Map)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowOnlineMapElement	Включает отображение элемента панели индикаторов **Онлайн карта (Online Map)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowImageElement	Включает отображение элемента панели индикаторов **Изображение (Image)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowTextElement	Включает отображение элемента панели индикаторов **Текст (Text)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowPanelElement	Включает отображение элемента панели индикаторов **Панель (Panel)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowShapeElement	Включает отображение элемента панели индикаторов **Геометрия (Shape)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowButtonElement	Включает отображение элемента панели индикаторов **Кнопка (Button)** на

© 2003-2026 Stimulsoft

	инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowListBoxElement	Включает отображение элемента панели индикаторов **Список (ListBox)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowComboBoxElement	Включает отображение элемента панели индикаторов **Выпадающий список (ComboBox)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowTreeViewElement	Включает отображение элемента панели индикаторов **Иерархический список (TreeView)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowTreeViewBoxElement	Включает отображение элемента панели индикаторов **Выпадающий иерархический список (TreeViewBox)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowDatePickerElement	Включает отображение элемента панели индикаторов **Выбор даты (Date Picker)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.

Кросс-секции (CrossBands)

Наименование	Описание
ShowCrossGroupHeaderBand	Включает отображение секции **Кросс-заголовок группы (Cross Group Header)** в

© 2003-2026 Stimulsoft

	меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowCrossGroupFooterBand	Включает отображение секции **Кросс-итог группы (Cross Group Footer)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowCrossHeaderBand	Включает отображение секции **Кросс-заголовок данных (Cross Header)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowCrossFooterBand	Включает отображение секции **Кросс-итог данных (Cross Footer)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowCrossDataBand	Включает отображение секции **Кросс-данные (Cross Data)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.

Панели индикаторов(Dashboards)

Наименование	Описание
ShowNewDashboardButton	Предоставляет возможность отображать или не отображать кнопку **Новая панель индикаторов (New Dashboard)** в дизайнере отчетов. По умолчанию свойство имеет значение **true**.

Страницы(Pages)

Наименование	Описание
ShowNewPageButton	Предоставляет возможность отображать или не отображать кнопку **Новая страница (New Page)** в дизайнере отчетов. По умолчанию

© 2003-2026 Stimulsoft

	свойство имеет значение **true**.

При разработке отчета или дашборда в дизайнере отчетов, можно также определить настройки экспорта (ExportOptions), параметры Email (EmailOptions) и панель инструментов (PreviewToolbarOptions) на вкладке **Просмотр (Preview)**. Эти опции аналогичны опциям вьювера отчетов.

6 Reports and Dashboards for ASP.NET Core Razor

.NET Core представляет собою кросс-платформенную технологию создания Web приложений для Windows, Linux и macOS. **ASP.NET Core Razor Pages** представляют собой технологию, альтернативную системе Model-View-Controller (MVC). **Razor Pages** позволяют создавать страницы с кодом, которые могут обрабатывать различные запросы. Данная технология делает создание сценариев для страниц проще и эффективнее по сравнению с использованием MVC решения. Компания Stimulsoft предоставляет инструменты создания, отображения, преобразования отчетов и панелей индикаторов (дашбордов) с использованием этой технологии.

Инструменты создания и редактирования отчетов:	Инструменты отображения и конвертации отчетов:	Инструменты создания и редактирования дашбордов:	Инструменты отображения и конвертации дашбордов:
HTML5 дизайнер	HTML5 вьювер	HTML5 дизайнер	HTML5 вьювер

6.1 HTML5 вьювер

Примеры

Ознакомьтесь с примерами по работе с компонентом ASP.NET Core Razor HTML5 Viewer на GitHub. Все примеры представляют собой отдельные проекты, сгруппированные в один solution для Visual Studio.

© 2003-2026 Stimulsoft

Компонент **HTML5 Viewer (StiNetCoreViewer)** предназначен для просмотра отчетов в окне браузера. При этом не требуется устанавливать у клиента .NET Framework, ActiveX компоненты или какие-либо специальные плагины. Все, что необходимо – это любой современный Web-браузер.

При помощи **HTML5 Viewer** можно просматривать, печатать отчеты, выполнять экспортирование отчетов на любом компьютере с любой установленной операционной системой. Так как вьювер использует только HTML и JavaScript технологии, он может быть запущен на устройствах, где нет поддержки Flash или Silverlight – планшеты, смартфоны. Также вьювер поддерживает Mobile и Touch интерфейсы, которые автоматически включаются при использовании мобильных устройств и мониторов с сенсорным экраном.

Компонент **HTML5 Viewer** использует технологию AJAX для выполнения всех действий (загрузки отчета, листания страниц, масштабирования, интерактивности в отчетах и др.), что позволяет избавиться от перезагрузки всей страницы, а также экономить Web-трафик и повысить скорость работы. Для построения отчетов используется ядро генератора отчетов, построенное с использованием .NET Core технологии. Данная технология является кроссплатформенной, что позволяет разворачивать приложение на серверах, использующих операционные системы Windows, macOS и Linux.

HTML5 Viewer поддерживает множество тем оформления, анимированный интерфейс, закладки, интерактивные отчеты, редактирование элементов отчета на странице, полноэкранный режим, поиск, и другие, необходимые для просмотра отчетов, возможности.

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для использования **HTML5 Viewer** в Web-проекте, необходимо установить NuGet пакет Stimulsoft.Reports.Web.NetCore:

© 2003-2026 Stimulsoft

› Выберите пункт "Manage NuGet Packages..." в контекстном меню проекта;
› На вкладке Browse, в строке поиска, укажите Stimulsoft.Reports.Web.NetCore;
› Выделите элемент, определите версию пакета и нажмите кнопку **Установить (Install)**. При обновлении пакета, следует нажать кнопку **Обновить (Update)**.

Для того, чтобы добавить возможности просмотра и экспорта дашбордов в Web-проекте, установить NuGet пакет Stimulsoft.Dashboards.Web.NetCore, (данный пакет связан с пакетом Stimulsoft.Reports.Web.NetCore, при его отсутствии он будет установлен автоматически):
› Выберите пункт "Manage NuGet Packages..." в контекстном меню проекта;
› На вкладке Browse, в строке поиска, укажите Stimulsoft.Dashboards.Web.NetCore;
› Выделите элемент, определите версию пакета и нажмите кнопку **Установить (Install)**. При обновлении пакета, следует нажать кнопку **Обновить (Update)**.

- Как это работает
- Активация
- Отображение отчета и дашборда
- Подключение данных
- Локализация вьювера
- Печать отчета
- Экспорт отчета и дашборда
- Режимы отображения
- Работам с параметрами
- Настройки вьювера
- Интерактивные отчеты
- Время ожидания
- Редактирование построенного отчета
- Отправка отчета по Email
- Вызов дизайнера из вьювера
- Использование тем
- Работа с закладками
- Вспомогательные методы
- Экспорт и печать из кода

© 2003-2026 Stimulsoft

6.1.1 Как это работает

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для запуска вьювера требуется разместить на странице компонент **StiNetCoreViewer**, задать ему необходимые свойства, а в обработчике событий страницы определить необходимые действия. При запуске вьювера отчетов происходят следующие действия:

- .NET Core компонент генерирует HTML и JavaScript код, необходимый для отображения и работы вьювера;
- По окончании вывода компонента запускается JavaScript метод, который запрашивает на стороне сервера первую страницу отчета, весь отчет (в зависимости от выбранного режима) и его необходимые параметры;
- Каждое действие во вьювере (например, листание страниц, печать отчета, экспорт отчета и др.) вызывает определенное действие на стороне сервера, в котором можно выполнить необходимые манипуляции с отчетом;
- Для ускорения работы вьювер сохраняет отчет в кэше или сессии сервера, что позволяет исключить повторное построение отчета.

6.1.2 Активация

После приобретения продукта Stimulsoft вам необходимо активировать лицензию для используемых компонентов. Это можно выполнить указав лицензионный ключ или загрузив файл с лицензионным ключом. Ниже представлен пример активации компонента **StiNetCoreViewer**.

© 2003-2026 Stimulsoft

Index.cshtml.cs

```
...
//Activation with using license code
public class IndexModel : PageModel
{
   static IndexModel()
   {
      Stimulsoft.Base.StiLicense.Key = "Your activation code...";
   }
}

//Activation with using license file
public class IndexModel : PageModel
{
   public IndexModel(IWebHostEnvironment webHostEnvironment)
   {
      var path = Path.Combine(webHostEnvironment.ContentRootPath, "Content\
      \license.key");
      Stimulsoft.Base.StiLicense.LoadFromFile(path);
   }
}
...
```

Получить лицензионный ключ или скачать файл с лицензионным ключом, вы можете в личном кабинете пользователя. Для авторизации в кабинете используйте имя пользователя и пароль, указанные при покупке подписки продукта.

6.1.3 Отображение отчета и дашборда

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

© 2003-2026 Stimulsoft

Примечание

При присвоении отчета вьювер компоненту, происходит автоматическое построение отчета. Вызов метода `Report.Render()` необходим только в том случае, если требуется провести какие-либо действия с построенным отчетом перед его отображением во вьювере. Аналогично, при использовании режима компиляции, вызов метода `Report.Compile()` требуется только в том случае, если требуется провести какие-либо действия с откомпилированным отчетом перед его построением и отображением во вьювере.

Для отображения отчета необходимо добавить на страницу компонент **StiNetCoreViewer** и задать ему минимально необходимые свойства, а в обработчике событий страницы определить необходимые действия.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Actions =
   {
      GetReport = "GetReport",
      ViewerEvent = "ViewerEvent"
   }
})
...
```

Index.cshtml.cs

```
...
public IActionResult OnPostGetReport()
{
   // Create the report object
   StiReport report = new StiReport();

   // Load report or dashboard
   report.Load(StiNetCoreHelper.MapPath(this, "Reports/SimpleList.mrt"));
   //report.Load(StiNetCoreHelper.MapPath(this, "Reports/Dashboard.mrt"));

   return StiNetCoreViewer.GetReportResult(this, report);
}

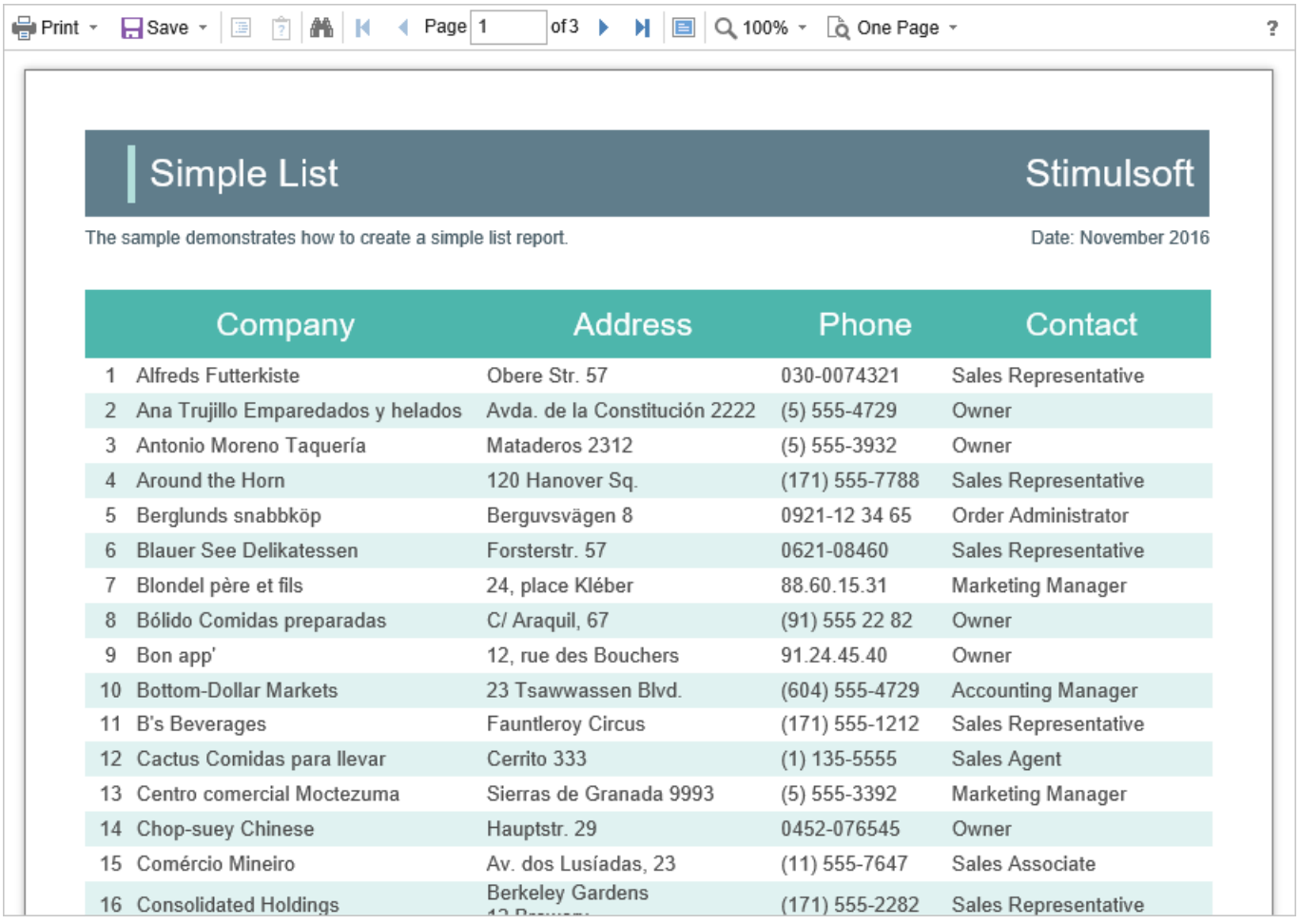
public IActionResult OnGetViewerEvent()
{
   return StiNetCoreViewer.ViewerEventResult(this);
}
```

© 2003-2026 Stimulsoft

```
public IActionResult OnPostViewerEvent()
{
   return StiNetCoreViewer.ViewerEventResult(this);
}
...
```

Информация

Действие **ViewerEvent** является обязательным. Без него корректная работа вьювера невозможна. Действие вызывается для двух типов запросов: **OnGet** - компонент запрашивает необходимые для работы ресурсы, такие как CSS-стили, JS-скрипты и изображения; **OnPost** - все остальные действия вьювера.



3

© 2003-2026 Stimulsoft

Если отчет не был построен перед отображением, то компонент **HTML5 Viewer** построит его автоматически. Таким образом, для отображения отчета разрешено использовать шаблоны отчетов и построенные отчеты.

Index.cshtml.cs

```
...
public IActionResult OnPostGetReport()
{
   StiReport report = new StiReport();
   report.LoadDocument(StiNetCoreHelper.MapPath(this, "Reports/
   SimpleList.mdc"));

   return StiNetCoreViewer.GetReportResult(this, report);
}
...
```

Так как дашборд не является статическим документов и для его работы требуются данные, то формат построенного документа MDC для него не доступен. Вместо этого предоставлена возможность использовать снимок отчета в формате MRT, который содержит в себе все необходимые для работы дашборда данные, и может быть корректно отображен во вьювере.

Index.cshtml.cs

```
...
public ActionResult OnPostGetReport()
{
   StiReport report = new StiReport();
   report.Load(StiNetCoreHelper.MapPath("~/Reports/Snapshot.mrt"));

   return StiNetCoreViewer.GetReportResult(report);
}
...
```

Подключение пользовательских шрифтов

Вы можете подключить пользовательские шрифты с использованием класса **StiFontCollection**, указав файл содержащий шрифт. Для этого необходимо в конструкторе страницы вызвать статический метод для загрузки шрифта.

© 2003-2026 Stimulsoft

Index.cshtml.cs

```
...
public class IndexModel : PageModel
{
   static IndexModel()
   {
      Stimulsoft.Base.StiFontCollection.AddFontFile(StiNetCoreHelper.MapPat
      h(this,"/fonts/my-font/font-name.ttf"));
   }
}
...
```

6.1.4 Подключение данных

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Данные для построения отчета могут быть подключены различными способами. Самый простой – хранить настройки подключения в самом шаблоне отчета. Так же данные можно подключить из кода, это можно сделать при загрузке отчета в действии **GetReport**.

Index.cshtml.cs

```
...
public IActionResult OnPostGetReport()
{
   DataSet ds = new DataSet();
   ds.ReadXml(StiNetCoreHelper.MapPath(this, "Data/Demo.xml"));

   StiReport report = new StiReport();
   report.Load(StiNetCoreHelper.MapPath(this, "Reports/
   TwoSimpleLists.mrt"));
   report.Dictionary.Databases.Clear();
   report.RegData("Demo", ds);

   return StiNetCoreViewer.GetReportResult(this, report);
}
...
```

© 2003-2026 Stimulsoft

Данные для отчета можно подключать не только при загрузке отчета. Например, можно подключить новые данные в момент интерактивных действий во вьювере (применение параметров отчета, сортировка, детализация, сворачивание). Для этого необходимо задать действие **Interaction** для компонента **HTML5 Viewer**, а в обработчике действия подключить данные для текущего отчета. Таким же способом можно подключить данные в других действиях вьювера.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Actions =
   {
      GetReport = "GetReport",
      ViewerEvent = "ViewerEvent",
      Interaction = "ViewerInteraction"
   }
   })
...
```

Index.cshtml.cs

```
...
public IActionResult OnPostViewerInteraction()
{
   DataSet data = new DataSet();
   data.ReadXml(StiNetCoreHelper.MapPath(this, "Data/Demo.xml"));

   StiReport report = StiNetCoreViewer.GetReportObject(this);
   report.RegData("Demo", data);

   return StiNetCoreViewer.InteractionResult(this, report);
}
...
```

При необходимости подключить новые данные только для определенного интерактивного действия вьювера, например, только при применении параметров отчета, можно воспользоваться параметрами вьювера. Параметры вьювера представлены в виде объекта класса **StiRequestParams**, они передаются при любом запросе на сторону сервера, и содержат все необходимые сведения и состояния клиентской части вьювера. Для определения типа действия вьювера достаточно проверить свойство **Action** у параметров вьювера.

© 2003-2026 Stimulsoft

Index.cshtml.cs

```
...
public IActionResult OnPostViewerInteraction()
{
   StiRequestParams requestParams =
   StiNetCoreViewer.GetRequestParams(this);
   if (requestParams.Action == StiAction.Variables)
   {
      DataSet data = new DataSet();
      data.ReadXml(StiNetCoreHelper.MapPath(this, "Data/Demo.xml"));

      StiReport report = StiNetCoreViewer.GetReportObject(this);
      report.RegData("Demo", data);

      return StiNetCoreViewer.InteractionResult(this, report);
   }

   return StiNetCoreViewer.InteractionResult(this);
}
...
```

SQL источники данных

Параметры подключения к SQL источнику данных, как и к любому другому, можно хранить в самом шаблоне отчета. Если требуется задать параметры подключения из кода перед построением отчета (например, по причине безопасности или в зависимости от авторизованного пользователя), можно воспользоваться приведенным ниже примером.

© 2003-2026 Stimulsoft

Index.cshtml.cs

```
...
public IActionResult OnPostGetReport()
{
   OracleConnection connection = new OracleConnection("Data
   Source=Oracle8i;Integrated Security=yes");
   connection.Open();
   OracleDataAdapter adapter = new OracleDataAdapter();
   adapter.SelectCommand = new OracleCommand("SELECT * FROM Products",
   connection);

   DataSet dataSet = new DataSet("productsDataSet");
   adapter.Fill(dataSet, "Products");

   StiReport report = new StiReport();
   report.Load(StiNetCoreHelper.MapPath(this, "Reports/
   SqlSampleReport.mrt"));
   report.RegData("Products", dataSet);

   return StiNetCoreViewer.GetReportResult(this, report);
}
...
```

Также, для SQL источников данных используемых в отчете, можно указать **Время ожидания запроса (Query Timeout)** в секундах. Значение данного свойства сохраняется в самом шаблоне отчёта для каждого SQL подключения отдельно.

Ниже приведен пример кода, который предоставляет возможность изменить строку соединения для MS SQL, скорректировать запрос, установить время ожидания запроса для уже созданного соединения и источников данных в отчете.

© 2003-2026 Stimulsoft

Index.cshtml.cs

```
...
public IActionResult OnPostGetReport()
{

   StiReport report = new StiReport();
   report.Load(Server.MapPath("Report.mrt"));
   ((StiSqlDatabase)
   report.Dictionary.Databases["Connection"]).ConnectionString = @"Data
   Source=server;Integrated Security=True;Initial Catalog=DataBase";
   ((StiSqlSource)
   report.Dictionary.DataSources["DataSourceName"]).SqlCommand = "select *
   from Table where Column = 100";
   ((StiSqlSource)
   report.Dictionary.DataSources["DataSourceName"]).CommandTimeout = 1000;

   return StiNetCoreViewer.GetReportResult(this, report);
}
...
```

Информация

Для SQL источников данных других типов подключение создается аналогично, при этом подключается адаптер, соответствующий типу источника данных. Например, для MS SQL источника необходимо подключать SqlDataAdapter, для Oracle необходим OracleDataAdapter. Также, при этом следует указать строку соединения, соответствующую типу подключения.

Вы также можете использовать данные для разработки отчетов и дашбордов полученные из OData хранилищ данных. В этом случае, авторизация может быть выполнена с использованием имени, пароля пользователя или с использованием токена. Параметры авторизации указываются в строке соединения к OData хранилищу с использованием разделителя ";".

© 2003-2026 Stimulsoft

Index.cshtml.cs

```
...
public IActionResult OnPostGetReport()
{

   StiReport report = new StiReport();

   //Authorization using a user account
   var oDataDatabase = new StiODataDatabase("OData", "OData", @"https://
   services.odata.org/V4/Northwind/
   Northwind.svc;AddressBearer=adress;UserName=UserName;Password=Password;C
   lient_Id=Your Client ID", false, null);

   //Authorization using a user token
   var oDataDatabase = new StiODataDatabase("OData", "OData", @"https://
   services.odata.org/V4/Northwind/Northwind.svc;Token=Enter your token",
   false, null);

   report.Dictionary.Databases.Add(oDataDatabase);
   oDataDatabase.Synchronize(report);

   //Query with data filter
   ((StiSqlSource)report.Dictionary.DataSources["Products"]).SqlCommand =
   "Products?$filter=ProductID eq 2";

   return StiNetCoreViewer.GetReportResult(this, report);
}
...
```

В таблице ниже представлены шаблоны строк подключения для различных типов источников данных.

Источник данных	Шаблон строки соединения
MS SQL	Integrated Security=False; Data Source=myServerAddress;Initial Catalog=myDataBase; User ID=myUsername; Password=myPassword;
MySQL	Server=myServerAddress; Database=myDataBase;UserId=myUsername; Pwd=myPassword;
ODBC	Driver={SQL Server}; Server=myServerAddress;Database=myDataBase ; Uid=myUsername; Pwd=myPassword;
OLE DB	Provider=SQLOLEDB.1; Integrated

© 2003-2026 Stimulsoft

	Security=SSPI;Persist Security Info=False; Initial Catalog=myDataBase;Data Source=myServerAddress
Oracle	Data Source=TORCL;User Id=myUsername;Password=myPassword;
MS Access	Provider=Microsoft.Jet.OLEDB.4.0;User ID=Admin;Password=pass;Data Source=C:\ \myAccessFile.accdb;
PostgreSQL	Server=myServerAddress; Port=5432; Database=myDataBase;User Id=myUsername; Password=myPassword;
Firebird	User=SYSDBA; Password=masterkey; Database=SampleDatabase.fdb;DataSource=my ServerAddress; Port=3050; Dialect=3; Charset=NONE;Role=; Connection lifetime=15; Pooling=true; MinPoolSize=0;MaxPoolSize=50; Packet Size=8192; ServerType=0;
SQL CE	Data Source=c:\MyData.sdf; Persist Security Info=False;
SQLite	Data Source=c:\mydb.db; Version=3;
DB2	Server=myAddress:myPortNumber;Database=m yDataBase;UID=myUsername;PWD=myPassword; Max Pool Size=100;Min Pool Size=10;
Infomix	Database=myDataBase;Host=192.168.10.10;Serv er=db_engine_tcp;Service=1492;Protocol=onsoc tcp;UID=myUsername;Password=myPassword;
Sybase	Data Source=myASEserver;Port=5000;Database=myD ataBase;Uid=myUsername;Pwd=myPassword;
Teradata	Data Source=myServerAddress;User ID=myUsername;Password=myPassword;
VistaDB	Data Source=D:\folder \myVistaDatabaseFile.vdb4;Open Mode=ExclusiveReadWrite;
Universal(dotConnect)	Provider=Oracle;direct=true;data

© 2003-2026 Stimulsoft

	source=192.168.0.1;port=1521;sid=sid;user=user;password=pass
MongoDB	mongodb://<user>:<password>@localhost/test
OData	http://services.odata.org/v3/odata/OData.svc/

Информация

В таблице представлены наиболее часто используемые шаблоны для строки соединения. Посмотреть различные варианты строк соединения вы можете на специализированном сайте.

Данные из XML, JSON, Excel файлов

Подключение к XML и JSON источникам данных можно хранить в шаблоне отчета. Если требуется указать файлы данных из кода, можно воспользоваться приведенным ниже примером.

Index.cshtml.cs

```
...
public IActionResult OnPostGetReport()
{
   DataSet data = new DataSet();
   data.ReadXml(StiNetCoreHelper.MapPath(this, "Data/Demo.xml"));

   StiReport report = new StiReport();
   report.Load(StiNetCoreHelper.MapPath(this, "Reports/SimpleList.mrt"));
   report.RegData(data);

   return StiNetCoreViewer.GetReportResult(this, report);
}
...
```

© 2003-2026 Stimulsoft

Index.cshtml.cs

```
...
public IActionResult OnPostGetReport()
{
   DataSet data
   = StiJsonToDataSetConverterV2.GetDataSetFromFile(StiNetCoreHelper.MapPat
   h(this, "Data/Demo.json"));

   StiReport report = new StiReport();
   report.Load(StiNetCoreHelper.MapPath(this, "Reports/SimpleList.mrt"));
   report.RegData(data);

   return StiNetCoreViewer.GetReportResult(this, report);
}
...
```

Информация

Во вьювере присутствует возможность получения данных из Excel файла. Для этого можно воспользоваться указанным ниже методом.

```
DataSet dataSet = StiExcelConnector.Get().GetDataSet(new
StiExcelOptions(array, this.FirstRowIsHeader));
```

6.1.5 Локализация

Компонент **HTML5 Viewer** поддерживает полную локализацию своего интерфейса. Для того чтобы локализовать интерфейс вьювера отчетов на необходимый язык, используется специальное свойство **Localization**. В качестве значения данного свойства необходимо указать путь к XML-файлу локализации (относительный либо абсолютный).

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Localization = "Localization/en.xml"
})
...
```

При загрузке вьювера отчетов файл локализации будет загружен автоматически.

© 2003-2026 Stimulsoft

6.1.6 Использование тем

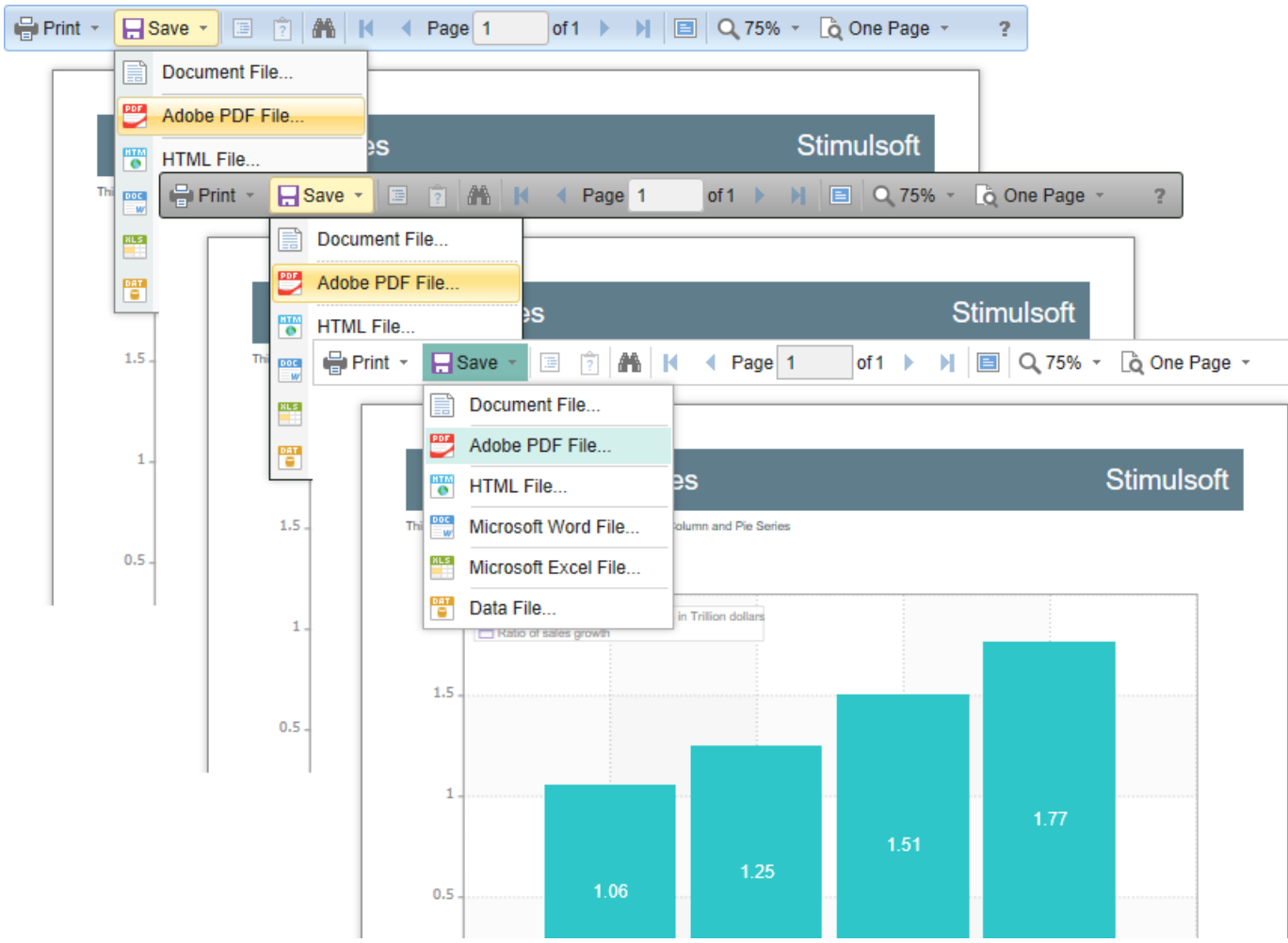
Компонент **HTML5 Viewer** имеет возможность смены тем оформления визуальных элементов управления. Для изменения темы используется свойство **Theme**, которое может принимать одно из значений перечисления **StiViewerTheme**.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Theme = StiViewerTheme.Office2022WhiteTeal
})
...
```

На данный момент доступно **8 тем** оформления с различными цветовыми акцентами. В результате, доступно более **60** вариантов оформления. Это позволяет настроить внешний вид вьювера практически под любое оформление Web проекта.

© 2003-2026 Stimulsoft



По умолчанию вьювер имеет только верхнюю панель инструментов, на которой находятся все элементы управления отчетом. При необходимости, панель инструментов можно разделить на верхнюю и нижнюю. На верхней панели будут находиться меню печати и экспорта отчета, а также кнопки работы с параметрами и закладками. Нижняя панель инструментов будет содержать элементы переключения между страницами отчета и меню управления масштабом. Для включения указанного режима предназначено свойство **DisplayMode**, которое может иметь значение **Simple** (стандартный простой режим) и **Separated** (раздельный режим).

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Appearance =
   {
      ScrollbarsMode = true
   },
   Toolbar =
   {
      DisplayMode = StiToolbarDisplayMode.Separated
   }
})
...
```

Дополнительно предусмотрена возможность установки параметров оформления основных элементов вьювера. Например, можно изменить шрифт и цвет надписей панели управления вьювера, установить фон вьювера, задать цвет границ страницы и др. Ниже представлен список доступных свойств, изменяющих оформление вьювера, и их значения по умолчанию.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Appearance =
   {
      BackgroundColor = Color.White,
      PageBorderColor = Color.Blue,
      ShowPageShadow = true
   },
   Toolbar =
   {
      BackgroundColor = Color.White,
      BorderColor = Color.Gray,
      FontColor = Color.Black,
      FontFamily = "Arial"
   }
})
...
```

© 2003-2026 Stimulsoft

6.1.7 Основные возможности

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

К основным возможностям вьювера можно отнести следующие операции: переключение между страницами отчета, изменение масштаба и режима отображения отчета, печать отчета, экспортирование отчета, применение параметров отчета, сортировку, сворачивание и детализацию. Все указанные операции выполняются в AJAX-режиме без перезагрузки страницы браузера. Для корректной работы этих операций необходимо определить специальное действие **ViewerEvent**.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Actions =
   {
      ViewerEvent = "ViewerEvent"
   }
})
...
```

© 2003-2026 Stimulsoft

Index.cshtml.cs

```
...
public IActionResult OnGetViewerEvent()
{
   // Some code before loading the viewer resources
   // ...

   return StiNetCoreViewer.ViewerEventResult(this);
}

public IActionResult OnPostViewerEvent()
{
   // Some code before the viewer actions
   // ...

   return StiNetCoreViewer.ViewerEventResult(this);
}
...
```

Информация

Действие **ViewerEvent** является обязательным. Без него корректная работа вьювера невозможна. Действие вызывается для двух типов запросов: **OnGet** - компонент запрашивает необходимые для работы ресурсы, такие как CSS-стили, JS-скрипты и изображения; **OnPost** - все остальные действия вьювера.

Действие **ViewerEvent** возвращает подготовленную HTML страницу отчета (или набор страниц), построенную с учетом текущего состояния вьювера. При необходимости в указанном действии можно изменить параметры текущего отчета, а также обновить данные отчета в случае интерактивных действий вьювера.

6.1.8 Печать отчета

Информация

Обратите внимание, что возможность печати доступна только для отчетов, и недоступна для дашбордов.

В компоненте **HTML5 Viewer** предусмотрено несколько вариантов печати

© 2003-2026 Stimulsoft

отчета. Каждый имеет свои особенности, достоинства и недостатки.

Печать в PDF (Print to PDF)
Печать будет осуществляться посредством экспортирования отчета в **PDF формат**. К достоинствам можно отнести большую точность расположения и печати элементов отчета по сравнению с остальными вариантами печати. Из недостатков можно упомянуть обязательное наличие установленного в браузере плагина для просмотра PDF файлов (современные браузеры имеют встроенное средство просмотра и печати PDF файлов).

Печать с просмотром (Print with Preview)
Печать отчета будет осуществлена в отдельном всплывающем окне браузера в **HTML формат**. Отчет можно предварительно просмотреть, а затем отправить на принтер или скопировать в другое место в виде текста или HTML-кода. К достоинствам можно отнести кроссбраузерность при печати, отсутствие необходимости установки специальных плагинов. Недостатком является относительно невысокая точность расположения элементов отчета, обусловленная особенностями реализации HTML-форматирования.

Печать без просмотра (Print without Preview)
Печать отчета будет осуществлена непосредственно на принтер без предварительного просмотра. После выбора этого пункта меню отображается системный диалог печати. Так как печать в этом режиме осуществляется в HTML формат, то качество печати аналогично качеству печати отчета с предварительным просмотром.

Информация

При печати в **HTML формат** необходимо убедиться в соответствии параметров страницы отчета и параметров страницы принтера (размер бумаги, ориентация, поля, отступы), а также проверить настройки печати браузера, такие как отступы, колонтитулы, печать фоновых изображений, цветная печать.

Для работы функции печати не требуется дополнительных настроек вьювера. Если необходимо выполнить какие-либо действия перед печатью отчета, то можно определить специальное действие **PrintReport**.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Actions =
   {
      PrintReport = "PrintReport"
   }
})
...
```

Index.cshtml.cs

```
...
public IActionResult OnPostPrintReport()
{
   // Some code before print
   // ...

   return StiNetCoreViewer.PrintReportResult(this);
}
...
```

Настройки печати

При выборе печати отчета на панели вьювера, отображается меню с выбором варианта печати. Компонент **HTML5 Viewer** имеет возможность принудительно установить требуемый режим печати. Для этого достаточно установить свойство **PrintDestination** в одно из указанных ниже значений перечисления **StiPrintDestination**.

- **Default** – при выборе печати будет отображено меню (значение свойства по умолчанию);
- **Pdf** – печать в PDF формат;
- **Direct** – печать в HTML формат непосредственно на принтер, будет отображен системный диалог печати;
- **WithPreview** – печать в HTML формат с предварительным просмотром во всплывающем окне.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Toolbar =
   {
      PrintDestination = StiPrintDestination.Default
   }
})
...
```

Компонент **HTML5 Viewer** имеет возможность полностью отключить печать отчета, если она не требуется. Для этого необходимо установить значение **false** для свойства **ShowPrintButton**.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Toolbar =
   {
      ShowPrintButton = false
   }
})
...
```

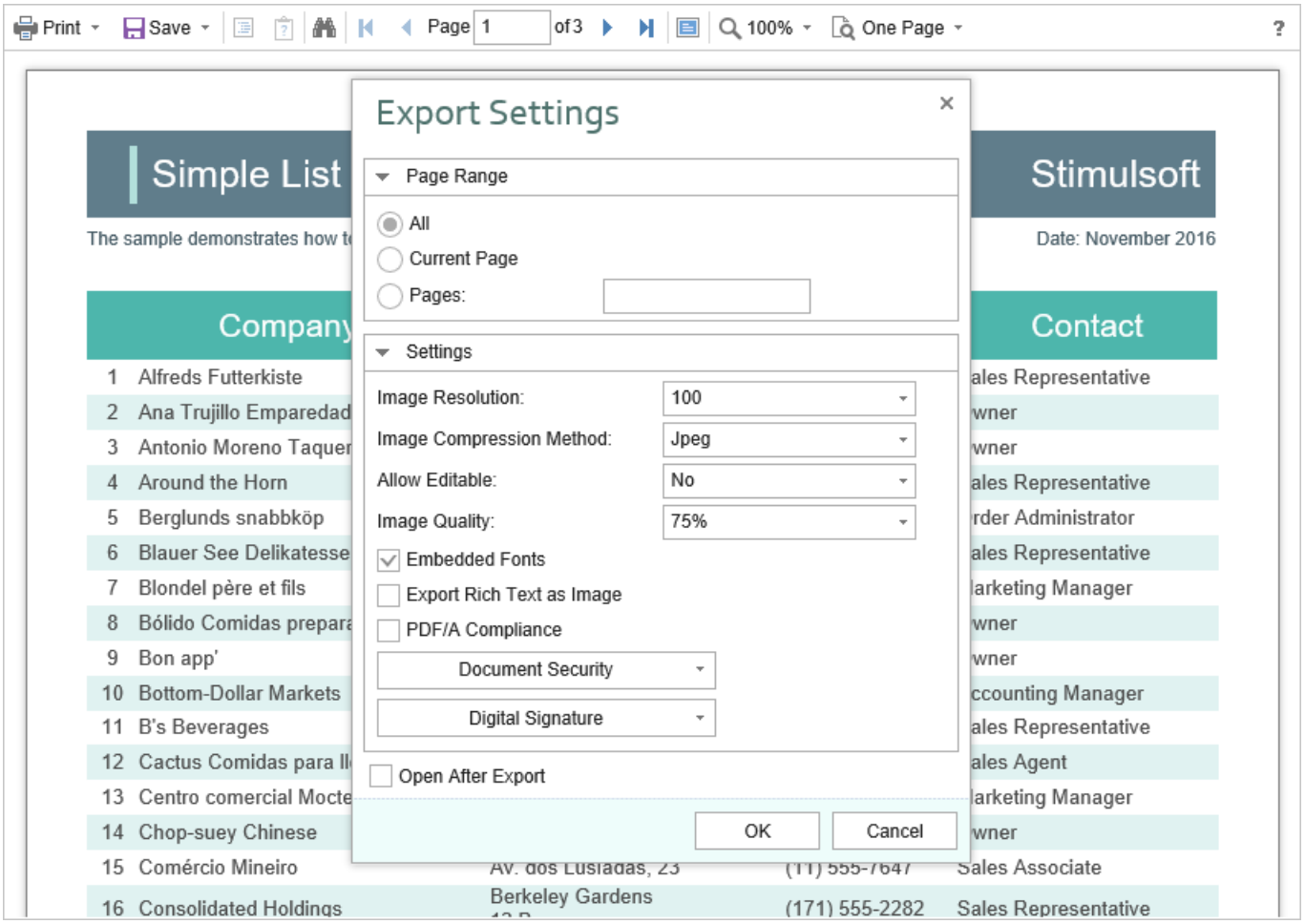
6.1.9 Экспорт отчета и дашборда

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Компонент **HTML5 Viewer** позволяет экспортировать отображаемый отчет в три десятка различных форматов, таких как **PDF**, **HTML**, **Word**, **Excel**, **XPS**, **RTF**, изображения, текст и другие. Экспорт панели индикаторов (дашборда) возможен в PDF, Excel, файлы изображений. Для работы функции экспорта не требуется дополнительных настроек вьювера.

© 2003-2026 Stimulsoft



Для работы функции экспорта не требуется дополнительных настроек вьювера. Если необходимо выполнить какие-либо действия перед экспортированием отчета, то можно определить специальное действие **ExportReport**.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Actions =
   {
      ExportReport = "ExportReport"
   }
})
...
```

© 2003-2026 Stimulsoft

Index.cshtml.cs

```
...
public IActionResult OnPostExportReport()
{
   // Some code before export
   // ...

   return StiNetCoreViewer.ExportReportResult(this);
}
...
```

Настройки экспорта

Каждый формат экспорта отчета компонента **HTML5 Viewer** имеет множество настроек, и каждая настройка имеет свои значения по умолчанию. Иногда требуется установить другие значения по умолчанию. Для этого предназначено специальное свойство вьювера **DefaultSettings**, которое представляет собой контейнер всех настроек экспортов, используемых по умолчанию.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Exports =
   {
      DefaultSettings =
      {
         ExportToPdf =
         {
            ImageQuality = 0.75f,
            ImageFormat = Stimulsoft.Report.Export.StiImageFormat.Color
         },
         ExportToHtml =
         {
            ExportMode = Stimulsoft.Report.Export.StiHtmlExportMode.Div,
            UseEmbeddedImages = true
         }
      }
   }
})
...
```

Если требуется, можно полностью убрать отображение диалоговых окон экспорта, экспортирование всегда будет осуществляться с настройками по умолчанию. Для этого достаточно установить значение **false** для свойства **ShowExportDialog**.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Exports =
   {
      ShowExportDialog = false
   }
})
...
```

Компонент **HTML5 Viewer** содержит порядка 30 различных форматов экспорта, и иногда требуется отключить не используемые форматы. Это позволяет разгрузить интерфейс и упростить использование вьювера. Для отключения не используемых форматов экспортов достаточно установить значение **false** для соответствующих свойств вьювера, представленных в списке ниже.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Exports =
   {
      ShowExportToDocument = true,
      ShowExportToPdf = true,
      ShowExportToXps = true,
      ShowExportToPowerPoint = true,
      ShowExportToHtml = true,
      ShowExportToHtml5 = true,
      ShowExportToMht = true,
      ShowExportToText = true,
      ShowExportToRtf = true,
      ShowExportToWord = true,
      ShowExportToOpenDocumentWriter = true,
      ShowExportToExcel = true,
      ShowExportToExcelXml = true,
      ShowExportToExcelBiff = true,
      ShowExportToOpenDocumentCalc = true,
      ShowExportToCsv = true,
      ShowExportToDbf = true,
      ShowExportToXml = true,
      ShowExportToDif = true,
      ShowExportToSylk = true,
      ShowExportToImageBmp = true,
      ShowExportToImageGif = true,
      ShowExportToImageJpeg = true,
      ShowExportToImagePcx = true,
      ShowExportToImagePng = true,
      ShowExportToImageTiff = true,
      ShowExportToImageMetafile = true,
      ShowExportToImageSvg = true,
      ShowExportToImageSvgz = true
   }
})
...
```

Компонент **HTML5 Viewer** имеет возможность полностью отключить меню экспорта отчета, если оно не требуется. Для этого необходимо установить значение **false** для свойства **ShowSaveButton**.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Toolbar =
   {
      ShowSaveButton = false
   }
})
...
```

© 2003-2026 Stimulsoft

6.1.10 Режимы отображения

В компоненте **HTML5 Viewer** предусмотрено два режима отображения отчета: с полосами прокрутки и без них. По умолчанию установлен режим просмотра без полос прокрутки. Для включения режима просмотра с полосами прокрутки достаточно установить значение **true** для свойства **ScrollbarsMode**.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Appearance =
   {
      ScrollbarsMode = true
   }
})
...
```

В первом режиме (без полос прокрутки), вьювер отображает страницу или отчет целиком, автоматически растягивая область просмотра. Если заданы размеры по ширине и высоте, то вьювер будет осуществлять обрезку вышедшей за границы страницы. Во втором режиме, в отличие от первого, при выходе страницы за границы размеров вьювера, обрезка осуществляться не будет. Вместо этого появятся полосы прокрутки, при помощи которых можно просмотреть страницу или отчет целиком.

Информация

В режиме просмотра отчета с полосами прокрутки необходимо задать высоту вьювера, иначе будет задана высота по умолчанию, равная **650 пикселей**.

В компоненте **HTML5 Viewer** предусмотрен режим полноэкранного отображения отчета или дашборда. По умолчанию включен стандартный режим просмотра, вьювер имеет заданные в настройках размеры. Для включения полноэкранного режима просмотра достаточно установить значение **true** для свойства **FullScreenMode**.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Appearance =
   {
      FullScreenMode = true
   }
})
...
```

Также, для включения либо отключения полноэкранного режима можно воспользоваться соответствующей кнопкой на панели управления вьювера.

В компоненте **HTML5 Viewer** доступно три режима отображения отчета: постраничное отображение, отчет целиком в виде ленты, и табличное отображение страниц отчета. Для управления режимами предназначено свойство **ViewMode**, которое соответственно принимает одно из указанных значений: **SinglePage**, **Continuous**, **MultiplePages**.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Toolbar =
   {
      ViewMode = StiWebViewMode.SinglePage
   }
})
...
```

В компоненте **HTML5 Viewer** реализована поддержка работы как с обычным компьютером, так и с сенсорными экранами, мобильными устройствами. Для управления режимами интерфейса предназначено свойство **InterfaceType**, которое принимает одно из следующих значений:

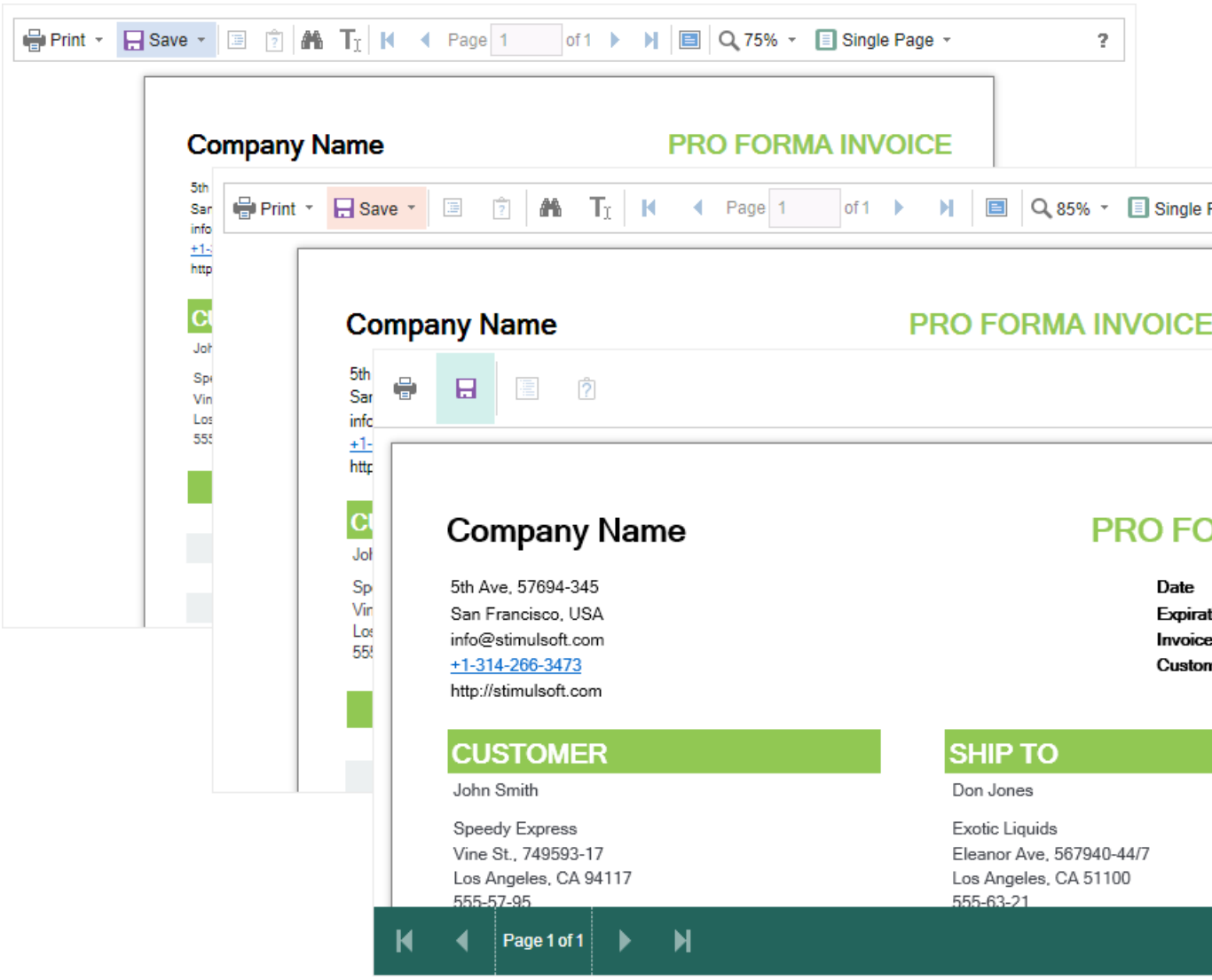
- **Auto** – тип интерфейса вьювера будет выбран автоматически в зависимости от используемого устройства (значение по умолчанию);
- **Mouse** – принудительное использование стандартного интерфейса для управления вьювером при помощи мыши;
- **Touch** – принудительное использование Touch интерфейса для управления вьювером при помощи сенсорного экрана монитора, в этом режиме элементы интерфейса вьювера имеют увеличенные размеры для удобства управления;
- **Mobile** – принудительное использование Mobile интерфейса для управления

© 2003-2026 Stimulsoft

вьювером при помощи экрана смартфона, в этом режиме интерфейс вьювера имеет упрощенный вид и адаптирован для управления при помощи мобильного устройства.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Appearance =
   {
      InterfaceType = StiInterfaceType.Auto
   }
})
...
```



© 2003-2026 Stimulsoft

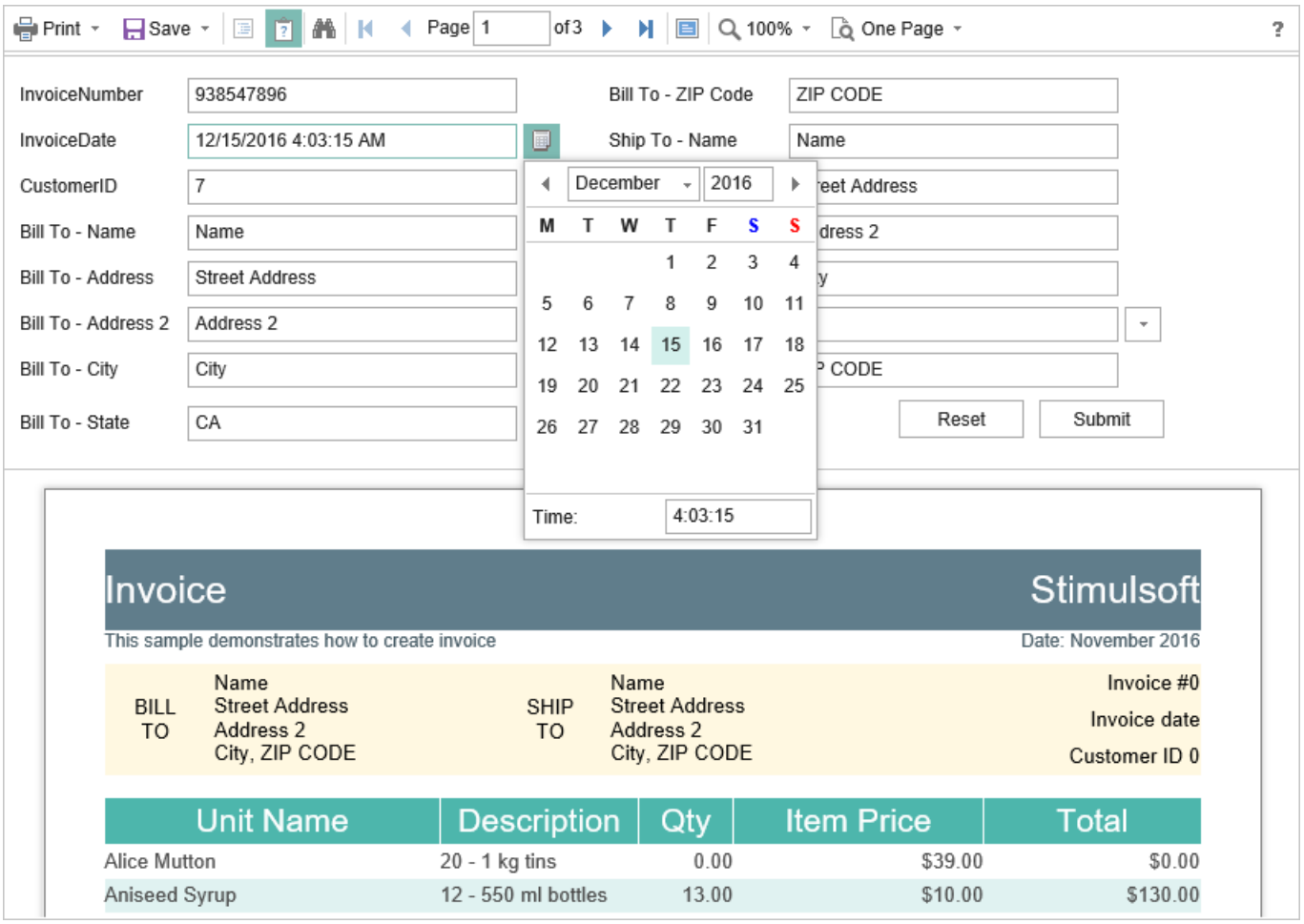
6.1.11 Работа с параметрами

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для работы с параметрами отчета в **HTML5 Viewer** реализована поддержка специальной панели параметров. Для добавления параметра на панель необходимо в отчете определить переменную, запрашиваемую у пользователя. При просмотре отчета во вьювере такая переменная будет автоматически добавлена на панель параметров. Поддерживаются все типы переменных отчета (обычные переменные, дата и время, границы, списки и др.).

© 2003-2026 Stimulsoft



Для работы отчетов с параметрами не требуется дополнительных настроек вьювера. Если необходимо провести какие-либо действия перед применением параметров, то можно определить специальное действие **Interaction**.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Actions =
   {
      Interaction = "ViewerInteraction"
   }
})
...
```

Index.cshtml.cs

```
...
public IActionResult OnPostViewerInteraction()
```

© 2003-2026 Stimulsoft

```
{
   // Some code before any interaction
   // ...

   return StiNetCoreViewer.InteractionResult(this);
}
...
```

Данное действие вызывается при любых интерактивных действиях вьювера. При необходимости выполнить какие-либо действия только при применении параметров отчета, можно воспользоваться параметрами вьювера. Параметры вьювера представлены в виде объекта класса **StiRequestParams**, они передаются при любом запросе на сторону сервера, и содержат все необходимые сведения и состояния клиентской части вьювера. Для определения типа действия вьювера достаточно проверить свойство **Action** у параметров вьювера.

Index.cshtml.cs

```
...
public IActionResult OnPostViewerInteraction()
{
   StiRequestParams requestParams =
   StiNetCoreViewer.GetRequestParams(this);
   if (requestParams.Action == StiAction.Variables)
   {
      // Some code before apply parameters
   }

   return StiNetCoreViewer.InteractionResult(this);
}
...
```

Если работа с параметрами не требуется, то можно полностью отключить данную возможность. Для этого предназначено свойство **ShowParametersButton** в разделе свойств **Toolbar**, для которого необходимо установить значение **false**.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Toolbar =
   {
      ShowParametersButton = false
   }
})
...
```

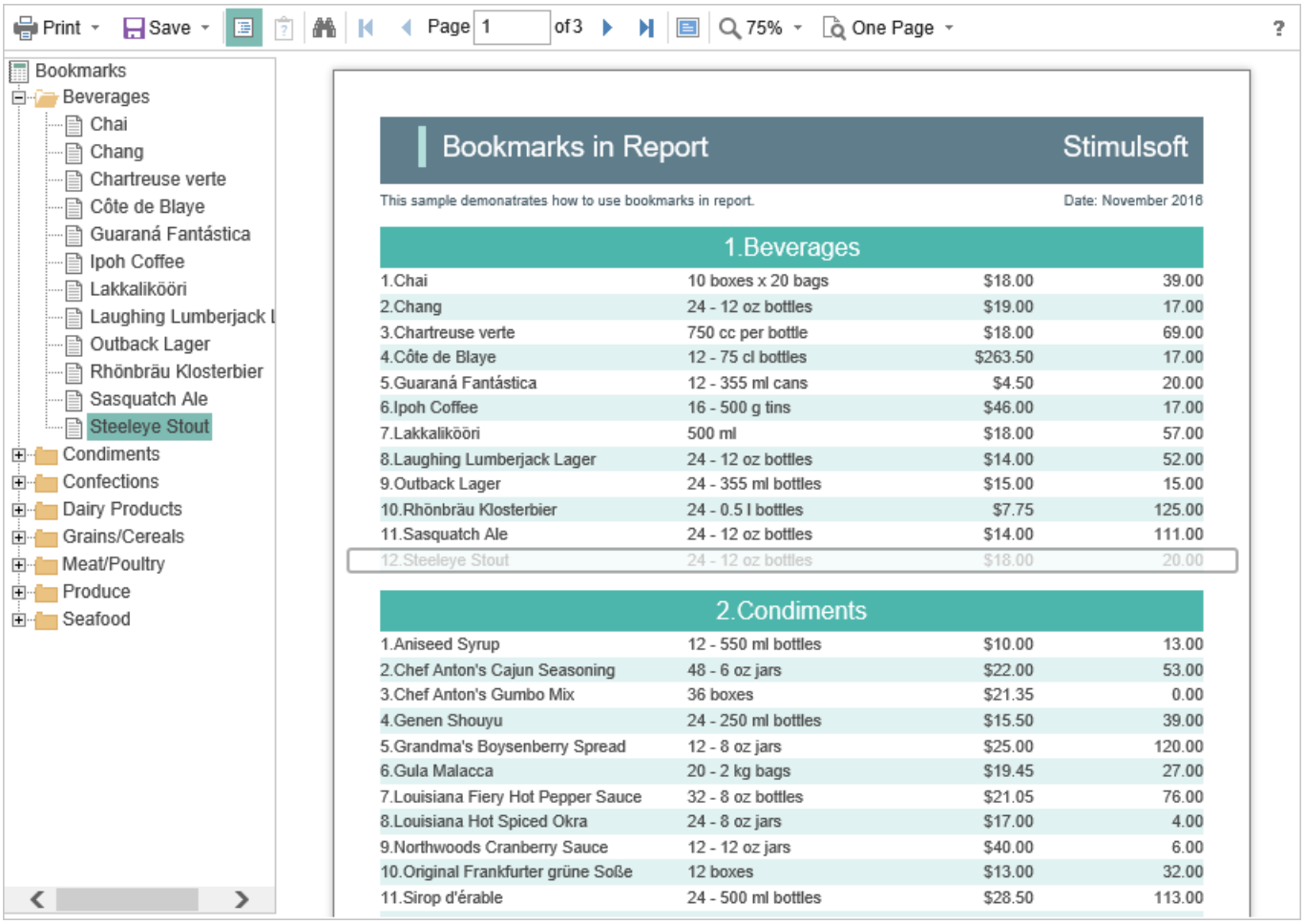
Информация

При такой конфигурации вьювера панель параметров не будет показана даже в том случае, если параметры присутствуют в отображаемом отчете.

6.1.12 Работа с закладками

В компоненте **HTML5 Viewer** реализована поддержка закладок отчета. При отображении такого отчета слева от страницы будет отображена панель с закладками. При выборе закладки отчета вьювер осуществит автоматический переход на нужную страницу, а элемент отчета с закладкой будет подсвечен.

© 2003-2026 Stimulsoft



По умолчанию ширина панели закладок равна 180 пикселей, компонент **HTML5 Viewer** позволяет изменить это значение. Для этого предназначено свойство **BookmarksTreeWidth**, значение которого указывается в пикселях.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Appearance =
   {
      BookmarksTreeWidth = 200
   }
})
...
```

Если работа с закладками отчета не требуется, то можно полностью отключить данную возможность. Для этого предназначено свойство

© 2003-2026 Stimulsoft

ShowBookmarksButton, которое необходимо установить в значение **false**.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Toolbar =
   {
      ShowBookmarksButton = false
   }
})
...
```

Информация

В этом случае закладки отчета не будут показаны, даже если они присутствуют в отображаемом отчете. Данное свойство не оказывает влияние на печать и экспортирование отчета.

При печати отчета с закладками дерево закладок будет скрыто. Если помимо самого отчета требуется распечатать и закладки, то необходимо установить свойство **BookmarksPrint** в значение **true**.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions () {
   Appearance =
   {
      BookmarksPrint = true
   }
})
...
```

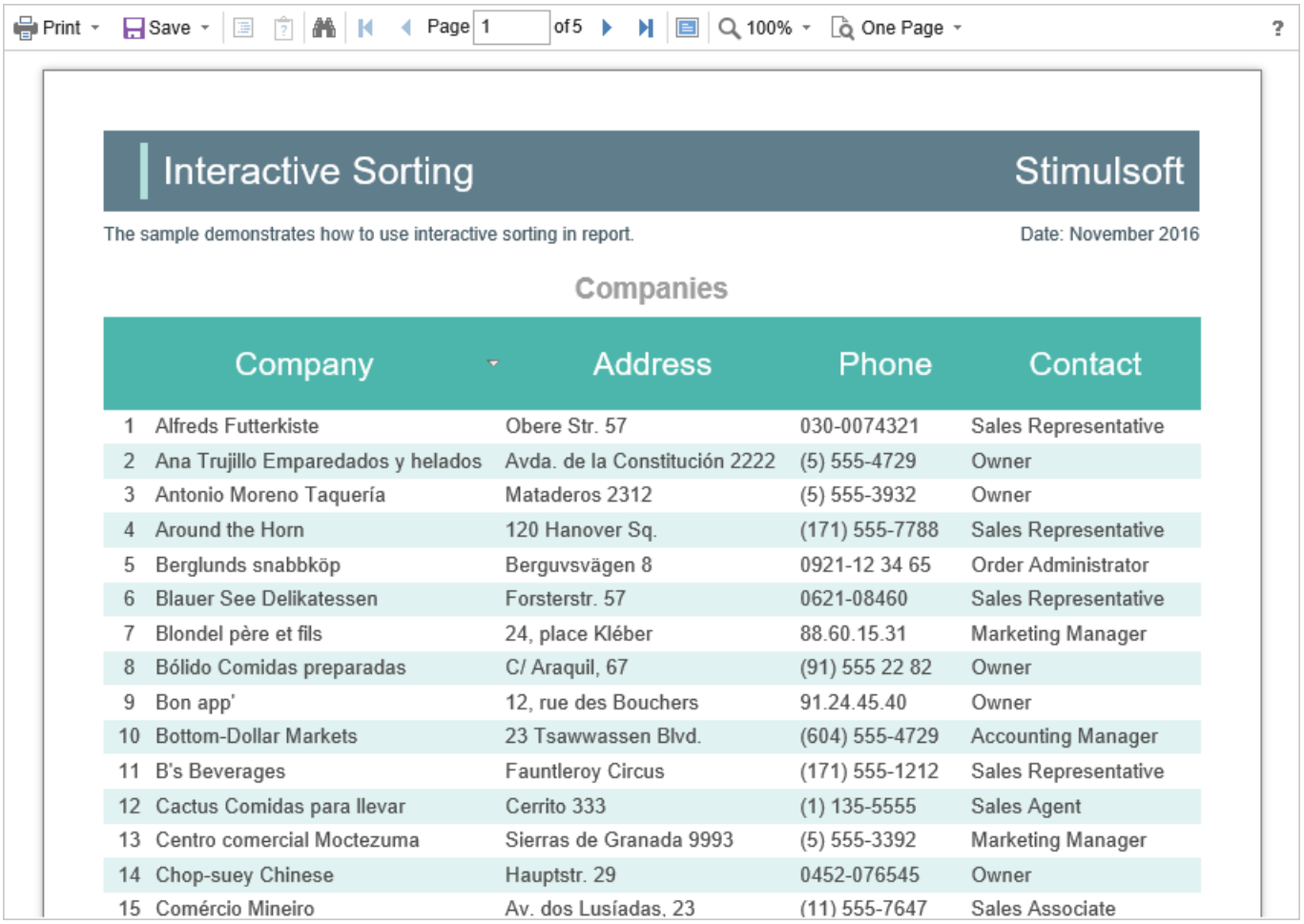
6.1.13 Динамическое сворачивание, сортировка и детализация

В компоненте **HTML5 Viewer** реализована поддержка динамической сортировки, сворачивания и детализации отчетов. Динамическая сортировка предоставляет возможность изменять направление сортировки в построенном отчете. Для этого следует щелкнуть по компоненту, у которого была установлена динамическая сортировка. Динамическая сортировка осуществляется в следующих направлениях: **По возрастанию (Ascending)** и **По убыванию (Descending)**. Каждый раз при щелчке по компоненту направление

© 2003-2026 Stimulsoft

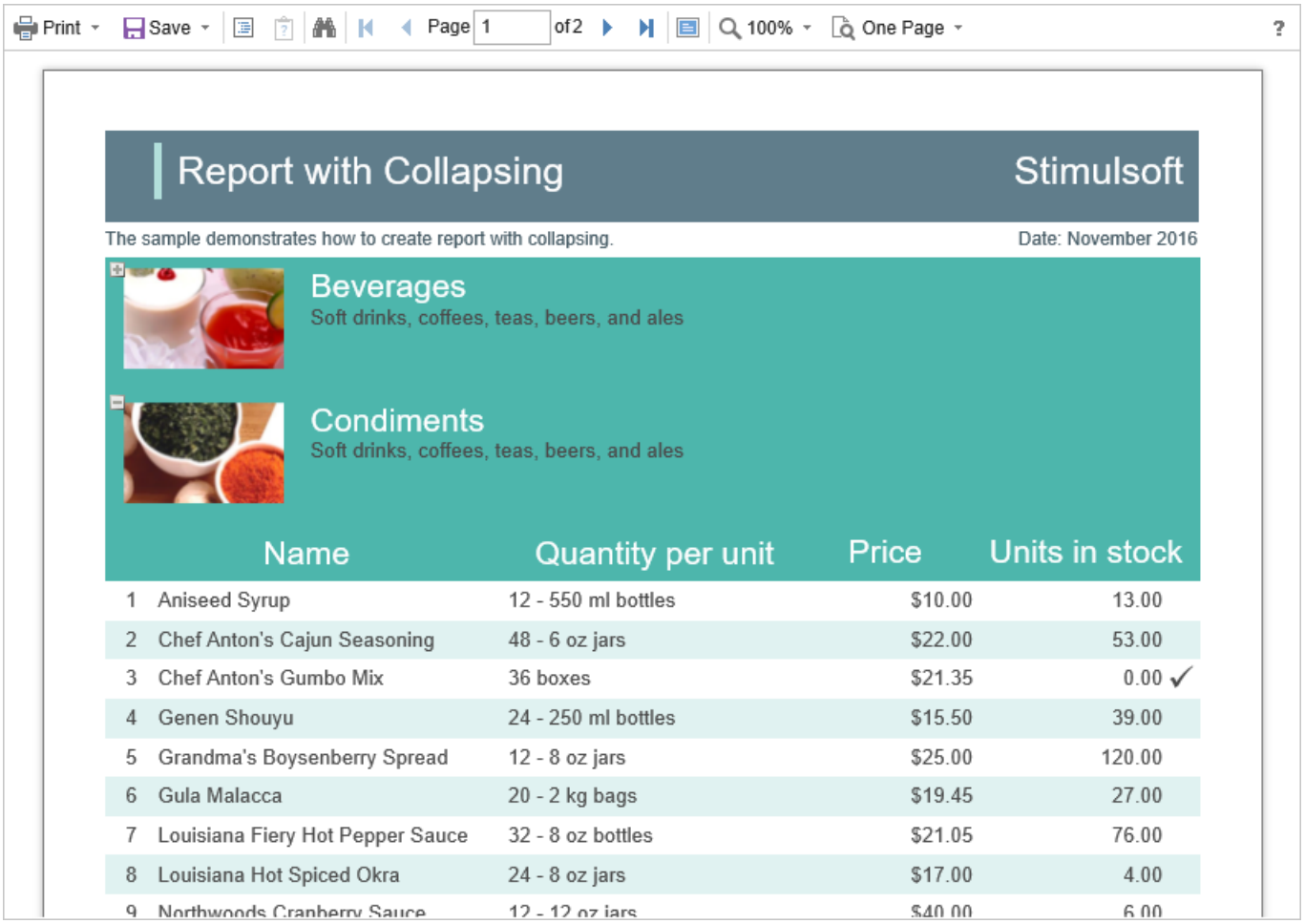
меняется на противоположное.

Допускается многоуровневая сортировка в отчете. Для этого необходимо удерживать клавишу **Ctrl** и последовательно нажимать на сортируемые компоненты отчета. Для сброса сортировки можно нажать на любой сортируемый компонент без удержания клавиши **Ctrl**.



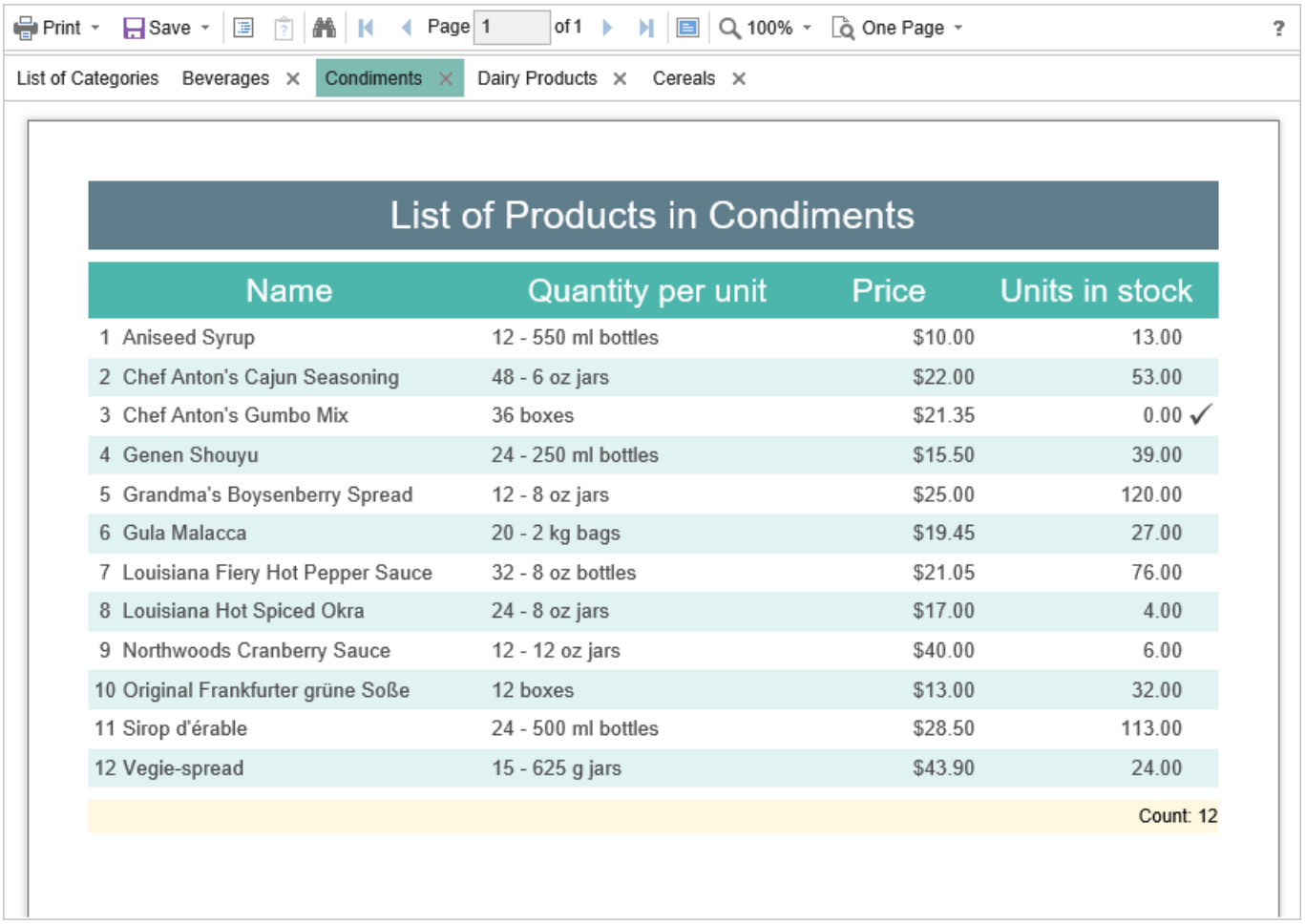
Отчет с динамическим сворачиванием представляет собой интерактивный отчет, в котором сворачивающиеся блоки могут сворачивать/разворачивать свое содержимое при нажатии на заголовок блока. Элементы отчета, которые можно свернуть/развернуть, обозначены специальной иконкой со знаком **[-]** или **[+]**.

© 2003-2026 Stimulsoft



При детализации данных под главной панелью вьювера будет отображена панель детализации с закладками детализированных отчетов. Отображенный в данный момент отчет будет подсвечен.

© 2003-2026 Stimulsoft



Для работы динамической сортировки, сворачивания и детализации отчетов не требуется дополнительных настроек вьювера. Для выполнения каких-либо действий перед сортировкой, сворачиванием или детализацией отчета предназначено специальное действие **Interaction**, которое будет вызвано при любом интерактивном действии вьювера.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Actions =
   {
      Interaction = "ViewerInteraction"
   }
})
...
```

© 2003-2026 Stimulsoft

Index.cshtml.cs

```
...
public IActionResult OnPostViewerInteraction()
{
   // Some code before any interaction
   // ...

   return StiNetCoreViewer.InteractionResult(this);
}
...
```

Для получения типа действия можно воспользоваться параметрами вьювера. Параметры вьювера представлены в виде объекта класса **StiRequestParams**, они передаются при любом запросе на сторону сервера, и содержат все необходимые сведения и состояния клиентской части вьювера. Для каждого вида интерактивности вьювера предусмотрен определенный тип действия:

- **Sorting** – при использовании сортировки колонок;
- **DrillDown** – при использовании детализации отчета;
- **Collapsing** – при использовании сворачивания блоков отчета.

Index.cshtml.cs

```
...
public IActionResult OnPostViewerInteraction()
{
   StiRequestParams requestParams =
   StiNetCoreViewer.GetRequestParams(this);
   switch (requestParams.Action)
   {
      case StiAction.Sorting:
      break;

      case StiAction.DrillDown:
      break;

      case StiAction.Collapsing:
      break;
   }

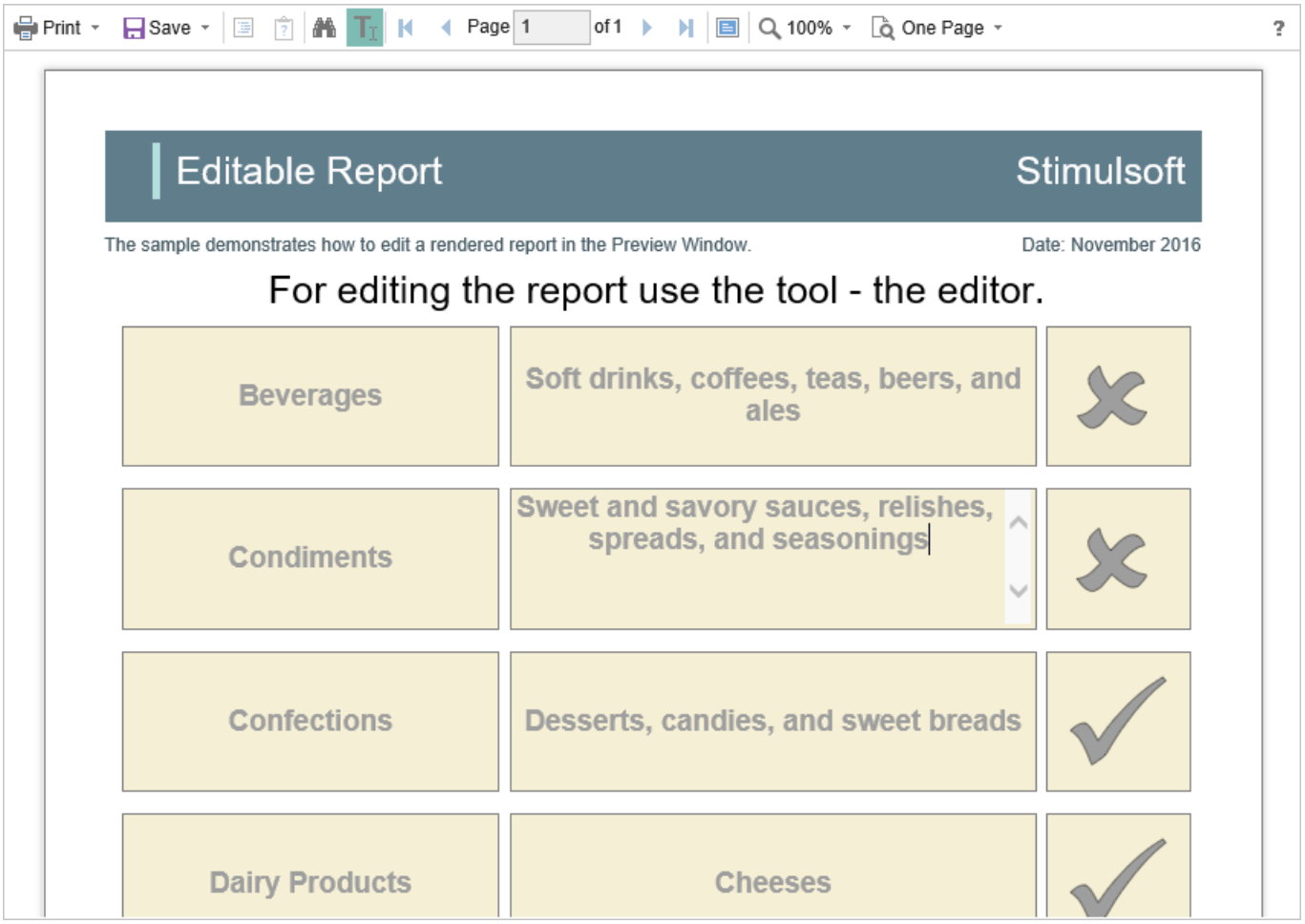
   return StiNetCoreViewer.InteractionResult(this);
}
...
```

6.1.14 Редактирование отчета

Компонент **HTML5 Viewer** имеет возможность редактировать элементы построенного отчета, такие как текстовые поля и чекбоксы. Для того, чтобы редактирование стало возможным, в самом шаблоне отчета необходимо

© 2003-2026 Stimulsoft

пометить необходимые компоненты как редактируемые. После отображения отчета во вьювере, для начала редактирования необходимо нажать на соответствующую кнопку на панели вьювера. После завершения редактирования необходимо нажать указанную кнопку еще раз, и все внесенные изменения будут примерены к отчету.



Для работы режима редактирования отчета не требуется никаких специальных настроек вьювера.

© 2003-2026 Stimulsoft

Информация

Отредактированные значения будут применены при печати либо экспортировании отчета, при этом исходный отчет останется не тронутым. После перезагрузки вьювера все значения будут возвращены к исходным.

6.1.15 Отправка отчета по Email

Информация

Обратите внимание, что возможность Отправить отчет по Email доступна только для отчетов, и недоступна для дашбордов.

В компоненте **HTML5 Viewer** предусмотрена возможность отправки отчета по Email. Для активации данной возможности необходимо установить свойство вьювера **ShowSendEmailButton** в значение **true** и определить действие **EmailReport**.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Actions =
   {
      EmailReport = "EmailReport"
   },
   Toolbar =
   {
      ShowSendEmailButton = true
   }
})
...
```

Index.cshtml.cs

```
...
public IActionResult OnPostEmailReport()
{
   StiEmailOptions options = StiNetCoreViewer.GetEmailOptions(this);

   // Passed from the viewer, can be checked and changed
   // options.AddressTo = "";
   // options.Subject = "";
```

© 2003-2026 Stimulsoft

```
    // options.Body = "";

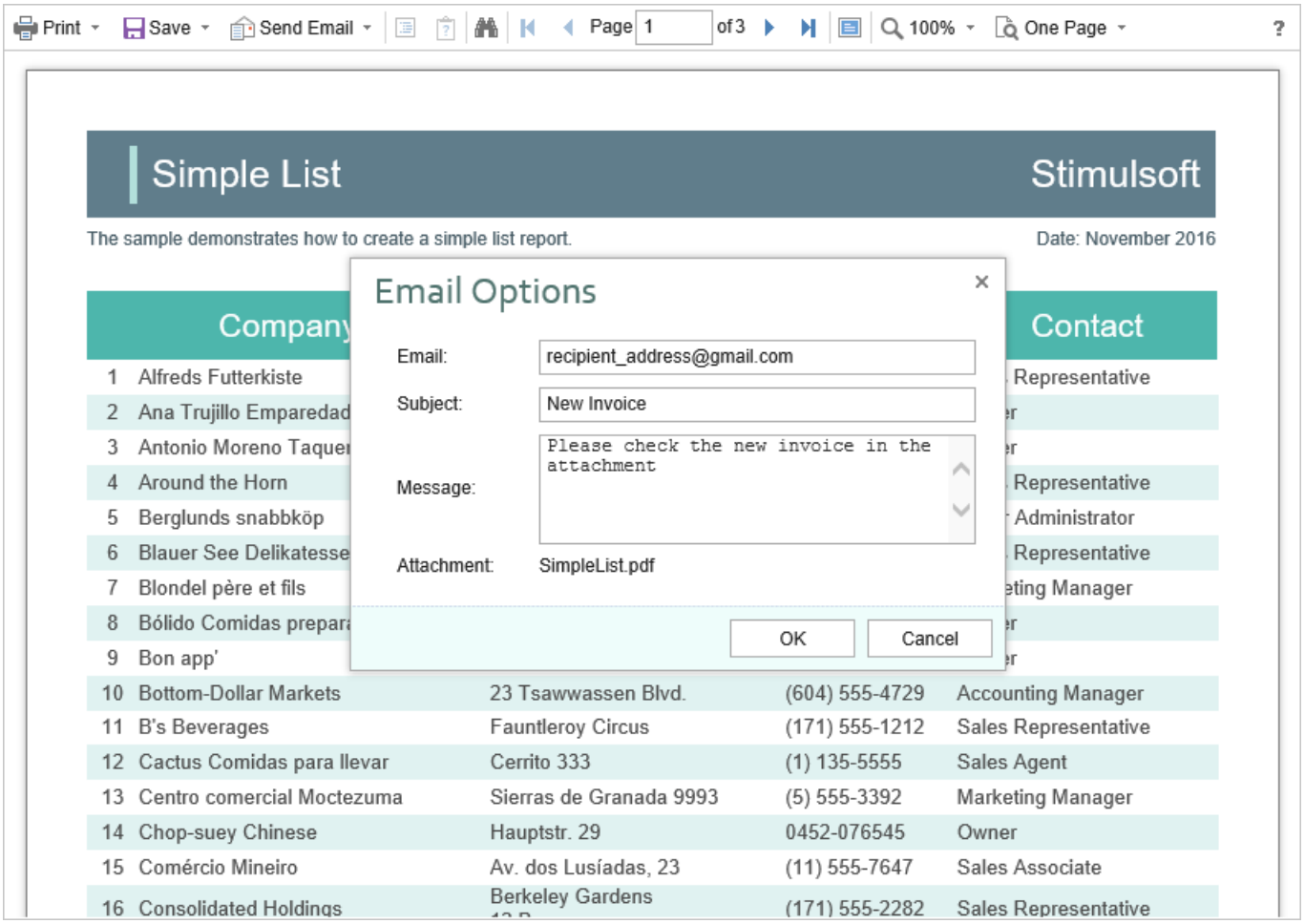
    // Should be filled here
    options.AddressFrom = "admin_address@test.com";
    options.Host = "smtp.test.com";
    options.Port = 465;
    options.UserName = "admin_address@test.com";
    options.Password = "admin_password";

    // options.CC.Add("email@test.com");
    // options.BCC.Add("email@test.com");
    // options.EnableSsl = true;

    return StiNetCoreViewer.EmailReportResult(this, options);
}
...
```

При отправке отчета по Email отображается меню выбора формата вложения, которое соответствует меню выбора формата экспортирования отчета. После выбора формата отобразиться диалог ввода параметров отправки, таких как Email получателя, тема и текст письма.

© 2003-2026 Stimulsoft



После подтверждения отправки будет вызвано описанное выше действие **EmailReport**, в котором можно проверить и скорректировать данные, введенные в этой форме. Экспортированный файл отчета будет прикреплен к письму автоматически.

Компонент **HTML5 Viewer** позволяет установить значения по умолчанию для формы отправки Email. Для этого предназначены свойства **DefaultEmailAddress**, **DefaultEmailSubject** и **DefaultEmailMessage**. По умолчанию данные свойства являются пустыми.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Email =
   {
      DefaultEmailAddress = "recipient_address@gmail.com",
      DefaultEmailSubject = "New Invoice",
      DefaultEmailMessage = "Please check the new invoice in the
      attachment"
   }
})
...
```

6.1.16 Вызов дизайнера из вьювера

Компонент **HTML5 Viewer** имеет возможность вызвать дизайнер отчетов. Для этого предназначена специальная кнопка **Дизайн (Design)** на панели инструментов вьювера (по умолчанию данная кнопка отключена). Для использования этой возможности необходимо установить свойство **ShowDesignButton** в значение **true**, а также определить действие **DesignReport**.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Actions =
   {
      DesignReport = "DesignReport"
   },
   Toolbar =
   {
      ShowDesignButton = true
   }
})
...
```

Index.cshtml.cs

```
...
public IActionResult OnPostDesignReport()
{
   StiReport report = StiNetCoreViewer.GetReportObject(this);
   TempData["ReportName"] = report.ReportName;

   return RedirectToPage("Designer");
}
...
```

Информация

© 2003-2026 Stimulsoft

Сам вьювер не запускает дизайнер, он лишь вызывает указанное действие, в котором можно получить все необходимые параметры. Далее в действии можно выполнить перенаправление на другой View, который содержит дизайнер отчетов.

6.1.17 Кэширование

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Компонент **HTML5 Viewer** позволяет использовать кэш сервера для хранения построенного отчета. Если кэширование не используется, то при каждом запросе страницы необходимо загружать отчет, подключать данные и строить его заново. Если использовать кэширование, то при следующем обновлении страницы, ранее построенный отчет будет загружен из кэша.

При использовании кэширования стоит учитывать, что каждый сохраненный отчет в кэше занимает память сервера и при большом количестве запросов к отчетам это может стать критичным фактором. Поэтому необходимо выбирать из двух вариантов – или маленькие потребности к памяти, но большие к производительности, или небольшие потребности к производительности, но большие к памяти.

Управлять кэшированием можно при помощи следующих приведенных ниже свойств.

Свойство CacheMode

© 2003-2026 Stimulsoft

Данное свойство вьювера включает кэширование и устанавливает его тип. Может принимать одно из значений, указанных в перечислении **StiServerCacheMode**:

- **None** – кэширование отключено, каждое действие вьювера требует загрузки отчета и, если это шаблон отчета, последующее его построение;
- **ObjectCache** – для кэширования используется кэш сервера, в котором сохраняется объект отчета (значение по умолчанию);
- **StringCache** - для кэширования используется кэш сервера, в котором сохраняется отчет в виде упакованной строки;
- **ObjectSession** - для кэширования используется текущая сессия, в которой сохраняется объект отчета;
- **StringSession** - для кэширования используется текущая сессия, в которой сохраняется отчет в виде упакованной строки.

Свойство CacheItemPriority

Данное свойство устанавливает приоритет хранимого в кэше сервера отчета, влияет на автоматическую очистку памяти сервера в случае ее нехватки. Чем меньше приоритет, тем больше шанс удаления информации из памяти.

Свойство CacheTimeout

Данное свойство указывает величину времени в минутах, в течение которого необходимо сохранять отчет в кэше сервера. Если при использовании кэширования требуемый отчет не найден в кэше (т.е. истекло время хранения объекта), то он будет запрошен заново при помощи специального действия **GetReport** с последующим подключением данных отчета, и их построением.

Компонент **HTML5 Viewer** предоставляет возможность определить собственные методы работы с кэшированием отчета. Для этого предназначен специальный класс **StiCacheHelper**, который содержит методы получения отчета из кэша и сохранения отчета в кэш. Необходимо создать новый класс, унаследованный от **StiCacheHelper**, и перегрузить указанные выше методы,

© 2003-2026 Stimulsoft

которые соответственно имеют названия **GetReport**, **SaveReport** и **RemoveReport**.

Index.cshtml.cs

```
...
public class IndexModel : PageModel
{
   public class StiMyCacheHelper : StiCacheHelper
   {
      public override StiReport GetReport(string guid)
      {
         var path = Path.Combine(HttpContext.Server.MapPath("CacheFiles"),
         guid);
         if (File.Exists(path))
         {
            StiReport report = new StiReport();
            var packedReport = File.ReadAllText(path);
            if (guid.EndsWith(GUID_ReportTemplate))
            report.LoadPackedReportFromString(packedReport);
            else report.LoadPackedDocumentFromString(packedReport);

            return report;
         }
         return null;
      }

      public override void SaveReport(StiReport report, string guid)
      {
         var packedReport = guid.EndsWith(GUID_ReportTemplate) ?
         report.SavePackedReportToString() :
         report.SavePackedDocumentToString();
         var path = Path.Combine(HttpContext.Server.MapPath("CacheFiles"),
         guid);
         File.WriteAllText(path, packedReport);
      }

      public override void RemoveReport(string guid)
      {
         var path = Path.Combine(HttpContext.Server.MapPath("CacheFiles"),
         guid);
         if (File.Exists(path))
            File.Delete(path);
      }
   }

   static IndexModel()
   {
      StiNetCoreViewer.CacheHelper = new StiMyCacheHelper();
   }
}
...
```

Для инициализации работы с кэшированием отчета при помощи созданного

© 2003-2026 Stimulsoft

класса, достаточно задать его в качестве значения статического свойства **StiNetCoreViewer.CacheHelper** в конструкторе страницы.

Информация

При отключенном кэшировании отчета (свойство вьювера **CacheMode** имеет значение **None**), указанный класс задействован не будет.

6.1.18 Вспомогательные методы

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для **HTML5 Viewer** предусмотрено несколько вспомогательных методов, которые предназначены для получения объекта просматриваемого в данный момент отчета, параметров текущего состояния вьювера и других полезных данных. Эти методы можно использовать в любых действиях вьювера.

Метод GetReportObject()

© 2003-2026 Stimulsoft

Возвращает объект отчета, с которым в данный момент работает вьювер. Допускается производить с ним необходимые манипуляции - регистрировать новые наборы данных, изменять свойства отчета, присваивать параметры либо загружать в объект другой отчет. Затем отчет можно вернуть вьюверу, указав его в качестве параметра в результирующем методе действия.

Index.cshtml.cs

```
...
public IActionResult OnPostViewerInteraction()
{
   StiReport report = StiNetCoreViewer.GetReportObject(this);
   report.ReportName = "MyReportName";

   return StiNetCoreViewer.InteractionResult(this, report);
}
...
```

Метод GetFormValues()

Возвращает значения формы, которая инициировала (открыла POST-запросом) страницу вьювера. Таким образом предоставляется возможность получить коллекцию параметров формы в любом действии вьювера.

Index.cshtml.cs

```
...
public IActionResult OnPostViewerInteraction()
{
   NameValueCollection formValues = StiNetCoreViewer.GetFormValues(this);

   return StiNetCoreViewer.InteractionResult(this);
}
...
```

По умолчанию данная возможность отключена в целях оптимизации запросов клиентской стороны вьювера на сервер. Для включения достаточно установить свойство PassFormValues в значение true.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Server =
   {
      PassFormValues = true
   }
})
...
```

Метод GetRequestParams()

Возвращает все параметры текущего состояния вьювера, переданные на сторону сервера. Они могут быть полезны для определения типа действия, которое в данный момент выполняет вьювер – например, для определения типа экспорта, а также всех параметров действия.

Index.cshtml.cs

```
...
public IActionResult OnPostExportReport()
{
   StiRequestParams requestParams =
   StiNetCoreViewer.GetRequestParams(this);
   if (requestParams.ExportFormat == StiExportFormat.Pdf)
   {
      StiReport report = StiNetCoreViewer.GetReportObject(this);

      // Some action with report for the PDF export
      // ...

      return StiNetCoreViewer.ExportReportResult(this, report);
   }

   return StiNetCoreViewer.ExportReportResult(this);
}
...
```

Допускается изменять значения некоторых параметров. После внесения изменений, для корректной работы вьювера необходимо передать измененный объект параметров на вход результирующего метода.

© 2003-2026 Stimulsoft

Index.cshtml.cs

```
...
public IActionResult OnPostViewerInteraction()
{
   StiRequestParams requestParams =
   StiNetCoreViewer.GetRequestParams(this);
   if (requestParams.Action == StiAction.Variables)
   {
      requestParams.Interaction.Variables["Variable1"] = "MyValue";
      return StiNetCoreViewer.InteractionResult(this, requestParams);
   }

   return StiNetCoreViewer.InteractionResult(this);
}
...
```

Метод GetExportSettings()

Возвращает все параметры текущего экспортирования отчета. Тип объекта параметров будет соответствовать типу экспорта, выбранному в меню вьювера. Любые параметры экспорта можно изменить и передать на вход результирующего метода. В этом случае, экспортирование отчета будет произведено с переданными параметрами.

Index.cshtml.cs

```
...
public IActionResult OnPostExportReport()
{
   StiExportSettings settings = StiNetCoreViewer.GetExportSettings(this);
   if (settings.GetExportFormat() == StiExportFormat.Pdf)
   {
      StiPdfExportSettings pdfSettings = (StiPdfExportSettings)settings;
      pdfSettings.EmbeddedFonts = true;
      pdfSettings.AllowEditable = StiPdfAllowEditable.No;
      return StiNetCoreViewer.ExportReportResult(this, settings);
   }

   return StiNetCoreViewer.ExportReportResult(this);
}
...
```

Методы MapPath() и MapWebRootPath()

Возвращает абсолютный путь соответственно к директории приложения или к директории wwwroot. Можно использовать для загрузки файлов шаблонов

© 2003-2026 Stimulsoft

отчетов, файлов данных и др. Данные методы расположены в статическом классе **StiNetCoreHelper**.

Index.cshtml.cs

```
...
public IActionResult OnPostGetReport()
{
   StiReport report = new StiReport();
   report.Load(StiNetCoreHelper.MapPath(this, "Reports/SimpleList.mrt"));

   return StiNetCoreViewer.GetReportResult(this, report);
}
...
```

6.1.19 Экспорт и печать из кода

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет». Кроме этого, следует понимать, что возможность печати из кода доступна только для отчетов, и недоступна для дашбордов.

Компонент **HTML5 Viewer** предоставляет возможность печатать отчет различными способами и выполнять экспортирование отчета в различные форматы. Эти действия выполняются при помощи меню вьювера. Если требуется напечатать отчет или выполнить экспорт при помощи кода отчета, например, в определенном событии страницы, то можно воспользоваться специальным классом **StiNetCoreReportResponse**. Данный класс содержит набор статических методов, позволяющих выполнить печать отчета либо экспорт отчета из кода, при этом вьювер отчетов не требуется.

Index.cshtml

```
page "{handler?}"
...
<a href="PrintReport">Print Report from Code</a>
<br />
<a href="ExportReport">Export Report from Code</a>
...
```

© 2003-2026 Stimulsoft

Index.cshtml.cs

```
...
private StiReport LoadSimpleList()
{
   DataSet dataSet = new DataSet();
   dataSet.ReadXml(Server.MapPath("Reports/Demo.xml"));

   StiReport report = new StiReport();
   report.Load(Server.MapPath("Reports/SimpleList.mrt"));
   report.RegData(dataSet);

   return report;
}

public IActionResult OnGetPrintReport()
{
   StiReport report = LoadSimpleList();

   return StiNetCoreReportResponse.PrintAsPdf(report);
   //return StiNetCoreReportResponse.PrintAsHtml(report);
}

public IActionResult OnGetExportReport()
{
   StiReport report = LoadSimpleList();

   return StiNetCoreReportResponse.ResponseAsPdf(report);
   //return StiNetCoreReportResponse.ResponseAsExcel2007(report);
   //return StiNetCoreReportResponse.ResponseAsText(report);
   //StiNetCoreReportResponse.ResponseAsJson(report);
}
...
```

Класс **StiNetCoreReportResponse** содержит методы печати отчета в формате PDF и HTML, а также методы экспортирования отчета в любой из поддерживаемых форматов. В качестве аргументов методы могут принимать различные настройки экспортов, режимы отображения и варианты сохранения полученных файлов.

6.1.20 Время ожидания

При работе с компонентом **StiNetCoreViewer** можно установить время ожидания (timeout) выполнения различных операций - хранение отчета в кэше, ответ сервера, выполнение запроса. Настройка времени ожидания выполняется при помощи свойств компонента и опций отчета.

© 2003-2026 Stimulsoft

Свойство CacheTimeout

Предоставляет возможность установить время в минутах, которое сервер будет хранить построенный отчет в кэше с момента последнего действия вьювера. По умолчанию установлено значение 10 минут.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Server =
   {
      CacheTimeout = 10
   }
})
...
```

Использование кэша увеличивает скорость работы вьювера с отчётом. Более подробно можно ознакомиться в главе Кэширование.

Свойство RequestTimeout

Предоставляет возможность установить время ожидания ответа от сервера в секундах, по истечении которого будет выдана ошибка связи с сервером. По умолчанию установлено значение 30 секунд. Для больших отчетов рекомендуется увеличить данное значение.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Server =
   {
      RequestTimeout = 30
   }
})
...
```

Опция CommandTimeout

Предоставляет возможность установить время ожидания запроса в секундах, при использовании SQL источников данных в отчете. Значение данного свойства сохраняется в самом шаблоне отчёта для каждого SQL подключения отдельно.

© 2003-2026 Stimulsoft

Ниже приведен пример кода, который предоставляет возможность установить время ожидания запроса для уже созданного MS SQL соединения и источников данных в отчете.

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Actions =
   {
      GetReport = "GetReport",
      ViewerEvent = "ViewerEvent"
   }
   })
...
```

Index.cshtml.cs

```
...
public IActionResult OnPostGetReport()
{
   StiReport report = new StiReport();
   report.Load(Server.MapPath("Report.mrt"));
   ((StiSqlSource)
   report.Dictionary.DataSources["DataSourceName"]).CommandTimeout = 1000;

   return StiNetCoreViewer.GetReportResult(this, report);
}

public IActionResult OnGetViewerEvent()
{
   return StiNetCoreViewer.ViewerEventResult(this);
}

public IActionResult OnPostViewerEvent()
{
   return StiNetCoreViewer.ViewerEventResult(this);
}
...
```

6.1.21 Настройки вьювера

Настройка **HTML5 Viewer** выполняется при помощи свойств, находящихся в классе **StiNetCoreViewerOptions**. Все свойства разделены на группы, некоторые из групп содержат свои подгруппы для удобства использования. Ниже приведен пример установки свойств вьювера.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.StiNetCoreViewer(new StiNetCoreViewerOptions() {
   Theme = StiViewerTheme.Office2022WhiteTeal,
   Localization = "Localization/en.xml",
   Actions =
   {
      GetReport = "GetReport",
      ViewerEvent = "ViewerEvent"
   },
   Appearance =
   {
      ReportDisplayMode = StiReportDisplayMode.Auto,
      InterfaceType = StiInterfaceType.Auto,
      ScrollbarsMode = true,
      ShowTooltips = false
   },
   Exports =
   {
      DefaultSettings =
      {
         ExportToPdf =
         {
            CreatorString = "Company Name",
            ImageQuality = 0.75f
         }
      },
      ShowExportToDbf = false,
      ShowExportToDif = false
   }
})
...
```

Обратите внимание, что все элементы дашбордов имеют собственные кнопки экспорта и полноэкранного режима просмотра. Специальных опций для управления их отображением не предусмотрено, но их можно выключить через свойства элемента. Нижеприведенный код необходимо добавить после загрузки отчета, перед его передачей вьюверу.

Index.cshtml.cs

```
...
var dbsElementInteraction = (report.GetComponentByName("RegionMap1") as
Stimulsoft.Report.Dashboard.IStiElementInteraction).DashboardInteraction;
(dbsElementInteraction as
Stimulsoft.Report.Dashboard.IStiInteractionLayout).ShowFullScreenButton =
false;
(dbsElementInteraction as
Stimulsoft.Report.Dashboard.IStiInteractionLayout).ShowSaveButton = false;
...
```

© 2003-2026 Stimulsoft

Главные настройки (без группы)

Наименование	Описание
Theme	Задает тему оформления вьювера. Список доступных тем оформления находится в перечислении **StiViewerTheme**. По умолчанию установлено значение **Office2022WhiteBlue**.
Localization	Задает путь к XML файлу локализации. Путь может быть абсолютным либо относительным. По умолчанию используется английская локализация, которая встроена во вьювер и не требует дополнительных XML файлов.
Width	Задает ширину компонента в необходимых единицах измерения, которые определены в классе **Unit**. Доступна установка значения в пикселях: **Unit.Pixel()**, поинтах: **Unit.Point()** и процентах: **Unit.Percentage()**. По умолчанию установлена ширина 100%.
Height	Задает высоту компонента в необходимых единицах измерения, которые определены в классе **Unit**. Доступна установка значения в пикселях: **Unit.Pixel()**, поинтах: **Unit.Point()** и процентах: **Unit.Percentage()**. По умолчанию установлена автоматическая высота в зависимости от размера страницы отчета, либо 650 пикселей в режиме отображения вьювера с полосами прокрутки.

Действия (Actions)

Наименование	Описание
GetReport	Задает название метода действия подготовки построенного отчета. Если включено

© 2003-2026 Stimulsoft

	кэширование отчета, то данное действие будет вызвано только один раз при первом запросе отчета, либо в случае, если требуемый отчет не найден в кэше сервера.
PrintReport	Задает название метода действия печати отчета.
ExportReport	Задает название метода действия экспорта отчета в необходимый формат.
EmailReport	Задает название метода действия отправки отчета по Email.
Interaction	Задает название метода действия работы вьювера с интерактивными операциями, такими как использование параметров, динамическая сортировка, сворачивание и детализация отчета.
DesignReport	Задает название метода действия для перехода к нужному представлению при нажатии кнопки Дизайн (Design) на панели вьювера.
ViewerEvent	Задает название метода действия основных событий вьювера и обработки действий вьювера, таких как печать и экспортирование отчета, работа с параметрами и интерактивность, если эти действия не заданы отдельно. Дополнительно данное действие используется для загрузки скриптов и стилей вьювера. Это действие является обязательным.

Работа с сервером (Server)

Наименование	Описание
RouteTemplate	Задает шаблон (route template) вызова действий вьювера отчетов. Если свойство не установлено, будет использоваться шаблон,

© 2003-2026 Stimulsoft

	установленный в Razor проекте. Если свойство `UseRelativeUrls` установлено в true, для этого свойства не будет учитываться `BasePath`. По умолчанию это свойство не установлено.
AllowAutoUpdateCookies	Предоставляет возможность вьюверу автоматически обновлять файлы cookies при каждом запросе на сервер. По умолчанию cookies устанавливаются при создании вьювера, если они не указаны в отчете. По умолчанию свойство имеет значение **false**.
AllowAntiforgeryToken	Предоставляет возможность вьюверу автоматически запрашивать и отправлять токен защиты от подделки. По умолчанию свойство имеет значение **true**.
RequestTimeout	Устанавливает время ожидания ответа от сервера в секундах, по истечении которого будет выдана ошибка. По умолчанию установлено значение 30 секунд. Для больших отчетов рекомендуется увеличить данное значение.
CacheTimeout	Устанавливает время в минутах, которое сервер будет хранить построенный отчет с момента последнего действия вьювера. По умолчанию установлено значение 10 минут.
CacheMode	Устанавливает режим кэширования отчета. Может принимать одно из указанных ниже значений перечисления **StiServerCacheMode**: › **None** – кэширование отключено, каждое действие вьювера требует загрузки отчета и, если это шаблон отчета, последующее его построение; › **ObjectCache** – в качестве хранилища используется кэш, отчет хранится как объект (значение свойства по умолчанию); › **ObjectSession** – в качестве хранилища

© 2003-2026 Stimulsoft

	используется сессия, отчет хранится как объект; › **StringCache** – в качестве хранилища используется кэш сервера, отчет сериализуется в упакованную строку; › **StringSession** – в качестве хранилища используется сессия, отчет сериализуется в упакованную строку.
CacheItemPriority	Устанавливает приоритет хранимого в кэше сервера отчета. Это свойство влияет на автоматическую очистку памяти сервера в случае ее нехватки. Чем меньше приоритет, тем больше шанс удаления информации из памяти.
AllowAutoUpdateCache	Устанавливает режим автоматического обновления кэша. Отчет, хранящийся в кэше либо сессии сервера, будет автоматически пересохранен через определенный промежуток времени при бездействии вьювера (примерно каждые 3 минуты). По умолчанию свойство имеет значение **true**.
UseRelativeUrls	Устанавливает режим вьювера, в котором для AJAX запросов на сервер используются относительные ссылки. По умолчанию свойство имеет значение **true**.
PortNumber	Получает или задает значение, которое указывает номер порта для использования в URL. Значение 0 определяет автоматическое обнаружение (установлено по умолчанию). Значение -1 удаляет номер порта.
PassQueryParametersForResources	Включает передачу всех параметров URL запроса при формировании ссылок на ресурсы вьювера. При значении **false** для запроса ресурсов вьювера используются только необходимые параметры, что способствует более корректной работе кэша браузера. По умолчанию свойство имеет

© 2003-2026 Stimulsoft

	значение **true**.
PassQueryParametersToReport	Включает использование всех параметров URL запроса в качестве значений одноименных переменных в отчете. По умолчанию свойство имеет значение **false**.
PassFormValues	Включает передачу значений POST-формы на сторону клиента, если эти значения требуется использовать в действиях вьювера. При включении данной возможности, вспомогательный метод **GetFormValues()** будет возвращать коллекцию параметров формы. По умолчанию свойство имеет значение **false**.
ShowServerErrorPage	Включает отображение HTML-страницы с детализацией ошибки, возникшей на стороне сервера. При включенном свойстве детализация ошибки будет отображена в окне вьювера. Если свойство выключено, будет отображен только числовой код ошибки и краткий текст ошибки в диалоговом окне. По умолчанию свойство имеет значение **true**.
UseCompression	Включает сжатие запросов вьювера в GZip поток. Это позволяет уменьшить объем интернет-трафика, но немного замедляет работу вьювера. По умолчанию свойство имеет значение **false**.
UseCacheForResources	Включает кэширование на стороне сервера ресурсов компонента, таких как скрипты, стили, изображения. Позволяет ускорить загрузку компонента и уменьшить нагрузку на сервер при большом количестве клиентов. По умолчанию свойство имеет значение **true**.
UseLocalizedCache	Предоставляет возможность использовать различные кэши в зависимости от выбранной локализации. По умолчанию свойство имеет значение **false**.

© 2003-2026 Stimulsoft

AllowLoadingCustomFontsToClientSide	Позволяет передавать пользовательские шрифты на клиентскую сторону и преобразовывать их в CSS стиль для корректного отображения текста на стороне HTML с заданным шрифтом. По умолчанию свойство имеет значение **false**.

Внешний вид (Appearance)

Наименование	Описание
IconSet	Предоставляет возможность установить набор иконок: › StiWebUIIconSet.Auto (значение по умолчанию) - устанавливает набор иконок автоматически. Для тем **Office2022** - используется набор иконок в стиле **Monoline**, для тем **Office2013** - используется набор иконок в стиле **Regular**. › StiWebUIIconSet.Monoline - устанавливает набор иконок в стиле **Monoline**; › StiWebUIIconSet.Regular - устанавливает набор иконок в стиле **Regular**.
CustomCss	Задает путь к CSS файлу стилей вьювера. Если данное свойство установлено, то стандартные стили выбранной темы загружены не будут. По умолчанию установлено пустое значение.
BackgroundColor	Задает цвет подложки вьювера. По умолчанию свойство имеет значение **White**.
PageBorderColor	Задает цвет границ страницы отчета. По умолчанию свойство имеет значение **Gray**.
RightToLeft	Устанавливает режим отображения **Right to Left** для элементов управления вьювера. По умолчанию свойство имеет значение **false**.
FullScreenMode	Устанавливает полноэкранный режим отображения вьювера. По умолчанию

© 2003-2026 Stimulsoft

	свойство имеет значение **false**.
ScrollbarsMode	Устанавливает режим отображения отчета с полосами прокрутки. По умолчанию свойство имеет значение **false**.
OpenLinksWindow	Задает целевое окно для открытия ссылок, содержащихся в отчете. По умолчанию свойство имеет значение **Blank** (новое окно).
OpenExportedReportWindow	Задает целевое окно для открытия файла экспорта из вьювера. По умолчанию свойство имеет значение **Blank** (новое окно).
DesignWindow	Задает целевое окно для открытия дизайнера отчетов. По умолчанию свойство имеет значение **Self** (текущее окно).
ShowTooltips	Включает отображение подсказок для элементов управления вьювера при наведении мыши. По умолчанию свойство имеет значение **true**.
ShowTooltipsHelp	Включает отображение ссылки на онлайн документацию в подсказках для элементов управления вьювера. По умолчанию свойство имеет значение **true**.
ShowDialogsHelp	Предоставляет возможность отображать или не отображать кнопку вызова справки в различных меню. По умолчанию, установлено значение **true**.
PageAlignment	Задает позицию страницы отчета в окне вьювера. Может принимать одно из указанных ниже значений перечисления **StiContentAlignment**: › **Left** – страница будет пристыкована к левому краю; › **Center** – страница будет центрирована (значение по умолчанию); › **Right** – страница будет пристыкована к правому краю.

© 2003-2026 Stimulsoft

ShowPageShadow	Включает отображение тени для страниц отчета. По умолчанию свойство имеет значение **true**.
BookmarksPrint	Включает печать закладок отчета на принтере (помимо самого отчета). По умолчанию свойство имеет значение **false**.
BookmarksTreeWidth	Устанавливает ширину панели закладок в пикселях. По умолчанию ширина имеет значение 180 пикселей.
ParametersPanelPosition	Задает позицию панели параметров отчета. Может принимать одно из указанных ниже значений перечисления **StiParametersPanelPosition**: › **Top** – панель будет пристыкована к верхнему краю (значение по умолчанию); › **Left** – панель будет пристыкована к левому краю.
ParametersPanelMaxHeight	Устанавливает максимальную высоту панели параметров в пикселях. По умолчанию максимальная высота имеет значение 300 пикселей.
ParametersPanelColumnsCount	Устанавливает количество колонок для отображения параметров отчета. По умолчанию разбиение производится на 2 колонки.
ParametersPanelSortDataItems	Устанавливает или выключает режим сортировки значений переменной. По умолчанию, опция установлена в значение **true**, т.е. значения переменной сортируются.
ParametersPanelDateFormat	Устанавливает формат даты и времени для переменных соответствующего типа на панели параметров. По умолчанию используется формат даты и времени, установленный браузером.
InterfaceType	Устанавливает тип используемого интерфейса

© 2003-2026 Stimulsoft

	вьювера. Может принимать одно из указанных ниже значений перечисления **StiInterfaceType**: › **Auto** – тип интерфейса вьювера будет выбран автоматически в зависимости от используемого устройства (значение по умолчанию); › **Mouse** – принудительное использование стандартного интерфейса для управления вьювером при помощи мыши; › **Touch** – принудительное использование Touch интерфейса для управления вьювером при помощи сенсорного экрана монитора, в этом режиме элементы интерфейса вьювера имеют увеличенные размеры для удобства управления; › **Mobile** – принудительное использование Mobile интерфейса для управления вьювером при помощи экрана смартфона, в этом режиме интерфейс вьювера имеет упрощенный вид и адаптирован для управления при помощи мобильного устройства.
AllowMobileMode	Включает или выключает возможность отображения отчета или дашборда в мобильном режиме. Если опция установлена в значение **false**, то мобильный режим просмотра не будет использоваться ни при каких обстоятельства. Если опция установлена в значение **true**, то мобильный режим просмотра будет использоваться запуске вьювера на мобильных устройствах. По умолчанию, опция установлена в значение **true**.
ChartRenderType	Устанавливает режим отображения диаграмм на странице отчета. Может принимать одно из указанных ниже значений перечисления

© 2003-2026 Stimulsoft

	StiChartRenderType: › **Image** – диаграммы отображаются в виде статичных изображений; › **Vector** – диаграммы отображаются в векторном режиме в виде SVG объекта; › **AnimatedVector** - диаграммы отображаются в векторном режиме в виде SVG объекта, элементы диаграммы появляются с анимацией (значение по умолчанию).
ReportDisplayMode	Устанавливает режим экспорта для отображения страниц отчета. Может принимать одно из указанных ниже значений перечисления **StiReportDisplayMode**: › **FromReport** - режим экспорта элементов отчета определяется из настроек шаблона - Div или Table; › **Table** – элементы отчета экспортируются с использованием HTML таблиц (значение по умолчанию); › **Div** – элементы отчета экспортируются с использованием DIV разметки; › **Span** - элементы отчета экспортируются с использованием SPAN разметки.
DatePickerFirstDayOfWeek	Устанавливает первый день недели для элемента выбора даты. Может принимать одно из указанных ниже значений перечисления **StiFirstDayOfWeek**: › **Auto** – автоматическое определение первого дня недели из настроек браузера (значение по умолчанию); › **Monday** – первым днем недели является понедельник; › **Sunday** – первым днем недели является воскресенье.

© 2003-2026 Stimulsoft

DatePickerIncludeCurrentDayForRanges	Предоставляет возможность включать или не включать текущий день в диапазон значений элемента **Выбор даты (Date Picker)**. По умолчанию, опция установлена в значение **false**, т.е. текущий день не включается в диапазон значений элемента.
AllowTouchZoom	Включает возможность изменения масштаба страницы отчета специальным двух пальцевым жестом (Pinch to Zoom) на сенсорном экране устройства. По умолчанию свойство имеет значение **true**.
ShowReportIsNotSpecifiedMessage	Предоставляет возможность отображать или не отображать уведомление, если отчет не задан. По умолчанию свойство имеет значение **true**.
PrintToPdfMode	Устанавливает режим печати в PDF документ. Может принимать значения: › **StiPrintToPdfMode.Hidden** - скрытый режим печати (по умолчанию); › **StiPrintToPdfMode.Popup** - документ PDF будет отображет перед печатью во всплывающем окне.
ImagesQuality	Получает или задает качество изображения, которое будет использоваться на странице вьювера. Может принимать значения: › **StiImagesQuality.Low** - низкое качество, используется для ускорения загрузки отчета и экономии памяти › **StiImagesQuality.Normal** - обычное качество, подходит для большинства ситуаций, установлено по умолчанию › **StiImagesQuality.High** - высокое качество, используется для мониторов высокой четкости, но может замедлять загрузку страниц
CombineReportPages	Предоставляет возможность объединить обработанные страницы шаблона отчета в

© 2003-2026 Stimulsoft

	один шаблон или представить каждую страницу шаблона отдельной вкладкой во вьювере. По умолчанию, опция установлена в значение **false**, т.е. каждая страница шаблона отчета будет представлена отдельной вкладкой во вьювере.
AllowPropagationEvents	Предоставляет возможность разрешить распространение событий нажатия клавиш, когда вьювер отчетов не в фокусе. По умолчанию свойство имеет значение **true**.
DashboardFilterElementItemHeight	Предоставляет возможность изменить высоту флажка в элементе дашборда **Список (List Box)**.

Панель инструментов (Toolbar)

Наименование	Описание
Visible	Включает отображение панели инструментов вьювера. По умолчанию свойство имеет значение **true**.
DisplayMode	Задает режим отображения панели инструментов вьювера. Может принимать одно из указанных ниже значений перечисления **StiToolbarDisplayMode**: › **Simple** – простой режим отображения, все элементы управления расположены на одной панели управления (значение по умолчанию); › **Separated** – раздельный режим отображения, панель управления разделена на верхнюю и нижнюю.
BackgroundColor	Задает цвет панели инструментов вьювера. По умолчанию используется цветовое оформление выбранной темы.
BorderColor	Задает цвет границ панели инструментов

© 2003-2026 Stimulsoft

	вьювера. По умолчанию используется цветовое оформление выбранной темы.
FontColor	Задает цвет текста для панели инструментов и меню вьювера. По умолчанию используется цветовое оформление выбранной темы.
FontFamily	Задает шрифт для панели инструментов и меню вьювера. По умолчанию используется шрифт выбранной темы.
Alignment	Устанавливает режим выравнивания элементов управления на панели инструментов вьювера. Может принимать одно из указанных ниже значений перечисления **StiContentAlignment**: › **Left** – элементы будут пристыкованы к левому краю; › **Center** – элементы будут располагаться по центру панели; › **Right** – элементы будут пристыкованы к правому краю; › **Default** – выравнивание зависит от свойства RightToLeft (значение по умолчанию)
ShowButtonCaptions	Включает отображение текста кнопок панели инструментов вьювера. По умолчанию свойство имеет значение **true**.
ShowPrintButton	Включает отображение кнопки **Печать (Print)** на панели инструментов вьювера. По умолчанию свойство имеет значение **true**.
ShowOpenButton	Включает отображение кнопки **Открыть (Open)** на панели инструментов вьювера при просмотре отчетов или панели индикаторов (дашбордов). По умолчанию свойство имеет значение **true**.
ShowSaveButton	Включает отображение кнопки **Сохранить (Save)** на панели инструментов вьювера при

© 2003-2026 Stimulsoft

	просмотре отчетов или панели индикаторов (дашбордов). По умолчанию свойство имеет значение **true**.
ShowSendEmailButton	Включает отображение кнопки **Отправиль Email (Send Email)** на панели инструментов вьювера. По умолчанию свойство имеет значение **false**. Также, необходимо определить действие EmailReport.
ShowFindButton	Включает отображение кнопки **Поиск (Find)** на панели инструментов вьювера. По умолчанию свойство имеет значение **true**.
ShowBookmarksButton	Включает отображение кнопки **Закладки (Bookmarks)** на панели инструментов вьювера. По умолчанию свойство имеет значение **true**. Если указанная кнопка скрыта, то панель закладок не будет отображена даже при наличии закладок в отчете.
ShowParametersButton	Включает отображение кнопки **Параметры (Parameters)** на панели инструментов вьювера. По умолчанию свойство имеет значение **true**. Если указанная кнопка скрыта, то панель параметров не будет отображена даже при наличии параметров в отчете.
ShowResourcesButton	Включает отображение кнопки **Ресурсы (Resources)** на панели инструментов вьювера. По умолчанию свойство имеет значение **true**. Если указанная кнопка скрыта, то панель ресурсов не будет отображена.
ShowEditorButton	Включает отображение кнопки **Редактор (Editor)** на панели инструментов вьювера при просмотре отчетов. По умолчанию свойство имеет значение **true**.
ShowFullScreenButton	Включает отображение кнопки **Полноэкранный просмотр (Full Screen)** на панели инструментов вьювера при просмотре отчетов или панели индикаторов (дашбордов). По умолчанию свойство имеет

© 2003-2026 Stimulsoft

	значение **true**.
ShowFirstPageButton	Включает отображение кнопки **Первая страница (First Page)** на панели инструментов вьювера. По умолчанию свойство имеет значение **true**.
ShowPreviousPageButton	Включает отображение кнопки **Предыдущая страница (Previous Page)** на панели инструментов вьювера. По умолчанию свойство имеет значение **true**.
ShowCurrentPageControl	Включает отображение индикатора текущей страницы отчета. По умолчанию свойство имеет значение **true**.
ShowNextPageButton	Включает отображение кнопки **Следующая страница (Next Page)** на панели инструментов вьювера. По умолчанию свойство имеет значение **true**.
ShowLastPageButton	Включает отображение кнопки **Последняя страница (Last Page)** на панели инструментов вьювер. По умолчанию свойство имеет значение **true**.
ShowZoomButton	Включает отображение кнопки выбора масштаба отчета. По умолчанию свойство имеет значение **true**.
ShowViewModeButton	Включает отображение кнопки выбора режима отображения страниц отчета. По умолчанию свойство имеет значение **true**.
ShowDesignButton	Включает отображение кнопки **Дизайн (Design)** на панели инструментов вьюверапри просмотре отчетов или панели индикаторов (дашбордов). По умолчанию свойство имеет значение **false**.
ShowAboutButton	Включает отображение кнопки **О программе (About)** на панели инструментов вьювера. По умолчанию свойство имеет значение **true**.
ShowRefreshButton	Включает или выключает отображение

© 2003-2026 Stimulsoft

	кнопки **Обновить (Refresh)** на панели инструментов вьювера при просмотре панелей индикаторов (дашбордов). По умолчанию свойство имеет значение **true**.
ShowPinToolbarButton	Включает отображение кнопки **Прикрепить панель (Pin Toolbar)** на панели инструментов вьювера. Кнопка доступна только в мобильном режиме вьювера. По умолчанию свойство имеет значение **true**.
PrintDestination	Устанавливает режим печати отчета. Может принимать одно из указанных ниже значений перечисления **StiPrintDestination**: › **Default** – будет отображено меню с предоставлением выбора режима печати (значение по умолчанию); › **Pdf** – печать будет осуществлена в формате PDF; › **Direct** – печать будет осуществлена в формате HTML непосредственно на принтер, будет отображен системный диалог печати; › **PopupWindow** – печать будет осуществлена в формате HTML через всплывающее окно предварительного просмотра отчета.
ViewMode	Устанавливает режим отображения страниц отчета. Может принимать одно из указанных ниже значений перечисления **StiWebViewMode**: › **SinglePage** – отображается одна страница отчета, выбранная на панели инструментов вьювера (значение по умолчанию); › **Continuous** – отображаются все страницы отчета в виде ленты; › **MultiplePages** – отображаются все страницы отчета в виде таблицы.
Zoom	Задает масштаб отображения страниц отчета.

© 2003-2026 Stimulsoft

	По умолчанию установлен масштаб 100 процентов. Допускаются значения от 10 до 500 процентов. Также допускается устанавливать одно из следующих значений: › **StiZoomMode.PageWidth** – при запуске вьювера будет установлен масштаб, необходимый для отображения отчета по ширине страницы; › **StiZoomMode.PageHeight** – при запуске вьювера будет установлен масштаб, необходимый для отображения отчета по высоте страницы.
MenuAnimation	Включает анимацию при появлении/скрытии меню вьювера. По умолчанию свойство имеет значение **true**.
ShowMenuMode	Устанавливает режим отображения меню вьювера. Может принимать одно из указанных ниже значений перечисления **StiShowMenuMode**: › **Click** – отображение по клику мыши (значение по умолчанию); › **Hover** – отображение по наведении курсора мыши.
AutoHide	Включает автоматическое сворачивание панели инструментов вьювера. Свойство работает только в мобильном режиме вьювера. По умолчанию свойство имеет значение **false**.

Экспорт отчета (Export)

Наименование	Описание
DefaultSettings	Данная группа свойств предоставляет возможность задать настройки экспортов по

© 2003-2026 Stimulsoft

	умолчанию для каждого типа экспорта. Эти настройки будут применены к диалоговым окнам экспортов при запуске вьювера, либо к отчету в случае отключения отображения диалоговых окон экспортов.
StoreExportSettings	Включает режим сохранения выбранных настроек в диалоговых окнах экспорта. Настройки будут храниться в cookies браузера. По умолчанию свойство имеет значение **true**.
ShowExportDialog	Включает отображение диалогового окна параметров экспорта. Если свойство имеет значение **false**, то экспортирование будет производиться с заданными по умолчанию настройками. По умолчанию свойство имеет значение **true**.
ShowExportToDocument	Включает отображение пункта меню экспорта **Файл документа (Document File)**. По умолчанию свойство имеет значение **true**.
ShowExportToPdf	Включает отображение пункта меню экспорта **Adobe PDF файл (Adobe PDF File)** при просмотре отчетов, и пункт **Adobe PDF** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToXps	Включает отображение пункта меню экспорта **Microsoft XPS файл (Microsoft XPS File)**. По умолчанию свойство имеет значение **false**.
ShowExportToPowerPoint	Включает отображение пункта меню экспорта **Microsoft PowerPoint 2007/2010 файл (Microsoft PowerPoint 2007/2010 File)**. По умолчанию свойство имеет значение **true**.
ShowExportToHtml	Включает отображение пункта меню экспорта **HTML файл (HTML File)**. По умолчанию свойство имеет значение **true**.
ShowExportToHtml5	Включает отображение пункта меню

© 2003-2026 Stimulsoft

	экспорта **HTML5 файл (HTML5 File)**. По умолчанию свойство имеет значение **true**.
ShowExportToMht	Включает отображение пункта меню экспорта **MHT Web архив (MHT Web Archive)**. По умолчанию свойство имеет значение **true**.
ShowExportToText	Включает отображение пункта меню экспорта **Text файл (Text File)**. По умолчанию свойство имеет значение **true**.
ShowExportToRtf	Включает отображение пункта меню экспорта **Rich Text файл (Rich Text File)**. По умолчанию свойство имеет значение **true**.
ShowExportToWord	Включает отображение пункта меню экспорта **Microsoft Word 2007/2010 файл (Microsoft Word 2007/2010 File)**. По умолчанию свойство имеет значение **true**.
ShowExportToOpenDocumentWriter	Включает отображение пункта меню экспорта **OpenDocument Writer файл (OpenDocument Writer File)**. По умолчанию свойство имеет значение **true**.
ShowExportToExcelBiff	Включает отображение пункта меню экспорта **Microsoft Excel файл (Microsoft Excel File)**. По умолчанию свойство имеет значение **true**.
ShowExportToExcelXml	Включает отображение пункта меню экспорта **Microsoft Excel Xml файл (Microsoft Excel Xml File)**. По умолчанию свойство имеет значение **true**.
ShowExportToExcel	Включает отображение пункта меню экспорта **Microsoft Excel 2007/2010 файл (Microsoft Excel 2007/2010 File)** при просмотре отчетов, и пункт **Microsoft Excel** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToOpenDocumentCalc	Включает отображение пункта меню экспорта **OpenDocument Calc файл**

© 2003-2026 Stimulsoft

	(OpenDocument Calc File). По умолчанию свойство имеет значение **true**.
ShowExportToCsv	Включает отображение пункта меню экспорта **CSV файл (CSV File)**. По умолчанию свойство имеет значение **true**.
ShowExportToDbf	Включает отображение пункта меню экспорта **DBF файл (DBF File)**. По умолчанию свойство имеет значение **true**.
ShowExportToXml	Включает отображение пункта меню экспорта **XML файл (XML File)**. По умолчанию свойство имеет значение **true**.
ShowExportToDif	Включает отображение пункта меню экспорта **Data Interchange формат (DIF) файл (Data Interchange Format (DIF) File)**. По умолчанию свойство имеет значение **true**.
ShowExportToSylk	Включает отображение пункта меню экспорта **Symbolic Link (SYLK) файл (Symbolic Link (SYLK) File)**. По умолчанию свойство имеет значение **true**.
ShowExportToJson	Включает отображение пункта меню экспорта **JSON файл (JSON File)**. По умолчанию свойство имеет значение **true**.
ShowExportToImageBmp	Включает отображение пункта меню экспорта **BMP рисунок (BMP Image)** при просмотре отчетов, и пункт **BMP** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToImageGif	Включает отображение пункта меню экспорта **GIF рисунок (GIF Image)** при просмотре отчетов, и пункт **GIF** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToImageJpeg	Включает отображение пункта меню экспорта **JPEG рисунок (JPEG Image)** при просмотре отчетов, и пункт **JPEG** при просмотре панелей индикаторов. По

© 2003-2026 Stimulsoft

	умолчанию свойство имеет значение **true**.
ShowExportToImagePcx	Включает отображение пункта меню экспорта **PCX рисунок (PCX Image)** при просмотре отчетов, и пункт **PCX** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToImagePng	Включает отображение пункта меню экспорта **PNG рисунок (PNG Image)** при просмотре отчетов, и пункт **PNG** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToImageTiff	Включает отображение пункта меню экспорта **TIFF рисунок (TIFF Image)** при просмотре отчетов, и пункт **TIFF** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToImageSvg	Включает отображение пункта меню экспорта **Scalable Vector Graphics (SVG) файл (Scalable Vector Graphics (SVG) File)** при просмотре отчетов, и пункт **SVG** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToImageSvgz	Включает отображение пункта меню экспорта **Compressed SVG (SVGZ) файл (Compressed SVG (SVGZ) File)** при просмотре отчетов, и пункт **SVGZ** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowOpenAfterExport	Предоставляет возможность отображать или не отображать параметр **Открыть после экспорта (Open After Export)** в меню настроек экспорта. По умолчанию свойство имеет значение **true**.

Отправка отчета по Email (Email)

© 2003-2026 Stimulsoft

Наименование	Описание
ShowEmailDialog	Включает отображение диалогового окна параметров отправки отчета по Email. Если диалоговое окно отключено, то отправка по Email будет производиться с настройками, заданными на стороне сервера в действии **EmailReport**. По умолчанию свойство имеет значение **true**.
ShowExportDialog	Включает отображение диалогового окна параметров экспорта при отправке Email. Если свойство имеет значение **false**, то экспортирование будет производиться с заданными по умолчанию настройками. По умолчанию свойство имеет значение **true**.
DefaultEmailAddress	Устанавливает Email получателя по умолчанию, т.е. адрес на который будет отправлено письмо с вложенным отчетом.
DefaultEmailSubject	Устанавливает тему (заголовок) электронного письма по умолчанию.
DefaultEmailMessage	Устанавливает сообщение (текст) электронного письма по умолчанию.
DefaultEmailReplyTo	Устанавливает сообщение (текст) по умолчанию для ответа на сообщение, созданное во вьювере.

6.2 HTML5 дизайнер

Примеры

Ознакомьтесь с примерами по работе с компонентом ASP.NET Core Razor HTML5 Designer GitHub. Все примеры представляют собой отдельные проекты, сгруппированные в один solution для Visual Studio.

Компонент **HTML5 Designer** (**StiNetCoreDesigner**) предназначен для редактирования отчетов в окне браузера. При этом не требуется устанавливать

© 2003-2026 Stimulsoft

у клиента .NET Framework, ActiveX компоненты или какие-либо специальные плагины. Все, что необходимо – это любой современный Web-браузер.

При помощи **HTML5 Designer** можно создавать, редактировать, сохранять, просматривать отчеты и печатать отчеты на любом компьютере с любой установленной операционной системой. Так как дизайнер использует только HTML и JavaScript технологии, он может быть запущен на устройствах, где нет поддержки Flash или Silverlight – планшеты, смартфоны. Также дизайнер поддерживает Touch интерфейс, который автоматически включается при использовании устройств с сенсорным экраном.

Компонент **HTML5 Designer** использует технологию AJAX для выполнения всех действий над редактируемым отчетом, что позволяет избавиться от перезагрузки всей страницы, а также экономить Web-трафик и повысить скорость работы. Для построения отчетов используется ядро генератора отчетов, построенное с использованием .NET Core технологии. Данная технология является кроссплатформенной, что позволяет разворачивать приложение на серверах, использующих операционные системы Windows, macOS и Linux.

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для использования **HTML5 Designer** в Web-проекте, необходимо установить NuGet пакет Stimulsoft.Reports.Web.NetCore:

- Выберите пункт "Manage NuGet Packages..." в контекстном меню проекта;
- На вкладке Browse, в строке поиска, укажите Stimulsoft.Reports.Web.NetCore;
- Выделите элемент, определите версию пакета и нажмите кнопку **Установить (Install)**. При обновлении пакета, следует нажать кнопку **Обновить (Update)**.

Для того, чтобы добавить возможности создания и редактирования дашбордов в Web-проекте, установить NuGet пакет Stimulsoft.Dashboards.Web.NetCore, (данный пакет связан с пакетом Stimulsoft.Reports.Web.NetCore, при его

© 2003-2026 Stimulsoft

отсутствии он будет установлен автоматически):
› Выберите пункт "Manage NuGet Packages..." в контекстном меню проекта;
› На вкладке Browse, в строке поиска, укажите Stimulsoft.Dashboards.Web.NetCore;
› Выделите элемент, определите версию пакета и нажмите кнопку **Установить (Install)**. При обновлении пакета, следует нажать кнопку **Обновить (Update)**.

- Как это работает
- Дополнительные возможности просмотра
- Активация
- Время ожидания
- Редактирование отчета и дашборда
- Локализация дизайнера отчетов
- Создание нового отчета и нового дашборда
- Использование тем
- Просмотр
- Кэширование
- Сохранение отчета и дашборда
- Настройки дизайнера отчетов
- Вспомогательные методы

6.2.1 Как это работает

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для запуска дизайнера требуется разместить на странице компонент **StiNetCoreDesigner**, задать ему необходимые настройки, а в обработчике событий страницы определить необходимые действия. При запуске дизайнера отчетов происходят следующие действия:
› .NET Core компонент генерирует HTML и JavaScript код, необходимый для отображения и работы дизайнера;

© 2003-2026 Stimulsoft

- По окончании вывода компонента запускается JavaScript метод, который запрашивает на стороне сервера шаблон отчета, и отображает его в окне редактирования;
- Различные действия, такие как предварительный просмотр отчета, сохранение шаблона отчета, экспорт отчета, применение параметров, сортировка и детализация отчета, вызывают определенное действие на стороне сервера, в котором можно выполнить необходимые манипуляции с отчетом.

6.2.2 Активация

После приобретения продукта Stimulsoft вам необходимо активировать лицензию для используемых компонентов. Это можно выполнить указав лицензионный ключ или загрузив файл с лицензионным ключом. Ниже представлен пример активации компонента **StiNetCoreDesigner**.

Index.cshtml.cs

```
...
//Activation with using license code
public class IndexModel : PageModel
{
   static IndexModel()
   {
      Stimulsoft.Base.StiLicense.Key = "Your activation code...";
   }
}

//Activation with using license file
public class IndexModel : PageModel
{
   public IndexModel(IWebHostEnvironment webHostEnvironment)
   {
      var path = Path.Combine(webHostEnvironment.ContentRootPath, "Content\
      \license.key");
      Stimulsoft.Base.StiLicense.LoadFromFile(path);
   }
}
...
```

Получить лицензионный ключ или скачать файл с лицензионным ключом, вы можете в личном кабинете пользователя. Для авторизации в кабинете используйте имя пользователя и пароль, указанные при покупке подписки продукта.

© 2003-2026 Stimulsoft

6.2.3 Редактирование отчета и дашборда

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для редактирования шаблона отчета необходимо добавить на страницу компонент **StiNetCoreDesigner** и задать ему минимально необходимые настройки, а в обработчике событий страницы определить необходимые действия.

Index.cshtml

```
...
@Html.StiNetCoreDesigner(new StiNetCoreDesignerOptions() {
   Actions =
   {
      GetReport = "GetReport",
      DesignerEvent = "DesignerEvent"
   }
})
...
```

© 2003-2026 Stimulsoft

Index.cshtml.cs

```
...
public IActionResult OnPostGetReport()
{
   // Create the report object
   StiReport report = new StiReport();

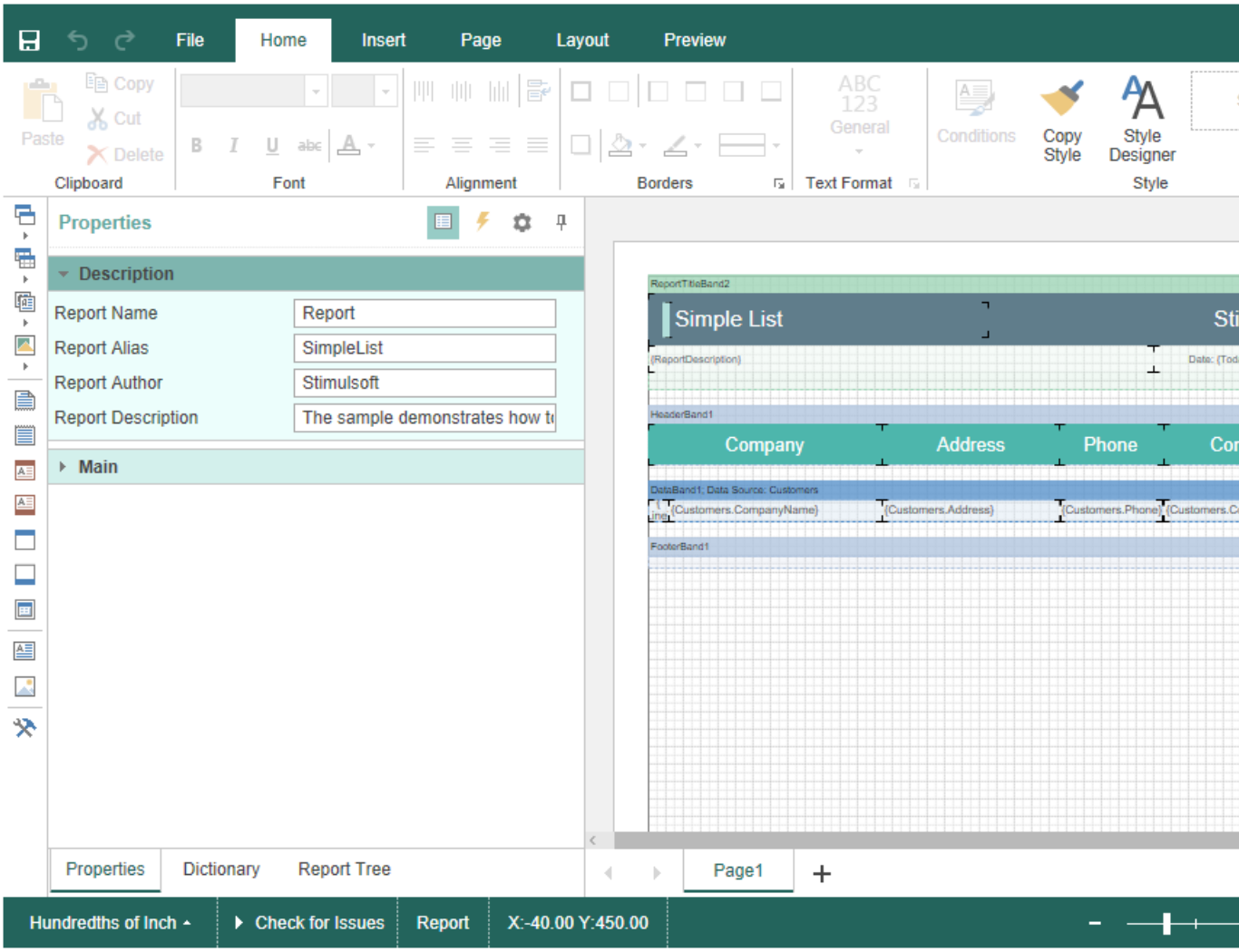
   // Load report or dashboard
   report.Load(StiNetCoreHelper.MapPath(this, "Reports/SimpleList.mrt"));
   //report.Load(StiNetCoreHelper.MapPath(this, "Reports/Dashboard.mrt"));

   return StiNetCoreDesigner.GetReportResult(this, report);
}

public IActionResult OnGetDesignerEvent()
{
   return StiNetCoreDesigner.DesignerEventResult(this);
}

public IActionResult OnPostDesignerEvent()
{
   return StiNetCoreDesigner.DesignerEventResult(this);
}
...
```

© 2003-2026 Stimulsoft



Действие **GetReport** предназначено для загрузки редактируемого шаблона отчета, оно вызывается автоматически после загрузки дизайнера отчетов. Действие **DesignerEvent** предназначено для обработки различных дополнительных действий дизайнера, таких как работа с данными, работа с компонентами, предварительный просмотр отчета и других.

© 2003-2026 Stimulsoft

Информация

Действие **DesignerEvent** является обязательным. Без него корректная работа дизайнера невозможна. Действие вызывается для двух типов запросов: **OnGet** - компонент запрашивает необходимые для работы ресурсы, такие как CSS-стили, JS-скрипты и изображения; **OnPost** - все остальные действия дизайнера.

6.2.4 Создание нового отчета и нового дашборда

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для запуска дизайнера с новым (пустым) отчетом достаточно создать новый отчет в действии **GetReport** и вернуть его дизайнеру. При необходимости, можно предварительно загрузить данные для отчета, либо выполнить какие-либо другие необходимые действия.

Index.cshtml

```
...
@Html.StiNetCoreDesigner(new StiNetCoreDesignerOptions() {
   Actions =
   {
      GetReport = "GetReport"
   }
})
...
```

© 2003-2026 Stimulsoft

Index.cshtml.cs

```
...
public IActionResult OnPostGetReport()
{
   StiReport report = new StiReport();
   //var newDashboard = StiReport.CreateNewDashboard();

   return StiNetCoreDesigner.GetReportResult(this, report);
   //return StiNetCoreDesigner.GetReportResult(this, newDashboard);
}
...
```

Также новый отчет можно создать, используя главное меню дизайнера. Для того, чтобы предварительно загрузить данные для нового отчета, либо выполнить какие-либо другие необходимые действия, предназначено действие **CreateReport**. Данное действие будет вызвано при создании нового пустого отчета из главного меню, либо при создании отчета с использованием мастера.

Index.cshtml

```
...
@Html.StiNetCoreDesigner(new StiNetCoreDesignerOptions() {
   Actions =
   {
      CreateReport = "CreateReport"
   }
})
...
```

Index.cshtml.cs

```
...
public IActionResult OnPostCreateReport()
{
   StiReport report = new StiReport();
   //var newDashboard = StiReport.CreateNewDashboard();

   // Register data for the new report, if necessary
   DataSet data = new DataSet("Demo");
   data.ReadXml(StiNetCoreHelper.MapPath(this, "Data/Demo.xml"));
   report.RegData(data);
   //newDashboard.RegData(data);
   report.Dictionary.Synchronize();
   //newDashboard.Dictionary.Synchronize();

   return StiNetCoreDesigner.GetReportResult(this, report);
   //return StiNetCoreDesigner.GetReportResult(this, newDashboard);
}
...
```

© 2003-2026 Stimulsoft

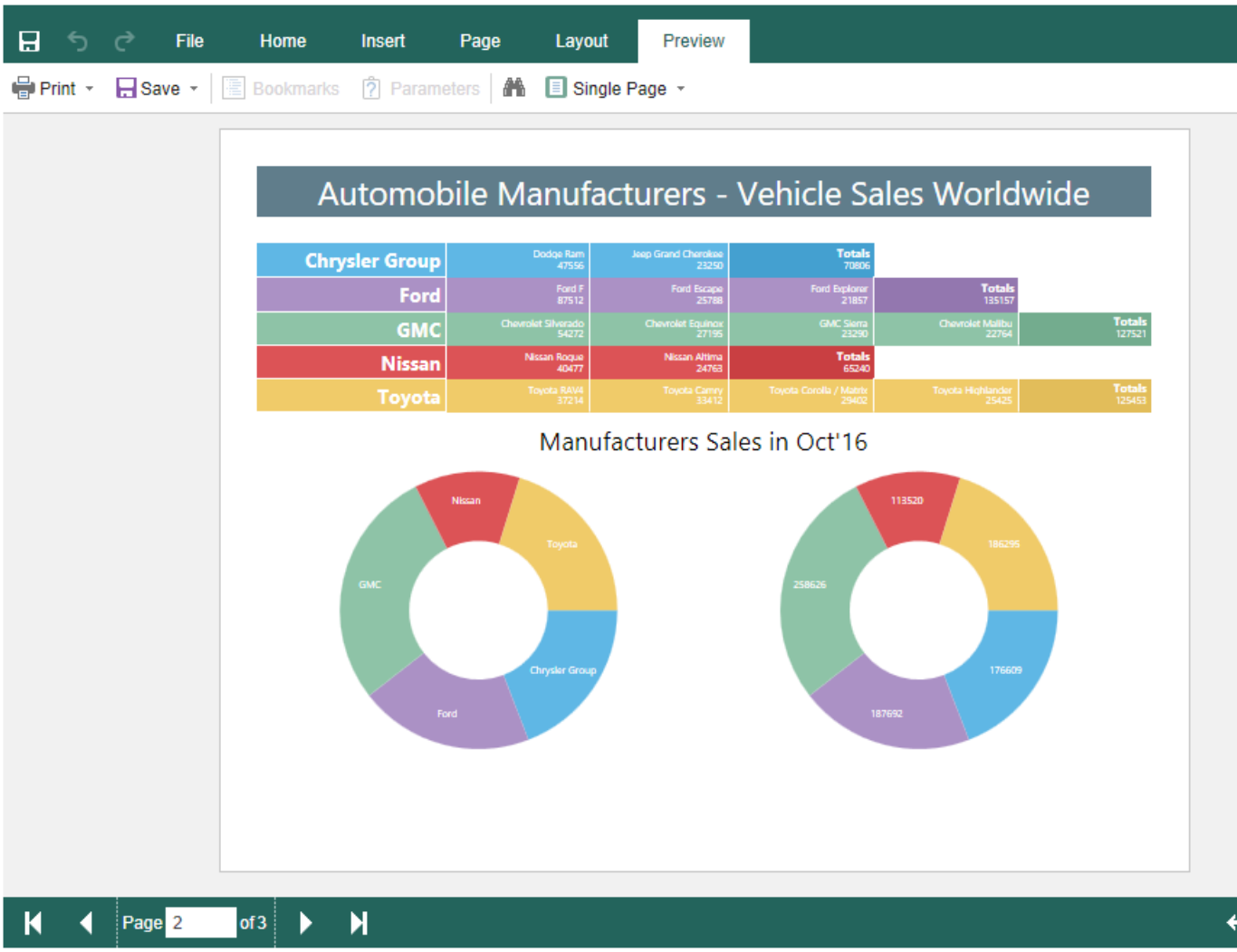
6.2.5 Просмотр

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

У компонента **HTML5 Designer** предусмотрен режим просмотра редактируемого отчета. Для просмотра достаточно перейти на соответствующую вкладку в окне дизайнера. Шаблон отчета будет передан на сторону сервера, построен и отображен во встроенном вьювере.

© 2003-2026 Stimulsoft



Перед просмотром отчета предусмотрена возможность выполнить какие-либо необходимые действия, например, подключить данные для отчета. Для этого достаточно определить специальное действие **PreviewReport**, которое будет вызвано перед просмотром отчета. Действие **PreviewReport** вызывается перед подготовкой и построением отчета для просмотра, до его сохранения в кэш.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.StiNetCoreDesigner(new StiNetCoreDesignerOptions() {
   Actions =
   {
      PreviewReport = "PreviewReport"
   }
})
...
```

Index.cshtml.cs

```
...
public IActionResult OnPostPreviewReport()
{
   StiReport report = StiNetCoreDesigner.GetActionReportObject(this);

   DataSet data = new DataSet("Demo");
   data.ReadXml(StiNetCoreHelper.MapPath(this, "Data/Demo.xml"));
   report.RegData(data);

   return StiNetCoreDesigner.PreviewReportResult(this, report);
}
...
```

При необходимости выполнить действия над отчетом непосредственно перед отображением отчета, можно воспользоваться специальным действием **GetPreviewReport**, которое вызывается после запроса подготовленного отчета из кэша.

Index.cshtml

```
...
@Html.StiNetCoreDesigner(new StiNetCoreDesignerOptions() {
   Actions =
   {
      GetPreviewReport = "GetPreviewReport"
   }
})
...
```

© 2003-2026 Stimulsoft

Index.cshtml.cs

```
...
public IActionResult OnPostGetPreviewReport()
{
   StiReport report = StiNetCoreDesigner.GetActionReportObject(this);

   DataSet data = new DataSet("Demo");
   data.ReadXml(StiNetCoreHelper.MapPath(this, "Data/Demo.xml"));
   report.RegData(data);
   //report.IsRendered = false;

   return StiNetCoreDesigner.PreviewReportResult(this, report);
}
...
```

Информация

Так как в данном событии передаётся уже подготовленный для просмотра отчет, то при необходимости выполнить повторное построение, необходимо установить флаг **report.IsRendered = false**.

6.2.6 Дополнительные возможности просмотра

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Окно предварительного просмотра отчета компонента **HTML5 Designer** представляет собой полноценный интерактивный **HTML5 Viewer**, который умеет осуществлять печать отчета и экспортирование отчета, поддерживает работу с параметрами отчета. Также поддерживаются интерактивные действия, такие как динамическая сортировка, детализация, сворачивание. Для использования указанных возможностей не требуется каких-либо дополнительных настроек дизайнера отчетов.

В любом из вышеперечисленных действий допускается производить манипуляции с шаблоном отчета, например, изменять их свойства и параметры,

© 2003-2026 Stimulsoft

подключать новые данные для построения.

Index.cshtml

```
...
@Html.StiNetCoreDesigner(new StiNetCoreDesignerOptions() {
   Actions =
   {
      ExportReport = "ExportReport"
   }
})
...
```

Index.cshtml.cs

```
...
public IActionResult OnPostExportReport()
{
   StiReport report = StiNetCoreDesigner.GetActionReportObject(this);
   // ...

   return StiNetCoreDesigner.ExportReportResult(this, report);
}
...
```

Информация

Если какие-либо из указанных дополнительных возможностей предварительного просмотра отчета не требуются (например, экспортирование или печать отчета), можно отключить их, используя соответствующие свойства компонента **HTML5 Designer**.

6.2.7 Сохранение отчета и дашборда

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

В компоненте **HTML5 Designer** предусмотрено два варианта сохранения отчета,

© 2003-2026 Stimulsoft

которые доступны в главном меню и на главной панели дизайнера: **Сохранить (Save)** и **Сохранить как (Save As)**. В свою очередь, каждый из этих вариантов сохранения имеет свои режимы и настройки.

Сохранение отчета или дашборда на стороне сервера

Для сохранения редактируемого отчета на стороне сервера необходимо определить действие **SaveReport**, которое будет вызвано при выборе пункта **Сохранить (Save)** в главном меню, либо нажатии на кнопку **Сохранить (Save)** на главной панели дизайнера.

Index.cshtml

```
...
@Html.StiNetCoreDesigner(new StiNetCoreDesignerOptions() {
   Actions =
   {
      SaveReport = "SaveReport"
   }
})
...
```

Index.cshtml.cs

```
...
public IActionResult OnPostSaveReport()
{
   StiReport report = StiNetCoreDesigner.GetReportObject(this);

   // Save the report template
   // ...

   return StiNetCoreDesigner.SaveReportResult(this);
}
...
```

Данное действие возвращает ответ клиентской стороне дизайнера о результате сохранения отчета. Если необходимо, после сохранения отчета предусмотрена возможность отобразить диалоговое окно с ошибкой или текстовым сообщением.

© 2003-2026 Stimulsoft

Index.cshtml.cs

```
...
public IActionResult OnPostSaveReport()
{
   StiReport report = StiNetCoreDesigner.GetReportObject(this);

   // Save the report template
   // ...

   // Completion of the report saving with message dialog box
   return StiNetCoreDesigner.SaveReportResult(this, "Some message after
   saving");
   //return Content("{"infoMessage":"Some info message after saving"}");
   //return Content("{\"warningMessage\":\"Some info message after saving
   \"}");
}
...
```

Если необходимо, вы можете получить доступ к оригинальному имени отчета или имени отчета из диалога сохранения.

Index.cshtml.cs

```
...
public IActionResult OnPostSaveReport()
{
   StiRequestParams requestParams = StiNetCoreDesigner.GetRequestParams();
   StiReport report = StiNetCoreDesigner.GetReportObject();

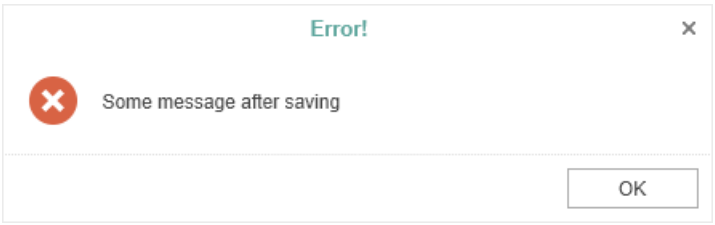
   //Report name from designer save dialog
   string savingReportName = requestParams.Designer.FileName;

   //Original report name from properties
   string originalReportName = report.ReportName;

   return StiNetCoreDesigner.SaveReportResult(this, "Some message after
   saving");
}
...
```

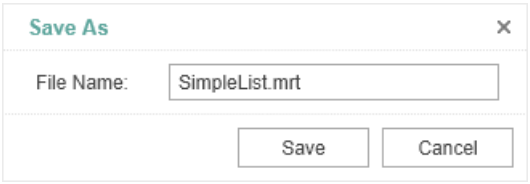
В этом случае, будет отображено диалоговое с указанным текстом. Текст может содержать как сообщение об ошибке сохранения либо предупреждение, так и любое другое сообщение.

© 2003-2026 Stimulsoft



Сохранение отчета или дашборда на стороне клиента

Для сохранения редактируемого отчета на стороне клиента в виде файла не требуется дополнительных настроек дизайнера. Достаточно выбрать пункт главного меню **Сохранить как (Save As)**, при нажатии на который отобразится диалог сохранения файла. В данном диалоге можно изменить имя файла отчета, после чего файл будет сохранен на локальном диске компьютера.



Компонент **HTML5 Designer** предоставляет возможность изменить поведение указанного варианта сохранения, для этого в настройках дизайнера предусмотрено специальное действие **SaveReportAs**. При определении этого действия сохранение отчета будет происходить на стороне сервера, работа данного действия будет аналогична действию **SaveReport**.

Index.cshtml

```
...
@Html.StiNetCoreDesigner(new StiNetCoreDesignerOptions() {
   Actions =
   {
      SaveReportAs = "SaveReportAs"
   }
})
...
```

© 2003-2026 Stimulsoft

Index.cshtml.cs

```
...
public IActionResult OnPostSaveReportAs()
{
   StiReport report = StiNetCoreDesigner.GetReportObject(this);

   // Save the report template
   // ...

   return StiNetCoreDesigner.SaveReportResult(this);
}
...
```

Чтобы получить наименование отчета из диалогового окна сохранения, используйте следующий код.

Index.cshtml.cs

```
public IActionResult OnPostSaveReportAs()
{
   StiReport report = StiNetCoreDesigner.GetReportObject(this);
   var requestParams = StiNetCoreDesigner.GetRequestParams(this);
   var reportName = requestParams.Designer.FileName;

   return StiNetCoreDesigner.SaveReportResult(this);
}
```

Настройки сохранения

Сохранение отчета производиться в фоновом режиме, т.е. без перезагрузки страницы в окне браузера. Если необходимо каким-либо образом визуально управлять процессом сохранения отчета, следует изменить значение свойства **SaveReportMode** (либо **SaveReportAsMode**) дизайнера на одно из трех указанных значений: **Hidden** (значение по умолчанию), **Visible** или **NewWindow**.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.StiNetCoreDesigner(new StiNetCoreDesignerOptions() {
   Actions =
   {
      SaveReportAs = "SaveReportAs"
   },
   Behavior =
   {
      SaveReportAsMode = StiSaveMode.Visible
   }
})
...
```

Если свойство **SaveReportMode** установлено в значение **Visible**, то действие сохранения отчета будет вызвано в текущем окне браузера в обычное (видимом) режиме при помощи POST-запроса. Если свойство **SaveReportMode** установлено в значение **NewWindow**, то действие сохранения отчета будет вызвано в новом окне браузера. По умолчанию данное свойство установлено в значение **Hidden**, т.е. действие сохранения отчета вызывается в фоне при помощи AJAX-запроса и никак не отображается в окне браузера. Для свойства **SaveReportAsMode** применимы те же значения и поведение.

6.2.8 Локализация дизайнера

Компонент **HTML5 Designer** поддерживает полную локализацию своего интерфейса. Для того чтобы локализовать интерфейс дизайнера отчетов на необходимый язык, используется специальное свойство **Localization**. В качестве значения данного свойства необходимо указать путь к XML-файлу локализации (относительный либо абсолютный).

Index.cshtml

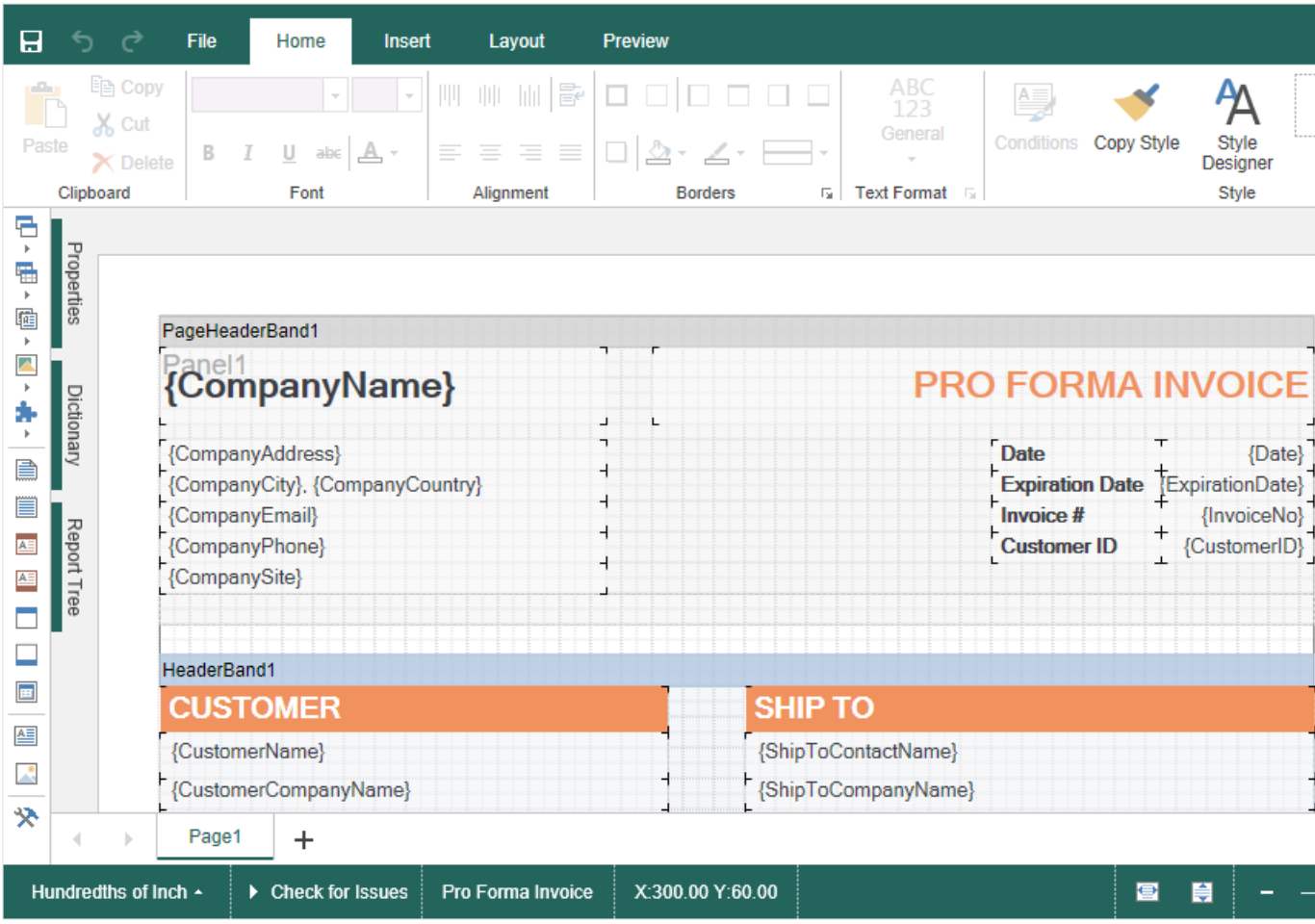
```
...
@Html.StiNetCoreDesigner(new StiNetCoreDesignerOptions() {
   Localization = "Localization/en.xml"
})
...
```

Интерфейс дизайнера отчетов позволяет выбрать необходимую локализацию из доступного списка. Для этого в качестве значения свойства **LocalizationDirectory** необходимо указать папку, в которой хранятся локализационные XML-файлы.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.StiNetCoreDesigner(new StiNetCoreDesignerOptions() {
   Localization = "Localization/en.xml",
   LocalizationDirectory = "Localization"
})
...
```



Информация

Если задано значение для свойства **Localization**, то при запуске дизайнера отчетов всегда будет применяться указанная в этом свойстве локализация. Если значение свойства не задано, то будет автоматически загружена локализация, выбранная из списка доступных локализаций на панели дизайнера отчетов.

© 2003-2026 Stimulsoft

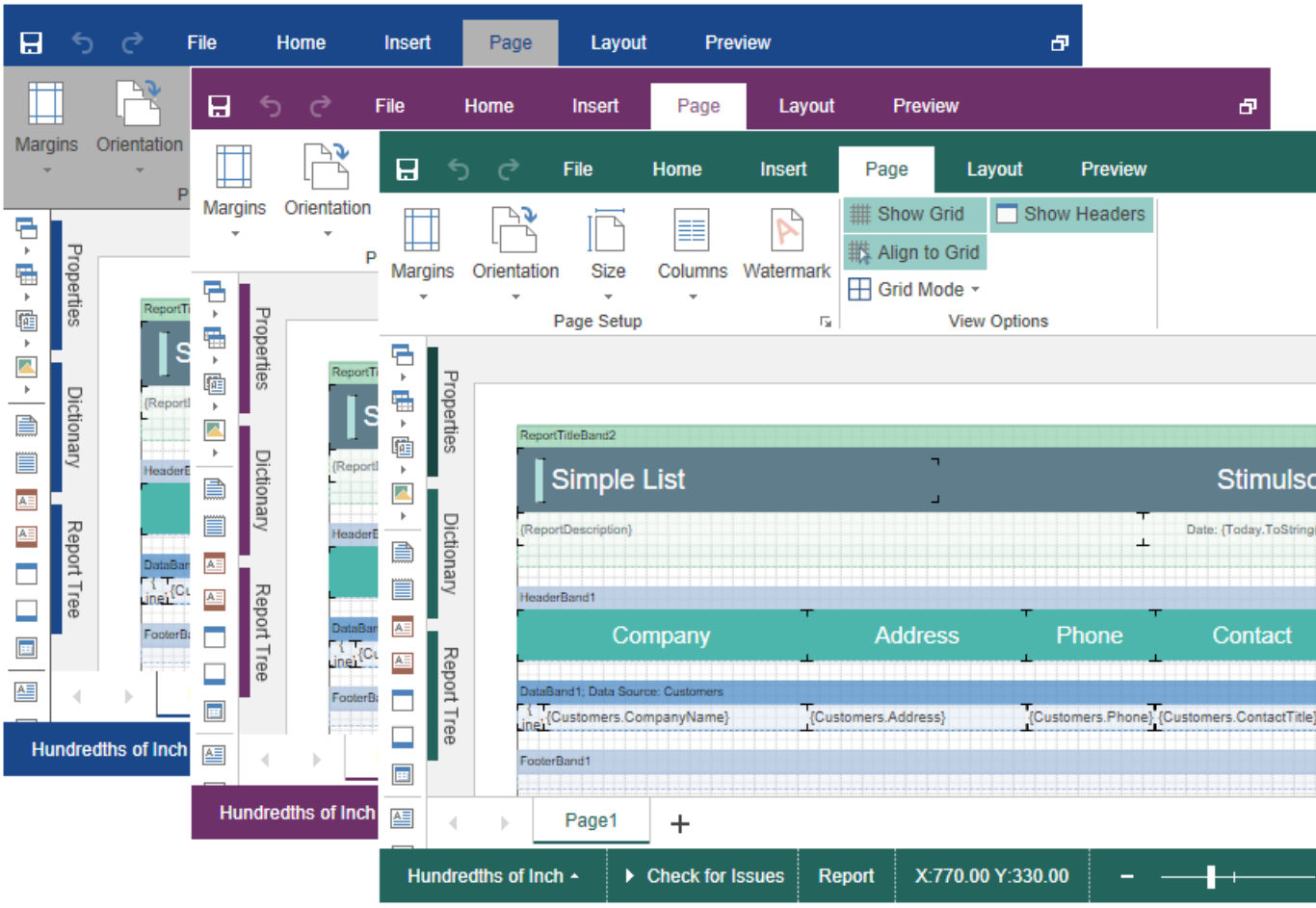
6.2.9 Использование тем

Компонент **HTML5 Designer** имеет возможность смены тем оформления визуальных элементов управления. Для изменения темы используется свойство **Theme**.

Index.cshtml

```
...
@Html.StiNetCoreDesigner(new StiNetCoreDesignerOptions() {
   Theme = StiDesignerTheme.Office2022WhiteTeal
})
...
```

На данный момент доступно **2 различные темы** со своими цветовыми акцентами. В результате, доступно **более 50 вариантов** оформления. Это позволяет настроить внешний вид дизайнера практически под любое оформление Web проекта.



© 2003-2026 Stimulsoft

6.2.10 Кэширование

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для работы **HTM5 Designer** использует кэш сервера для хранения редактируемого шаблона отчета. Это необходимо, так как клиентская часть дизайнера содержит только визуальное представление компонентов шаблона отчета, сам объект отчета со всеми параметрами и свойствами хранится на стороне сервера.

Управлять кэшированием можно при помощи следующих приведенных ниже свойств.

Свойство CacheMode

Данное свойство дизайнера устанавливает тип кэширования. Может принимать одно из значений, указанных в перечислении **StiServerCacheMode**:

- **None** – не используется в компоненте **HTML5 Designer**;
- **ObjectCache** – для кэширования используется кэш сервера, в котором сохраняется объект отчета (значение по умолчанию);
- **StringCache** - для кэширования используется кэш сервера, в котором сохраняется отчет в виде упакованной строки;
- **ObjectSession** - для кэширования используется текущая сессия, в которой сохраняется объект отчета;
- **StringSession** - для кэширования используется текущая сессия, в которой сохраняется отчет в виде упакованной строки.

Свойство CacheItemPriority

© 2003-2026 Stimulsoft

Данное свойство устанавливает приоритет хранимого в кэше сервера отчета, влияет на автоматическую очистку памяти сервера в случае ее нехватки. Чем меньше приоритет, тем больше шанс удаления информации из памяти.

Свойство CacheTimeout

Данное свойство указывает величину времени в минутах, в течение которого необходимо сохранять отчет в кэше сервера. Если при использовании кэширования требуемый отчет не найден в кэше (т.е. истекло время хранения объекта), то он будет запрошен заново при помощи специального действия **GetReport**, в этом случае несохраненные изменения могут быть утеряны.

Компонент **HTML5 Designer** предоставляет возможность определить собственные методы работы с кэшированием отчета. Для этого предназначен специальный класс **StiCacheHelper**, который содержит методы получения отчета из кэша и сохранения отчета в кэш. Необходимо создать новый класс, унаследованный от **StiCacheHelper**, и перегрузить указанные выше методы, которые соответственно имеют названия **GetObject, SaveObject** и **RemoveObject**.

© 2003-2026 Stimulsoft

Index.cshtml.cs

```
...
public class IndexModel : PageModel
{
   public class StiMyCacheHelper : StiCacheHelper
   {
      public override object GetObject(string guid)
      {
         var path = Path.Combine(HttpContext.Server.MapPath("CacheFiles"),
         guid);
         if (File.Exists(path))
         {
            byte[] cacheData = System.IO.File.ReadAllBytes(path);
            return StiCacheHelper.GetObjectFromCacheData(cacheData);
         }
         return null;
      }

      public override void SaveObject(object obj, string guid)
      {
         byte[] cacheData = StiCacheHelper.GetCacheDataFromObject(obj);
         var path = Path.Combine(HttpContext.Server.MapPath("CacheFiles"),
         guid);
         File.WriteAllBytes(path, cacheData);
      }

      public override void RemoveObject(string guid)
      {
         var path = Path.Combine(HttpContext.Server.MapPath("CacheFiles"),
         guid);
         if (File.Exists(path))
            File.Delete(path);
      }
   }

   static IndexModel()
   {
      StiNetCoreDesigner.CacheHelper = new StiMyCacheHelper();
   }
}
...
```

Для инициализации работы с кэшированием отчета при помощи созданного класса, достаточно задать его в качестве значения статического свойства **StiNetCoreDesigner.CacheHelper** в конструкторе страницы.

© 2003-2026 Stimulsoft

6.2.11 Вспомогательные методы

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для **HTML5 Designer** предусмотрено несколько вспомогательных методов, которые предназначены для получения объекта редактируемого в данный момент отчета, параметров текущего состояния дизайнера и других полезных данных. Эти методы можно использовать в действиях дизайнера.

Метод GetReportObject()

Возвращает объект отчета, с которым в данный момент работает дизайнер. Допускается производить с ним необходимые манипуляции - регистрировать новые наборы данных, изменять свойства отчета, присваивать параметры либо загружать в объект другой отчет. Затем отчет можно вернуть дизайнеру, указав его в качестве параметра в результирующем методе действия.

Index.cshtml.cs

```
...
public IActionResult OnPostExportReport()
{
   StiReport report = StiNetCoreDesigner.GetReportObject(this);
   report.ReportName = "MyReportName";

   return StiNetCoreDesigner.ExportReportResult(this, report);
}
...
```

Метод GetActionReportObject()

Возвращает объект отчета, который будет использоваться для конкретного действия. Например, для действия **OpenReport** данный метод вернет отчет, загруженный с локального диска компьютера. Для действия **PreviewReport**

© 2003-2026 Stimulsoft

метод вернет подготовленную копию отчета для предварительного просмотра.

Index.cshtml.cs

```
...
public IActionResult OnPostOpenReport()
{
   StiReport report = StiNetCoreDesigner.GetActionReportObject(this);

   // Register data for the opened report, if necessary
   DataSet data = new DataSet("Demo");
   data.ReadXml(StiNetCoreHelper.MapPath(this, "Data/Demo.xml"));
   report.RegData(data);
   report.Dictionary.Synchronize();

   return StiNetCoreDesigner.GetReportResult(this, report);
}
...
```

Метод GetFormValues()

Возвращает значения формы, которая инициировала (открыла POST-запросом) страницу дизайнера. Таким образом предоставляется возможность получить коллекцию параметров формы в любом действии дизайнера.

Index.cshtml.cs

```
...
public IActionResult OnPostDesignerInteraction()
{
   NameValueCollection formValues = StiNetCoreDesigner.GetFormValues(this);

   return StiNetCoreDesigner.InteractionResult(this);
}
...
```

По умолчанию данная возможность отключена в целях оптимизации запросов клиентской стороны дизайнера на сервер. Для включения достаточно установить свойство **PassFormValues** в значение **true**.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.StiNetCoreDesigner(new StiNetCoreDesignerOptions() {
   Server =
   {
      PassFormValues = true
   }
})
...
```

Метод GetRequestParams()

Возвращает все параметры текущего состояния дизайнера, переданные на сторону сервера. Они могут быть полезны для определения типа действия, которое в данный момент выполняет дизайнер – например, для определения типа экспорта, а также всех параметров действия.

Index.cshtml.cs

```
...
public IActionResult OnPostExportReport()
{
   StiRequestParams requestParams =
   StiNetCoreDesigner.GetRequestParams(this);
   if (requestParams.ExportFormat == StiExportFormat.Pdf)
   {
      StiReport report = StiNetCoreDesigner.GetReportObject(this);

      // Some action with report for the PDF export
      // ...

      return StiNetCoreDesigner.ExportReportResult(this, report);
   }

   return StiNetCoreDesigner.ExportReportResult(this);
}
...
```

Метод GetExportSettings()

Возвращает все параметры текущего экспортирования отчета. Тип объекта параметров будет соответствовать типу экспорта, выбранному в меню предварительного просмотра отчета. Любые параметры экспорта можно изменить и передать на вход результирующего метода. В этом случае, экспортирование отчета будет произведено с переданными параметрами.

© 2003-2026 Stimulsoft

Index.cshtml.cs

```
...
public IActionResult OnPostExportReport()
{
   StiExportSettings settings = StiNetCoreDesigner.GetExportSettings(this);
   if (settings.GetExportFormat() == StiExportFormat.Pdf)
   {
      StiPdfExportSettings pdfSettings = (StiPdfExportSettings)settings;
      pdfSettings.EmbeddedFonts = true;
      pdfSettings.AllowEditable = StiPdfAllowEditable.No;
      return StiNetCoreDesigner.ExportReportResult(this, settings);
   }

   return StiNetCoreDesigner.ExportReportResult(this);
}
...
```

Методы MapPath() и MapWebRootPath()

Возвращает абсолютный путь соответственно к директории приложения или к директории wwwroot. Можно использовать для загрузки файлов шаблонов отчетов, файлов данных и др. Данные методы расположены в статическом классе **StiNetCoreHelper**.

Index.cshtml.cs

```
...
public IActionResult OnPostGetReport()
{
   StiReport report = new StiReport();
   report.Load(StiNetCoreHelper.MapPath(this, "Reports/SimpleList.mrt"));

   return StiNetCoreDesigner.GetReportResult(this, report);
}
...
```

6.2.12 Время ожидания

При работе с компонентом **StiNetCoreDesigner** можно установить время ожидания (timeout) выполнения различных операций - хранение отчета в кэше, ответ сервера, выполнение запроса. Настройка времени ожидания выполняется при помощи свойств компонента и опций отчета.

Свойство CacheTimeout

Предоставляет возможность установить время в минутах, которое сервер будет

© 2003-2026 Stimulsoft

хранить редактируемый отчет в кэше с момента последнего действия дизайнера. По умолчанию установлено значение 10 минут.

Index.cshtml

```
...
@Html.StiNetCoreDesigner(new StiNetCoreDesignerOptions() {
   Server =
   {
      CacheTimeout = 10
   }
})
...
```

Использование кэша увеличивает скорость работы дизайнера с отчетом. Более подробно можно ознакомиться в главе Кэширование.

Свойство RequestTimeout

Предоставляет возможность установить время ожидания ответа от сервера в секундах, по истечении которого будет выдана ошибка связи с сервером. По умолчанию установлено значение 30 секунд. Для больших отчетов рекомендуется увеличить данное значение.

Index.cshtml

```
...
@Html.StiNetCoreDesigner(new StiNetCoreDesignerOptions() {
   Server =
   {
      RequestTimeout = 30
   }
})
...
```

Опция CommandTimeout

Предоставляет возможность установить время ожидания запроса в секундах, при использовании SQL источников данных в отчете. Значение данного свойства сохраняется в самом шаблоне отчёта для каждого SQL подключения отдельно.

Ниже приведен пример кода, который предоставляет возможность установить время ожидания запроса для уже созданного MS SQL соединения и источников

© 2003-2026 Stimulsoft

данных в отчете.

Index.cshtml

```
...
@Html.StiNetCoreDesigner(new StiNetCoreDesignerOptions() {
   Actions =
   {
      GetReport = "GetReport",
      DesignerEvent = "DesignerEvent"
   }
})
...
```

Index.cshtml.cs

```
...
public IActionResult OnGetDesignerEvent()
{
   return StiNetCoreDesigner.DesignerEventResult(this);
}

public IActionResult OnPostDesignerEvent()
{
   return StiNetCoreDesigner.DesignerEventResult(this);
}
...
```

6.2.13 Добавление пользовательских функций

Информация

Ознакомьтесь с примером по добавлению пользовательской функции для компонента ASP.NET Core Razor HTML5 Designer GitHub.

При интеграции дизайнера отчетов в пользовательское приложение, присутствует возможность добавления custom функции в словарь данных дизайнера отчетов. После добавления функции, её можно использовать при разработке отчетов и панелей индикаторов (дашбордов). Ниже представлен пример добавления функции расчета итога суммирования.

© 2003-2026 Stimulsoft

Index.cshtml.cs

```
...
public static decimal MySum(object value)
{
   if (!ListExt.IsList(value))
   return Stimulsoft.Base.Helpers.StiValueHelper.TryToDecimal(value);

   return Stimulsoft.Data.Functions.Funcs.SkipNulls(ListExt.ToList(value))
   .TryCastToDecimal()
   .Sum();
}
...
static IndexModel()
{
   StiFunctions.AddFunction("MyCategory", "MySum",
   "description", typeof(DesignerPage),
   typeof(decimal), "Calculates a sum of the specified set of values.",
   new[] { typeof(object) },
   new[] { "values" },
   new[] { "A set of values" }).UseFullPath = false;
}
...
```

6.2.14 Настройки

Настройка **HTML5 Designer** выполняется при помощи свойств, находящихся в классе **StiNetCoreDesignerOptions**. Все свойства разделены на группы, некоторые из групп содержат свои подгруппы свойств для удобства использования. Ниже приведен пример установки некоторых свойств дизайнера.

© 2003-2026 Stimulsoft

Index.cshtml

```
...
@Html.StiNetCoreDesigner(new StiNetCoreDesignerOptions() {
   Theme = Stimulsoft.Report.Web.StiDesignerTheme.Office2022WhiteTeal,
   Localization = "Localization/en.xml",
   Actions =
   {
      GetReport = "GetReport",
      PreviewReport = "PreviewReport",
      SaveReport = "SaveReport",
      DesignerEvent = "DesignerEvent"
   },
   Appearance =
   {
      ReportDisplayMode = StiReportDisplayMode.Auto,
      ShowTooltipsHelp = false,
      ShowDialogsHelp = false,
      DefaultUnit = Stimulsoft.Report.StiReportUnitType.Centimeters
   },
   Dictionary =
   {
      PermissionBusinessObjects =
      Stimulsoft.Report.Web.StiDesignerPermissions.None,
      PermissionDataConnections =
      Stimulsoft.Report.Web.StiDesignerPermissions.View
   },
   Bands =
   {
      ShowChildBand = false,
      ShowEmptyBand = false,
      ShowOverlayBand = false
   }
})
...
```

Основные настройки (без группы)

Наименование	Описание
Theme	Задает тему оформления дизайнера. Список доступных тем оформления находится в перечислении **StiDesignerTheme**. По умолчанию установлено значение **Office2022WhiteBlue**.
Localization	Задает путь к XML файлу локализации. Путь может быть абсолютным либо относительным. По умолчанию используется английская локализация, которая встроена в

© 2003-2026 Stimulsoft

	дизайнер и не требует дополнительных XML файлов.
LocalizationDirectory	Задает путь к директории с XML файлами локализаций. Находящиеся в указанной папке локализационные файлы будут загружены в список выбора локализации на панели дизайнера.
Width	Задает ширину компонента в необходимых единицах измерения, которые определены в классе **Unit**. Доступна установка значения в пикселях: **Unit.Pixel()**, поинтах: **Unit.Point()** и процентах: **Unit.Percentage()**. По умолчанию компонент разворачивается на всю область окна браузера.
Height	Задает высоту компонента в необходимых единицах измерения, которые определены в классе **Unit**. Доступна установка значения в пикселях: **Unit.Pixel()**, поинтах: **Unit.Point()** и процентах: **Unit.Percentage()**. По умолчанию компонент разворачивается на всю область окна браузера.

Действия (Actions)

Наименование	Описание
GetReport	Задает название метода действия подготовки шаблона отчета при загрузке дизайнера.
OpenReport	Задает название метода действия открытия шаблона отчета из меню дизайнера.
CreateReport	Задает название метода действия подготовки шаблона отчета при создании нового отчета в дизайнере.
SaveReport	Задает название метода действия сохранения шаблона отчета на стороне сервера.
SaveReportAs	Задает название метода действия сохранения

© 2003-2026 Stimulsoft

	шаблона отчета на стороне сервера при использовании пункта меню **Сохранить как (Save As)**. Если действие не задано, будет использоваться встроенный метод сохранения шаблона отчета на локальный диск компьютера.
PreviewReport	Задает название метода действия подготовки построенного отчета в окне просмотра.
GetPreviewReport	Задает название метода действия непосредственно перед отображением отчета в окне просмотра.
ExportReport	Задает название метода действия экспорта отчета в необходимый формат.
Exit	Задает название метода действия для перехода к нужному представлению при нажатии кнопки Выход (Exit) в главном меню дизайнера.
DesignerEvent	Задает название метода действия дизайнера отчетов, служащего для обработки различных дополнительных действий дизайнера, таких как работа с данными, работа с компонентами отчета, а так же для загрузки скриптов и стилей дизайнера.

Работа с сервером (Server)

Наименование	Описание
RouteTemplate	Задает шаблон (route template) вызова действий дизайнера отчетов. Если свойство не установлено, будет использоваться шаблон, установленный в Razor проекте. Если свойство `UseRelativeUrls` установлено в true, для этого свойства не будет учитываться `BasePath`. По умолчанию это свойство не установлено.

© 2003-2026 Stimulsoft

ShowServerErrorPage	Включает отображение HTML-страницы с детализацией ошибки, возникшей на стороне сервера. При включенном свойстве детализация ошибки будет отображена в окне предварительного просмотра. Если свойство выключено, будет отображен только числовой код ошибки и краткий текст ошибки в диалоговом окне. По умолчанию свойство имеет значение **true**.
AllowAutoUpdateCookies	Предоставляет возможность дизайнеру автоматически обновлять файлы cookies при каждом запросе на сервер. По умолчанию cookies устанавливаются при создании дизайнера, если они не указаны в отчете. По умолчанию свойство имеет значение **false**.
AllowAntiforgeryToken	Предоставляет возможность дизайнеру автоматически запрашивать и отправлять токен защиты от подделки. По умолчанию свойство имеет значение **true**.
RequestTimeout	Устанавливает время ожидания ответа от сервера в секундах, по истечении которого будет выдана ошибка. По умолчанию установлено значение 30 секунд. Для больших отчетов рекомендуется увеличить данное значение.
CacheTimeout	Устанавливает время в минутах, которое сервер будет хранить редактируемый отчет с момента последнего действия дизайнера. По умолчанию установлено значение 10 минут.
CacheMode	Устанавливает режим кэширования отчета. Может принимать одно из указанных ниже значений перечисления **StiServerCacheMode**: › **None** – не используется в компоненте **HTML5 Designer**; › **ObjectCache** – в качестве хранилища используется кэш, отчет хранится как объект

© 2003-2026 Stimulsoft

	(значение свойства по умолчанию); › **ObjectSession** – в качестве хранилища используется сессия, отчет хранится как объект; › **StringCache** – в качестве хранилища используется кэш сервера, отчет сериализуется в упакованную строку; › **StringSession** – в качестве хранилища используется сессия, отчет сериализуется в упакованную строку.
CacheItemPriority	Устанавливает приоритет хранимого в кэше сервера отчета. Это свойство влияет на автоматическую очистку памяти сервера в случае ее нехватки. Чем меньше приоритет, тем больше шанс удаления информации из памяти.
AllowAutoUpdateCache	Устанавливает режим автоматического обновления кэша. Отчет, хранящийся в кэше либо сессии сервера, будет автоматически пересохранен через определенный промежуток времени при бездействии дизайнера (примерно каждые 3 минуты). По умолчанию свойство имеет значение **true**.
UseRelativeUrls	Устанавливает режим дизайнера, в котором для запросов на сервер используются относительные ссылки. По умолчанию свойство имеет значение **true**.
PortNumber	Получает или задает значение, которое указывает номер порта для использования в URL. Значение 0 определяет автоматическое обнаружение (установлено по умолчанию). Значение -1 удаляет номер порта.
PassQueryParametersForResources	Включает передачу всех параметров URL запроса при формировании ссылок на ресурсы дизайнера. При значении **false** для запроса ресурсов дизайнера используются только необходимые параметры, что

© 2003-2026 Stimulsoft

	способствует более корректной работе кэша браузера. По умолчанию свойство имеет значение **true**.
PassFormValues	Включает передачу значений POST-формы на сторону клиента, если эти значения требуется использовать в действиях дизайнера. При включении данной возможности, вспомогательный метод **GetFormValues()** будет возвращать коллекцию параметров формы. По умолчанию свойство имеет значение **false**.
UseCompression	Включает сжатие запросов дизайнера в GZip поток. Это позволяет уменьшить объем интернет-трафика, но немного замедляет работу дизайнера. По умолчанию свойство имеет значение **false**.
UseCacheForResources	Включает кэширование на стороне сервера ресурсов компонента, таких как скрипты, стили, изображения. Позволяет ускорить загрузку компонента и уменьшить нагрузку на сервер при большом количестве клиентов. По умолчанию свойство имеет значение **true**.
AllowLoadingCustomFontsToClientSide	Позволяет передавать пользовательские шрифты на клиентскую сторону и преобразовывать их в CSS стиль для корректного отображения текста на стороне HTML с заданным шрифтом. По умолчанию свойство имеет значение **false**.

Внешний вид (Appearance)

Наименование	Описание
IconSet	Предоставляет возможность установить набор иконок: › StiWebUIIconSet.Auto (значение по

© 2003-2026 Stimulsoft

	умолчанию) - устанавливает набор иконок автоматически. Для тем **Office2022** - используется набор иконок в стиле **Monoline**, для тем **Office2013** - используется набор иконок в стиле **Regular**. › StiWebUIIconSet.Monoline - устанавливает набор иконок в стиле **Monoline**; › StiWebUIIconSet.Regular - устанавливает набор иконок в стиле **Regular**.
CustomCss	Задает путь к CSS файлу стилей дизайнера. Если данное свойство установлено, то стандартные стили выбранной темы загружены не будут. По умолчанию установлено пустое значение.
DefaultUnit	Устанавливает единицы измерения размеров для отчета и всех его компонентов. По умолчанию используются сантиметры.
Zoom	Задает масштаб отображения страниц отчета. По умолчанию установлен масштаб 100 процентов. Допускаются значения от 10 до 200 процентов. Может принимать одно из указанных ниже значений перечисления **StiZoomMode**: › **PageWidth** – при запуске дизайнера будет установлен масштаб, необходимый для отображения отчета по ширине страницы; › **PageHeight** – при запуске дизайнера будет установлен масштаб, необходимый для отображения отчета по высоте страницы.
ShowAnimation	Включает анимацию элементов интерфейса дизайнера. По умолчанию свойство имеет значение **true**.
ShowOpenDialog	Устанавливает режим открытия с диалогом или через событие open. По умолчанию свойство имеет значение **true**, т.е. диалог открытия отображается.

© 2003-2026 Stimulsoft

ShowTooltips	Включает отображение подсказок для элементов управления дизайнера при наведении мыши. По умолчанию свойство имеет значение **true**.
ShowTooltipsHelp	Включает отображение ссылки на онлайн документацию в подсказках для элементов управления дизайнера. По умолчанию свойство имеет значение **true**.
ShowDialogsHelp	Включает отображение ссылки на онлайн документацию на заголовках диалоговых форм дизайнера. По умолчанию свойство имеет значение **true**.
InterfaceType	Устанавливает тип используемого интерфейса дизайнера. Может принимать одно из указанных ниже значений перечисления **StiInterfaceType**: › **Auto** – тип интерфейса дизайнера будет выбран автоматически в зависимости от используемого устройства (значение по умолчанию); › **Mouse** – принудительное использование интерфейса для управления дизайнером при помощи мыши; › **Touch** – принудительное использование Touch интерфейса для управления дизайнером при помощи сенсорного экрана (мобильные устройства), также в этом режиме для удобства управления элементы интерфейса увеличены.
DatePickerFirstDayOfWeek	Устанавливает первый день недели для элемента выбора даты. Может принимать одно из указанных ниже значений перечисления **StiFirstDayOfWeek**: › **Auto** - автоматическое определение первого дня недели из настроек браузера (значение по умолчанию);

© 2003-2026 Stimulsoft

	› **Monday** – первым днем недели является понедельник; › **Sunday** – первым днем недели является воскресенье.
DatePickerIncludeCurrentDayForRanges	Предоставляет возможность включать или не включать текущий день в элементе выбора даты. По умолчанию свойство имеет значение **false**.
FormatForDateControls	Предоставляет возможность переопределить формат для элементов управления дат. По умолчанию, значение для текущей опции не задан и формат для дат определяется из локали браузера.
ShowReportTree	Включает отображение дерева компонентов отчета. По умолчанию свойство имеет значение **true**.
ChartRenderType	Устанавливает режим отображения диаграмм в отчете на вкладке **Просмотр (Preview)**. Может принимать одно из указанных ниже значений перечисления **StiChartRenderType**: › **Image** – диаграммы отображаются в виде статичных изображений; › **Vector** – диаграммы отображаются в векторном режиме в виде SVG объекта; › **AnimatedVector** - диаграммы отображаются в векторном режиме в виде SVG объекта, элементы диаграммы появляются с анимацией (значение по умолчанию).
ReportDisplayMode	Устанавливает режим экспорта для отображения страниц отчета на вкладке предварительного просмотра. Может принимать одно из указанных ниже значений перечисления **StiReportDisplayMode**: › **FromReport** - режим экспорта элементов

© 2003-2026 Stimulsoft

	отчета определяется из настроек шаблона - Div или Table; › **Table** – элементы отчета экспортируются с использованием HTML таблиц (значение по умолчанию); › **Div** – элементы отчета экспортируются с использованием DIV разметки; › **Span** - элементы отчета экспортируются с использованием SPAN разметки.
ParametersPanelDateFormat	Устанавливает формат даты и времени для переменных соответствующего типа на панели параметров. По умолчанию используется формат даты и времени, установленный браузером.
CloseDesignerWithoutAsking	Предоставляет возможность установить режим закрытия дизайнера отчетов без подтверждения изменений в текущем отчете. По умолчанию свойство имеет значение **false**, т.е. дизайнер будет запрашивать разрешение на сохранение изменений в текущем отчете.
ShowSystemFonts	Предоставляет возможность отображать или не отображать системные шрифты в списке шрифтов дизайнера отчетов. По умолчанию свойство имеет значение **true**, т.е. системные шрифты отображаются в списке шрифтов.
WizardTypeRunningAfterLoad	Предоставляет возможность вызывать мастер создания отчетов после запуска дизайнера отчетов. Может принимать одно из указанных ниже значений перечисления **StiWizardType**: › **None** - дизайнер отчетов будет запущен без вызова мастера создания отчета; › **StandardReport** - дизайнер отчетов будет запущен с вызовом стандартного мастера создания отчета; › **MasterDetailReport** - дизайнер отчетов

© 2003-2026 Stimulsoft

	будет запущен с вызовом master-detail мастера создания отчета; › **LabelReport** - дизайнер отчетов будет запущен с вызовом мастера создания отчета с этикетками; › **InvoicesReport** - дизайнер отчетов будет запущен с вызовом мастера создания инвойса; › **OrdersReport** - дизайнер отчетов будет запущен с вызовом мастера создания ордера; › **QuotationReport** - дизайнер отчетов будет запущен с вызовом мастера создания квоты.
AllowWordWrapTextEditors	Предоставляет возможность установить режим переноса в текстовых редакторах. По умолчанию свойство имеет значение **true**.
DefaultRibbonType	Предоставляет возможность установить используемый по умолчанию стиль панели инструментов дизайнера. Может принимать одно из указанных ниже значений перечисления: › **StiDesignerRibbonType.Classic** - Будет установлен классический стиль (значение по умолчанию); › **StiDesignerRibbonType.SingleLine** – Будет установлен компактный однострочный стиль.
AllowPropagationEvents	Предоставляет возможность разрешить распространение событий нажатия клавиш, когда дизайнер отчетов не в фокусе. По умолчанию свойство имеет значение **true**.
PropertiesPanelViewMode	Предоставляет возможность закрепить или открепить панель свойств, словарь отчета, дерево отчета. Может принимать одно из указанных ниже значений перечисления: › **Pinned** - панели закреплены (значение по умолчангию); › **Unpinned** - панели открепленны.

© 2003-2026 Stimulsoft

Поведение (Behavior)

Наименование	Описание
ShowSaveDialog	Включает отображение диалога ввода имени отчета при его сохранении. Имя отчета будет передано в параметрах дизайнера. По умолчанию свойство имеет значение **true**.
UndoMaxLevel	Устанавливает максимальную глубину отмены действий с отчетом (функция Undo/ Redo). Большое значение этого свойства будет потреблять больше памяти на стороне сервера для хранения параметров отмены. По умолчанию установлено значение **6**.
AllowChangeWindowTitle	Разрешает использование заголовка окна браузера для отображения имени файла редактируемого отчета. По умолчанию свойство имеет значение **true**.
SaveReportMode	Задает режим сохранения отчета. Может принимать одно из трех значений перечисления **StiSaveMode**. › **Hidden** - действие сохранения отчета вызывается в фоне при помощи AJAX-запроса и никак не отображается в окне браузера (значение по умолчанию); › **Visible** - действие сохранения отчета будет вызвано в текущем окне браузера в видимом режиме при помощи POST-запроса; › **NewWindow** - действие сохранения отчета будет вызвано в новом окне (вкладке) браузера.
SaveReportAsMode	Задает режим сохранения отчета. Может принимать одно из трех значений перечисления **StiSaveMode**. › **Hidden** - действие сохранения отчета вызывается в фоне при помощи AJAX-запроса

© 2003-2026 Stimulsoft

	и никак не отображается в окне браузера (значение по умолчанию); › **Visible** - действие сохранения отчета будет вызвано в текущем окне браузера в видимом режиме при помощи POST-запроса; › **NewWindow** - действие сохранения отчета будет вызвано в новом окне (вкладке) браузера.
CheckReportBeforePreview	Предоставляет возможность проверять или не проверять отчет встроенным инструментом перед его просмотром. По умолчанию свойство имеет значение **true**.

Главное меню (FileMenu)

Наименование	Описание
Visible	Включает отображение главного меню дизайнера отчетов. По умолчанию свойство имеет значение **true**.
ShowNew	Включает отображение пункта главного меню **Создать (New)**. По умолчанию свойство имеет значение **true**.
ShowFileMenuNewReport	Включает отображение кнопки **Пустой отчет (Blank Report)** в меню **Создать**. По умолчанию свойство имеет значение **true**.
ShowFileMenuNewDashboard	Включает отображение кнопки **Пустая панель индикаторов (Blank Dashboard)** в меню **Создать**. По умолчанию свойство имеет значение **true**.
ShowOpen	Включает отображение пункта главного меню **Открыть (Open)**. По умолчанию свойство имеет значение **true**.
ShowSave	Включает отображение пункта главного меню **Сохранить (Save)**. По умолчанию свойство имеет значение **true**.

© 2003-2026 Stimulsoft

ShowSaveAs	Включает отображение пункта главного меню **Сохранить как (Save As)**. По умолчанию свойство имеет значение **true**.
ShowClose	Включает отображение пункта главного меню **Закрыть (Close)**. По умолчанию свойство имеет значение **true**.
ShowExit	Включает отображение пункта главного меню **Выход (Exit)**. По умолчанию свойство имеет значение **false**.
ShowReportSetup	Включает отображение пункта главного меню **Свойства отчета (Report Setup)**. По умолчанию свойство имеет значение **true**.
ShowOptions	Включает отображение пункта главного меню **Параметры (Options)**. По умолчанию свойство имеет значение **true**.
ShowInfo	Включает отображение пункта главного меню **Сведения (Info)**. По умолчанию свойство имеет значение **true**.
ShowAbout	Включает отображение пункта главного меню **О программе (About)**. По умолчанию свойство имеет значение **true**.
ShowHelp	Включает отображение пункта главного меню **Помощь (Help)**. По умолчанию свойство имеет значение **true**.

Словарь данных (Dictionary)

Наименование	Описание
Visible	Включает отображение словаря данных отчета. По умолчанию свойство имеет значение **true**.
UseAliases	Предоставляет возможность использовать псевдонимы в словаре данных. Может принимать одно из трех значений

© 2003-2026 Stimulsoft

	перечисления **StiUseAliases**: › **Auto** - определяет режим использования псевдонимов из сохраненного значения в файлах cookies (значение по умолчанию); › **True** - устанавливает режим использования псевдонимов в словаре данных; › **False** - выключает режим использования псевдонимов в словаре данных.
NewReportDictionary	Предоставляет возможность создавать новый словарь данных или объединить с существующим, при создании нового отчета в дизайнере. Может принимать одно из трех значений перечисления **StiNewReportDictionary**: › **Auto** - определяет режим создания или объединения словаря данных из сохраненного значения в файлах cookies(значение по умолчанию); › **DictionaryNew** - устанавливает режим создания нового словаря данных, при создании нового отчета; › **DictionaryMerge** - устанавливает режим объединения существующего словаря данных с новым, при создании нового отчета в дизайнере.
ShowDictionaryContextMenuProperties	Устанавливает видимость пункта **Свойства (Properties)** в контекстном меню словаря данных. По умолчанию свойство имеет значение **true**.
ShowDictionaryActions	Устанавливает видимость меню **Действия (Actions)** на панели инструментов словаря данных. По умолчанию свойство имеет значение **true**.
PermissionDataConnections	Устанавливает доступные действия над подключениями к данным отчета. Может принимать одно или несколько значений из

© 2003-2026 Stimulsoft

	перечисления **StiDesignerPermissions**.
PermissionDataSources	Устанавливает доступные действия над источниками данных отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionDataColumns	Устанавливает доступные действия над колонками данных отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionBusinessObjects	Устанавливает доступные действия над бизнес-объектами отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionDataRelations	Устанавливает доступные действия над связями данных отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionVariables	Устанавливает доступные действия над переменными отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionResources	Устанавливает доступные действия над ресурсами в словаре данных отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionSqlParameters	Устанавливает доступные действия над параметрами в SQL запросах источников данных отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionDataTransformations	Устанавливает доступные действия над преобразованием данных. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionUserFunctions	Устанавливает доступные действия над пользовательскими функциями. Может

© 2003-2026 Stimulsoft

	принимать одно или несколько значений из перечисления **StiDesignerPermissions**.

В таблице ниже приведены все доступные значения для перечисления **StiDesignerPermissions**, которые можно установить для элементов словаря отчета.

Значение	**Описание**
None	Запрещает любое действие над элементом словаря данных.
All	Разрешает любые действия над элементом словаря данных.
Create	Разрешает создавать определенный элемент словаря данных.
Delete	Разрешает удалять определенный элемент словаря данных.
Modify	Разрешает редактировать определенный элемент словаря данных.
View	Разрешает просматривать определенный элемент словаря данных.
ModifyView	Разрешает редактировать и просматривать определенный элемент словаря данных.

Панель инструментов (Toolbar)

Наименование	**Описание**
ShowToolbar	Предоставляет возможность отображать или не отображать панель инструментов в дизайнере отчетов. По умолчанию свойство имеет значение **true**.
ShowSetupToolboxButton	Включает отображение кнопки вызова окна настроек боковой панели компонентов отчета. По умолчанию свойство имеет

© 2003-2026 Stimulsoft

	значение **true**.
ShowInsertButton	Включает отображение вкладки **Вставить (Insert)** на панели инструментов дизайнера. По умолчанию свойство имеет значение **true**.
ShowLayoutButton	Включает отображение вкладки **Формат (Layout)** на панели инструментов дизайнера. По умолчанию свойство имеет значение **true**.
ShowPageButton	Включает отображение вкладки **Страница (Page)** на панели инструментов дизайнера. По умолчанию свойство имеет значение **true**.
ShowPreviewButton	Включает отображение вкладки **Предварительный просмотр (Preview)** на панели инструментов дизайнера. По умолчанию свойство имеет значение **true**.
ShowSaveButton	Включает отображение кнопки **Сохранить (Save)** на панели инструментов дизайнера. По умолчанию свойство имеет значение **true**.
ShowAboutButton	Включает отображение кнопки **О программе (About)** на панели инструментов дизайнера. По умолчанию свойство имеет значение **false**.

Панель свойств (PropertiesGrid)

Наименование	Описание
Visible	Включает отображение панели свойств компонентов отчета. По умолчанию свойство имеет значение **true**.
Width	Задает ширину панели свойств компонентов отчета. По умолчанию установлена ширина **370** точек.
LabelWidth	Задает ширину надписей панели свойств компонентов отчета. По умолчанию установлена ширина **160** точек.

© 2003-2026 Stimulsoft

PropertiesGridPosition	Предоставляет возможность определить позицию панели свойств в дизайнере отчетов. Может принимать одно из трех значений перечисления **StiPropertiesGridPosition**. › **Left** - панель свойств будет располагаться слева; › **Right** - панель свойств будет располагаться справа.
ShowPropertiesWhichUsedFromStyles	Устанавливает видимость свойств, которые используются из стилей в дизайнере. По умолчанию свойство имеет значение **false**.

Компоненты отчета (Components)

Наименование	Описание
ShowText	Включает отображение компонента **Текст (Text)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowTextInCells	Включает отображение компонента **Текст в ячейках (Text in Cells)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowRichText	Включает отображение компонента **Rich текст (Rich Text)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowImage	Включает отображение компонента **Картинка (Image)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowBarCode	Включает отображение компонента **Штрих-код (Bar Code)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет

© 2003-2026 Stimulsoft

	значение **true**.
ShowShape	Включает отображение компонента **Геометрия (Shape)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowHorizontalLinePrimitive	Включает отображение компонента **Горизонтальная линия (Horizontal Line)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowVerticalLinePrimitive	Включает отображение компонента **Вертикальная линия (Vertical Line)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowRectanglePrimitive	Включает отображение компонента **Прямоугольник (Rectangle)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowRoundedRectanglePrimitive	Включает отображение компонента **Закругленный прямоугольник (Rounded Rectangle)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowPanel	Включает отображение компонента **Панель (Panel)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowClone	Включает отображение компонента **Клон (Clone)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowCheckBox	Включает отображение компонента **Флажок (Check Box)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowSubReport	Включает отображение компонента **Вложенный отчет (Sub Report)** в меню

© 2003-2026 Stimulsoft

	вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowZipCode	Включает отображение компонента **Почтовый индекс (Zip Code)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowTable	Включает отображение компонента **Таблица (Table)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowCrossTab	Включает отображение компонента **Кросс-таблица (Cross-Tab)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowChart	Включает отображение компонента **Диаграмма (Chart)** в меню вставки компонентов дизайнера. Действует сразу на все типы диаграмм. По умолчанию свойство имеет значение **true**.
ShowMap	Включает отображение компонента **Карта (Map)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **false**.
ShowGauge	Включает отображение компонента **Индикатор (Gauge)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **false**.
ShowSparkline	Включает отображение компонента **Спарклайн (Sparkline)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowMathFormula	Включает отображение компонента **Математическая формула (Math Formula)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowElectronicSignature	Включает отображение компонента

© 2003-2026 Stimulsoft

	Электронная подпись (Electronic Signature) в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowPdfDigitalSignature	Включает отображение компонента **Цифровая подпись PDF (PDF Digital Signature)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.

Секции (Bands)

Наименование	Описание
ShowReportTitleBand	Включает отображение пункта **Заголовок отчета (Report Title)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowReportSummaryBand	Включает отображение пункта **Итог отчета (Report Summary)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowPageHeaderBand	Включает отображение пункта **Заголовок страницы (Page Header)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowPageFooterBand	Включает отображение пункта **Итог страницы (Page Footer)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowGroupHeaderBand	Включает отображение пункта **Заголовок группы (Group Header)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowGroupFooterBand	Включает отображение пункта **Итог группы (Group Footer)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowHeaderBand	Включает отображение пункта **Заголовок данных (Header)** в списке меню **Секции**. По

© 2003-2026 Stimulsoft

	умолчанию свойство имеет значение **true**.
ShowFooterBand	Включает отображение пункта **Итог данных (Footer)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowColumnHeaderBand	Включает отображение пункта **Заголовок колонки (Column Header)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowColumnFooterBand	Включает отображение пункта **Итог колонки (Column Footer)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowDataBand	Включает отображение пункта **Данные (Data)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowHierarchicalBand	Включает отображение пункта **Дерево (Hierarchical)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowChildBand	Включает отображение пункта **Подчиненный (Child)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowEmptyBand	Включает отображение пункта **Пустые данные (Empty)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowOverlayBand	Включает отображение пункта **Водяной знак (Overlay)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowTableOfContents	Включает отображение пункта **Оглавление (Table Of Contents)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.

Элементы дашборда (DashboardElements)

Наименование	Описание

© 2003-2026 Stimulsoft

ShowTableElement	Включает отображение элемента панели индикаторов **Таблица (Table)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowCardsElement	Включает отображение элемента панели индикаторов **Карточки (Cards)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowChartElement	Включает отображение элемента панели индикаторов **Диаграмма (Chart)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowGaugeElement	Включает отображение элемента панели индикаторов **Измерительный прибор (Gauge)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowPivotTableElement	Включает отображение элемента панели индикаторов **Сводная таблица (Pivot)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowIndicatorElement	Включает отображение элемента панели индикаторов **Индикатор (Indicator)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowProgressElement	Включает отображение элемента панели индикаторов **Прогресс (Progress)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowRegionMapElement	Включает отображение элемента панели

© 2003-2026 Stimulsoft

	индикаторов **Региональная карта (Region Map)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowOnlineMapElement	Включает отображение элемента панели индикаторов **Онлайн карта (Online Map)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowImageElement	Включает отображение элемента панели индикаторов **Изображение (Image)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowTextElement	Включает отображение элемента панели индикаторов **Текст (Text)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowPanelElement	Включает отображение элемента панели индикаторов **Панель (Panel)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowShapeElement	Включает отображение элемента панели индикаторов **Геометрия (Shape)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowButtonElement	Включает отображение элемента панели индикаторов **Кнопка (Button)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowListBoxElement	Включает отображение элемента панели индикаторов **Список (ListBox)** на

© 2003-2026 Stimulsoft

	инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowComboBoxElement	Включает отображение элемента панели индикаторов **Выпадающий список (ComboBox)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowTreeViewElement	Включает отображение элемента панели индикаторов **Иерархический список (TreeView)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowTreeViewBoxElement	Включает отображение элемента панели индикаторов **Выпадающий иерархический список (TreeViewBox)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowDatePickerElement	Включает отображение элемента панели индикаторов **Выбор даты (Date Picker)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.

Кросс-секции (CrossBands)

Наименование	Описание
ShowCrossGroupHeaderBand	Включает отображение секции **Кросс-заголовок группы (Cross Group Header)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowCrossGroupFooterBand	Включает отображение секции **Кросс-итог группы (Cross Group Footer)** в меню вставки компонентов дизайнера. По умолчанию

© 2003-2026 Stimulsoft

	свойство имеет значение **true**.
ShowCrossHeaderBand	Включает отображение секции **Кросс-заголовок данных (Cross Header)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowCrossFooterBand	Включает отображение секции **Кросс-итог данных (Cross Footer)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowCrossDataBand	Включает отображение секции **Кросс-данные (Cross Data)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.

Панели индикаторов(Dashboards)

Наименование	Описание
ShowNewDashboardButton	Предоставляет возможность отображать или не отображать кнопку **Новая панель индикаторов (New Dashboard)** в дизайнере отчетов. По умолчанию свойство имеет значение **true**.

Страницы(Pages)

Наименование	Описание
ShowNewPageButton	Предоставляет возможность отображать или не отображать кнопку **Новая страница (New Page)** в дизайнере отчетов. По умолчанию свойство имеет значение **true**.

При разработке отчета или дашборда в дизайнере отчетов, можно также определить настройки экспорта (ExportOptions), параметры Email (EmailOptions)

© 2003-2026 Stimulsoft

и панель инструментов (PreviewToolbarOptions) на вкладке **Просмотр (Preview)**. Эти опции аналогичны опциям вьювера отчетов.

7 Reports and Dashboards for Blazor

Blazor представляет собою кросс-платформенную технологию создания Web приложений для Windows, Linux и macOS. Компания Stimulsoft предоставляет инструменты создания, отображения, преобразования отчетов с использованием этой технологии. Поддерживаются две разновидности данной технологии: **Blazor Server** – распределённая система, в которой вся логика выполняется на стороне сервера, а визуальная часть отображается на клиенте, и **Blazor WebAssembly** – все модули продукта загружаются и выполняются прямо в окне Web браузера.

Инструменты создания и редактирования отчетов:	Инструменты отображения и конвертации отчетов:
Blazor дизайнер	Blazor вьювер

7.1 Вьювер

Примеры

Ознакомьтесь с примерами по работе с компонентом **Blazor Viewer** на GitHub, доступны технологии Blazor Server и Blazor WebAssembly. Все примеры представляют собой отдельные проекты, сгруппированные в один solution для Visual Studio.

Компонент **Blazor Viewer** (**StiBlazorViewer**) предназначен для просмотра отчетов в окне браузера. При этом не требуется устанавливать у клиента компоненты или какие-либо специальные плагины. Все, что необходимо – это современный Web-браузер. При помощи **Blazor Viewer** можно просматривать, печатать, выполнять экспортирование отчетов на любом компьютере с любой установленной операционной системой.

Компонент **Blazor Viewer** разработан как универсальный компонент, и может работать как с использованием технологии **Blazor Server**, так и с использованием технологию **WebAssembly**. При выполнении всех

© 2003-2026 Stimulsoft

интерактивных действий над отчётами, компонент запрашивает только необходимые данные, что позволяет избавиться от перезагрузки всей страницы, а также экономить Web-трафик и повысить скорость работы.

Blazor Viewer поддерживает множество тем оформления, анимированный интерфейс, закладки, интерактивные отчеты, редактирование элементов отчета на странице, полноэкранный режим, поиск, и другие, необходимые для просмотра отчетов, возможности.

Для использования **Blazor Viewer** в Web-проекте, необходимо установить NuGet пакет Stimulsoft.Reports.Blazor либо Stimulsoft.Dashboards.Blazor:
- Выберите пункт "Manage NuGet Packages..." в контекстном меню проекта;
- На вкладке Browse, в строке поиска, укажите Stimulsoft.Reports.Blazor;
- Выделите элемент, определите версию пакета и нажмите кнопку **Установить (Install)**. При обновлении пакета, следует нажать кнопку **Обновить (Update)**.

Если по какой-либо причине это невозможно, необходимо добавить в проект указанные ниже сборки:
Stimulsoft.Base.dll
Stimulsoft.Blockly.dll
Stimulsoft.Data.dll
Stimulsoft.Drawing.dll
Stimulsoft.Map.dll
Stimulsoft.Report.dll
Stimulsoft.Report.Blazor.dll
Stimulsoft.Report.Check.dll
Stimulsoft.Report.Helper.dll
Stimulsoft.Report.Web.dll
Stimulsoft.Report.WebDesign.dll
Stimulsoft.System.dll
Stimulsoft.System.Web.dll

Для работы с дашбордами, необходимо дополнительно добавить в проект указанные ниже сборки:
Stimulsoft.Dashboard.dll
Stimulsoft.Dashboard.Drawing.dll
Stimulsoft.Dashboard.Export.dll

Активация Динамическое сворачивание,

© 2003-2026 Stimulsoft

- Отображение отчета
- Подключение данных
- Локализация вьювера
- Печать отчета
- Экспорт отчета
- Редактирование построенного отчета
- Режимы отображения
- Работам с параметрами
- сортировка и детализация
- Работа с закладками
- Отправка отчета по Email
- Вызов дизайнера из вьювера
- Экспорт и печать из кода
- Основные возможности
- События вьювера
- Настройки вьювера

7.1.1 Активация

После приобретения продукта Stimulsoft вам необходимо активировать лицензию для используемых компонентов. Это можно выполнить, указав лицензионный ключ или загрузив файл с лицензионным ключом. Ниже представлен пример активации компонента **StiBlazorViewer**.

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorViewer />

@code
{
   protected override void OnInitialized()
   {
      //Activation with using license code
      Stimulsoft.Base.StiLicense.Key = "Your activation code...";

      //Activation with using license file
      Stimulsoft.Base.StiLicense.LoadFromFile("Content/license.key");

      base.OnInitialized();
   }
}
```

© 2003-2026 Stimulsoft

Получить лицензионный ключ или скачать файл с лицензионным ключом, вы можете в личном кабинете пользователя. Для авторизации в кабинете используйте имя пользователя и пароль, указанные при покупке подписки продукта.

7.1.2 Отображение отчета

Примечание

При присвоении отчета вьювер компоненту, происходит автоматическое построение отчета. Вызов метода `Report.Render()` необходим только в том случае, если требуется провести какие-либо действия с построенным отчетом перед его отображением во вьювере. Аналогично, при использовании режима компиляции, вызов метода `Report.Compile()` требуется только в том случае, если требуется провести какие-либо действия с откомпилированным отчетом перед его построением и отображением во вьювере.

Для отображения отчета необходимо добавить компонент **StiBlazorViewer** на **Razor** страницу, добавить объект **StiReport** и присвоить его вьюверу, используя свойство Report. После загрузки отчета из файла, он будет отображен во вьювере автоматически.

© 2003-2026 Stimulsoft

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorViewer Report="@Report" />

@code
{
   //Report object to use in viewer
   private StiReport Report;

   protected override void OnInitialized()
   {
      base.OnInitialized();

      //Create empty report object
      var report = new StiReport();

      //Load report template
      report.Load("Reports/TwoSimpleLists.mrt");

      //Assing report object to viewer
      Report = report;
   }
}
```

Если отчет не был построен перед отображением, то компонент **Blazor Viewer** построит его автоматически. Таким образом, для отображения разрешено использовать различные типы отчетов: шаблоны отчетов и уже построенные отчеты.

Загрузка шрифта

Blazor не имеет доступа к шрифтам, установленным на компьютере, поэтому для корректного построения отчета требуется указать используемые в отчете шрифты. Это можно сделать при помощи статического класса **StiFontCollection**, указав файл либо Base64 строку, содержащую необходимый шрифт.

© 2003-2026 Stimulsoft

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Web

<StiBlazorViewer Report="@Report" />

@code
{
   //Report object to use in viewer
   private StiReport Report;

   protected override void OnInitialized()
   {
      base.OnInitialized();

      //Init base font as a file
      Stimulsoft.Base.StiFontCollection.AddFontFile("Fonts/Microsoft Sans
      Serif.ttf", "Segoe UI");

      //Init base font as a Base64 string
      var fontBase64 = "AAEAAAAVAQAABABQRFNJR/W/YxAAA814A...";
      Stimulsoft.Base.StiFontCollection.AddFontBase64(fontBase64, "Segoe
      UI");

      var report = new StiReport();

      //Load report template
      report.Load("Reports/TwoSimpleLists.mrt");

      //Assing report object to viewer
      Report = report;
   }
}
```

7.1.3 Подключение данных

Данные для построения отчета могут быть подключены различными способами. Самый простой – хранить настройки подключения в самом шаблоне отчета. Так же данные можно подключить из кода, это можно сделать перед присвоением отчета вьюверу.

© 2003-2026 Stimulsoft

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorViewer Report="@Report" />

@code
{
   //Report object to use in viewer
   private StiReport Report;

   protected override void OnInitialized()
   {
      base.OnInitialized();

      //Load new data from XML file
      var data = new System.Data.DataSet();
      data.ReadXml("Data/Demo.xml");

      //Create empty report object
      var report = new StiReport();

      //Load report template
      report.Load("Reports/TwoSimpleLists.mrt");
      report.Dictionary.Databases.Clear();
      report.RegData("Demo", data);

      //Assing report object to viewer
      Report = report;
   }
}
```

Информация

В данный момент SQL источники данных поддерживаются поддерживаются только для Blazor Server проектов.

SQL источники данных

Параметры подключения к SQL источнику данных, как и к любому другому, можно хранить в самом шаблоне отчета. Если требуется задать параметры подключения из кода перед построением отчета (например, по причине безопасности или в зависимости от авторизованного пользователя), можно воспользоваться приведенным ниже примером.

© 2003-2026 Stimulsoft

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorViewer Report="@Report" />

@code
{
   //Report object to use in viewer
   private StiReport Report;

   protected override void OnInitialized()
   {
      base.OnInitialized();

      OracleConnection connection = new OracleConnection("Data
      Source=Oracle8i;Integrated Security=yes");
      connection.Open();

      OracleDataAdapter adapter = new OracleDataAdapter();
      adapter.SelectCommand = new OracleCommand("SELECT * FROM Products",
      connection);

      var dataSet = new DataSet("productsDataSet");
      adapter.Fill(dataSet, "Products");

      var report = new StiReport();
      report.Load("Reports/TwoSimpleLists.mrt");
      report.Dictionary.Databases.Clear();
      report.RegData("Products", dataSet);

      //Assing report object to viewer
      Report = report;
   }
}
```

Также, для SQL источников данных используемых в отчете, можно указать **Время ожидания запроса (Query Timeout)** в секундах. Значение данного свойства сохраняется в самом шаблоне отчёта для каждого SQL подключения отдельно.

Ниже приведен пример кода, который предоставляет возможность изменить строку соединения для MS SQL, скорректировать запрос, установить время ожидания запроса для уже созданного соединения и источников данных в отчете.

© 2003-2026 Stimulsoft

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorViewer Report="@Report" />

@code
{
   //Report object to use in viewer
   private StiReport Report;

   protected override void OnInitialized()
   {
      base.OnInitialized();

      var report = new StiReport();
      report.Load("Reports/Report.mrt");
      ((StiSqlDatabase)
      report.Dictionary.Databases["Connection"]).ConnectionString = @"Data
      Source=server;Integrated Security=True;Initial Catalog=DataBase";
      ((StiSqlSource)
      report.Dictionary.DataSources["DataSourceName"]).SqlCommand = "select
      * from Table where Column = 100";
      ((StiSqlSource)
      report.Dictionary.DataSources["DataSourceName"]).CommandTimeout =
      1000;

      //Assing report object to viewer
      Report = report;
   }
}
```

Информация

Для SQL источников данных других типов подключение создается аналогично, при этом подключается адаптер, соответствующий типу источника данных. Например, для MS SQL источника необходимо подключать SqlDataAdapter, для Oracle необходим OracleDataAdapter. Также, при этом следует указать строку соединения, соответствующую типу подключения.

Вы также можете использовать данные для разработки отчетов и дашбордов полученные из OData хранилищ данных. В этом случае, авторизация может быть выполнена с использованием имени, пароля пользователя или с использованием токена. Параметры авторизации указываются в строке соединения к OData хранилищу с использованием разделителя ";".

© 2003-2026 Stimulsoft

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorViewer Report="@Report" />

@code
{
   //Report object to use in viewer
   private StiReport Report;

   protected override void OnInitialized()
   {
      base.OnInitialized();

      var report = new StiReport();

      //Authorization using a user account
      var oDataDatabase = new StiODataDatabase("OData", "OData", @"https://
      services.odata.org/V4/Northwind/
      Northwind.svc;AddressBearer=adress;UserName=UserName;Password=Passwor
      d;Client_Id=Your Client ID", false, null);

      //Authorization using a user token
      var oDataDatabase = new StiODataDatabase("OData", "OData", @"https://
      services.odata.org/V4/Northwind/Northwind.svc;Token=Enter your
      token", false, null);

      report.Dictionary.Databases.Add(oDataDatabase);
      oDataDatabase.Synchronize(report);

      //Query with data filter
      ((StiSqlSource)report.Dictionary.DataSources["Products"]).SqlCommand
      = "Products?$filter=ProductID eq 2";

      //Assing report object to viewer
      Report = report;
   }
}
```

В таблице ниже представлены шаблоны строк подключения для различных типов источников данных.

Источник данных	Шаблон строки соединения
MS SQL	Integrated Security=False; Data Source=myServerAddress;Initial Catalog=myDataBase; User ID=myUsername;

© 2003-2026 Stimulsoft

	Password=myPassword;
MySQL	Server=myServerAddress; Database=myDataBase;UserId=myUsername; Pwd=myPassword;
ODBC	Driver={SQL Server}; Server=myServerAddress;Database=myDataBase ; Uid=myUsername; Pwd=myPassword;
OLE DB	Provider=SQLOLEDB.1; Integrated Security=SSPI;Persist Security Info=False; Initial Catalog=myDataBase;Data Source=myServerAddress
Oracle	Data Source=TORCL;User Id=myUsername;Password=myPassword;
MS Access	Provider=Microsoft.Jet.OLEDB.4.0;User ID=Admin;Password=pass;Data Source=C:\ \myAccessFile.accdb;
PostgreSQL	Server=myServerAddress; Port=5432; Database=myDataBase;User Id=myUsername; Password=myPassword;
Firebird	User=SYSDBA; Password=masterkey; Database=SampleDatabase.fdb;DataSource=my ServerAddress; Port=3050; Dialect=3; Charset=NONE;Role=; Connection lifetime=15; Pooling=true; MinPoolSize=0;MaxPoolSize=50; Packet Size=8192; ServerType=0;
SQL CE	Data Source=c:\MyData.sdf; Persist Security Info=False;
SQLite	Data Source=c:\mydb.db; Version=3;
DB2	Server=myAddress:myPortNumber;Database=m yDataBase;UID=myUsername;PWD=myPassword; Max Pool Size=100;Min Pool Size=10;
Infomix	Database=myDataBase;Host=192.168.10.10;Serv er=db_engine_tcp;Service=1492;Protocol=onsoc tcp;UID=myUsername;Password=myPassword;
Sybase	Data

© 2003-2026 Stimulsoft

	Source=myASEserver;Port=5000;Database=myDataBase;Uid=myUsername;Pwd=myPassword;
Teradata	Data Source=myServerAddress;User ID=myUsername;Password=myPassword;
VistaDB	Data Source=D:\folder\myVistaDatabaseFile.vdb4;Open Mode=ExclusiveReadWrite;
Universal(dotConnect)	Provider=Oracle;direct=true;data source=192.168.0.1;port=1521;sid=sid;user=user;password=pass
MongoDB	mongodb://<user>:<password>@localhost/test
OData	http://services.odata.org/v3/odata/OData.svc/

Информация

В таблице представлены наиболее часто используемые шаблоны для строки соединения. Посмотреть различные варианты строк соединения вы можете на специализированном сайте.

Данные из XML, JSON, Excel файлов

Подключения к XML и JSON источникам данных можно хранить в шаблоне отчета. Если требуется указать файлы данных из кода, можно воспользоваться приведенным ниже примером.

© 2003-2026 Stimulsoft

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorViewer Report="@Report" />

@code
{
   private StiReport Report;

   protected override void OnInitialized()
   {
      base.OnInitialized();

      //Init base font
      Stimulsoft.Base.StiFontCollection.AddFontFile("Fonts/Microsoft Sans
      Serif.ttf", "Segoe UI");

      //Load new data from XML file
      var dataSet = new System.Data.DataSet();
      dataSet.ReadXml("Data/Demo.xml");

      //Create empty report object
      var report = new StiReport();

      //Load report template
      report.Load("Reports/TwoSimpleLists.mrt");

      //Register data for the report
      report.RegData("Demo", dataSet);

      //Assing report object to viewer
      Report = report;
   }
}
```

© 2003-2026 Stimulsoft

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorViewer Report="@Report" />

@code
{
   private StiReport Report;

   protected override void OnInitialized()
   {
      base.OnInitialized();

      //Init base font
      Stimulsoft.Base.StiFontCollection.AddFontFile("Fonts/Microsoft Sans
      Serif.ttf", "Segoe UI");

      //Load new data from JSON file
      var dataSet = StiJsonToDataSetConverterV2.GetDataSetFromFile("Data/
      Demo.json");

      var report = new StiReport();

      //Load report template
      report.Load("Reports/TwoSimpleLists.mrt");

      //Register data for the report
      report.RegData(dataSet);

      //Assing report object to viewer
      Report = report;
   }
}
```

Информация

Во вьювере присутствует возможность получения данных из Excel файла. Для этого можно воспользоваться указанным ниже методом.

```
var dataSet = StiExcelConnector.Get().GetDataSet(new
StiExcelOptions(array, this.FirstRowIsHeader));
```

7.1.4 Локализация вьювера

Компонент **Blazor Viewer** поддерживает полную локализацию своего интерфейса. Для того чтобы локализовать интерфейс вьювера отчетов на необходимый язык, используется специальное свойство **Localization**. В качестве

© 2003-2026 Stimulsoft

значения данного свойства необходимо указать путь к XML-файлу локализации (относительный либо абсолютный).

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorViewer Localization="Localization/en.xml" />
```

При загрузке вьювера отчетов файл локализации будет загружен автоматически.

7.1.5 Использование тем

Компонент **Blazor Viewer** имеет возможность смены тем оформления визуальных элементов управления. Для изменения темы используется свойство **Theme** в опциях компонента, которое может принимать одно из значений перечисления **StiViewerTheme**.

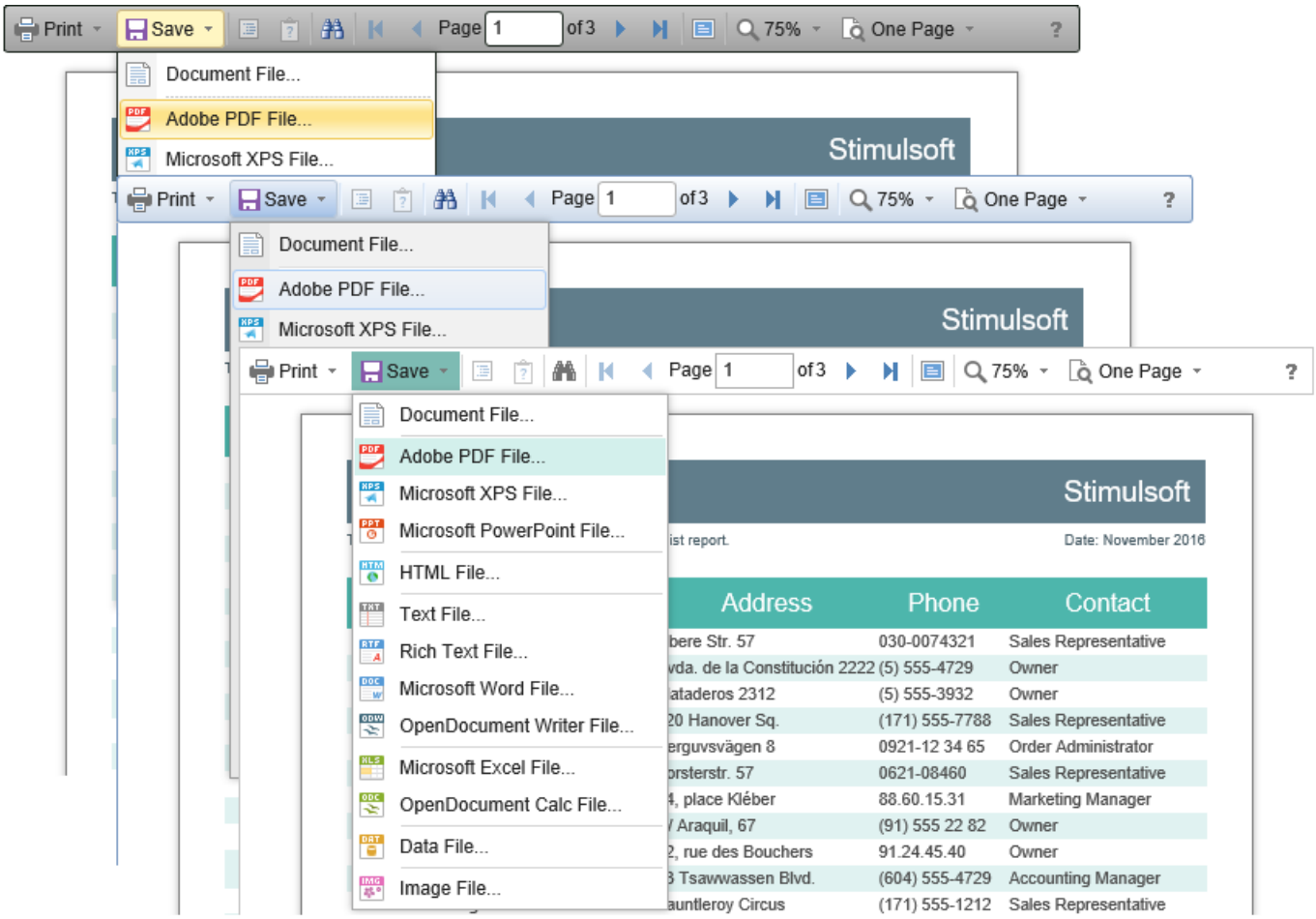
Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorViewer Theme="StiViewerTheme.Office2022WhiteCarmine" />
```

На данный момент доступно **8 тем** оформления с различными цветовыми акцентами. В результате, доступно более **60** вариантов оформления. Это позволяет настроить внешний вид вьювера практически под любое оформление Web проекта.

© 2003-2026 Stimulsoft



По умолчанию вьювер отображает только верхнюю панель инструментов, на которой находятся все элементы управления отчетом. При необходимости, панель инструментов можно разделить на верхнюю и нижнюю. На верхней панели будут находиться меню печати и экспорта отчета, а также кнопки работы с параметрами и закладками. Нижняя панель инструментов будет содержать элементы переключения между страницами отчета и меню управления масштабом. Для включения указанного режима предназначено свойство **DisplayMode**, которое может иметь значение **Simple** (стандартный простой режим) и **Separated** (раздельный режим).

© 2003-2026 Stimulsoft

Index.razor

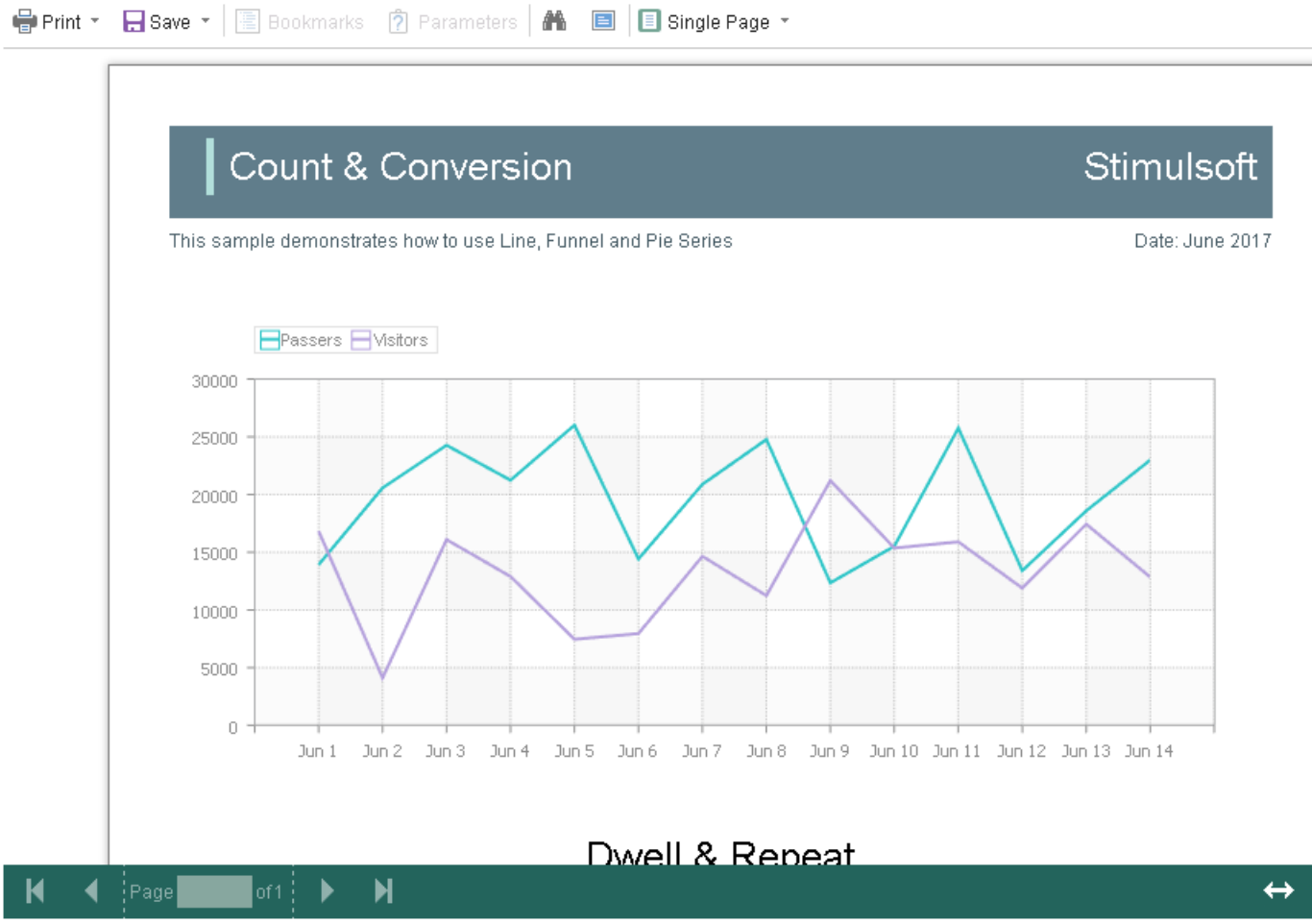
```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorViewer Options="@Options" />

@code
{
   //Options object
   private StiBlazorViewerOptions Options;

   protected override void OnInitialized()
   {
      base.OnInitialized();

      //Init options object
      Options = new StiBlazorViewerOptions();
      Options.Appearance.ScrollbarsMode = true;
      Options.Toolbar.DisplayMode = StiToolbarDisplayMode.Separated;
   }
}
```



© 2003-2026 Stimulsoft

Дополнительно предусмотрена возможность установки параметров оформления основных элементов вьювера. Например, можно изменить шрифт и цвет надписей панели управления вьювера, установить фон вьювера, задать цвет границ страницы и др. Ниже представлен список доступных свойств, изменяющих оформление вьювера, и их значения по умолчанию.

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web
@using System.Drawing

<StiBlazorViewer Options="@Options" />

@code
{
   //Options object
   private StiBlazorViewerOptions Options;

   protected override void OnInitialized()
   {
      base.OnInitialized();

      //Init options object
      Options = new StiBlazorViewerOptions();

      Options.Appearance.BackgroundColor = Color.White;
      Options.Appearance.PageBorderColor = Color.Blue;
      Options.Appearance.ShowPageShadow = true;

      Options.Toolbar.BackgroundColor = Color.White;
      Options.Toolbar.BorderColor = Color.Gray;
      Options.Toolbar.FontColor = Color.Black;
      Options.Toolbar.FontFamily = "Arial";
   }
}
```

7.1.6 Основные возможности

К основным возможностям **Blazor Viewer** можно отнести следующие операции: отображение отчета, переключение между страницами отчета, изменение масштаба и режима отображения отчета. Все указанные операции выполняются без перезагрузки страницы браузера. Для их выполнения не требуется установки никаких специальных опций либо событий.

Вьювер отчетов имеет специальное событие **OnViewerEvent**, которое будет вызвано при любом действии вьювера. В данном событии можно узнать тип

© 2003-2026 Stimulsoft

действия, которое вьювер выполняет в данный момент, а также получить все параметры вьювера, переданные на сторону сервера.

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorViewer OnViewerEvent="@OnViewerEvent" />

@code
{
   private void OnViewerEvent(StiReportDataEventArgs args)
   {
      var action = args.Action;
      var report = args.Report;
      var parameters = args.RequestParams;
   }
}
```

Информация

Событие не будет вызвано для событий, имеющих свои собственные обработчики - печать, экспорт, интерактивные действия над отчетом и т.п. Эти события описаны отдельно в соответствующих разделах документации.

7.1.7 Печать отчета

В компоненте **Blazor Viewer** предусмотрено несколько вариантов печати отчета. Каждый имеет свои особенности, достоинства и недостатки.

Печать в PDF (Print to PDF)

Печать будет осуществляться посредством экспортирования отчета в **PDF формат**. К достоинствам можно отнести большую точность расположения и печати элементов отчета по сравнению с остальными вариантами печати. Из недостатков можно упомянуть обязательное наличие установленного в браузере плагина для просмотра PDF файлов (современные браузеры имеют встроенное средство просмотра и печати PDF файлов).

Печать с просмотром (Print with Preview)

Печать отчета будет осуществлена в отдельном всплывающем окне браузера в **Blazor Viewer**. Отчет можно предварительно просмотреть, а затем отправить на принтер или скопировать в другое место в виде текста или HTML-кода. К

© 2003-2026 Stimulsoft

достоинствам можно отнести кроссбраузерность при печати, отсутствие необходимости установки специальных плагинов. Недостатком является относительно невысокая точность расположения элементов отчета, обусловленная особенностями реализации HTML-форматирования.

Печать без просмотра (Print without Preview)
Печать отчета будет осуществлена непосредственно на принтер без предварительного просмотра. После выбора этого пункта меню отображается системный диалог печати. Так как печать в этом режиме осуществляется в HTML формат, то качество печати аналогично качеству печати отчета с предварительным просмотром.

Информация

При печати в **HTML формат** необходимо убедиться в соответствии параметров страницы отчета и параметров страницы принтера (размер бумаги, ориентация, поля, отступы), а также проверить настройки печати браузера, такие как отступы, колонтитулы, печать фоновых изображений, цветная печать.

Для работы функции печати не требуется дополнительных настроек вьювера. Если необходимо выполнить какие-либо действия перед печатью отчета, то можно определить специальное событие **OnPrintReport**.

© 2003-2026 Stimulsoft

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorViewer OnPrintReport="@OnPrintReport" />

@code
{
   //Options object
   private StiBlazorViewerOptions Options;

   private void OnPrintReport(StiPrintReportEventArgs args)
   {
      // Some code before print
      // ...
   }
}
```

Настройка печати отчета

При выборе печати отчета на панели вьювера, отображается меню с выбором варианта печати. Компонент **Blazor Viewer** имеет возможность принудительно установить требуемый режим печати. Для этого достаточно установить свойство **PrintDestination** в одно из указанных ниже значений перечисления **StiPrintDestination**.

- **Default** – при выборе печати будет отображено меню (значение свойства по умолчанию);
- **Pdf** – печать в PDF формат;
- **Direct** – печать в HTML формат непосредственно на принтер, будет отображен системный диалог печати;
- **WithPreview** – печать в HTML формат с предварительным просмотром во всплывающем окне.

© 2003-2026 Stimulsoft

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Web

<StiBlazorViewer Options="@Options" />

@code
{
   //Options object
   private StiBlazorViewerOptions Options;


   protected override void OnInitialized()
   {
      base.OnInitialized();

      Options = new StiBlazorViewerOptions();
      Options.Toolbar.PrintDestination = StiPrintDestination.Default;
   }
}
```

Компонент **HTML5 Viewer** имеет возможность полностью отключить печать отчета, если она не требуется. Для этого необходимо установить значение **false** для свойства **ShowPrintButton**.

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Web

<StiBlazorViewer Options="@Options"></StiBlazorViewer>

@code
{
   //Options object
   private StiBlazorViewerOptions Options;

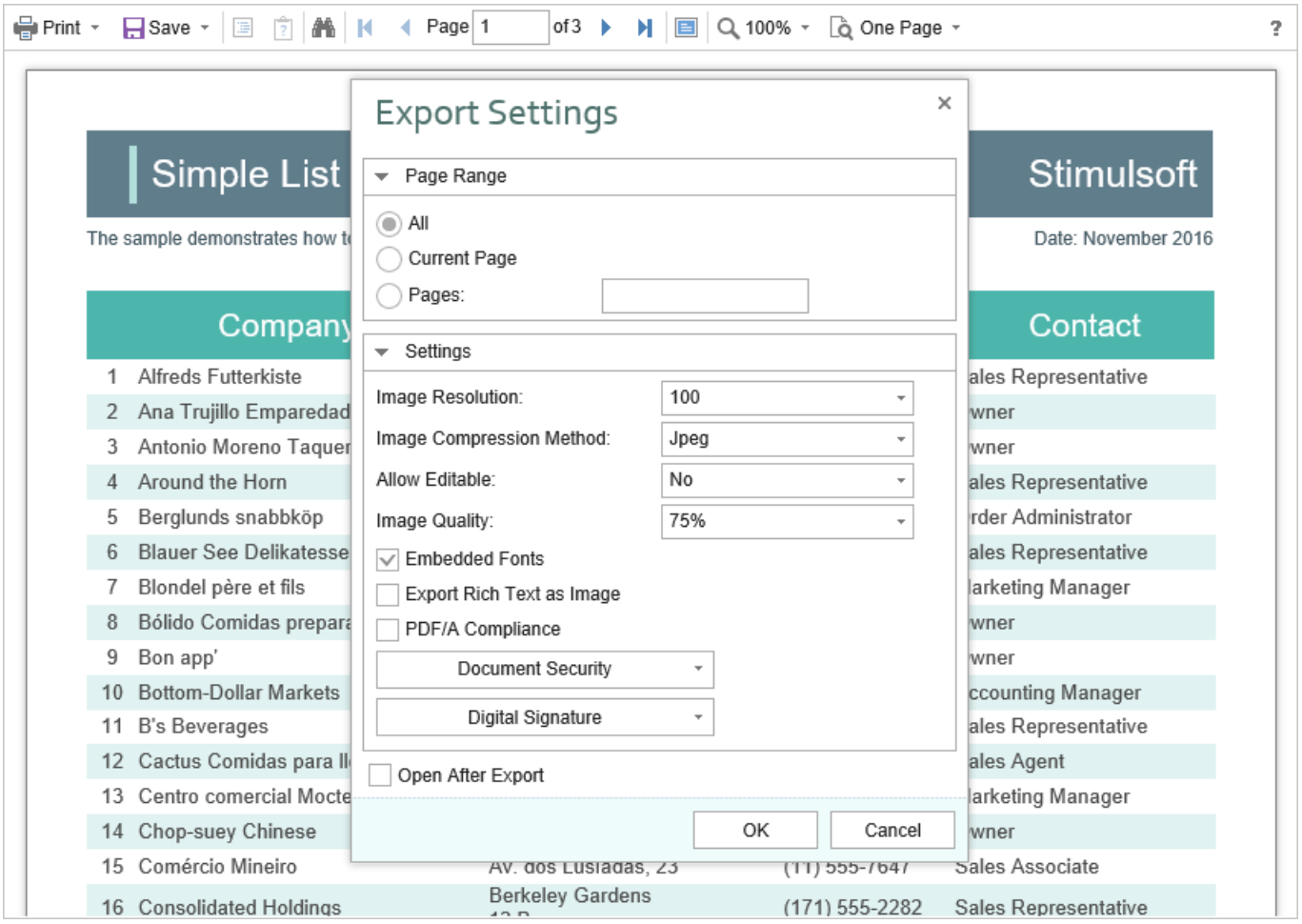

   protected override void OnInitialized()
   {
      base.OnInitialized();

      Options = new StiBlazorViewerOptions();
      Options.Toolbar.ShowPrintButton = true;
   }
}
```

7.1.8 Экспорт отчета

Компонент **Blazor Viewer** позволяет экспортировать отображаемый отчет в различные форматы, такие как **PDF**, **HTML**, **Word**, **Excel**, текст и другие. Для работы функции экспорта не требуется дополнительных настроек вьювера.

© 2003-2026 Stimulsoft



Если необходимо выполнить какие-либо действия перед экспортированием отчета, то можно определить специальное событие **OnExportReport**.

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorViewer OnExportReport="@OnExportReport" />

@code
{
   private void OnExportReport(StiExportReportEventArgs args)
   {
      // Some code before export
      // ...
   }
}
```

© 2003-2026 Stimulsoft

Настройки экспорта

Каждый формат экспорта отчета компонента **Blazor Viewer** имеет множество настроек, и каждая настройка имеет свои значения по умолчанию. Иногда требуется установить другие значения по умолчанию. Для этого предназначено специальное свойство вьювера DefaultSettings, которое можно найти в опциях экспорта. Это свойство представляет собой контейнер всех настроек экспортов, используемых по умолчанию.

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorViewer Options="@Options" />

@code
{
   //Options object
   private StiBlazorViewerOptions Options;

   protected override void OnInitialized()
   {
      base.OnInitialized();

      //Init options object
      Options = new StiBlazorViewerOptions();

      //PDF default settings
      Options.Exports.DefaultSettings.ExportToPdf.ImageQuality = 0.75f;
      Options.Exports.DefaultSettings.ExportToPdf.ImageFormat =
      Stimulsoft.Report.Export.StiImageFormat.Color;

      //HTML default settings
      Options.Exports.DefaultSettings.ExportToHtml.UseEmbeddedImages =
      true;
      Options.Exports.DefaultSettings.ExportToHtml.ExportMode =
      Stimulsoft.Report.Export.StiHtmlExportMode.Div;
   }
}
```

Если требуется, можно полностью убрать отображение диалоговых окон экспорта, экспортирование всегда будет осуществляться с настройками по умолчанию. Для этого достаточно установить значение **false** для свойства **ShowExportDialog**.

© 2003-2026 Stimulsoft

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorViewer Options="@Options" />

@code
{
   //Options object
   private StiBlazorViewerOptions Options;

   protected override void OnInitialized()
   {
      base.OnInitialized();

      //Init options object
      Options = new StiBlazorViewerOptions();
      Options.Exports.ShowExportDialog = false;
   }
}
```

Компонент **Blazor Viewer** содержит порядка 20 различных форматов экспорта, и иногда требуется отключить не используемые форматы. Это позволяет разгрузить интерфейс и упростить использование вьювера. Для отключения не используемых форматов экспортов достаточно установить значение **false** для соответствующих свойств вьювера, представленных в списке ниже.

© 2003-2026 Stimulsoft

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorViewer Options="@Options" />

@code
{
   //Options object
   private StiBlazorViewerOptions Options;

   protected override void OnInitialized()
   {
      base.OnInitialized();

      //Init options object
      Options = new StiBlazorViewerOptions();
      Options.Exports.ShowExportDialog = true;
      Options.Exports.ShowExportToDocument = true;
      Options.Exports.ShowExportToPdf = true;
      Options.Exports.ShowExportToXps = true;
      Options.Exports.ShowExportToPowerPoint = true;
      Options.Exports.ShowExportToHtml = true;
      Options.Exports.ShowExportToHtml5 = true;
      Options.Exports.ShowExportToMht = true;
      Options.Exports.ShowExportToText = true;
      Options.Exports.ShowExportToRtf = true;
      Options.Exports.ShowExportToWord = true;
      Options.Exports.ShowExportToOpenDocumentWriter = true;
      Options.Exports.ShowExportToExcel = true;
      Options.Exports.ShowExportToExcelXml = true;
      Options.Exports.ShowExportToExcelBiff = true;
      Options.Exports.ShowExportToOpenDocumentCalc = true;
      Options.Exports.ShowExportToCsv = true;
      Options.Exports.ShowExportToDbf = true;
      Options.Exports.ShowExportToXml = true;
      Options.Exports.ShowExportToDif = true;
      Options.Exports.ShowExportToSylk = true;
   }
}
```

Компонент **Blazor Viewer** имеет возможность полностью отключить меню экспорта отчета, если оно не требуется. Для этого необходимо установить значение **false** для свойства **ShowSaveButton**.

© 2003-2026 Stimulsoft

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorViewer Options="@Options" />

@code
{
   //Options object
   private StiBlazorViewerOptions Options;

   protected override void OnInitialized()
   {
      base.OnInitialized();

      //Init options object
      Options = new StiBlazorViewerOptions();
      Options.Toolbar.ShowSaveButton = false;
   }
}
```

7.1.9 Режимы отображения

В компоненте **Blazor Viewer** предусмотрено два режима отображения отчета: с полосами прокрутки и без них. По умолчанию установлен режим просмотра без полос прокрутки. Для включения режима просмотра с полосами прокрутки достаточно установить значение **true** для свойства **ScrollbarsMode**.

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorViewer Options="@Options" />

@code
{
   //Options object
   private StiBlazorViewerOptions Options;

   protected override void OnInitialized()
   {
      base.OnInitialized();

      //Init options object
      Options = new StiBlazorViewerOptions();
      Options.Appearance.ScrollbarsMode = false;
   }
}
```

© 2003-2026 Stimulsoft

В первом режиме (без полос прокрутки), вьювер отображает страницу или отчет целиком, автоматически растягивая область просмотра. Если заданы размеры по ширине и высоте, то вьювер будет осуществлять обрезку вышедшей за границы страницы. Во втором режиме, в отличие от первого, при выходе страницы за границы размеров вьювера, обрезка осуществляться не будет. Вместо этого появятся полосы прокрутки, при помощи которых можно просмотреть страницу или отчет целиком.

Информация

В режиме просмотра отчета с полосами прокрутки необходимо задать высоту вьювера, иначе будет задана высота по умолчанию, равная **650 пикселей**.

В компоненте **Blazor Viewer** предусмотрен режим полноэкранного отображения отчета или дашборда. По умолчанию включен стандартный режим просмотра, вьювер имеет заданные в настройках размеры. Для включения полноэкранного режима просмотра достаточно установить значение **true** для свойства **FullScreenMode**.

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorViewer Options="@Options" />

@code
{
   //Options object
   private StiBlazorViewerOptions Options;

   protected override void OnInitialized()
   {
      base.OnInitialized();

      //Init options object
      Options = new StiBlazorViewerOptions();
      Options.Appearance.FullScreenMode = true;
   }
}
```

© 2003-2026 Stimulsoft

Также, для включения либо отключения полноэкранного режима можно воспользоваться соответствующей кнопкой на панели управления вьювера.

В компоненте **Blazor Viewer** доступно три режима отображения отчета: постраничное отображение, отчет целиком в виде ленты, и табличное отображение страниц отчета. Для управления режимами предназначено свойство **ViewMode**, которое соответственно принимает одно из указанных значений: **SinglePage**, **Continuous**, **MultiplePages**.

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorViewer Options="@Options" />

@code
{
   //Options object
   private StiBlazorViewerOptions Options;

   protected override void OnInitialized()
   {
      base.OnInitialized();

      //Init options object
      Options = new StiBlazorViewerOptions();
      Options.Toolbar.ViewMode = StiWebViewMode.SinglePage;
   }
}
```

В компоненте **Blazor Viewer** реализована поддержка работы как с обычным компьютером, так и с сенсорными экранами, мобильными устройствами. Для управления режимами интерфейса предназначено свойство **InterfaceType**, которое принимает одно из следующих значений:

- **Auto** – тип интерфейса вьювера будет выбран автоматически в зависимости от используемого устройства (значение по умолчанию);
- **Mouse** – принудительное использование стандартного интерфейса для управления вьювером при помощи мыши;
- **Touch** – принудительное использование Touch интерфейса для управления вьювером при помощи сенсорного экрана монитора, в этом режиме элементы интерфейса вьювера имеют увеличенные размеры для удобства управления;
- **Mobile** – принудительное использование Mobile интерфейса для управления вьювером при помощи экрана смартфона, в этом режиме интерфейс вьювера

© 2003-2026 Stimulsoft

имеет упрощенный вид и адаптирован для управления при помощи мобильного устройства.

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

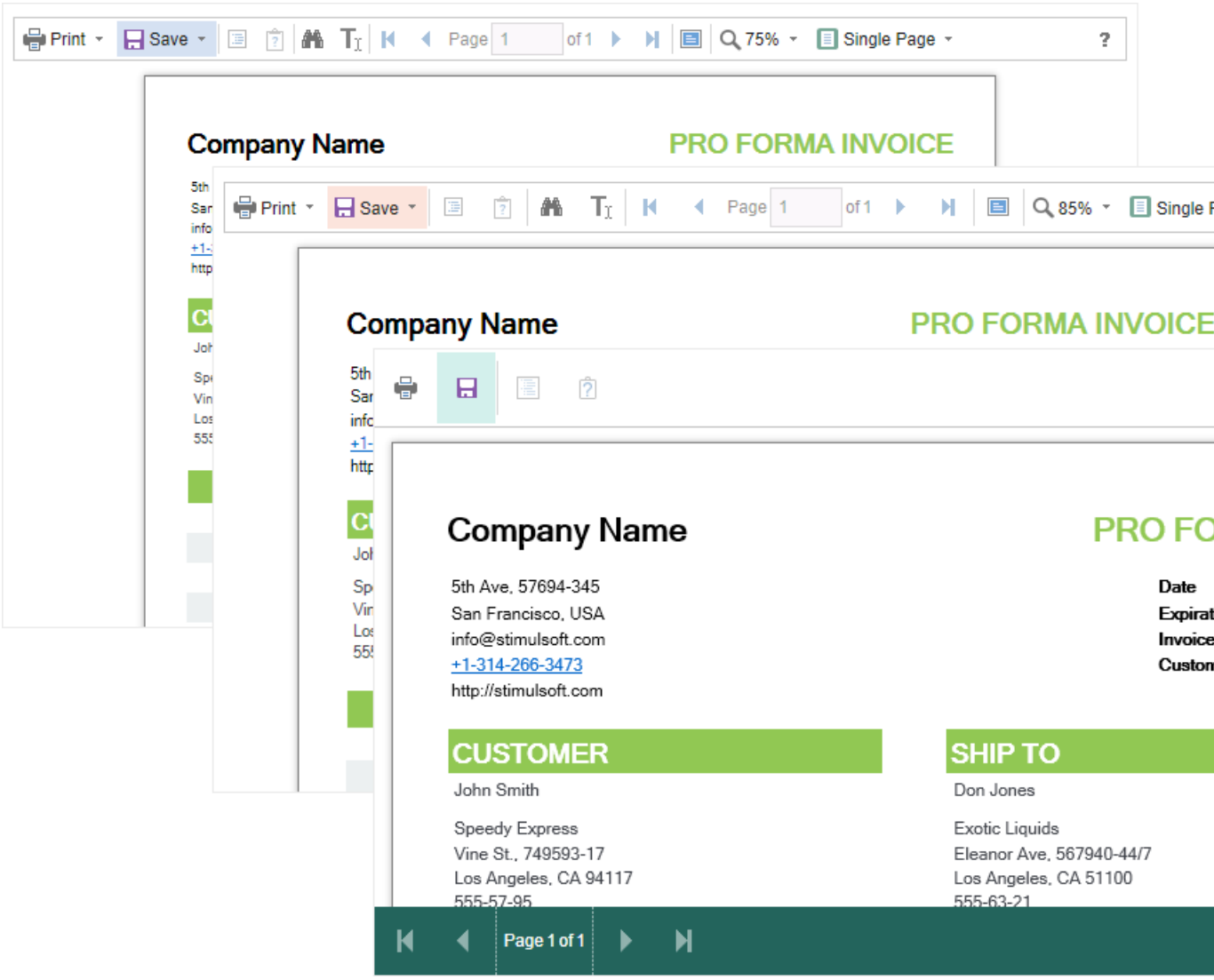
<StiBlazorViewer Options="@Options" />

@code
{
   //Options object
   private StiBlazorViewerOptions Options;

   protected override void OnInitialized()
   {
      base.OnInitialized();

      //Init options object
      Options = new StiBlazorViewerOptions();
      Options.Appearance.InterfaceType = StiInterfaceType.Auto;
   }
}
```

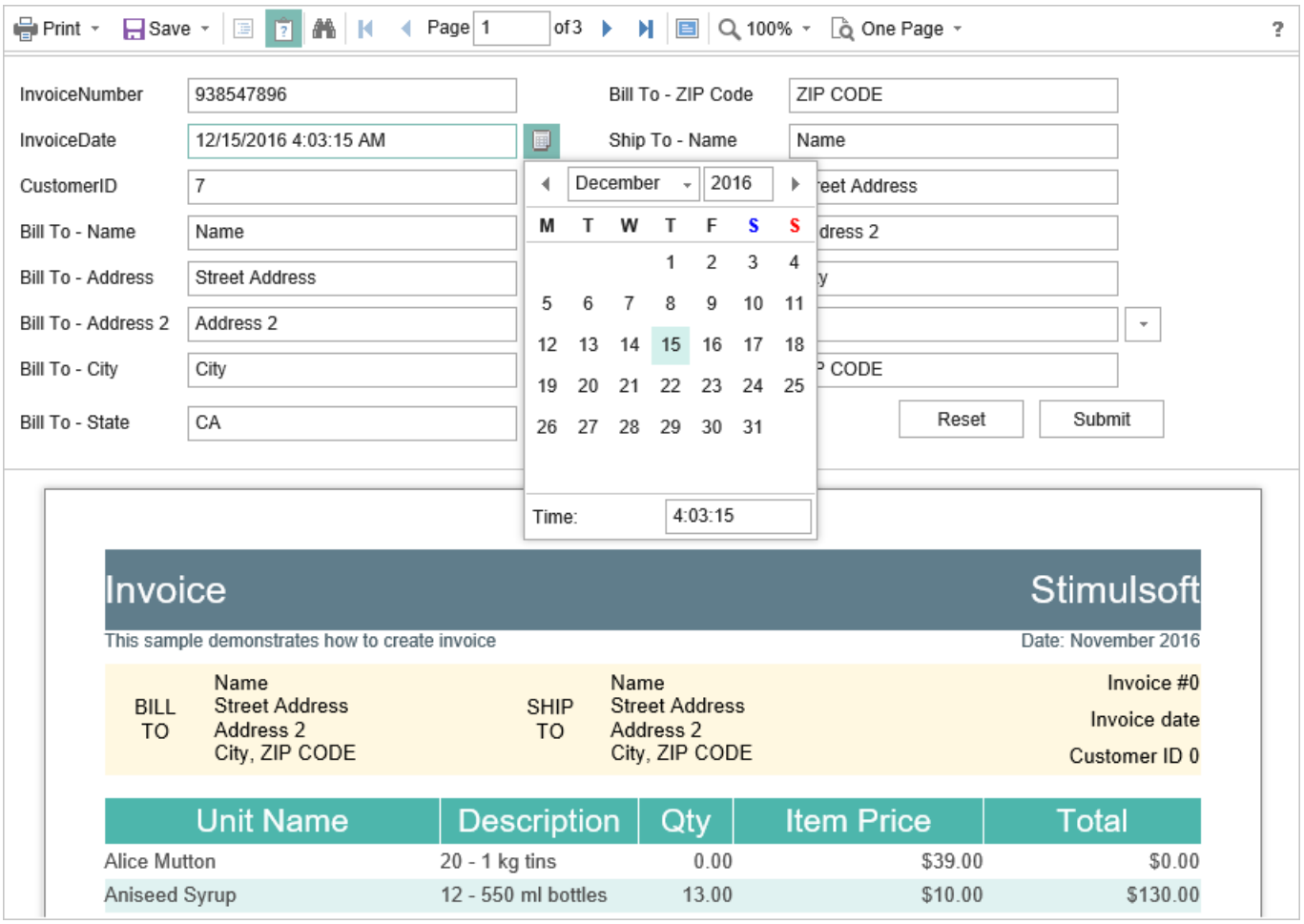
© 2003-2026 Stimulsoft



7.1.10 Работа с параметрами

Для работы с параметрами отчета в **Blazor Viewer** реализована поддержка специальной панели параметров. Для добавления параметра на панель необходимо в отчете определить переменную, запрашиваемую у пользователя. При просмотре отчета во вьюверe такая переменная будет автоматически добавлена на панель параметров. Поддерживаются все типы переменных отчета (обычные переменные, дата и время, границы, списки и др.).

© 2003-2026 Stimulsoft



Для работы отчетов с параметрами не требуется дополнительных настроек вьювера. Если необходимо провести какие-либо действия перед применением параметров, то можно определить специальное событие **OnInteraction**.

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorViewer OnInteraction="@OnInteraction" />

@code
{
   private void OnInteraction(StiReportDataEventArgs args)
   {
      // Some code before any interaction
      // ...
   }
}
```

© 2003-2026 Stimulsoft

Данное событие вызывается при любых интерактивных действиях вьювера. При необходимости выполнить какие-либо действия только при применении параметров отчета, можно воспользоваться значениями, переданными в аргументах события. Аргументы содержат все необходимые сведения и состояния клиентской части вьювера. Для определения типа действия вьювера достаточно проверить свойство **Action**.

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorViewer OnInteraction="@OnInteraction" />

@code
{
   private void OnInteraction(StiReportDataEventArgs args)
   {
      if (args.Action == StiAction.Variables)
      {
         // Some code before apply parameters
      }
   }
}
```

Если работа с параметрами не требуется, то можно полностью отключить данную возможность. Для этого предназначено свойство **ShowParametersButton** в разделе свойств **Toolbar**, для которого необходимо установить значение **false**.

© 2003-2026 Stimulsoft

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorViewer Options="@Options" />

@code
{
   //Options object
   private StiBlazorViewerOptions Options;

   protected override void OnInitialized()
   {
      base.OnInitialized();

      //Init options object
      Options = new StiBlazorViewerOptions();
      Options.Toolbar.ShowParametersButton = false;
   }
}
```

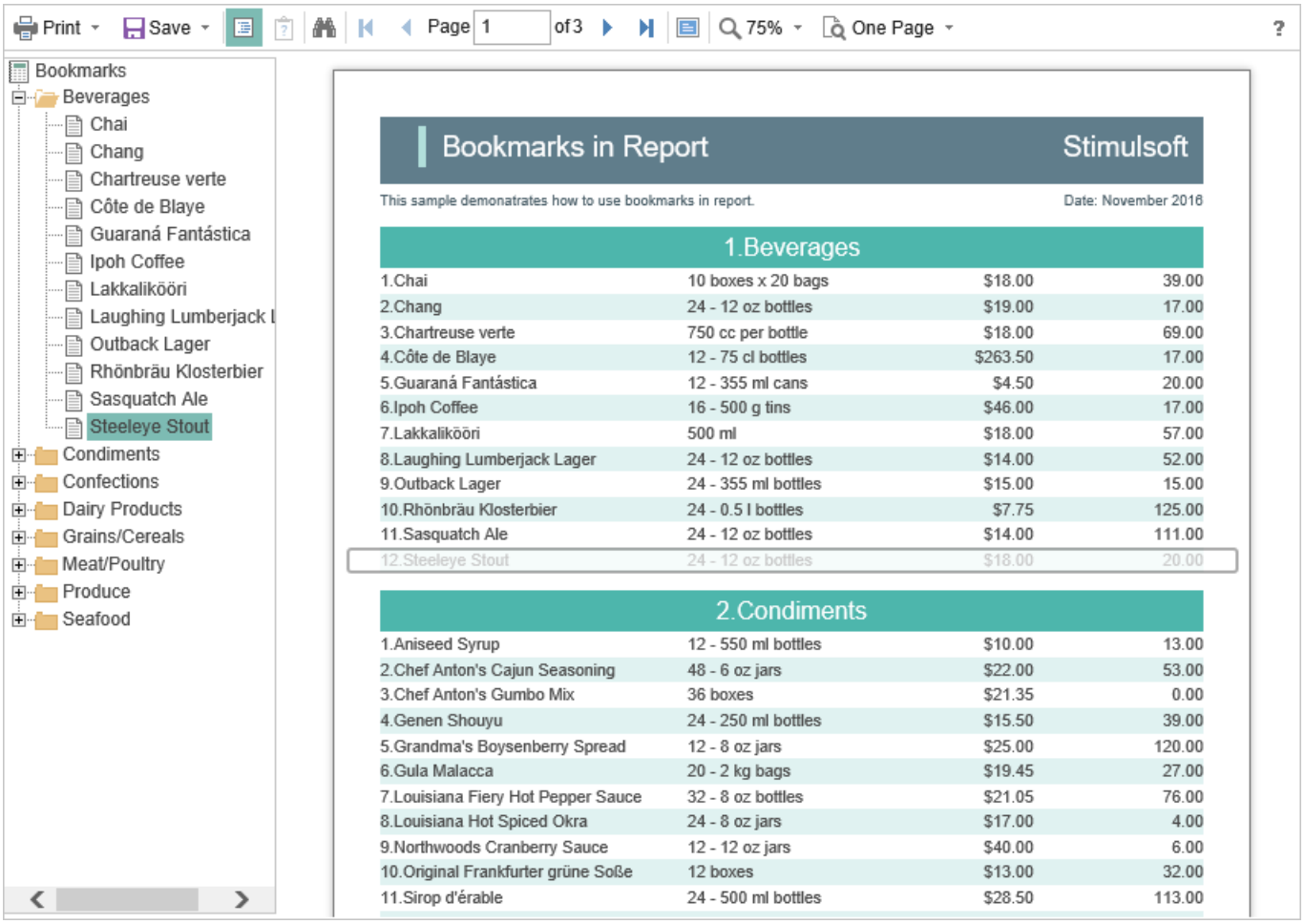
Информация

При такой конфигурации вьювера панель параметров не будет показана даже в том случае, если параметры присутствуют в отображаемом отчете.

7.1.11 Работа с закладками

В компоненте **Blazor Viewer** реализована поддержка закладок отчета. При отображении такого отчета слева от страницы будет отображена панель с закладками. При выборе закладки отчета вьювер осуществит автоматический переход на нужную страницу, а элемент отчета с закладкой будет подсвечен.

© 2003-2026 Stimulsoft



По умолчанию ширина панели закладок равна 180 пикселей, компонент **Blazor Viewer** позволяет изменить это значение. Для этого предназначено свойство **BookmarksTreeWidth**, значение которого указывается в пикселях.

© 2003-2026 Stimulsoft

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorViewer Options="@Options" />

@code
{
   //Options object
   private StiBlazorViewerOptions Options;

   protected override void OnInitialized()
   {
      base.OnInitialized();

      //Init options object
      Options = new StiBlazorViewerOptions();
      Options.Appearance.BookmarksTreeWidth = 200;
   }
}
```

Если работа с закладками отчета не требуется, то можно полностью отключить данную возможность. Для этого предназначено свойство **ShowBookmarksButton**, которое необходимо установить в значение **false**.

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorViewer Options="@Options" />

@code
{
   //Options object
   private StiBlazorViewerOptions Options;

   protected override void OnInitialized()
   {
      base.OnInitialized();

      //Init options object
      Options = new StiBlazorViewerOptions();
      Options.Toolbar.ShowBookmarksButton = false;
   }
}
```

© 2003-2026 Stimulsoft

Информация

В этом случае закладки отчета не будут показаны, даже если они присутствуют в отображаемом отчете. Данное свойство не оказывает влияние на печать и экспортирование отчета.

При печати отчета с закладками дерево закладок будет скрыто. Если помимо самого отчета требуется распечатать и закладки, то необходимо установить свойство **BookmarksPrint** в значение **true**.

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorViewer Options="@Options" />

@code
{
   //Options object
   private StiBlazorViewerOptions Options;

   protected override void OnInitialized()
   {
      base.OnInitialized();

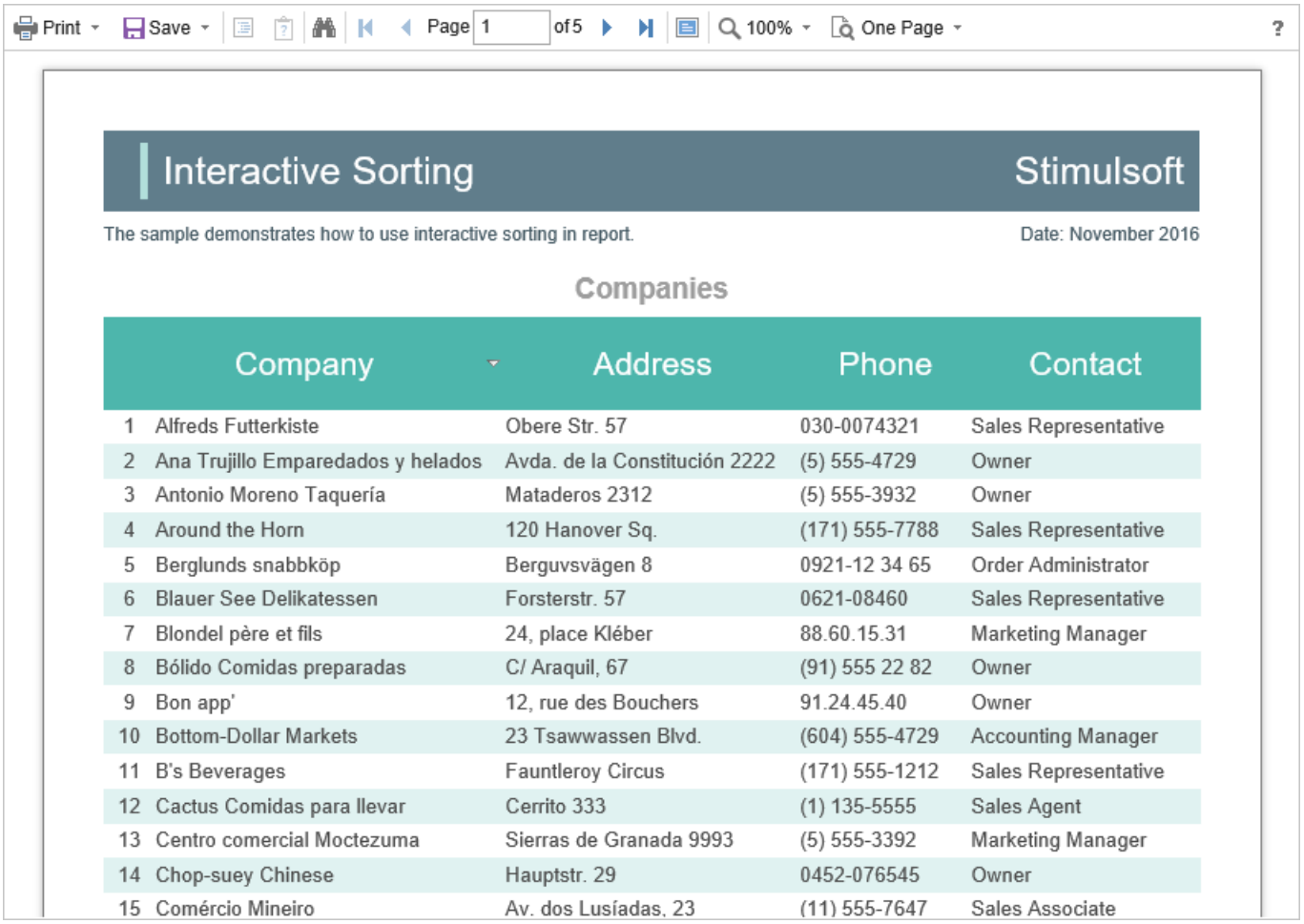
      //Init options object
      Options = new StiBlazorViewerOptions();
      Options.Appearance.BookmarksPrint = true;
   }
}
```

7.1.12 Динамическое сворачивание, сортировка и детализация

В компоненте **Blazor Viewer** реализована поддержка динамической сортировки, сворачивания и детализации отчетов. Динамическая сортировка предоставляет возможность изменять направление сортировки в построенном отчете. Для этого следует щелкнуть по компоненту, у которого была установлена динамическая сортировка. Динамическая сортировка осуществляется в следующих направлениях: **По возрастанию (Ascending)** и **По убыванию (Descending)**. Каждый раз при щелчке по компоненту направление меняется на противоположное.

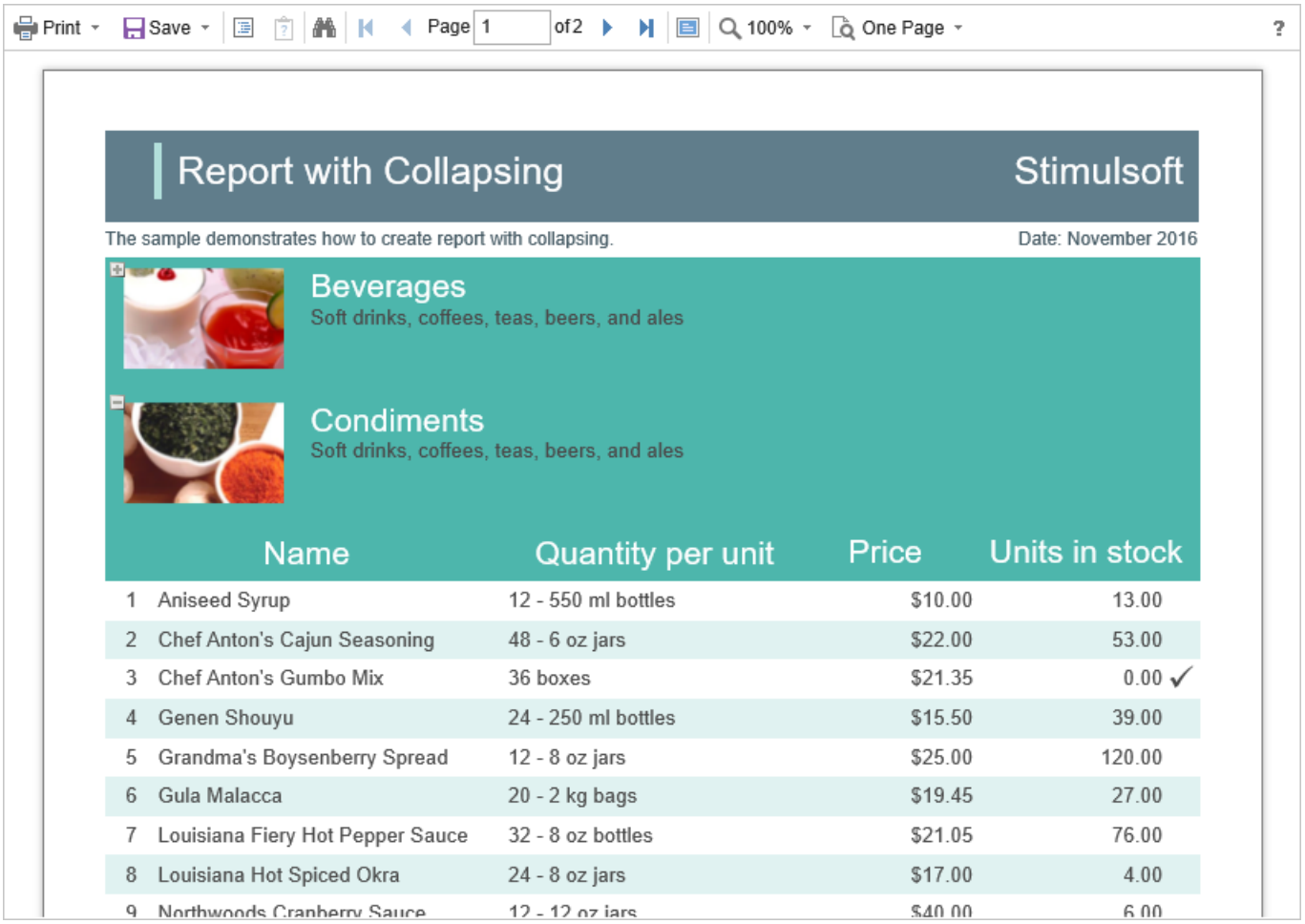
© 2003-2026 Stimulsoft

Допускается многоуровневая сортировка в отчете. Для этого необходимо удерживать клавишу **Ctrl** и последовательно нажимать на сортируемые компоненты отчета. Для сброса сортировки можно нажать на любой сортируемый компонент без удержания клавиши **Ctrl**.



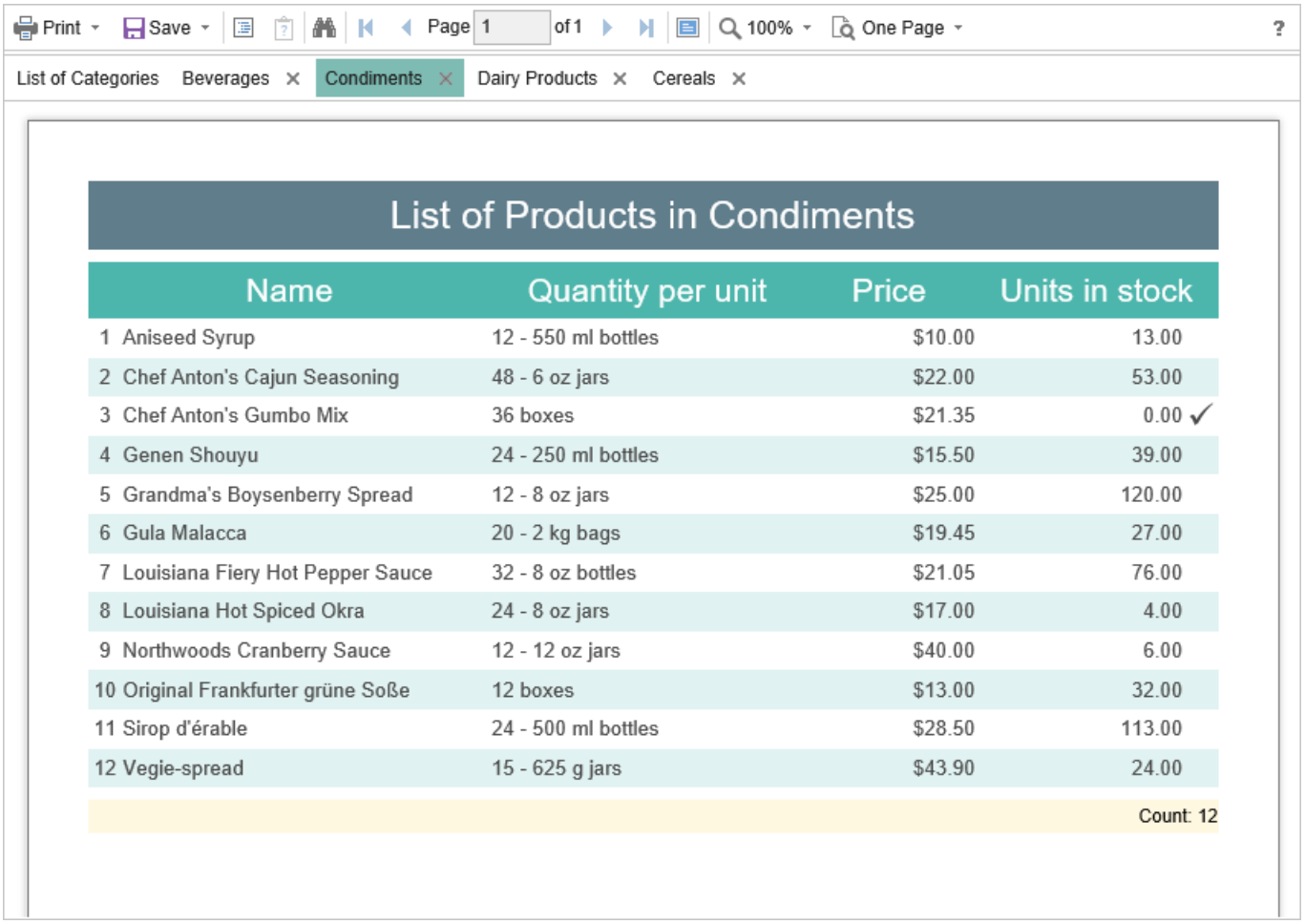
Отчет с динамическим сворачиванием представляет собой интерактивный отчет, в котором сворачивающиеся блоки могут сворачивать/разворачивать свое содержимое при нажатии на заголовок блока. Элементы отчета, которые можно свернуть/развернуть, обозначены специальной иконкой со знаком **[-]** или **[+]**.

© 2003-2026 Stimulsoft



При детализации данных под главной панелью вьювера будет отображена панель детализации с закладками детализированных отчетов. Отображенный в данный момент отчет будет подсвечен.

© 2003-2026 Stimulsoft



Для работы динамической сортировки, сворачивания и детализации отчетов не требуется дополнительных настроек вьювера. Если необходимо провести какие-либо действия перед сортировкой, сворачиванием или детализацией отчета, то можно определить специальное событие **OnInteraction**.

© 2003-2026 Stimulsoft

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorViewer OnInteraction="@OnInteraction" />

@code
{
   private void OnInteraction(StiReportDataEventArgs args)
   {
      // Some code before any interaction
      // ...
   }
}
```

Для получения типа действия можно воспользоваться аргументами события. Для каждого вида интерактивности вьювера предусмотрен определенный тип действия:

- **Sorting** – при использовании сортировки колонок;
- **DrillDown** – при использовании детализации отчета;
- **Collapsing** – при использовании сворачивания блоков отчета.

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorViewer OnInteraction="@OnInteraction" />

@code
{
   private void OnInteraction(StiReportDataEventArgs args)
   {
      switch (args.Action)
      {
         case StiAction.Sorting:
            break;

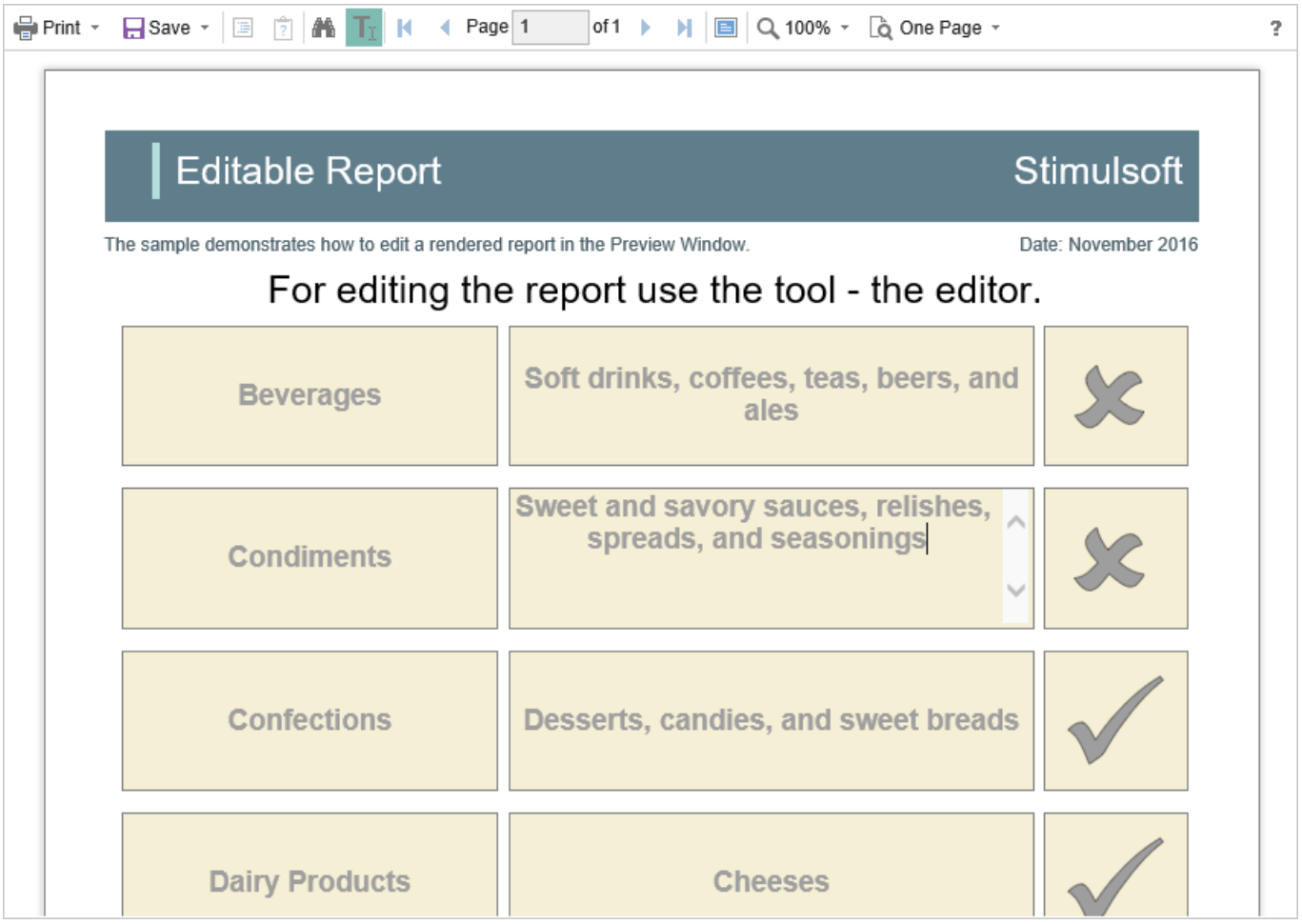
         case StiAction.DrillDown:
            break;

         case StiAction.Collapsing:
            break;
      }
   }
}
```

© 2003-2026 Stimulsoft

7.1.13 Редактирование построенного отчета

Компонент **Blazor Viewer** имеет возможность редактировать элементы построенного отчета, такие как текстовые поля и чекбоксы. Для того, чтобы редактирование стало возможным, в самом шаблоне отчета необходимо пометить нужные компоненты как редактируемые. После отображения отчета во вьювере, для начала редактирования необходимо нажать на соответствующую кнопку на панели вьювера. После завершения редактирования необходимо нажать указанную кнопку еще раз, и все внесенные изменения будут примерены к отчету.



Для работы режима редактирования отчета не требуется никаких специальных настроек вьювера.

© 2003-2026 Stimulsoft

Информация

Отредактированные значения будут применены при печати либо экспортировании отчета, при этом исходный отчет останется не тронутым. После перезагрузки вьювера все значения будут возвращены к исходным.

7.1.14 Отправка отчета по Email

В компоненте **Blazor Viewer** предусмотрена возможность отправки отчета по Email. Для активации данной возможности необходимо установить свойство вьювера **ShowSendEmailButton** в значение **true** и определить событие **OnEmailReport**.

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorViewer Options="@Options" OnEmailReport="@OnEmailReport" />

@code
{
   //Options object
   private StiBlazorViewerOptions Options;

   private void OnEmailReport(StiEmailReportEventArgs args)
   {
      //args.Options.AddressTo = "";
      //args.Options.Subject = "";
      //args.Options.Body = "";

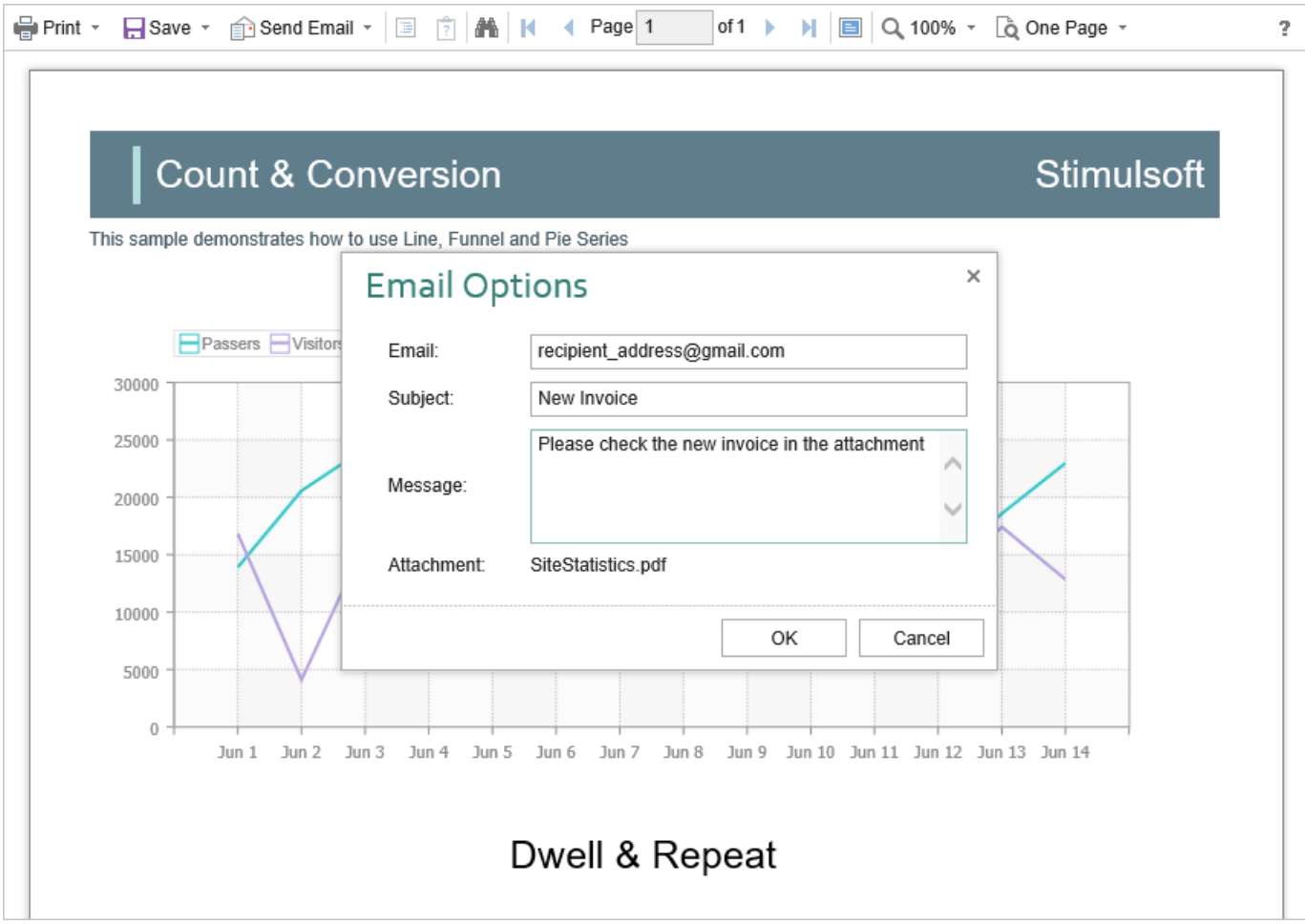
      // Should be filled here
      args.Options.AddressFrom = "admin_address@test.com";
      args.Options.Host = "smtp.test.com";
      args.Options.Port = 465;
      args.Options.UserName = "admin_address@test.com";
      args.Options.Password = "admin_password";

      //args.Options.CC.Add("email@test.com");
      //args.Options.BCC.Add("email@test.com");
      //args.Options.EnableSsl = true;
   }
}
```

При отправке отчета по Email отображается меню выбора формата вложения, которое соответствует меню выбора формата экспортирования отчета. После

© 2003-2026 Stimulsoft

выбора формата отобразиться диалог ввода параметров отправки, таких как Email получателя, тема и текст письма.



После подтверждения отправки будет вызвано описанное выше событие **OnEmailReport**, в котором можно проверить и скорректировать данные, введенные в этой форме. Экспортированный файл отчета будет прикреплен к письму автоматически.

Компонент **Blazor Viewer** позволяет установить значения по умолчанию для формы отправки Email. Для этого предназначены свойства **DefaultEmailAddress**, **DefaultEmailSubject** и **DefaultEmailMessage**. По умолчанию данные свойства являются пустыми.

© 2003-2026 Stimulsoft

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorViewer Options="@Options" />

@code
{
   //Options object
   private StiBlazorViewerOptions Options;

   protected override void OnInitialized()
   {
      base.OnInitialized();

      //Init options object
      Options = new StiBlazorViewerOptions();
      Options.Email.DefaultEmailAddress = "recipient_address@gmail.com";
      Options.Email.DefaultEmailSubject = "New Invoice";
      Options.Email.DefaultEmailMessage = "Please check the new invoice in
      the attachment";
   }
}
```

7.1.15 Вызов дизайнера из вьювера

Компонент **Blazor Viewer** имеет возможность вызвать дизайнер отчетов. Для использования этой возможности необходимо определить событие **OnDesignReport**.

© 2003-2026 Stimulsoft

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web
@inject NavigationManager NavigationManager

<StiBlazorViewer Report="@Report" OnDesignReport="@OnDesignReport" />

@code
{
   //Report object to use in viewer
   private StiReport Report;

   protected override void OnInitialized()
   {
      base.OnInitialized();

      var report = new StiReport();
      report.Load("Reports/TwoSimpleLists.mrt");
      Report = report;
   }

   protected void OnDesignReport(StiReportDataEventArgs args)
   {
      //Redirect to the Designer page
      NavigationManager.NavigateTo("Designer?report=" +
      args.Report.ReportName);
   }
}
```

Информация

Сам вьювер не запускает дизайнер, он лишь вызывает указанное событие, в котором можно получить все необходимые параметры. Далее в действии можно выполнить перенаправление на другую страницу, которая содержит дизайнер отчетов.

7.1.16 Экспорт и печать из кода

Компонент **Blazor Viewer** предоставляет возможность печатать отчет различными способами и выполнять экспортирование отчета в различные форматы. Эти действия выполняются при помощи меню вьювера. Если требуется напечатать отчет или выполнить экспорт при помощи кода, например, в событии нажатия кнопки, то можно воспользоваться специальным классом **StiReportResponse**. Данный класс содержит набор статических методов, позволяющих выполнить печать либо экспорт отчета из кода, при этом вьювер отчетов не требуется.

© 2003-2026 Stimulsoft

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<button @onclick="@OnClickPrintButton">Print PDF</button>
<button @onclick="@OnClickExportButton">Export PDF</button>

@code
{
   private StiReport LoadSimpleList()
   {
      var dataSet = new System.Data.DataSet();
      dataSet.ReadXml("Data/Demo.xml");

      var report = new StiReport();
      report.Load("Reports/SimpleList.mrt");
      report.RegData(dataSet);

      return report;
   }

   protected void OnClickPrintButton()
   {
      var report = LoadSimpleList();

      StiReportResponse.PrintAsPdf(report);
      //StiReportResponse.PrintAsHtml(report);
   }

   protected void OnClickExportButton()
   {
      var report = LoadSimpleList();

      StiReportResponse.ResponseAsPdf(report);
      //StiReportResponse.ResponseAsExcel2007(report);
      //StiReportResponse.ResponseAsPng(report);
      //StiNetCoreReportResponse.ResponseAsJson(report);
   }
}
```

Если на **Razor** странице отсутствуют компоненты по работе с отчётами (вьювер либо дизайнер), то необходимо выполнить предварительную инициализацию генератора отчётов. Это можно сделать, переопределив стандартное событие **OnInitializedAsync** страницы, добавив туда специальный статический метод **StiBlazorHelper.Initialize()**, на вход которого необходимо передать объект **JSRuntime**. После этого действия по экспорту и печати отчёта будут работать корректно.

© 2003-2026 Stimulsoft

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@inject IJSRuntime JSRuntime;

protected override Task OnInitializedAsync()
{
   StiBlazorHelper.Initialize(JSRuntime);

   return base.OnInitializedAsync();
}
```

7.1.17 События вьювера

Компонент **Blazor Viewer** поддерживает события, которые предоставляют возможность выполнить необходимые операции перед определенными действиями, такими как печать и экспортирование отчета, отправка отчета по Email, интерактивность, экспорт дашборда и др. Ниже приведен пример обработки событий вьювера.

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorViewer OnViewerReport="@OnViewerEvent" />

@code
{
   private void OnViewerEvent(StiReportDataEventArgs args)
   {
      var action = args.Action;
      var report = args.Report;
      var parameters = args.RequestParams;
   }
}
```

Список событий

Наименование	Описание
OnOpenReport	Событие возникает при открытии отчета.
OnPrintReport	Событие возникает при печати отчета.
OnExportReport	Событие возникает при экспортировании отчета.

© 2003-2026 Stimulsoft

OnEmailReport	Событие возникает при отправки отчета по Email.
OnInteraction	Событие возникает при работе вьювера с интерактивными операциями, такими как использование параметров, динамическая сортировка, сворачивание и детализация отчета.
OnViewerEvent	Событие возникает при любом действии во вьювере.
OnDesignReport	Событие возникает при нажатии на кнопку Дизайн (Design) на панели инструментов вьювера.
OnViewerAfterRender	Событие возникает после завершения построения HTML5 кода вьювера и всех его элементов управления.

7.1.18 Настройки вьювера

Настройка **Blazor Viewer** выполняется при помощи свойств компонента. Основные настройки устанавливаются у самого Razor компонента, все дополнительные настройки устанавливаются при помощи специального класса опций **StiBlazorViewerOptions**. Ниже приведен пример установки свойств вьювера.

© 2003-2026 Stimulsoft

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorViewer ID="Viewer1" Width="500px" Height="500px"
Localization="Lozalization/en.xml" Options="@Options"
Theme="StiViewerTheme.Office2022WhiteCarmine" />

@code
{
   //Options object
   private StiBlazorViewerOptions Options;

   protected override void OnInitialized()
   {
      base.OnInitialized();

      //Init options object
      Options = new StiBlazorViewerOptions();
      Options.Appearance.ScrollbarsMode = true;
      Options.Appearance.ShowTooltips = false;
      Options.Appearance.ReportDisplayMode =
      StiReportDisplayMode.FromReport;
      Options.Toolbar.DisplayMode = StiToolbarDisplayMode.Separated;
      Options.Exports.ShowExportToDbf = false;
      Options.Exports.ShowExportToDif = false;
      Options.Exports.DefaultSettings.ExportToPdf.CreatorString = "Company
      Name";
      Options.Exports.DefaultSettings.ExportToPdf.ImageQuality = 0.75f;
   }
}
```

Свойства вьювера

Наименование	Описание
ID	Строковый уникальный идентификатор компонента, по умолчанию **StiBlazorViewer**.
Report	Объект отчёта, который будет отображён во вьювере.
Localization	Задает путь к XML файлу локализации. Путь может быть абсолютным либо относительным. По умолчанию используется английская локализация, которая встроена во вьювер и не требует дополнительных XML файлов.
Width	Задает ширину компонента в необходимых

© 2003-2026 Stimulsoft

	единицах измерения – пикселях (по умолчанию), процентах, поинтах. Например, “500”, “500px”, “100%”, “300pt”. По умолчанию установлена ширина 100%.
Height	Задает высоту компонента в необходимых единицах измерения – пикселях (по умолчанию), процентах, поинтах. Например, “500”, “500px”, “100%”, “300pt”. По умолчанию установлена автоматическая высота в зависимости от размера страницы отчета, либо 650 пикселей в режиме отображения вьювера с полосами прокрутки.
Options	Представляет возможность задать объект класса **StiBlazorViewerOptions**, который содержит набор опций (настроек) компонента.
Theme	Задает тему оформления вьювера. Список доступных тем оформления находится в перечислении **StiViewerTheme**. По умолчанию установлено значение **Office2022WhiteBlue**.

Работа с сервером (Server)

Наименование	Описание
UseRelativeUrls	Устанавливает режим дизайнера, в котором для запросов на сервер используются относительные ссылки. По умолчанию свойство имеет значение **true**.
PortNumber	Получает или задает значение, которое указывает номер порта для использования в URL. Значение 0 определяет автоматическое обнаружение (установлено по умолчанию). Значение -1 удаляет номер порта.
PassQueryParametersToReport	Включает использование всех параметров URL запроса в качестве значений

© 2003-2026 Stimulsoft

	одноименных переменных в отчете. По умолчанию свойство имеет значение **false**.
PassQueryParametersForResources	Включает передачу всех параметров URL запроса при формировании ссылок на ресурсы дизайнера. При значении **false** для запроса ресурсов дизайнера используются только необходимые параметры, что способствует более корректной работе кэша браузера. По умолчанию свойство имеет значение **true**.
PassFormValues	Включает передачу значений POST-формы на сторону клиента, если эти значения требуется использовать в действиях дизайнера. При включении данной возможности, вспомогательный метод **GetFormValues()** будет возвращать коллекцию параметров формы. По умолчанию свойство имеет значение **false**.
AllowLoadingCustomFontsToClientSide	Позволяет передавать пользовательские шрифты на клиентскую сторону и преобразовывать их в CSS стиль для корректного отображения текста на стороне HTML с заданным шрифтом. По умолчанию свойство имеет значение **false**.

Внешний вид (Appearance)

Наименование	Описание
IconSet	Предоставляет возможность установить набор иконок: › StiWebUIIconSet.Auto (значение по умолчанию) - устанавливает набор иконок автоматически. Для тем **Office2022** - используется набор иконок в стиле **Monoline**, для тем **Office2013** - используется набор

© 2003-2026 Stimulsoft

	иконок в стиле **Regular**. › StiWebUIIconSet.Monoline - устанавливает набор иконок в стиле **Monoline**; › StiWebUIIconSet.Regular - устанавливает набор иконок в стиле **Regular**.
CustomCss	Задает путь к CSS файлу стилей вьювера. Если данное свойство установлено, то стандартные стили выбранной темы загружены не будут. По умолчанию установлено пустое значение.
BackgroundColor	Задает цвет подложки вьювера. По умолчанию свойство имеет значение **White**.
PageBorderColor	Задает цвет границ страницы отчета. По умолчанию свойство имеет значение **Gray**.
RightToLeft	Устанавливает режим отображения **Right to Left** для элементов управления вьювера. По умолчанию свойство имеет значение **false**.
FullScreenMode	Устанавливает полноэкранный режим отображения вьювера. По умолчанию свойство имеет значение **false**.
ScrollbarsMode	Устанавливает режим отображения отчета с полосами прокрутки. По умолчанию свойство имеет значение **false**.
OpenLinksWindow	Задает целевое окно для открытия ссылок, содержащихся в отчете. По умолчанию свойство имеет значение **Blank** (новое окно).
OpenExportedReportWindow	Задает целевое окно для открытия файла экспорта из вьювера. По умолчанию свойство имеет значение **Blank** (новое окно).
DesignWindow	Задает целевое окно для открытия дизайнера отчетов. По умолчанию свойство имеет значение **Self** (текущее окно).
ShowTooltips	Включает отображение подсказок для элементов управления вьювера при наведении мыши. По умолчанию свойство имеет значение **true**.

© 2003-2026 Stimulsoft

ShowTooltipsHelp	Включает отображение ссылки на онлайн документацию в подсказках для элементов управления вьювера. По умолчанию свойство имеет значение **true**.
ShowDialogsHelp	Предоставляет возможность отображать или не отображать кнопку вызова справки в различных меню. По умолчанию, установлено значение **true**.
PageAlignment	Задает позицию страницы отчета в окне вьювера. Может принимать одно из указанных ниже значений перечисления **StiContentAlignment**: › **Left** – страница будет пристыкована к левому краю; › **Center** – страница будет центрирована (значение по умолчанию); › **Right** – страница будет пристыкована к правому краю.
ShowPageShadow	Включает отображение тени для страниц отчета. По умолчанию свойство имеет значение **true**.
BookmarksPrint	Включает печать закладок отчета на принтере (помимо самого отчета). По умолчанию свойство имеет значение **false**.
BookmarksTreeWidth	Устанавливает ширину панели закладок в пикселях. По умолчанию ширина имеет значение 180 пикселей.
ParametersPanelPosition	Задает позицию панели параметров отчета. Может принимать одно из указанных ниже значений перечисления **StiParametersPanelPosition**: › **Top** – панель будет пристыкована к верхнему краю (значение по умолчанию); › **Left** – панель будет пристыкована к левому краю.

© 2003-2026 Stimulsoft

ParametersPanelMaxHeight	Устанавливает максимальную высоту панели параметров в пикселях. По умолчанию максимальная высота имеет значение 300 пикселей.
ParametersPanelColumnsCount	Устанавливает количество колонок для отображения параметров отчета. По умолчанию разбиение производится на 2 колонки.
ParametersPanelSortDataItems	Устанавливает или выключает режим сортировки значений переменной. По умолчанию, опция установлена в значение **true**, т.е. значения переменной сортируются.
ParametersPanelDateFormat	Устанавливает формат даты и времени для переменных соответствующего типа на панели параметров. По умолчанию используется формат даты и времени, установленный браузером.
InterfaceType	Устанавливает тип используемого интерфейса вьювера. Может принимать одно из указанных ниже значений перечисления **StiInterfaceType**: › **Auto** – тип интерфейса вьювера будет выбран автоматически в зависимости от используемого устройства (значение по умолчанию); › **Mouse** – принудительное использование стандартного интерфейса для управления вьювером при помощи мыши; › **Touch** – принудительное использование Touch интерфейса для управления вьювером при помощи сенсорного экрана монитора, в этом режиме элементы интерфейса вьювера имеют увеличенные размеры для удобства управления; › **Mobile** – принудительное использование Mobile интерфейса для управления вьювером при помощи экрана смартфона, в этом

© 2003-2026 Stimulsoft

	режиме интерфейс вьювера имеет упрощенный вид и адаптирован для управления при помощи мобильного устройства.
AllowMobileMode	Включает или выключает возможность отображения отчета или дашборда в мобильном режиме. Если опция установлена в значение **false**, то мобильный режим просмотра не будет использоваться ни при каких обстоятельства. Если опция установлена в значение **true**, то мобильный режим просмотра будет использоваться запуске вьювера на мобильных устройствах. По умолчанию, опция установлена в значение **true**.
ChartRenderType	Устанавливает режим отображения диаграмм на странице отчета. Может принимать одно из указанных ниже значений перечисления **StiChartRenderType**: › **Image** – диаграммы отображаются в виде статичных изображений; › **Vector** – диаграммы отображаются в векторном режиме в виде SVG объекта; › **AnimatedVector** - диаграммы отображаются в векторном режиме в виде SVG объекта, элементы диаграммы появляются с анимацией (значение по умолчанию).
ReportDisplayMode	Устанавливает режим экспорта для отображения страниц отчета. Может принимать одно из указанных ниже значений перечисления **StiReportDisplayMode**: › **FromReport** - режим экспорта элементов отчета определяется из настроек шаблона - Div или Table; › **Table** – элементы отчета экспортируются с

© 2003-2026 Stimulsoft

	использованием HTML таблиц (значение по умолчанию); › **Div** – элементы отчета экспортируются с использованием DIV разметки; › **Span** - элементы отчета экспортируются с использованием SPAN разметки.
DatePickerFirstDayOfWeek	Устанавливает первый день недели для элемента выбора даты. Может принимать одно из указанных ниже значений перечисления **StiFirstDayOfWeek**: › **Auto** – автоматическое определение первого дня недели из настроек браузера (значение по умолчанию); › **Monday** – первым днем недели является понедельник; › **Sunday** – первым днем недели является воскресенье.
DatePickerIncludeCurrentDayForRanges	Предоставляет возможность включать или не включать текущий день в диапазон значений элемента **Выбор даты (Date Picker)**. По умолчанию, опция установлена в значение **false**, т.е. текущий день не включается в диапазон значений элемента.
AllowTouchZoom	Включает возможность изменения масштаба страницы отчета специальным двух пальцевым жестом (Pinch to Zoom) на сенсорном экране устройства. По умолчанию свойство имеет значение **true**.
ShowReportIsNotSpecifiedMessage	Предоставляет возможность отображать или не отображать уведомление, если отчет не задан. По умолчанию свойство имеет значение **true**.
PrintToPdfMode	Устанавливает режим печати в PDF документ. Может принимать значения: › **StiPrintToPdfMode.Hidden** - скрытый режим печати (по умолчанию);

© 2003-2026 Stimulsoft

	› **StiPrintToPdfMode.Popup** - документ PDF будет отображет перед печатью во всплывающем окне.
ImagesQuality	Получает или задает качество изображения, которое будет использоваться на странице вьювера. Может принимать значения: › **StiImagesQuality.Low** - низкое качество, используется для ускорения загрузки отчета и экономии памяти › **StiImagesQuality.Normal** - обычное качество, подходит для большинства ситуаций, установлено по умолчанию › **StiImagesQuality.High** - высокое качество, используется для мониторов высокой четкости, но может замедлять загрузку страниц
CombineReportPages	Предоставляет возможность объединить обработанные страницы шаблона отчета в один шаблон или представить каждую страницу шаблона отдельной вкладкой во вьювере. По умолчанию, опция установлена в значение **false**, т.е. каждая страница шаблона отчета будет представлена отдельной вкладкой во вьювере.
AllowPropagationEvents	Предоставляет возможность разрешить распространение событий нажатия клавиш, когда вьювер отчетов не в фокусе. По умолчанию свойство имеет значение **true**.
DashboardFilterElementItemHeight	Предоставляет возможность изменить высоту флажка в элементе дашборда **Список (List Box)**.

Панель инструментов (Toolbar)

Наименование	Описание
Visible	Включает отображение панели инструментов

© 2003-2026 Stimulsoft

	вьювера. По умолчанию свойство имеет значение **true**.
DisplayMode	Задает режим отображения панели инструментов вьювера. Может принимать одно из указанных ниже значений перечисления **StiToolbarDisplayMode**: › **Simple** – простой режим отображения, все элементы управления расположены на одной панели управления (значение по умолчанию); › **Separated** – раздельный режим отображения, панель управления разделена на верхнюю и нижнюю.
BackgroundColor	Задает цвет панели инструментов вьювера. По умолчанию используется цветовое оформление выбранной темы.
BorderColor	Задает цвет границ панели инструментов вьювера. По умолчанию используется цветовое оформление выбранной темы.
FontColor	Задает цвет текста для панели инструментов и меню вьювера. По умолчанию используется цветовое оформление выбранной темы.
FontFamily	Задает шрифт для панели инструментов и меню вьювера. По умолчанию используется шрифт выбранной темы.
Alignment	Устанавливает режим выравнивания элементов управления на панели инструментов вьювера. Может принимать одно из указанных ниже значений перечисления **StiContentAlignment**: › **Left** – элементы будут пристыкованы к левому краю; › **Center** – элементы будут располагаться по центру панели;

© 2003-2026 Stimulsoft

	› **Right** – элементы будут пристыкованы к правому краю; › **Default** – выравнивание зависит от свойства RightToLeft (значение по умолчанию)
ShowButtonCaptions	Включает отображение текста кнопок панели инструментов вьювера. По умолчанию свойство имеет значение **true**.
ShowPrintButton	Включает отображение кнопки **Печать (Print)** на панели инструментов вьювера. По умолчанию свойство имеет значение **true**.
ShowOpenButton	Включает отображение кнопки **Открыть (Open)** на панели инструментов вьювера при просмотре отчетов или панели индикаторов (дашбордов). По умолчанию свойство имеет значение **true**.
ShowSaveButton	Включает отображение кнопки **Сохранить (Save)** на панели инструментов вьювера при просмотре отчетов или панели индикаторов (дашбордов). По умолчанию свойство имеет значение **true**.
ShowSendEmailButton	Включает отображение кнопки **Отправиль Email (Send Email)** на панели инструментов вьювера. По умолчанию свойство имеет значение **false**. Также, необходимо подписаться на событие OnEmailReport.
ShowFindButton	Включает отображение кнопки **Поиск (Find)** на панели инструментов вьювера. По умолчанию свойство имеет значение **true**.
ShowBookmarksButton	Включает отображение кнопки **Закладки (Bookmarks)** на панели инструментов вьювера. По умолчанию свойство имеет значение **true**. Если указанная кнопка скрыта, то панель закладок не будет отображена даже при наличии закладок в отчете.
ShowParametersButton	Включает отображение кнопки **Параметры**

© 2003-2026 Stimulsoft

	(Parameters) на панели инструментов вьювера. По умолчанию свойство имеет значение **true**. Если указанная кнопка скрыта, то панель параметров не будет отображена даже при наличии параметров в отчете.
ShowResourcesButton	Включает отображение кнопки **Ресурсы (Resources)** на панели инструментов вьювера. По умолчанию свойство имеет значение **true**. Если указанная кнопка скрыта, то панель ресурсов не будет отображена.
ShowEditorButton	Включает отображение кнопки **Редактор (Editor)** на панели инструментов вьювера при просмотре отчетов. По умолчанию свойство имеет значение **true**.
ShowFullScreenButton	Включает отображение кнопки **Полноэкранный просмотр (Full Screen)** на панели инструментов вьювера при просмотре отчетов или панели индикаторов (дашбордов). По умолчанию свойство имеет значение **true**.
ShowFirstPageButton	Включает отображение кнопки **Первая страница (First Page)** на панели инструментов вьювера. По умолчанию свойство имеет значение **true**.
ShowPreviousPageButton	Включает отображение кнопки **Предыдущая страница (Previous Page)** на панели инструментов вьювера. По умолчанию свойство имеет значение **true**.
ShowCurrentPageControl	Включает отображение индикатора текущей страницы отчета. По умолчанию свойство имеет значение **true**.
ShowNextPageButton	Включает отображение кнопки **Следующая страница (Next Page)** на панели инструментов вьювера. По умолчанию свойство имеет значение **true**.
ShowLastPageButton	Включает отображение кнопки **Последняя**

© 2003-2026 Stimulsoft

	страница (Last Page) на панели инструментов вьювер. По умолчанию свойство имеет значение **true**.
ShowZoomButton	Включает отображение кнопки выбора масштаба отчета. По умолчанию свойство имеет значение **true**.
ShowViewModeButton	Включает отображение кнопки выбора режима отображения страниц отчета. По умолчанию свойство имеет значение **true**.
ShowDesignButton	Включает отображение кнопки **Дизайн (Design)** на панели инструментов вьювера при просмотре отчетов или панели индикаторов (дашбордов). По умолчанию свойство имеет значение **false**.
ShowAboutButton	Включает отображение кнопки **О программе (About)** на панели инструментов вьювера. По умолчанию свойство имеет значение **true**.
ShowRefreshButton	Включает или выключает отображение кнопки **Обновить (Refresh)** на панели инструментов вьювера при просмотре панелей индикаторов (дашбордов). По умолчанию свойство имеет значение **true**.
ShowPinToolbarButton	Включает отображение кнопки **Прикрепить панель (Pin Toolbar)** на панели инструментов вьювера. Кнопка доступна только в мобильном режиме вьювера. По умолчанию свойство имеет значение **true**.
PrintDestination	Устанавливает режим печати отчета. Может принимать одно из указанных ниже значений перечисления **StiPrintDestination**: › **Default** – будет отображено меню с предоставлением выбора режима печати (значение по умолчанию); › **Pdf** – печать будет осуществлена в формате PDF; › **Direct** – печать будет осуществлена в

© 2003-2026 Stimulsoft

	формате HTML непосредственно на принтер, будет отображен системный диалог печати; › **PopupWindow** – печать будет осуществлена в формате HTML через всплывающее окно предварительного просмотра отчета.
ViewMode	Устанавливает режим отображения страниц отчета. Может принимать одно из указанных ниже значений перечисления **StiWebViewMode**: › **SinglePage** – отображается одна страница отчета, выбранная на панели инструментов вьювера (значение по умолчанию); › **Continuous** – отображаются все страницы отчета в виде ленты; › **MultiplePages** – отображаются все страницы отчета в виде таблицы.
Zoom	Задает масштаб отображения страниц отчета. По умолчанию установлен масштаб 100 процентов. Допускаются значения от 10 до 500 процентов. Также допускается устанавливать одно из следующих значений: › **StiZoomMode.PageWidth** – при запуске вьювера будет установлен масштаб, необходимый для отображения отчета по ширине страницы; › **StiZoomMode.PageHeight** – при запуске вьювера будет установлен масштаб, необходимый для отображения отчета по высоте страницы.
MenuAnimation	Включает анимацию при появлении/скрытии меню вьювера. По умолчанию свойство имеет значение **true**.
ShowMenuMode	Устанавливает режим отображения меню вьювера. Может принимать одно из указанных ниже значений перечисления

© 2003-2026 Stimulsoft

	StiShowMenuMode: › **Click** – отображение по клику мыши (значение по умолчанию); › **Hover** – отображение по наведении курсора мыши.
AutoHide	Включает автоматическое сворачивание панели инструментов вьювера. Свойство работает только в мобильном режиме вьювера. По умолчанию свойство имеет значение **false**.

Экспорт отчета (Export)

Наименование	Описание
DefaultSettings	Данная группа свойств предоставляет возможность задать настройки экспортов по умолчанию для каждого типа экспорта. Эти настройки будут применены к диалоговым окнам экспортов при запуске вьювера, либо к отчету в случае отключения отображения диалоговых окон экспортов.
StoreExportSettings	Включает режим сохранения выбранных настроек в диалоговых окнах экспорта. Настройки будут храниться в cookies браузера. По умолчанию свойство имеет значение **true**.
ShowExportDialog	Включает отображение диалогового окна параметров экспорта. Если свойство имеет значение **false**, то экспортирование будет производиться с заданными по умолчанию настройками. По умолчанию свойство имеет значение **true**.
ShowExportToDocument	Включает отображение пункта меню экспорта **Файл документа (Document File)**. По умолчанию свойство имеет значение **true**.

© 2003-2026 Stimulsoft

ShowExportToPdf	Включает отображение пункта меню экспорта **Adobe PDF файл (Adobe PDF File)** при просмотре отчетов, и пункт **Adobe PDF** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToXps	Включает отображение пункта меню экспорта **Microsoft XPS файл (Microsoft XPS File)**. По умолчанию свойство имеет значение **false**.
ShowExportToPowerPoint	Включает отображение пункта меню экспорта **Microsoft PowerPoint 2007/2010 файл (Microsoft PowerPoint 2007/2010 File)**. По умолчанию свойство имеет значение **true**.
ShowExportToHtml	Включает отображение пункта меню экспорта **HTML файл (HTML File)**. По умолчанию свойство имеет значение **true**.
ShowExportToHtml5	Включает отображение пункта меню экспорта **HTML5 файл (HTML5 File)**. По умолчанию свойство имеет значение **true**.
ShowExportToMht	Включает отображение пункта меню экспорта **MHT Web архив (MHT Web Archive)**. По умолчанию свойство имеет значение **true**.
ShowExportToText	Включает отображение пункта меню экспорта **Text файл (Text File)**. По умолчанию свойство имеет значение **true**.
ShowExportToRtf	Включает отображение пункта меню экспорта **Rich Text файл (Rich Text File)**. По умолчанию свойство имеет значение **true**.
ShowExportToWord	Включает отображение пункта меню экспорта **Microsoft Word 2007/2010 файл (Microsoft Word 2007/2010 File)**. По умолчанию свойство имеет значение **true**.
ShowExportToOpenDocumentWriter	Включает отображение пункта меню экспорта **OpenDocument Writer файл (OpenDocument Writer File)**. По умолчанию

© 2003-2026 Stimulsoft

	свойство имеет значение **true**.
ShowExportToExcelBiff	Включает отображение пункта меню экспорта **Microsoft Excel файл (Microsoft Excel File)**. По умолчанию свойство имеет значение **true**.
ShowExportToExcelXml	Включает отображение пункта меню экспорта **Microsoft Excel Xml файл (Microsoft Excel Xml File)**. По умолчанию свойство имеет значение **true**.
ShowExportToExcel	Включает отображение пункта меню экспорта **Microsoft Excel 2007/2010 файл (Microsoft Excel 2007/2010 File)** при просмотре отчетов, и пункт **Microsoft Excel** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToOpenDocumentCalc	Включает отображение пункта меню экспорта **OpenDocument Calc файл (OpenDocument Calc File)**. По умолчанию свойство имеет значение **true**.
ShowExportToCsv	Включает отображение пункта меню экспорта **CSV файл (CSV File)**. По умолчанию свойство имеет значение **true**.
ShowExportToDbf	Включает отображение пункта меню экспорта **DBF файл (DBF File)**. По умолчанию свойство имеет значение **true**.
ShowExportToXml	Включает отображение пункта меню экспорта **XML файл (XML File)**. По умолчанию свойство имеет значение **true**.
ShowExportToDif	Включает отображение пункта меню экспорта **Data Interchange формат (DIF) файл (Data Interchange Format (DIF) File)**. По умолчанию свойство имеет значение **true**.
ShowExportToSylk	Включает отображение пункта меню экспорта **Symbolic Link (SYLK) файл (Symbolic Link (SYLK) File)**. По умолчанию свойство имеет значение **true**.

© 2003-2026 Stimulsoft

ShowExportToJson	Включает отображение пункта меню экспорта **JSON файл (JSON File)**. По умолчанию свойство имеет значение **true**.
ShowExportToImageBmp	Включает отображение пункта меню экспорта **BMP рисунок (BMP Image)** при просмотре отчетов, и пункт **BMP** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToImageGif	Включает отображение пункта меню экспорта **GIF рисунок (GIF Image)** при просмотре отчетов, и пункт **GIF** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToImageJpeg	Включает отображение пункта меню экспорта **JPEG рисунок (JPEG Image)** при просмотре отчетов, и пункт **JPEG** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToImagePcx	Включает отображение пункта меню экспорта **PCX рисунок (PCX Image)** при просмотре отчетов, и пункт **PCX** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToImagePng	Включает отображение пункта меню экспорта **PNG рисунок (PNG Image)** при просмотре отчетов, и пункт **PNG** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToImageTiff	Включает отображение пункта меню экспорта **TIFF рисунок (TIFF Image)** при просмотре отчетов, и пункт **TIFF** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToImageSvg	Включает отображение пункта меню экспорта **Scalable Vector Graphics (SVG) файл (Scalable Vector Graphics (SVG) File)** при просмотре отчетов, и пункт **SVG** при

© 2003-2026 Stimulsoft

	просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowExportToImageSvgz	Включает отображение пункта меню экспорта **Compressed SVG (SVGZ) файл (Compressed SVG (SVGZ) File)** при просмотре отчетов, и пункт **SVGZ** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
ShowOpenAfterExport	Предоставляет возможность отображать или не отображать параметр **Открыть после экспорта (Open After Export)** в меню настроек экспорта. По умолчанию свойство имеет значение **true**.

Отправка отчета по Email (Email)

Наименование	Описание
ShowEmailDialog	Включает отображение диалогового окна параметров отправки отчета по Email. Если диалоговое окно отключено, то отправка по Email будет производиться с настройками, заданными на стороне сервера при использовании события **OnEmailReport**. По умолчанию свойство имеет значение **true**.
ShowExportDialog	Включает отображение диалогового окна параметров экспорта при отправке Email. Если свойство имеет значение **false**, то экспортирование будет производиться с заданными по умолчанию настройками. По умолчанию свойство имеет значение **true**.
DefaultEmailAddress	Устанавливает Email получателя по умолчанию, т.е. адрес на который будет отправлено письмо с вложенным отчетом.
DefaultEmailSubject	Устанавливает тему (заголовок) электронного письма по умолчанию.

© 2003-2026 Stimulsoft

DefaultEmailMessage	Устанавливает сообщение (текст) электронного письма по умолчанию.
DefaultEmailReplyTo	Устанавливает сообщение (текст) по умолчанию для ответа на сообщение, созданное во вьювере.

7.2 Дизайнер

Примеры

Ознакомьтесь с примерами по работе с компонентом **Blazor Designer** на GitHub, доступны технологии Blazor Server и Blazor WebAssembly. Все примеры представляют собой отдельные проекты, сгруппированные в один solution для Visual Studio.

Компонент **Blazor Designer** (**StiBlazorDesigner**) предназначен для редактирования отчетов в окне браузера. При этом не требуется устанавливать у клиента компоненты или какие-либо специальные плагины. Все, что необходимо – это современный Web-браузер. При помощи **Blazor Designer** можно создавать, редактировать, сохранять, просматривать отчеты, а также печатать отчеты на любом компьютере с любой установленной операционной системой.

Компонент **Blazor Designer** разработан как универсальный компонент, и может работать как с использованием технологии **Blazor Server**, так и с использованием технологию **WebAssembly**. При выполнении всех действий над отчётами, компонент запрашивает только необходимые данные, что позволяет избавиться от перезагрузки всей страницы, а также экономить Web-трафик и повысить скорость работы.

Для использования **Blazor Designer** в Web-проекте, необходимо установить NuGet пакет Stimulsoft.Reports.Blazor, , либо Stimulsoft.Dashboards.Blazor:

- Выберите пункт "Manage NuGet Packages..." в контекстном меню проекта;
- На вкладке Browse, в строке поиска, укажите Stimulsoft.Reports.Blazor;
- Выделите элемент, определите версию пакета и нажмите кнопку **Установить (Install)**. При обновлении пакета, следует нажать кнопку **Обновить (Update)**.

© 2003-2026 Stimulsoft

Если по какой-либо причине это невозможно, необходимо добавить в проект указанные ниже сборки:
Stimulsoft.Base.dll
Stimulsoft.Blockly.dll
Stimulsoft.Data.dll
Stimulsoft.Drawing.dll
Stimulsoft.Map.dll
Stimulsoft.Report.dll
Stimulsoft.Report.Blazor.dll
Stimulsoft.Report.Check.dll
Stimulsoft.Report.Helper.dll
Stimulsoft.Report.Web.dll
Stimulsoft.Report.WebDesign.dll
Stimulsoft.System.dll
Stimulsoft.System.Web.dll

Для работы с дашбордами, необходимо дополнительно добавить в проект указанные ниже сборки:
Stimulsoft.Dashboard.dll
Stimulsoft.Dashboard.Drawing.dll
Stimulsoft.Dashboard.Export.dll

- Активация
- Дополнительные возможности предварительного просмотра
- Редактирование отчета
- Просмотр
- Создание нового отчета
- Локализация дизайнера
- Сохранение отчета
- Использование тем
- Настройки
- События

7.2.1 Активация

После приобретения продукта Stimulsoft вам необходимо активировать лицензию для используемых компонентов. Это можно выполнить, указав лицензионный ключ или загрузив файл с лицензионным ключом. Ниже представлен пример активации компонента **Blazor Designer**.

© 2003-2026 Stimulsoft

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorDesigner />

@code
{
   protected override void OnInitialized()
   {
      //Activation with using license code
      Stimulsoft.Base.StiLicense.Key = "Your activation code...";

      //Activation with using license file
      Stimulsoft.Base.StiLicense.LoadFromFile("Content/license.key");

      base.OnInitialized();
   }
}
```

Получить лицензионный ключ или скачать файл с лицензионным ключом, вы можете в личном кабинете пользователя. Для авторизации в кабинете используйте имя пользователя и пароль, указанные при покупке подписки продукта.

7.2.2 Редактирование отчета

Для редактирования отчета необходимо добавить компонент **StiBlazorDesigner** на **Razor** страницу, добавить объект **StiReport** и присвоить его дизайнеру, используя свойство **Report**. После загрузки отчета из файла, он будет передан дизайнеру автоматически.

© 2003-2026 Stimulsoft

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorDesigner Report="@Report" />

@code
{
   //Report object to use in designer
   private StiReport Report;

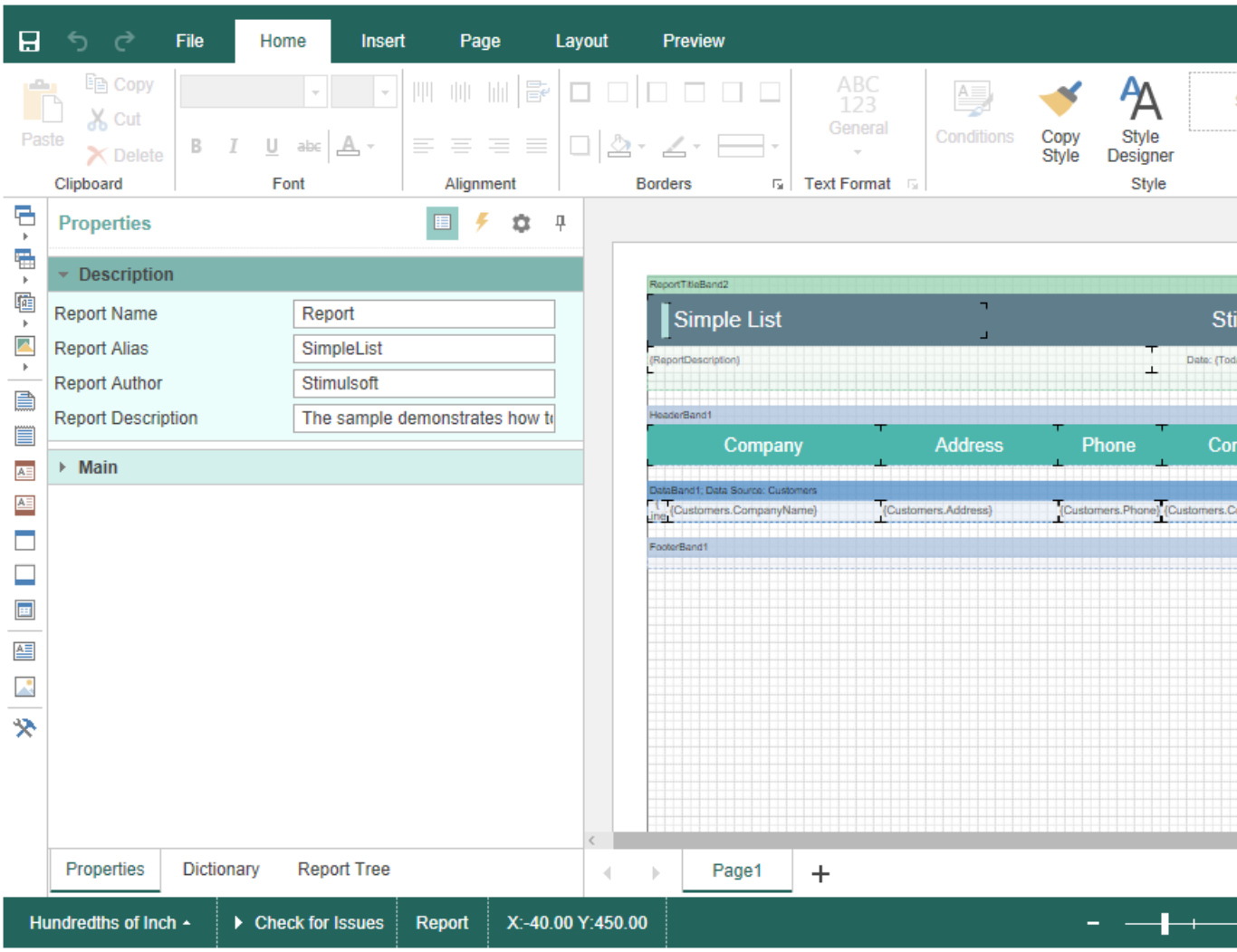
   protected override void OnInitialized()
   {
      base.OnInitialized();

      //Create empty report object
      var report = new StiReport();

      //Load report template
      report.Load("Reports/TwoSimpleLists.mrt");

      //Assing report object to viewer
      Report = report;
   }
}
```

© 2003-2026 Stimulsoft



7.2.3 Создание нового отчета

Для запуска дизайнера с новым отчетом достаточно создать новый объект в событии **OnInitialized**. При необходимости, можно предварительно загрузить данные для отчета, либо выполнить какие-либо другие необходимые действия.

© 2003-2026 Stimulsoft

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorDesigner Report="@Report" />

@code
{
   //Report object to use in designer
   private StiReport Report;

   protected override void OnInitialized()
   {
      base.OnInitialized();

      //Create empty report object
      var report = new StiReport();

      //Assing report object to designer
      Report = report;
   }
}
```

Также новый отчет можно создать, используя главное меню дизайнера. Для того, чтобы предварительно загрузить данные для нового отчета, либо выполнить какие-либо другие необходимые действия, можно использовать событие **OnCreateReport**. Это событие будет обработано при создании нового пустого отчета из главного меню, либо при создании отчета с использованием мастера.

© 2003-2026 Stimulsoft

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Web

<StiBlazorDesigner Report="@Report" OnCreateReport="@OnCreateReport"/>

@code
{
   //Report object to use in designer
   private StiReport Report;

   protected override void OnInitialized()
   {
      base.OnInitialized();

      //Create empty report object
      var report = new StiReport();

      //Load report template
      report.Load("Reports/Simple List.mrt");

      //Assing report object to designer
      Report = report;
   }

   private void OnCreateReport(StiReportDataEventArgs args)
   {
      //Delete connections in the report template
      args.Report.Dictionary.Databases.Clear();

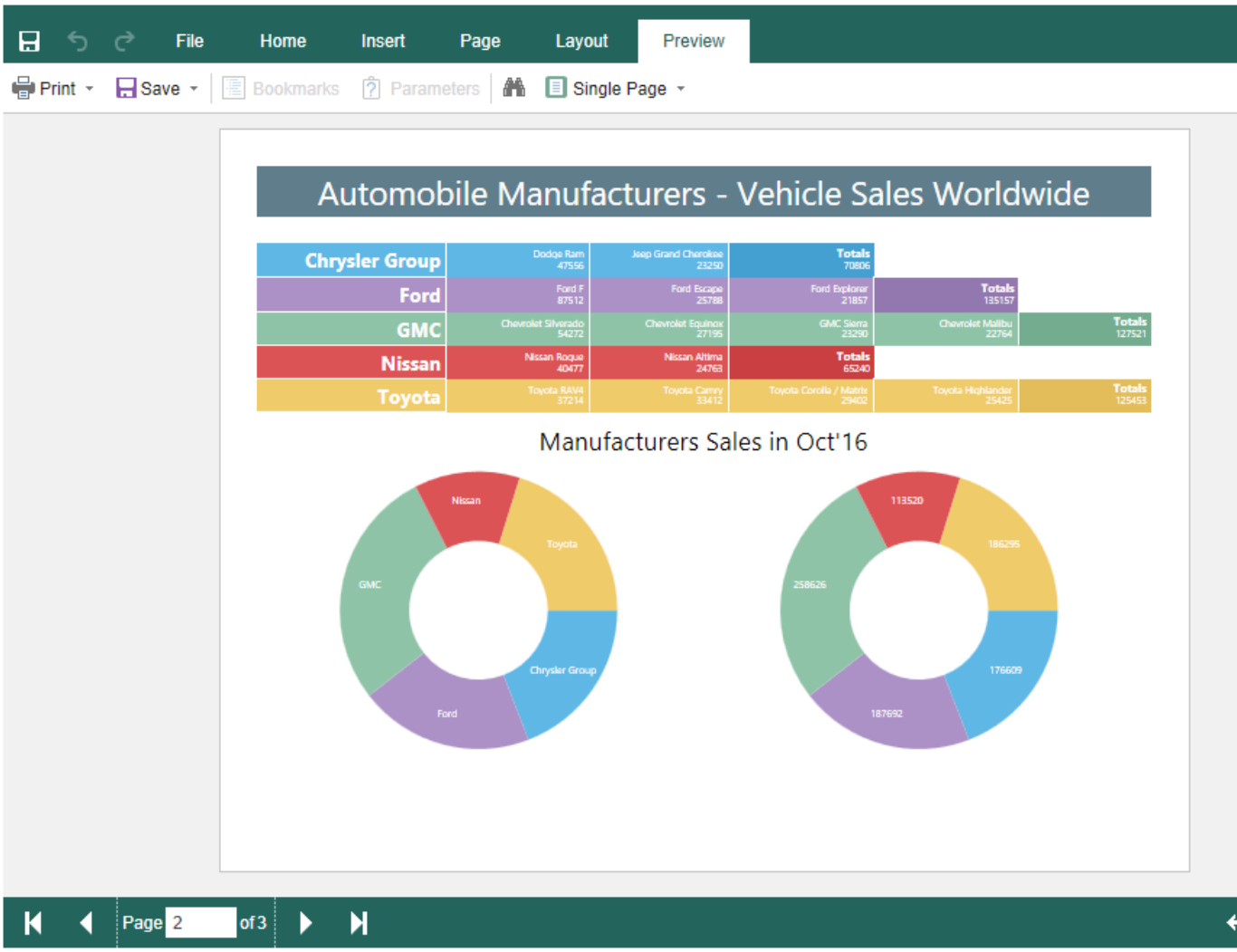
      //Load new data from XML file
      var data = new System.Data.DataSet();
      data.ReadXml("Data/Demo1.xml");

      args.Report.RegData(data);
      args.Report.Dictionary.Synchronize();
   }
}
```

7.2.4 Просмотр отчета

У компонента **Blazor Designer** предусмотрен режим просмотра редактируемого отчета. Для просмотра отчета, достаточно перейти на соответствующую вкладку в окне дизайнера. Шаблон будет построен и отображен во встроенном вьювере.

© 2003-2026 Stimulsoft



Перед предварительным просмотром отчета предусмотрена возможность выполнить какие-либо необходимые действия, например, подключить данные для отчета. Для этого достаточно определить специальное событие **OnPreviewReport**, которое будет вызвано перед просмотром отчета.

© 2003-2026 Stimulsoft

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorDesigner Report="@Report" OnPreviewReport="@OnPreviewReport"/>

@code
{
   //Report object to use in designer
   private StiReport Report;

   protected override void OnInitialized()
   {
      base.OnInitialized();

      //Create empty report object
      var report = new StiReport();

      //Load report template
      report.Load("Reports/Simple List.mrt");

      //Assing report object to designer
      Report = report;
   }

   private void OnPreviewReport(StiReportDataEventArgs args)
   {
      //Load new data from XML file
      var data = new System.Data.DataSet();
      data.ReadXml("Data/Demo1.xml");

      args.Report.RegData(data);
   }
}
```

7.2.5 Дополнительные возможности просмотра

Окно просмотра отчета компонента **Blazor Designer** представляет собой полноценный интерактивный вьювер, который умеет осуществлять печать отчета, экспортирование отчета, поддерживает работу с их параметрами. Также поддерживаются интерактивные действия, такие как динамическая сортировка, детализация, сворачивание. Для использования указанных возможностей не требуется каких-либо дополнительных настроек дизайнера отчетов.

В любом из вышеперечисленных действий допускается производить манипуляции с шаблоном отчета, например, изменять его свойства и параметры, подключать новые данные для построения. При экспортировании отчета можно получить формат экспорта, прочитать или изменить его настройки.

© 2003-2026 Stimulsoft

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorDesigner Report="@Report" OnExportReport="@OnExportReport"/>

@code
{
   //Report object to use in designer
   private StiReport Report;

   protected override void OnInitialized()
   {
      base.OnInitialized();

      //Create empty report object
      var report = new StiReport();

      //Load report template
      report.Load("Reports/Simple List.mrt");

      //Assing report object to designer
      Report = report;
   }

   private void OnExportReport(StiExportReportEventArgs args)
   {
      //Current export format
      var exportFormat = args.Format;

      //Current export settings
      var exportSettings = args.Settings;

      //Load new data from XML file
      var data = new System.Data.DataSet();
      data.ReadXml("Data/Demo1.xml");

      args.Report.RegData(data);
   }
}
```

Информация

Если какие-либо из указанных дополнительных возможностей предварительного просмотра отчета не требуются (например, экспортирование или печать отчета), можно отключить их, используя соответствующие опции компонента **Blazor Designer**.

© 2003-2026 Stimulsoft

7.2.6 Сохранение отчета

В компоненте **Blazor Designer** предусмотрено два варианта сохранения отчета, которые доступны в главном меню и на главной панели дизайнера: **Сохранить (Save)** и **Сохранить как (Save As)**. В свою очередь, каждый из этих вариантов сохранения имеет свои режимы и настройки.

Сохранение отчета на стороне сервера

Для сохранения редактируемого отчета на стороне сервера необходимо определить событие **OnSaveReport**, которое будет вызвано при выборе пункта **Сохранить (Save)** в главном меню, либо нажатии на кнопку **Сохранить (Save)** на главной панели дизайнера.

Index.razor

```
@using Stimulsoft.Base
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorDesigner Report="@Report" OnSaveReport="@OnSaveReport"/>

@code
{
   //Report object to use in designer
   private StiReport Report;

   protected override void OnInitialized()
   {
      base.OnInitialized();

      //Create empty report object
      var report = new StiReport();

      //Load report template
      report.Load("Reports/TwoSimpleLists.mrt");

      //Assing report object to designer
      Report = report;
   }

   private void OnSaveReport(StiSaveReportEventArgs args)
   {
      args.Report.Save("Reports/TwoSimpleLists.mrt");
   }
}
```

Событие **OnSaveReport** будет вызвано при нажатии на кнопку **Сохранить (Save)**. В аргументах события будет передан редактируемый отчёт. Если

© 2003-2026 Stimulsoft

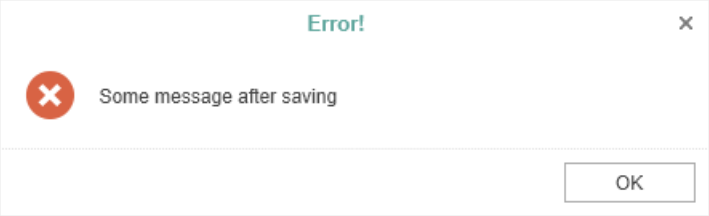
необходимо, после сохранения отчета предусмотрена возможность отобразить диалоговое окно с ошибкой или текстовым сообщением. Для этого предназначены свойства **ErrorCode** и **ErrorString** в аргументах события.

Index.razor

```
...
private void OnSaveReport(StiSaveReportEventArgs args)
{
   args.Report.Save("Reports/TwoSimpleLists.mrt");

   args.ErrorString = "Some message after saving";
}
...
```

В этом случае, будет отображено диалоговое с указанным текстом. Текст может содержать как сообщение об ошибке сохранения либо предупреждение, так и любое другое сообщение.



Если необходимо, вы можете получить доступ к оригинальному имени отчета или имени отчета из диалога сохранения.

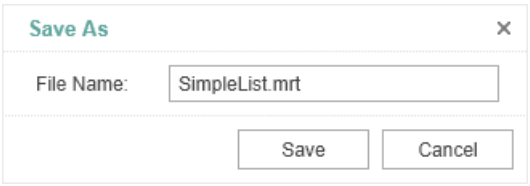
Index.razor

```
...
private void OnSaveReport(StiSaveReportEventArgs args)
{
   //Report name from designer save dialog
   var savingReportName = args.FileName;

   //Original report name from properties
   var originalReportName = args.Report.ReportName;
}
...
```

Сохранение на стороне клиента

Для сохранения редактируемого отчета на стороне клиента в виде файла не требуется дополнительных настроек дизайнера. Достаточно выбрать пункт главного меню **Сохранить как (Save As)**, при нажатии на который отобразится диалог сохранения файла. В данном диалоге можно изменить имя файла отчета, после чего файл будет сохранен на локальном диске компьютера.



Компонент **Blazor Designer** предоставляет возможность изменить поведение указанного варианта сохранения, для этого в настройках дизайнера предусмотрено специальное событие **OnSaveReportAs**. При определении этого действия сохранение отчета будет происходить на стороне сервера, работа данного действия будет аналогична событию **OnSaveReport**.

Index.razor

```
...
private void OnSaveReportAs(StiSaveReportAsEventArgs args)
{
   args.Report.Save("Reports/TwoSimpleLists.mrt");
}
...
```

Настройки сохранения

Сохранение отчета производиться в фоновом режиме, т.е. без перезагрузки страницы в окне браузера. Если необходимо каким-либо образом визуально управлять процессом сохранения отчета, следует изменить значение опции **SaveReportMode** (либо **SaveReportAsMode**) дизайнера на одно из трех указанных значений: **Hidden** (значение по умолчанию), **Visible** или **NewWindow**.

© 2003-2026 Stimulsoft

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorDesigner Report="@Report" Options="@Options" />

@code
{
   //Report object to use in designer
   private StiReport Report;

   private StiBlazorDesignerOptions Options;

   protected override void OnInitialized()
   {
      base.OnInitialized();

      //Init options object
      Options = new StiBlazorDesignerOptions();
      Options.Behavior.SaveReportMode = StiSaveMode.Visible;
      Options.Behavior.SaveReportAsMode = StiSaveMode.Visible;
   }
}
```

Если опция **SaveReportMode** установлена в значение **Visible**, то действие сохранения отчета будет вызвано в текущем окне браузера в обычном (видимом) режиме при помощи POST-запроса. Если опция **SaveReportMode** установлена в значение **NewWindow**, то событие сохранения отчета будет вызвано в новом окне браузера. По умолчанию данная опция установлена в значение **Hidden**, т.е. событие сохранения отчета вызывается в фоне при помощи AJAX-запроса и никак не отображается в окне браузера. Для опции **SaveReportAsMode** применимы те же значения и поведение.

7.2.7 Локализация дизайнера

Компонент **Blazor Designer** поддерживает полную локализацию своего интерфейса. Для того чтобы локализовать интерфейс дизайнера отчетов на необходимый язык, используется специальное свойство **Localization**. В качестве значения данного свойства необходимо указать путь к XML-файлу локализации (относительный либо абсолютный).

© 2003-2026 Stimulsoft

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorDesigner Localization="Localization/en.xml" />
```

7.2.8 Использование тем

Компонент **Blazor Designer** имеет возможность смены тем оформления визуальных элементов управления. Для изменения темы используется свойство **Theme**.

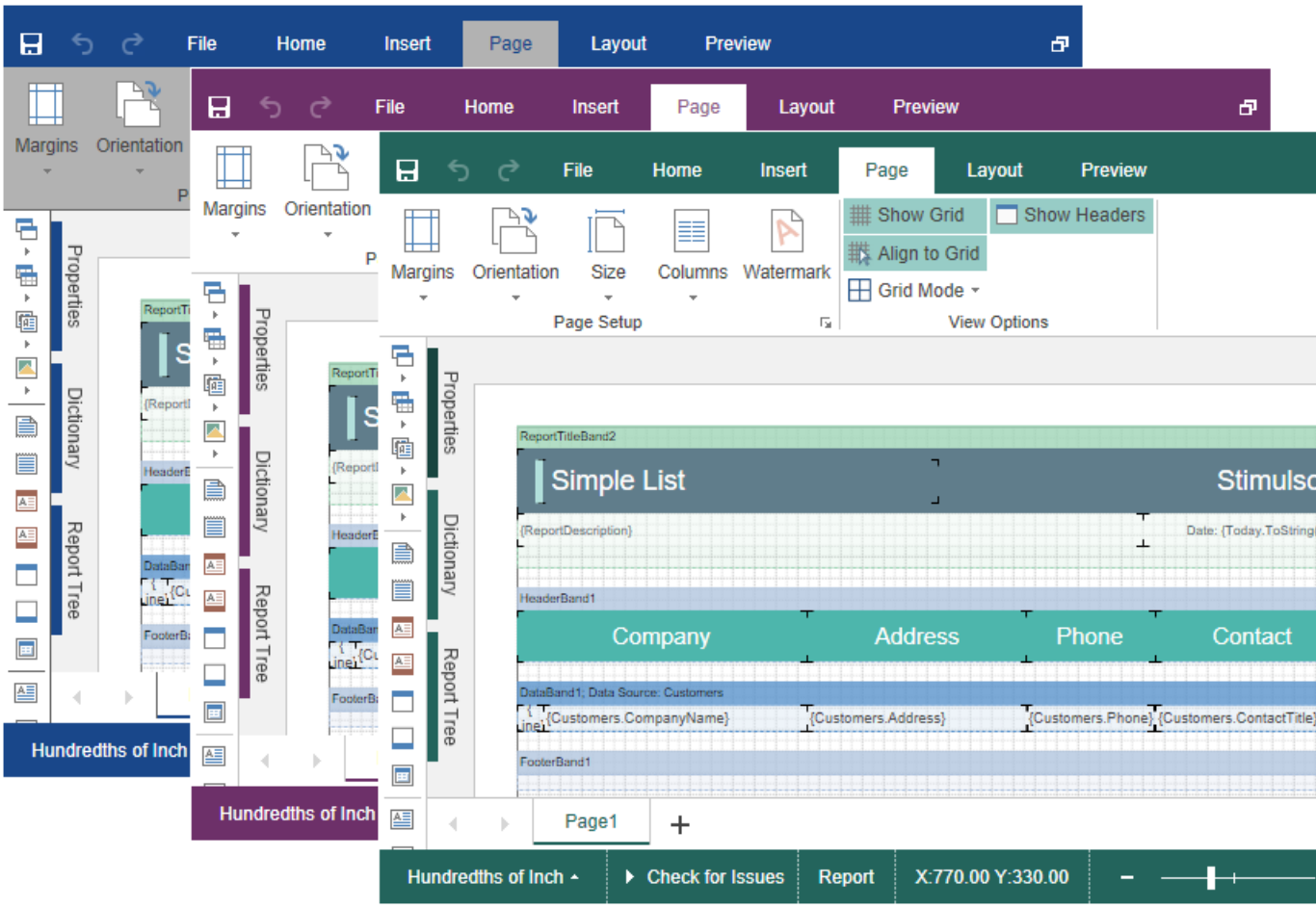
Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorDesigner Theme="StiDesignerTheme.Office2022WhiteCarmine" />
```

На данный момент доступно **2 различные темы** со своими цветовыми акцентами. В результате, доступно **более 50 вариантов** оформления. Это позволяет настроить внешний вид дизайнера практически под любое оформление Web проекта.

© 2003-2026 Stimulsoft



7.2.9 События дизайнера

Компонент **Blazor Designer** поддерживает события, которые предоставляют возможность выполнить необходимые операции перед определенными действиями, такими как создание, редактирование и сохранение шаблона отчета, предварительный просмотр, печать отчета, экспортирование отчета, интерактивность и др. Ниже приведен пример обработки событий дизайнера.

© 2003-2026 Stimulsoft

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorDesigner Report="@Report" OnPreviewReport="@OnPreviewReport"/>

@code
{
   //Report object to use in designer
   private StiReport Report;

   protected override void OnInitialized()
   {
      base.OnInitialized();

      //Create empty report object
      var report = new StiReport();

      //Load report template
      report.Load("Reports/TwoSimpleLists.mrt");

      //Assing report object to designer
      Report = report;
   }

   private void OnPreviewReport(StiReportDataEventArgs args)
   {
      //Load new data from XML file
      var data = new System.Data.DataSet();
      data.ReadXml("Data/Demo1.xml");

      args.Report.RegData(data);
   }
}
```

Список событий

Наименование	Описание
OnCreateReport	Событие возникает при создании нового отчета из меню дизайнера.
OnOpenReport	Событие возникает при открытии отчета из меню дизайнера. В аргументах события будет передан загруженный отчет или дашборд.
OnPreviewReport	Событие возникает при переходе на вкладку просмотра отчета, а также

© 2003-2026 Stimulsoft

	при интерактивных действиях, таких как использование переменных отчета, динамическое сворачивание, детализация и сортировка отчета при просмотре.
OnSaveReport	Событие возникает при нажатии на кнопку Сохранить (Save) на панели либо в главном меню дизайнера.
OnSaveReportAs	Событие возникает при нажатии на кнопку Сохранить как (Save As) либо в главном меню дизайнера. Если событие не определено, отчет будет сохранен на локальный диск компьютера.
OnExportReport	Событие возникает при экспортировании отчета.
OnDesignerEvent	Событие возникает при любой действии в дизайнере отчетов.
OnPrintReport	Событие возникает при печати отчета из окна предварительного просмотра.
OnExit	Событие возникает при нажатии на кнопку **Выход (Exit)** в главном меню дизайнера.
OnDesignerAfterRender	Событие возникает после завершения построения HTML5 кода дизайнера и всех его элементов управления.

7.2.10 Настройки

Настройка **Blazor Designer** выполняется при помощи свойств компонента. Основные настройки устанавливаются у самого Razor компонента, все дополнительные настройки устанавливаются при помощи специального класса опций **StiBlazorDesignerOptions**. Ниже приведен пример установки свойств дизайнера.

© 2003-2026 Stimulsoft

Index.razor

```
@using Stimulsoft.Report
@using Stimulsoft.Report.Blazor
@using Stimulsoft.Report.Web

<StiBlazorDesigner ID="Viewer1" Width="500px" Height="500px"
Localization="Localization/en.xml" Options="@Options"
Theme="StiDesignerTheme.Office2022WhiteCarmine" />

@code
{
   //Options object
   private StiBlazorDesignerOptions Options;

   protected override void OnInitialized()
   {
      base.OnInitialized();

      //Init options object
      Options = new StiBlazorDesignerOptions();
      Options.FileMenu.Visible = false;
      Options.Dictionary.Visible = false;
      Options.Toolbar.ShowToolbar = false;
      Options.Components.ShowText = false;
      Options.Bands.ShowEmptyBand = false;
      Options.DashboardElements.ShowTableElement = false;
      Options.Behavior.SaveReportMode = StiSaveMode.Hidden;
   }
}
```

Основные настройки (без группы)

Наименование	Описание
Theme	Задает тему оформления дизайнера. Список доступных тем оформления находится в перечислении **StiDesignerTheme**. По умолчанию установлено значение **Office2022WhiteBlue**.
ID	Строковый уникальный идентификатор компонента, по умолчанию **StiBlazorDesigner**.
Options	Представляет возможность задать объект класса **StiBlazorDesignerOptions**, который содержит набор опций (настроек) компонента.

© 2003-2026 Stimulsoft

Report	Объект отчёта, который будет отправлен на редактирование в дизайнере.
Localization	Задает путь к XML файлу локализации. Путь может быть абсолютным либо относительным. По умолчанию используется английская локализация, которая встроена в дизайнер и не требует дополнительных XML файлов.
Width	Задает ширину компонента в необходимых единицах измерения – пикселях (по умолчанию), процентах, поинтах. Например, “500”, “500px”, “100%”, “300pt”. По умолчанию компонент разворачивается на всю область окна браузера.
Height	Задает высоту компонента в необходимых единицах измерения – пикселях (по умолчанию), процентах, поинтах. Например, “500”, “500px”, “100%”, “300pt”. По умолчанию компонент разворачивается на всю область окна браузера.

Работа с сервером (Server)

Наименование	Описание
UseRelativeUrls	Устанавливает режим дизайнера, в котором для запросов на сервер используются относительные ссылки. По умолчанию свойство имеет значение **true**.
PortNumber	Получает или задает значение, которое указывает номер порта для использования в URL. Значение 0 определяет автоматическое обнаружение (установлено по умолчанию). Значение -1 удаляет номер порта.
PassFormValues	Включает передачу значений POST-формы на сторону клиента, если эти значения требуется использовать в действиях дизайнера. При

© 2003-2026 Stimulsoft

	включении данной возможности, вспомогательный метод **GetFormValues()** будет возвращать коллекцию параметров формы. По умолчанию свойство имеет значение **false**.
PassQueryParametersForResources	Включает передачу всех параметров URL запроса при формировании ссылок на ресурсы дизайнера. При значении **false** для запроса ресурсов дизайнера используются только необходимые параметры, что способствует более корректной работе кэша браузера. По умолчанию свойство имеет значение **true**.
AllowLoadingCustomFontsToClientSide	Позволяет передавать пользовательские шрифты на клиентскую сторону и преобразовывать их в CSS стиль для корректного отображения текста на стороне HTML с заданным шрифтом. По умолчанию свойство имеет значение **false**.

Внешний вид (Appearance)

Наименование	Описание
IconSet	Предоставляет возможность установить набор иконок: › StiWebUIIconSet.Auto (значение по умолчанию) - устанавливает набор иконок автоматически. Для тем **Office2022** - используется набор иконок в стиле **Monoline**, для тем **Office2013** - используется набор иконок в стиле **Regular**. › StiWebUIIconSet.Monoline - устанавливает набор иконок в стиле **Monoline**; › StiWebUIIconSet.Regular - устанавливает набор иконок в стиле **Regular**.

© 2003-2026 Stimulsoft

CustomCss	Задает путь к CSS файлу стилей дизайнера. Если данное свойство установлено, то стандартные стили выбранной темы загружены не будут. По умолчанию установлено пустое значение.
DefaultUnit	Устанавливает единицы измерения размеров для отчета и всех его компонентов. По умолчанию используются сантиметры.
Zoom	Задает масштаб отображения страниц отчета. По умолчанию установлен масштаб 100 процентов. Допускаются значения от 10 до 200 процентов. Может принимать одно из указанных ниже значений перечисления **StiZoomMode**: › **PageWidth** – при запуске дизайнера будет установлен масштаб, необходимый для отображения отчета по ширине страницы; › **PageHeight** – при запуске дизайнера будет установлен масштаб, необходимый для отображения отчета по высоте страницы.
ShowAnimation	Включает анимацию элементов интерфейса дизайнера. По умолчанию свойство имеет значение **true**.
ShowOpenDialog	Устанавливает режим открытия с диалогом или через событие open. По умолчанию свойство имеет значение **true**, т.е. диалог открытия отображается.
ShowTooltips	Включает отображение подсказок для элементов управления дизайнера при наведении мыши. По умолчанию свойство имеет значение **true**.
ShowTooltipsHelp	Включает отображение ссылки на онлайн документацию в подсказках для элементов управления дизайнера. По умолчанию свойство имеет значение **true**.
ShowDialogsHelp	Включает отображение ссылки на онлайн

© 2003-2026 Stimulsoft

	документацию на заголовках диалоговых форм дизайнера. По умолчанию свойство имеет значение **true**.
InterfaceType	Устанавливает тип используемого интерфейса дизайнера. Может принимать одно из указанных ниже значений перечисления **StiInterfaceType**: › **Auto** – тип интерфейса дизайнера будет выбран автоматически в зависимости от используемого устройства (значение по умолчанию); › **Mouse** – принудительное использование интерфейса для управления дизайнером при помощи мыши; › **Touch** – принудительное использование Touch интерфейса для управления дизайнером при помощи сенсорного экрана (мобильные устройства), также в этом режиме для удобства управления элементы интерфейса увеличены.
DatePickerFirstDayOfWeek	Устанавливает первый день недели для элемента выбора даты. Может принимать одно из указанных ниже значений перечисления **StiFirstDayOfWeek**: › **Auto** - автоматическое определение первого дня недели из настроек браузера (значение по умолчанию); › **Monday** – первым днем недели является понедельник; › **Sunday** – первым днем недели является воскресенье.
DatePickerIncludeCurrentDayForRanges	Предоставляет возможность включать или не включать текущий день в элементе выбора даты. По умолчанию свойство имеет значение **false**.
FormatForDateControls	Предоставляет возможность переопределить

© 2003-2026 Stimulsoft

	формат для элементов управления дат. По умолчанию, значение для текущей опции не задан и формат для дат определяется из локали браузера.
ShowReportTree	Включает отображение дерева компонентов отчета. По умолчанию свойство имеет значение **true**.
ChartRenderType	Устанавливает режим отображения диаграмм в отчете на вкладке **Просмотр (Preview)**. Может принимать одно из указанных ниже значений перечисления **StiChartRenderType**: › **Image** – диаграммы отображаются в виде статичных изображений; › **Vector** – диаграммы отображаются в векторном режиме в виде SVG объекта; › **AnimatedVector** - диаграммы отображаются в векторном режиме в виде SVG объекта, элементы диаграммы появляются с анимацией (значение по умолчанию).
ReportDisplayMode	Устанавливает режим экспорта для отображения страниц отчета на вкладке предварительного просмотра. Может принимать одно из указанных ниже значений перечисления **StiReportDisplayMode**: › **FromReport** - режим экспорта элементов отчета определяется из настроек шаблона - Div или Table; › **Table** – элементы отчета экспортируются с использованием HTML таблиц (значение по умолчанию); › **Div** – элементы отчета экспортируются с использованием DIV разметки; › **Span** - элементы отчета экспортируются с использованием SPAN разметки.
ParametersPanelDateFormat	Устанавливает формат даты и времени для

© 2003-2026 Stimulsoft

	переменных соответствующего типа на панели параметров. По умолчанию используется формат даты и времени, установленный браузером.
CloseDesignerWithoutAsking	Предоставляет возможность установить режим закрытия дизайнера отчетов без подтверждения изменений в текущем отчете. По умолчанию свойство имеет значение **false**, т.е. дизайнер будет запрашивать разрешение на сохранение изменений в текущем отчете.
ShowSystemFonts	Предоставляет возможность отображать или не отображать системные шрифты в списке шрифтов дизайнера отчетов. По умолчанию свойство имеет значение **true**, т.е. системные шрифты отображаются в списке шрифтов.
WizardTypeRunningAfterLoad	Предоставляет возможность вызывать мастер создания отчетов после запуска дизайнера отчетов. Может принимать одно из указанных ниже значений перечисления **StiWizardType**: › **None** - дизайнер отчетов будет запущен без вызова мастера создания отчета; › **StandardReport** - дизайнер отчетов будет запущен с вызовом стандартного мастера создания отчета; › **MasterDetailReport** - дизайнер отчетов будет запущен с вызовом master-detail мастера создания отчета; › **LabelReport** - дизайнер отчетов будет запущен с вызовом мастера создания отчета с этикетками; › **InvoicesReport** - дизайнер отчетов будет запущен с вызовом мастера создания инвойса; › **OrdersReport** - дизайнер отчетов будет запущен с вызовом мастера создания ордера;

© 2003-2026 Stimulsoft

	› **QuotationReport** - дизайнер отчетов будет запущен с вызовом мастера создания квоты.
AllowWordWrapTextEditors	Предоставляет возможность установить режим переноса в текстовых редакторах. По умолчанию свойство имеет значение **true**.
DefaultRibbonType	Предоставляет возможность установить используемый по умолчанию стиль панели инструментов дизайнера. Может принимать одно из указанных ниже значений перечисления: › **StiDesignerRibbonType.Classic** - Будет установлен классический стиль (значение по умолчанию); › **StiDesignerRibbonType.SingleLine** – Будет установлен компактный однострочный стиль.
AllowPropagationEvents	Предоставляет возможность разрешить распространение событий нажатия клавиш, когда дизайнер отчетов не в фокусе. По умолчанию свойство имеет значение **true**.
PropertiesPanelViewMode	Предоставляет возможность закрепить или открепить панель свойств, словарь отчета, дерево отчета. Может принимать одно из указанных ниже значений перечисления: › **Pinned** - панели закреплены (значение по умолчангию); › **Unpinned** - панели открепленны.

Поведение (Behavior)

Наименование	Описание
ShowSaveDialog	Включает отображение диалога ввода имени отчета при его сохранении. Имя отчета будет передано в параметрах дизайнера. По умолчанию свойство имеет значение **true**.
UndoMaxLevel	Устанавливает максимальную глубину отмены действий с отчетом (функция Undo/

© 2003-2026 Stimulsoft

	Redo). Большое значение этого свойства будет потреблять больше памяти на стороне сервера для хранения параметров отмены. По умолчанию установлено значение **6**.
AllowChangeWindowTitle	Разрешает использование заголовка окна браузера для отображения имени файла редактируемого отчета. По умолчанию свойство имеет значение **true**.
SaveReportMode	Задает режим сохранения отчета. Может принимать одно из трех значений перечисления **StiSaveMode**. › **Hidden** - действие сохранения отчета вызывается в фоне при помощи AJAX-запроса и никак не отображается в окне браузера (значение по умолчанию); › **Visible** - действие сохранения отчета будет вызвано в текущем окне браузера в видимом режиме при помощи POST-запроса; › **NewWindow** - действие сохранения отчета будет вызвано в новом окне (вкладке) браузера.
SaveReportAsMode	Задает режим сохранения отчета. Может принимать одно из трех значений перечисления **StiSaveMode**. › **Hidden** - действие сохранения отчета вызывается в фоне при помощи AJAX-запроса и никак не отображается в окне браузера (значение по умолчанию); › **Visible** - действие сохранения отчета будет вызвано в текущем окне браузера в видимом режиме при помощи POST-запроса; › **NewWindow** - действие сохранения отчета будет вызвано в новом окне (вкладке) браузера.
CheckReportBeforePreview	Предоставляет возможность проверять или не проверять отчет встроенным

© 2003-2026 Stimulsoft

	инструментом перед его просмотром. По умолчанию свойство имеет значение **true**.

Главное меню (FileMenu)

Наименование	Описание
Visible	Включает отображение главного меню дизайнера отчетов. По умолчанию свойство имеет значение **true**.
ShowNew	Включает отображение пункта главного меню **Создать (New)**. По умолчанию свойство имеет значение **true**.
ShowFileMenuNewReport	Включает отображение кнопки **Пустой отчет (Blank Report)** в меню **Создать**. По умолчанию свойство имеет значение **true**.
ShowFileMenuNewDashboard	Включает отображение кнопки **Пустая панель индикаторов (Blank Dashboard)** в меню **Создать**. По умолчанию свойство имеет значение **true**.
ShowOpen	Включает отображение пункта главного меню **Открыть (Open)**. По умолчанию свойство имеет значение **true**.
ShowSave	Включает отображение пункта главного меню **Сохранить (Save)**. По умолчанию свойство имеет значение **true**.
ShowSaveAs	Включает отображение пункта главного меню **Сохранить как (Save As)**. По умолчанию свойство имеет значение **true**.
ShowClose	Включает отображение пункта главного меню **Закрыть (Close)**. По умолчанию свойство имеет значение **true**.
ShowExit	Включает отображение пункта главного меню **Выход (Exit)**. По умолчанию свойство имеет значение **false**.

© 2003-2026 Stimulsoft

ShowReportSetup	Включает отображение пункта главного меню **Свойства отчета (Report Setup)**. По умолчанию свойство имеет значение **true**.
ShowOptions	Включает отображение пункта главного меню **Параметры (Options)**. По умолчанию свойство имеет значение **true**.
ShowInfo	Включает отображение пункта главного меню **Сведения (Info)**. По умолчанию свойство имеет значение **true**.
ShowAbout	Включает отображение пункта главного меню **О программе (About)**. По умолчанию свойство имеет значение **true**.
ShowHelp	Включает отображение пункта главного меню **Помощь (Help)**. По умолчанию свойство имеет значение **true**.

Словарь данных (Dictionary)

Наименование	Описание
Visible	Включает отображение словаря данных отчета. По умолчанию свойство имеет значение **true**.
UseAliases	Предоставляет возможность использовать псевдонимы в словаре данных. Может принимать одно из трех значений перечисления **StiUseAliases**: › **Auto** - определяет режим использования псевдонимов из сохраненного значения в файлах cookies (значение по умолчанию); › **True** - устанавливает режим использования псевдонимов в словаре данных; › **False** - выключает режим использования псевдонимов в словаре данных.
NewReportDictionary	Предоставляет возможность создавать новый

© 2003-2026 Stimulsoft

	словарь данных или объединить с существующим, при создании нового отчета в дизайнере. Может принимать одно из трех значений перечисления **StiNewReportDictionary**: › **Auto** - определяет режим создания или объединения словаря данных из сохраненного значения в файлах cookies(значение по умолчанию); › **DictionaryNew** - устанавливает режим создания нового словаря данных, при создании нового отчета; › **DictionaryMerge** - устанавливает режим объединения существующего словаря данных с новым, при создании нового отчета в дизайнере.
ShowDictionaryContextMenuProperties	Устанавливает видимость пункта **Свойства (Properties)** в контекстном меню словаря данных. По умолчанию свойство имеет значение **true**.
ShowDictionaryActions	Устанавливает видимость меню **Действия (Actions)** на панели инструментов словаря данных. По умолчанию свойство имеет значение **true**.
PermissionDataConnections	Устанавливает доступные действия над подключениями к данным отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionDataSources	Устанавливает доступные действия над источниками данных отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionDataColumns	Устанавливает доступные действия над колонками данных отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.

© 2003-2026 Stimulsoft

PermissionBusinessObjects	Устанавливает доступные действия над бизнес-объектами отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionDataRelations	Устанавливает доступные действия над связями данных отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionVariables	Устанавливает доступные действия над переменными отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionResources	Устанавливает доступные действия над ресурсами в словаре данных отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionSqlParameters	Устанавливает доступные действия над параметрами в SQL запросах источников данных отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionDataTransformations	Устанавливает доступные действия над преобразованием данных. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
PermissionUserFunctions	Устанавливает доступные действия над пользовательскими функциями. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.

В таблице ниже приведены все доступные значения для перечисления **StiDesignerPermissions**, которые можно установить для элементов словаря отчета.

Значение	**Описание**

© 2003-2026 Stimulsoft

None	Запрещает любое действие над элементом словаря данных.
All	Разрешает любые действия над элементом словаря данных.
Create	Разрешает создавать определенный элемент словаря данных.
Delete	Разрешает удалять определенный элемент словаря данных.
Modify	Разрешает редактировать определенный элемент словаря данных.
View	Разрешает просматривать определенный элемент словаря данных.
ModifyView	Разрешает редактировать и просматривать определенный элемент словаря данных.

Панель инструментов (Toolbar)

Наименование	Описание
ShowToolbar	Предоставляет возможность отображать или не отображать панель инструментов в дизайнере отчетов. По умолчанию свойство имеет значение **true**.
ShowSetupToolboxButton	Включает отображение кнопки вызова окна настроек боковой панели компонентов отчета. По умолчанию свойство имеет значение **true**.
ShowInsertButton	Включает отображение вкладки **Вставить (Insert)** на панели инструментов дизайнера. По умолчанию свойство имеет значение **true**.
ShowLayoutButton	Включает отображение вкладки **Формат (Layout)** на панели инструментов дизайнера. По умолчанию свойство имеет значение **true**.
ShowPageButton	Включает отображение вкладки **Страница (Page)** на панели инструментов дизайнера.

© 2003-2026 Stimulsoft

	По умолчанию свойство имеет значение **true**.
ShowPreviewButton	Включает отображение вкладки **Предварительный просмотр (Preview)** на панели инструментов дизайнера. По умолчанию свойство имеет значение **true**.
ShowSaveButton	Включает отображение кнопки **Сохранить (Save)** на панели инструментов дизайнера. По умолчанию свойство имеет значение **true**.
ShowAboutButton	Включает отображение кнопки **О программе (About)** на панели инструментов дизайнера. По умолчанию свойство имеет значение **false**.

Панель свойств (PropertiesGrid)

Наименование	Описание
Visible	Включает отображение панели свойств компонентов отчета. По умолчанию свойство имеет значение **true**.
Width	Задает ширину панели свойств компонентов отчета. По умолчанию установлена ширина **370** точек.
LabelWidth	Задает ширину надписей панели свойств компонентов отчета. По умолчанию установлена ширина **160** точек.
PropertiesGridPosition	Предоставляет возможность определить позицию панели свойств в дизайнере отчетов. Может принимать одно из трех значений перечисления **StiPropertiesGridPosition**. › **Left** - панель свойств будет располагаться слева; › **Right** - панель свойств будет располагаться справа.

© 2003-2026 Stimulsoft

ShowPropertiesWhichUsedFromStyles	Устанавливает видимость свойств, которые используются из стилей в дизайнере. По умолчанию свойство имеет значение **false**.

Компоненты отчета (Components)

Наименование	Описание
ShowText	Включает отображение компонента **Текст (Text)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowTextInCells	Включает отображение компонента **Текст в ячейках (Text in Cells)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowRichText	Включает отображение компонента **Rich текст (Rich Text)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowImage	Включает отображение компонента **Картинка (Image)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowBarCode	Включает отображение компонента **Штрих-код (Bar Code)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowShape	Включает отображение компонента **Геометрия (Shape)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowHorizontalLinePrimitive	Включает отображение компонента **Горизонтальная линия (Horizontal Line)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.

© 2003-2026 Stimulsoft

ShowVerticalLinePrimitive	Включает отображение компонента **Вертикальная линия (Vertical Line)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowRectanglePrimitive	Включает отображение компонента **Прямоугольник (Rectangle)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowRoundedRectanglePrimitive	Включает отображение компонента **Закругленный прямоугольник (Rounded Rectangle)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowPanel	Включает отображение компонента **Панель (Panel)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowClone	Включает отображение компонента **Клон (Clone)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowCheckBox	Включает отображение компонента **Флажок (Check Box)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowSubReport	Включает отображение компонента **Вложенный отчет (Sub Report)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowZipCode	Включает отображение компонента **Почтовый индекс (Zip Code)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowTable	Включает отображение компонента **Таблица (Table)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет

© 2003-2026 Stimulsoft

	значение **true**.
ShowCrossTab	Включает отображение компонента **Кросс-таблица (Cross-Tab)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowChart	Включает отображение компонента **Диаграмма (Chart)** в меню вставки компонентов дизайнера. Действует сразу на все типы диаграмм. По умолчанию свойство имеет значение **true**.
ShowMap	Включает отображение компонента **Карта (Map)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **false**.
ShowGauge	Включает отображение компонента **Индикатор (Gauge)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **false**.
ShowSparkline	Включает отображение компонента **Спарклайн (Sparkline)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowMathFormula	Включает отображение компонента **Математическая формула (Math Formula)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowElectronicSignature	Включает отображение компонента **Электронная подпись (Electronic Signature)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowPdfDigitalSignature	Включает отображение компонента **Цифровая подпись PDF (PDF Digital Signature)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.

© 2003-2026 Stimulsoft

Секции (Bands)

Наименование	Описание
ShowReportTitleBand	Включает отображение пункта **Заголовок отчета (Report Title)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowReportSummaryBand	Включает отображение пункта **Итог отчета (Report Summary)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowPageHeaderBand	Включает отображение пункта **Заголовок страницы (Page Header)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowPageFooterBand	Включает отображение пункта **Итог страницы (Page Footer)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowGroupHeaderBand	Включает отображение пункта **Заголовок группы (Group Header)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowGroupFooterBand	Включает отображение пункта **Итог группы (Group Footer)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowHeaderBand	Включает отображение пункта **Заголовок данных (Header)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowFooterBand	Включает отображение пункта **Итог данных (Footer)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowColumnHeaderBand	Включает отображение пункта **Заголовок колонки (Column Header)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.

© 2003-2026 Stimulsoft

ShowColumnFooterBand	Включает отображение пункта **Итог колонки (Column Footer)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowDataBand	Включает отображение пункта **Данные (Data)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowHierarchicalBand	Включает отображение пункта **Дерево (Hierarchical)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowChildBand	Включает отображение пункта **Подчиненный (Child)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowEmptyBand	Включает отображение пункта **Пустые данные (Empty)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowOverlayBand	Включает отображение пункта **Водяной знак (Overlay)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.
ShowTableOfContents	Включает отображение пункта **Оглавление (Table Of Contents)** в списке меню **Секции**. По умолчанию свойство имеет значение **true**.

Элементы дашборда (DashboardElements)

Наименование	Описание
ShowTableElement	Включает отображение элемента панели индикаторов **Таблица (Table)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowCardsElement	Включает отображение элемента панели индикаторов **Карточки (Cards)** на инструментарии или вкладке **Вставить**

© 2003-2026 Stimulsoft

	(Insert) в дизайнере. По умолчанию свойство имеет значение **true**.
ShowChartElement	Включает отображение элемента панели индикаторов **Диаграмма (Chart)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowGaugeElement	Включает отображение элемента панели индикаторов **Измерительный прибор (Gauge)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowPivotTableElement	Включает отображение элемента панели индикаторов **Сводная таблица (Pivot)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowIndicatorElement	Включает отображение элемента панели индикаторов **Индикатор (Indicator)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowProgressElement	Включает отображение элемента панели индикаторов **Прогресс (Progress)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowRegionMapElement	Включает отображение элемента панели индикаторов **Региональная карта (Region Map)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowOnlineMapElement	Включает отображение элемента панели индикаторов **Онлайн карта (Online Map)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию

© 2003-2026 Stimulsoft

	свойство имеет значение **true**.
ShowImageElement	Включает отображение элемента панели индикаторов **Изображение (Image)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowTextElement	Включает отображение элемента панели индикаторов **Текст (Text)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowPanelElement	Включает отображение элемента панели индикаторов **Панель (Panel)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowShapeElement	Включает отображение элемента панели индикаторов **Геометрия (Shape)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowButtonElement	Включает отображение элемента панели индикаторов **Кнопка (Button)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowListBoxElement	Включает отображение элемента панели индикаторов **Список (ListBox)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowComboBoxElement	Включает отображение элемента панели индикаторов **Выпадающий список (ComboBox)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.

© 2003-2026 Stimulsoft

ShowTreeViewElement	Включает отображение элемента панели индикаторов **Иерархический список (TreeView)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowTreeViewBoxElement	Включает отображение элемента панели индикаторов **Выпадающий иерархический список (TreeViewBox)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
ShowDatePickerElement	Включает отображение элемента панели индикаторов **Выбор даты (Date Picker)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.

Кросс-секции (CrossBands)

Наименование	Описание
ShowCrossGroupHeaderBand	Включает отображение секции **Кросс-заголовок группы (Cross Group Header)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowCrossGroupFooterBand	Включает отображение секции **Кросс-итог группы (Cross Group Footer)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowCrossHeaderBand	Включает отображение секции **Кросс-заголовок данных (Cross Header)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
ShowCrossFooterBand	Включает отображение секции **Кросс-итог данных (Cross Footer)** в меню вставки компонентов дизайнера. По умолчанию

© 2003-2026 Stimulsoft

	свойство имеет значение **true**.
ShowCrossDataBand	Включает отображение секции **Кросс-данные (Cross Data)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.

Панели индикаторов(Dashboards)

Наименование	Описание
ShowNewDashboardButton	Предоставляет возможность отображать или не отображать кнопку **Новая панель индикаторов (New Dashboard)** в дизайнере отчетов. По умолчанию свойство имеет значение **true**.

Страницы(Pages)

Наименование	Описание
ShowNewPageButton	Предоставляет возможность отображать или не отображать кнопку **Новая страница (New Page)** в дизайнере отчетов. По умолчанию свойство имеет значение **true**.

При разработке отчета или дашборда в дизайнере отчетов, можно также определить настройки экспорта (ExportOptions), параметры Email (EmailOptions) и панель инструментов (PreviewToolbarOptions) на вкладке **Просмотр (Preview)**. Эти опции аналогичны опциям вьювера отчетов.

8 Reports and Dashboards for JS

JavaScript представляет собою прототипно-ориентированный сценарный язык программирования. Компания Stimulsoft предоставляет инструменты создания, отображения, преобразования отчетов и панелей индикаторов (дашбордов) на различных устройствах с использованием этого языка.

© 2003-2026 Stimulsoft

8.1 Быстрый старт

В этом разделе собраны пошаговые руководства для быстрого развертывания компонентов Stimulsoft в приложениях JavaScript:

- Vanilla JavaScript;
- Angular JS;
- React JS;
- Vue JS;
- Node JS.

8.1.1 Чистый JavaScript

В этой главе будет рассмотрен пример быстрого развертывания Stimulsoft в приложениях pure JavaScript. Такое приложение состоит из HTML страницы и JS скриптов.

Создайте index.html файл

Это может быть любой html файл, но по умолчанию точкой входа считается `index.html`.

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>

</body>
</html>
```

Установите компоненты Stimulsoft

Для начала необходимо скачать пакет Stimulsoft. Если необходимы только инструменты отчетности, то следует скачать пакет Stimulsoft Reports.JS. Если же требуются и средства отчетности и дашборды, то пакет Stimulsoft Dashboards.JS. Затем, следует подключить скрипты Stimulsoft в файле `index.html`.

© 2003-2026 Stimulsoft

index.html

```
...
<script type="text/javascript" src="scripts/stimulsoft.reports.js"></
script>
<script type="text/javascript" src="scripts/stimulsoft.dashboards.js"></
script>
<script type="text/javascript" src="scripts/stimulsoft.designer.js"></
script>
<script type="text/javascript" src="scripts/stimulsoft.viewer.js"></
script>
<script type="text/javascript" src="scripts/
stimulsoft.blockly.editor.js"></script>
...
```

Создайте функцию запуска

Например, функция вызова дизайнера отчетов с пустым отчетом.

index.html

```
...
<script type="text/javascript">
   function onLoad() {
      var report = new Stimulsoft.Report.StiReport();

      var designer = new Stimulsoft.Designer.StiDesigner();
      designer.renderHtml('content');
      designer.report = report;
   }
</script>

...

<body onload="onLoad()">
   <div id="content"></div>
</body>
...
```

Или, функция вызова вьювера отчетов с ранее созданным шаблоном отчета.

© 2003-2026 Stimulsoft

index.html

```
...
<script type="text/javascript">
   function onLoad() {
      var report = new Stimulsoft.Report.StiReport();
      report.loadFile('reports/Report.mrt');

      var viewer = new Stimulsoft.Viewer.StiViewer();
      viewer.renderHtml('content');
      viewer.report = report;
   }
</script>

...

<body onload="onLoad()">
   <div id="content"></div>
</body>
...
```

Первый запуск

По умолчанию, браузер не имеет доступа к файловой системе. Это обусловлено политикой безопасности браузеров. Для того, чтобы локальный проект запускался корректно, следует использовать различные веб-серверы. Например, можно установить глобально http-server или serve, и затем запустить веб-сервер из командной строки в корневой папке проекта. В этом случае, `index.html` будет открыт в бразузере, с дизайнером или вьювером Stimulsoft.

8.1.2 JavaScript и CDN сервисы

В этой главе будет рассмотрен пример быстрого развертывания Stimulsoft в приложениях pure JavaScript с подл.чением файлов скриптов через CDN сервисы. Такое приложение состоит из HTML страницы и ссылок для JS скриптов.

Создайте index.html файл

Это может быть любой html файл, но по умолчанию точкой входа считается `index.html`.

© 2003-2026 Stimulsoft

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>

</body>
</html>
```

Подключите файлы скриптов Stimulsoft

На данный момент, можно использовать следующие сервисы:

- cdn.jsdelivr.net;
- unpkg.com.

При помощи этих сервисов можно получить доступ к файлам скриптов по URL из npm пакетов stimulsoft-reports-js и stimulsoft-dashboards-js . Таким образом, можно подключить файлы скриптов по URL в файле `index.html`. Например, при помощи сервиса `cdn.jsdelivr.net`:

index.html

```
...
<script type="text/javascript" src="https://cdn.jsdelivr.net/npm/
stimulsoft-reports-js/Scripts/stimulsoft.reports.js"></script>
<script type="text/javascript" src="https://cdn.jsdelivr.net/npm/
stimulsoft-reports-js/Scripts/stimulsoft.designer.js"></script>
<script type="text/javascript" src="https://cdn.jsdelivr.net/npm/
stimulsoft-reports-js/Scripts/stimulsoft.viewer.js"></script>
<script type="text/javascript" src="https://cdn.jsdelivr.net/npm/
stimulsoft-reports-js/Scripts/stimulsoft.blockly.editor.js"></script>
...
```

или, при использовании сервиса `unpkg.com`:

© 2003-2026 Stimulsoft

index.html

```
...
<script type="text/javascript" src="https://www.unpkg.com/stimulsoft-dashboards-js/Scripts/stimulsoft.reports.js"></script>
<script type="text/javascript" src="https://www.unpkg.com/stimulsoft-dashboards-js/Scripts/stimulsoft.dashboards.js"></script>
<script type="text/javascript" src="https://www.unpkg.com/stimulsoft-dashboards-js/Scripts/stimulsoft.designer.js"></script>
<script type="text/javascript" src="https://www.unpkg.com/stimulsoft-dashboards-js/Scripts/stimulsoft.viewer.js"></script>
<script type="text/javascript" src="https://www.unpkg.com/stimulsoft-dashboards-js/Scripts/stimulsoft.blockly.editor.js"></script>
...
```

Информация

Обратите внимание, при помощи сервисов `cdn.jsdelivr.net` и `unpkg.com` можно подключить различные скрипты, которые поставляются в пакетах stimulsoft-reports-js и stimulsoft-dashboards-js. Также можно подключить файлы скриптов определенной версии npm пакета, указав ее в URL через символ `@`. Например, https://cdn.jsdelivr.net/npm/stimulsoft-reports-js@2024.4.1/Scripts/stimulsoft.reports.js или https://www.unpkg.com/stimulsoft-reports-js@2024.4.1/Scripts/stimulsoft.reports.js.

Если же версия пакета в URL не будет указана, то файлы скриптов будут загружены из последней доступной версии пакета stimulsoft-reports-js или stimulsoft-dashboards-js

Создайте функцию запуска

Например, функция вызова дизайнера отчетов с пустым отчетом.

© 2003-2026 Stimulsoft

index.html

```
...
<script type="text/javascript">
   function onLoad() {
      var report = new Stimulsoft.Report.StiReport();

      var designer = new Stimulsoft.Designer.StiDesigner();
      designer.renderHtml('content');
      designer.report = report;
   }
</script>

...

<body onload="onLoad()">
   <div id="content"></div>
</body>
...
```

Или, функция вызова вьювера отчетов с ранее созданным шаблоном отчета.

index.html

```
...
<script type="text/javascript">
   function onLoad() {
      var report = new Stimulsoft.Report.StiReport();
      report.loadFile('reports/Report.mrt');

      var viewer = new Stimulsoft.Viewer.StiViewer();
      viewer.renderHtml('content');
      viewer.report = report;
   }
</script>

...

<body onload="onLoad()">
   <div id="content"></div>
</body>
...
```

Первый запуск

По умолчанию, браузер не имеет доступа к файловой системе. Это обусловлено политикой безопасности браузеров. Для того, чтобы локальный проект запускался корректно, следует использовать различные веб-серверы. Например, можно установить глобально http-server или serve, и затем запустить веб-сервер из командной строки в корневой папке проекта. В этом случае,

© 2003-2026 Stimulsoft

`index.html` будет открыт в бразузере, с дизайнером или вьювером Stimulsoft.

8.1.3 Angular JS

В этой главе будет рассмотрен пример использования компонентов Stimulsoft в приложениях Angular.

Создайте проект Angular

В терминале перейдите в каталог, в котором необходимо разместить будущий проект. Затем следует выполнить команду.

terminal

```
ng new sti-angular-js --no-standalone --routing --ssr=false --style css
```

Установите компоненты Stimulsoft

Для начала необходимо скачать пакет Stimulsoft. Если необходимы только инструменты отчетности, то следует скачать пакет Stimulsoft Reports.JS. Если же требуются и средства отчетности и дашборды, то пакет Stimulsoft Dashboards.JS. Далее, следует скопировать скрипты Stimulsoft в проект по пути `sti-angular-js./src/scripts.`

Настройте конфигурацию Angular проект

Для этого в секции `scripts` файла **angular.json** следует указать массив путей к файлам скриптов в проекте.

angular.json

```
...
"scripts": [
   "src/scripts/stimulsoft.reports.engine.js",
   "src/scripts/stimulsoft.reports.export.js",
   "src/scripts/stimulsoft.reports.chart.js",
   "src/scripts/stimulsoft.reports.maps.js",
   "src/scripts/stimulsoft.reports.import.xlsx.js",
   "src/scripts/stimulsoft.viewer.js",
   "src/scripts/stimulsoft.designer.js",
   "src/scripts/stimulsoft.blockly.editor.js"
]
...
```

© 2003-2026 Stimulsoft

В секции `architect` для параметра `builder` определите тип как `browser`, а не `application`. Кроме этого параметр в `options` следует переименовать с `browser` на `main`.

angular.json

```
...
"architect": {
   "build": {
      "builder": "@angular-devkit/build-angular:browser",
      "options": {
         "main": "src/main.ts"
      }
   }
}
...
```

Если подразумевается использование во вьювере или дизайнере уже готовых отчетов, то следует добавить путь к ним в секции `assets`.

angular.json

```
...
"assets": [
   "src/reports"
]
...
```

Добавьте модуль HttpClientModule

Необходимо импортировать и затем подключить модуль `HttpClientModule` в файле **app.module.ts**.

© 2003-2026 Stimulsoft

app.module.ts

```
import { HttpClientModule } from '@angular/common/http';

@NgModule({
...
  imports: [
    BrowserModule,
    AppRoutingModule,
    HttpClientModule
  ],
  ...
})
```

Разместите компонент Stimulsoft

В файле **app.component.ts** также необходимо импортировать `HttpClientModule`. Затем, можно импортировать Stimulsoft через директиву `declare var Stimulsoft: any;`. И в классе `AppComponent` следует определить вызов компонентов Stimulsoft. Например, дизайнер с пустым отчетом.

© 2003-2026 Stimulsoft

app.component.ts

```
import { Component } from '@angular/core';
import { HttpClientModule } from '@angular/common/http';

declare var Stimulsoft: any;

@Component({
   selector: 'app-root',
   template: `<div>
               <h2>Stimulsoft Reports.JS - Invoice.mrt - Designer</h2>
               <div id="content"></div>
             </div>`
})

export class AppComponent {
   designer: any = new Stimulsoft.Designer.StiDesigner(false,
   "StiDesigner", false);

  ngOnInit() {
     var report = new Stimulsoft.Report.StiReport();

     this.designer.report = report;
     this.designer.renderHtml("content");
   }

   constructor(private http: HttpClientModule) {

   }
}
```

Или, вызвать вьювер с ранее созданным отчетом.

© 2003-2026 Stimulsoft

index.html

```
...
export class AppComponent {
   viewer: any = new Stimulsoft.Viewer.StiViewer(false, "StiViewer",
   false);

  ngOnInit() {
     var report = new Stimulsoft.Report.StiReport();
     report.loadFile("reports/Invoice.mrt");

     this.viewer.report = report;
     this.viewer.renderHtml("content");
   }

   constructor(private http: HttpClientModule) {

   }
}
...
```

Первый запуск

Ангуляр приложение по умолчанию определяет стартовую команду в файле **package.json**. Поэтому для запуска проекта, достаточно в корневой папке проекта из терминала выполнить команду.

console

```
npm start
```

8.1.4 React JS

В этой главе будет рассмотрен пример использования компонентов Stimulsoft в приложениях React.

Создайте проект React

В терминале перейдите в каталог, в котором необходимо разместить будущий проект. Затем следует выполнить команду.

terminal

```
npx create-react-app my-app
```

© 2003-2026 Stimulsoft

Установите компоненты Stimulsoft

Для начала необходимо скачать пакет Stimulsoft. Если необходимы только инструменты отчетности, то следует скачать пакет stimulsoft-reports-js. Если же требуются и средства отчетности и дашборды, то пакет stimulsoft-dashboards-js. Для этого следует выполнить команду в терминале.

terminal

```
npm install stimulsoft-dashboards-js
```

Интегрируйте Stimulsoft в приложение

Для этого следует отредактировать файл **App.js** в папке **src** проекта. Для начала, следует импортировать модуль `Stimulsoft` и класс `React`. Далее, следует создать класс `App`, унаследованный от `React.Component`, переопределить конструктор класса, методы `render()` и `componentDidMount()`. Например, будет вызван дизайнер отчетов с пустым отчетом.

© 2003-2026 Stimulsoft

App.js

```
...
import React from 'react';
import { Stimulsoft } from 'stimulsoft-dashboards-js/Scripts/
stimulsoft.designer';

class App extends React.Component {
   constructor() {
      super();
      this.designer = new Stimulsoft.Designer.StiDesigner(false,
      "StiDesigner", false);
   }

   render() {
      return (
             <div className="App">
                <h2>Stimulsoft Designer</h2>
                <div id="content"></div>
             </div>
      );
   }

   componentDidMount() {
      var report = new Stimulsoft.Report.StiReport();

      this.designer.report = report;
      this.designer.renderHtml("content");
   }
}

export default App;
...
```

Или, вызвать вьювер с ранее созданным отчетом. Файлы отчетов следует предварительно скопировать в папку **reports** в директории проекта **./public**.

© 2003-2026 Stimulsoft

App.js

```
...
class App extends React.Component {
   constructor() {
      super();
      this.viewer = new Stimulsoft.Viewer.StiViewer(false, "StiViewer",
      false);
   }

   render() {
      return (
             <div className="App">
                <h2>Stimulsoft Viewer</h2>
                <div id="content"></div>
             </div>
      );
   }

   componentDidMount() {
      var report = new Stimulsoft.Report.StiReport();
      report.loadFile("reports/Invoice.mrt");

      this.viewer.report = report;
      this.viewer.renderHtml("content");
   }
}
...
```

Первый запуск

React приложение по умолчанию определяет стартовую команду в файле **package.json**. Поэтому для запуска проекта, достаточно в корневой папке проекта из терминала выполнить команду.

console

```
npm start
```

8.1.5 Vue JS

В этой главе будет рассмотрен пример использования компонентов Stimulsoft в приложениях Vue.

Создайте проект Vue

В терминале перейдите в каталог, в котором необходимо разместить будущий проект. Затем следует выполнить команду.

© 2003-2026 Stimulsoft

terminal

```
npm create vue@latest my-app
```

Параметры настройки приложения можно оставить по умолчанию, или в случае необходимости изменить.

Установите компоненты Stimulsoft

Для начала необходимо скачать пакет Stimulsoft. Если необходимы только инструменты отчетности, то следует скачать пакет stimulsoft-reports-js. Если же требуются и средства отчетности и дашборды, то пакет stimulsoft-dashboards-js. Для этого следует выполнить команду в терминале.

terminal

```
npm install stimulsoft-dashboards-js
```

Интегрируйте Stimulsoft в приложение

Для этого следует отредактировать файл **App.vue** в папке **src** проекта. Для начала, следует импортировать модуль Stimulsoft и метод `onMounted()`. Здесь же можно изменить блок `template` и `styles`. В метод `onMounted()` передать `callback` функцию с инициализацией Stimulsoft компонентов. Например, будет вызван дизайнер отчетов с пустым отчетом.

© 2003-2026 Stimulsoft

App.vue

```
<script setup lang="ts">
   import { onMounted } from "vue";
   import { Stimulsoft } from "stimulsoft-dashboards-js/Scripts/
   stimulsoft.designer.js";

   onMounted(() => {
      let designer = new Stimulsoft.Designer.StiDesigner(false,
      "StiDesigner", false);
      let report = new Stimulsoft.Report.StiReport();

      designer.report = report;
      designer.renderHtml("content");
   });
</script>

<template>
   <div id="app">
      <div>
         <h2 id="app-title">Stimulsoft Vue JS</h2>
         <div id="content"></div>
      </div>
   </div>
</template>

<style>
   #app {
      font-family: Avenir, Helvetica, Arial, sans-serif;
      -webkit-font-smoothing: antialiased;
      -moz-osx-font-smoothing: grayscale;
      color: #2c3e50;
      margin-top: 60px;
   }

   #app-title {
      text-align: center;
   }
</style>
```

Или, вызвать вьювер с ранее созданным отчетом. Файлы отчетов следует предварительно скопировать в папку **reports** в директории проекта **./public**.

© 2003-2026 Stimulsoft

App.vue

```
...
<script setup lang="ts">
   import { onMounted } from "vue";
   import { Stimulsoft } from "stimulsoft-dashboards-js/Scripts/
   stimulsoft.viewer.js";

   onMounted(() => {
      let viewer = new Stimulsoft.Viewer.StiViewer(false, "StiViewer",
      false);
      let report = new Stimulsoft.Report.StiReport();
      report.loadFile("reports/SimpleList.mrt");

      viewer.report = report;
      viewer.renderHtml("content");
   });
</script>
...
```

Первый запуск

Vue приложение по умолчанию определяет команды в файле **package.json**. Поэтому для сборки и запуска проекта, достаточно в корневой папке проекта из терминала выполнить команду.

console

```
npm run dev
```

8.1.6 Node JS

В этой главе будет рассмотрен пример использования Stimulsoft в приложениях Node.js. В этом случае, можно использовать только ядро генератора отчетов, без использования визуальных компонентов - дизайнер и вьювер отчетов.

Создайте проект Node.js

В терминале перейдите в каталог, в котором необходимо разместить будущий проект. Затем следует выполнить команду.

terminal

```
npm init
```

© 2003-2026 Stimulsoft

Параметры настройки приложения можно оставить по умолчанию, или в случае необходимости изменить.

Установите компоненты Stimulsoft

Для начала необходимо скачать пакет Stimulsoft. Если необходимы только инструменты отчетности, то следует скачать пакет stimulsoft-reports-js. Если же требуются и средства отчетности и дашборды, то пакет stimulsoft-dashboards-js. Для этого следует выполнить команду в терминале.

terminal

```
npm install stimulsoft-dashboards-js
```

Создайте файл проекта

Это может быть любой js файл, например, `index.js`.

Добавьте отчеты в проект

Скопируйте папку с отчетами в проект. Например, папка **reports** будет содержать файл отчета **Invoice.mrt**.

Интегрируйте Stimulsoft в приложение

Для этого следует отредактировать файл `index.js` в папке проекта. Для начала, следует импортировать модуль Stimulsoft. Затем, создать объект `StiReport()`, загрузить в него отчет, построить и экспортировать в PDF документ.

© 2003-2026 Stimulsoft

index.js

```
var Stimulsoft = require("stimulsoft-dashboards-js");

var report = new Stimulsoft.Report.StiReport();
report.loadFile("reports/Invoice.mrt");

report.renderAsync(function () {
   report.exportDocumentAsync(function (data) {
      var buffer = new Buffer.from(data, "utf-8");
      var fs = require("fs");
      fs.writeFileSync("Invoice.pdf", buffer);
   }, Stimulsoft.Report.StiExportFormat.Pdf);
});
```

Первый запуск

Для этого следует запустить файл проекта в среде Node.js.

console

```
node index.js
```

8.2 HTML5 вьювер

YouTube

Посмотрите видеоуроки по работе с компонентом JS HTML5 Viewer.
Подписывайтесь на канал Stimulsoft и узнайте первыми о новых видеоуроках.
Вопросы и предложения оставляйте в комментариях к видео.

Примеры

Ознакомьтесь с примерами по работе с компонентом JS HTML5 Viewer GitHub.
Все примеры представляют собой отдельные проекты, сгруппированные в отдельных папках по используемым технологиям.

Компонент **HTML5 Viewer (StiViewer)** предназначен для просмотра отчетов в

© 2003-2026 Stimulsoft

окне браузера. При этом Вам не нужен мощный сервер, а на стороне клиента не требуется устанавливать .NET Framework, ActiveX компоненты или какие-либо специальные плагины. Все, что необходимо – это любой современный Web-браузер.

При помощи **HTML5 Viewer** можно просматривать, печатать и выполнять экспортирование отчетов на любом компьютере с любой установленной операционной системой. Так как вьювер использует только HTML5 и JavaScript технологии, он может быть запущен на устройствах, где нет поддержки Flash или Silverlight – планшеты, смартфоны. Также вьювер поддерживает Mobile и Touch интерфейсы, которые автоматически включаются при использовании мобильных устройств и мониторов с сенсорным экраном.

Компонент **HTML5 Viewer** использует технологию **JavaScript** для выполнения всех действий (загрузка и построение отчета, листания страниц и масштабирования, интерактивности в отчетах, печать, экспортирование и др.), что позволяет использовать практически любую серверную сторону, а на стороне клиента избавиться от перезагрузки всей страницы и повысить скорость работы.

HTML5 Viewer поддерживает множество тем оформления, анимированный интерфейс, закладки, интерактивные отчеты, редактирование элементов отчета на странице, полноэкранный режим, поиск, и другие, необходимые для просмотра отчетов, возможности.

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для использования **HTML5 Viewer** в проекте, необходимо установить npm пакет stimulsoft-reports-js:

```
npm install stimulsoft-reports-js
```

Если данный вариант не подходит, можно добавить указанные ниже скрипты

© 2003-2026 Stimulsoft

вручную в проект.

viewer.html

```
...
<script src="scripts/stimulsoft.report.js"></script>
<script scr="scripts/stimulsoft.viewer.js"></script>
...
```

Для использования **HTML5 Viewer** в проекте с возможностью просмотра отчетов и дашбордов , необходимо установить npm пакет stimulsoft-dashboards-js:

```
npm install stimulsoft-dashboards-js
```

Если данный вариант не подходит, можно добавить указанные ниже скрипты вручную в проект.

viewer.html

```
...
<script src="scripts/stimulsoft.report.js"></script>
<script src="scripts/stimulsoft.dashboards.js"></script>
<script scr="scripts/stimulsoft.viewer.js"></script>
...
```

Информация

О вариантах скриптов, которые вы можете использовать в своих проектах, можно ознакомиться в главах Скрипты пакета Reports.JS и Скрипты пакета Dashboards.JS.

8.2.1 Как это работает

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

© 2003-2026 Stimulsoft

Компонент **StiViewer** разработан с использованием только HTML5 и JavaScript технологии, и для его работы не требуется сервер (он необходим только для размещения файлов проекта). При запуске вьювера отчетов происходят следующие действия:

- JavaScript компонент добавляет на текущую страницу HTML код интерфейса вьювера;
- Если был присвоен объект отчета, то отчет будет построен, после чего отобразится первая страница отчета;
- Каждое действие во вьювере (например, листание страниц, печать либо экспорт отчета и др.) вызывает определенное JavaScript событие, в котором можно выполнить необходимые манипуляции с отчетом сервера, что позволяет исключить повторное построение отчета.

8.2.2 Отображение отчета

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Примечание

При присвоении отчета вьювер компоненту, происходит автоматическое построение отчета. Вызов метода `report.render()` необходим только в том случае, если требуется провести какие-либо действия с построенным отчетом перед его отображением во вьювере.

Для отображения отчета необходимо добавить скрипты, необходимые для компонента **StiViewer**, на HTML страницу проекта.

© 2003-2026 Stimulsoft

viewer.html

```
...
<script src="scripts/stimulsoft.reports.js" type="text/javascript"></
script>
<script src="scripts/stimulsoft.dashboards.js"></script>
<script src="scripts/stimulsoft.viewer.js" type="text/javascript"></
script>
...
```

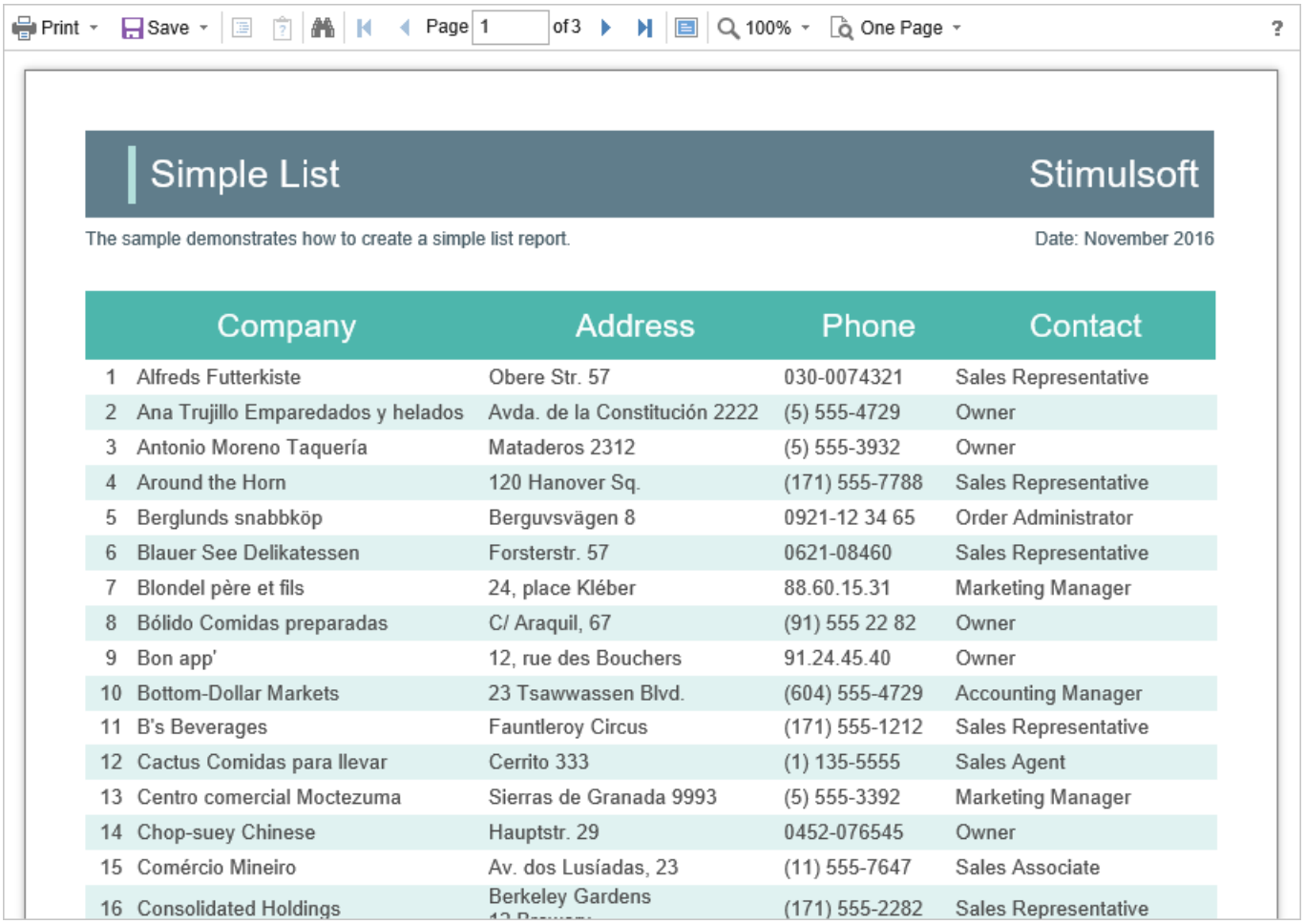
Затем необходимо добавить на HTML страницу JavaScript код загрузки отчета, и присвоить полученный объект вьюверу. В данном случае, вьювер будет развернут в текущем DOM элементе в том месте, где расположен скрипт.

viewer.html

```
...
<script type="text/javascript">
   var report = new Stimulsoft.Report.StiReport();
   report.loadFile("SimpleList.mrt");
   //report.loadFile("Dashboard.mrt");

   var viewer = new Stimulsoft.Viewer.StiViewer();
   viewer.report = report;
</script>
...
```

© 2003-2026 Stimulsoft



Объект **StiViewer** можно создать при помощи конструктора **Stimulsoft.Viewer.StiViewer()**, который может принимать на вход указанные не обязательные аргументы:

› **options**: представляет собой набор опций, находящихся в классе **Stimulsoft.Viewer.StiViewerOptions**. Все опции разделены на категории. Подробное описание категорий и опций находится в разделе Настройки вьювера.

› **viewerId**: идентификатор вьювера, используется при развертывании компонента как DOM объекта, по умолчанию используется значение “StiViewer”

› **renderAfterCreate**: определяет расположение вьювера. Если установлено значение **true**, то вьювер будет отображаться в том же месте DOM дерева, в котором располагается код создания объекта. Если установлено значение **false**, то вьювер будет располагаться в том месте где вызван метод **renderHtml()**. Например, инициализация вьювера в заголовке страницы.

© 2003-2026 Stimulsoft

viewer.html

```
...
<script type="text/javascript">
   var viewer = new Stimulsoft.Viewer.StiViewer(null, "StiViewer", false);
</script>
...
```

И последующий вывод вьювера в текущий элемент DIV.

viewer.html

```
...
<div>Page content</div>
<div>
   <script type="text/javascript">
   // Render the report viewer in this place
      viewer.renderHtml();
   </script>
</div>
...
```

В качестве аргумента метода вывода вьювера **renderHtml(id)** допускается указать идентификатор элемента HTML страницы, в котором необходимо вывести вьювер.

viewer.html

```
...
<script type="text/javascript" >
   var viewer = new Stimulsoft.Viewer.StiViewer(null, "StiViewer", false);
   viewer.renderHtml("content");
</script>
...
```

Указанный элемент должен находиться на HTML странице, на которой используется вьювер отчетов.

viewer.html

```
...
<div id="content"></div>
...
```

© 2003-2026 Stimulsoft

8.2.3 Использование тем

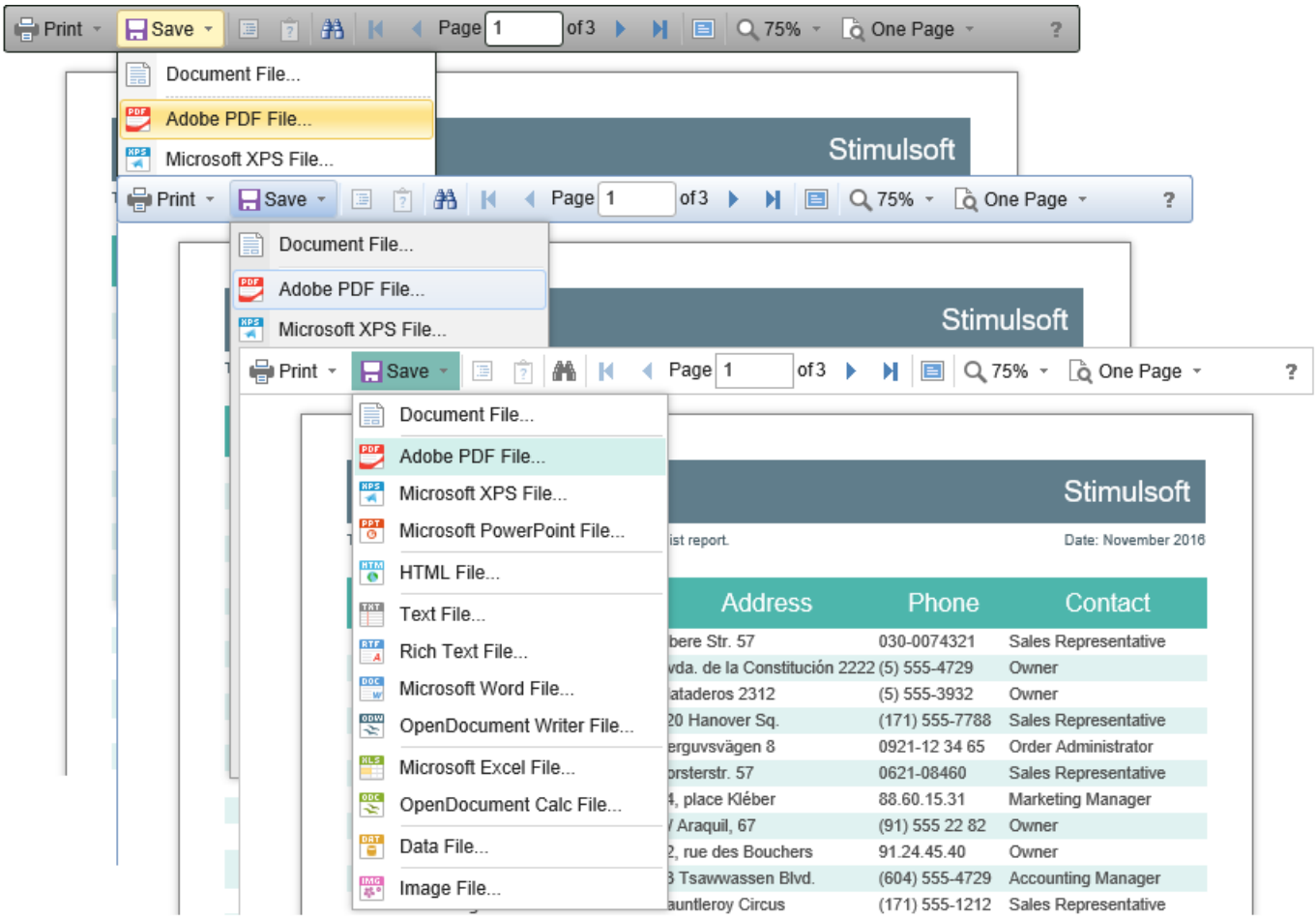
Компонент **HTML5 Viewer** имеет возможность смены тем оформления визуальных элементов управления. Для этого можно использовать опцию компонента **theme** или метод **setTheme()**.

viewer.html

```
...
var options = new Stimulsoft.Viewer.StiViewerOptions();
options.appearance.theme =
Stimulsoft.Viewer.StiViewerTheme.Office2022WhiteBlue;
...
viewer.setTheme(Stimulsoft.Viewer.StiViewerTheme.Office2022WhiteBlue);
...
```

На данный момент доступно **8 тем** оформления с различными цветовыми акцентами. В результате, доступно более **60** вариантов оформления. Это позволяет настроить внешний вид вьювера практически под любое оформление Web проекта.

© 2003-2026 Stimulsoft

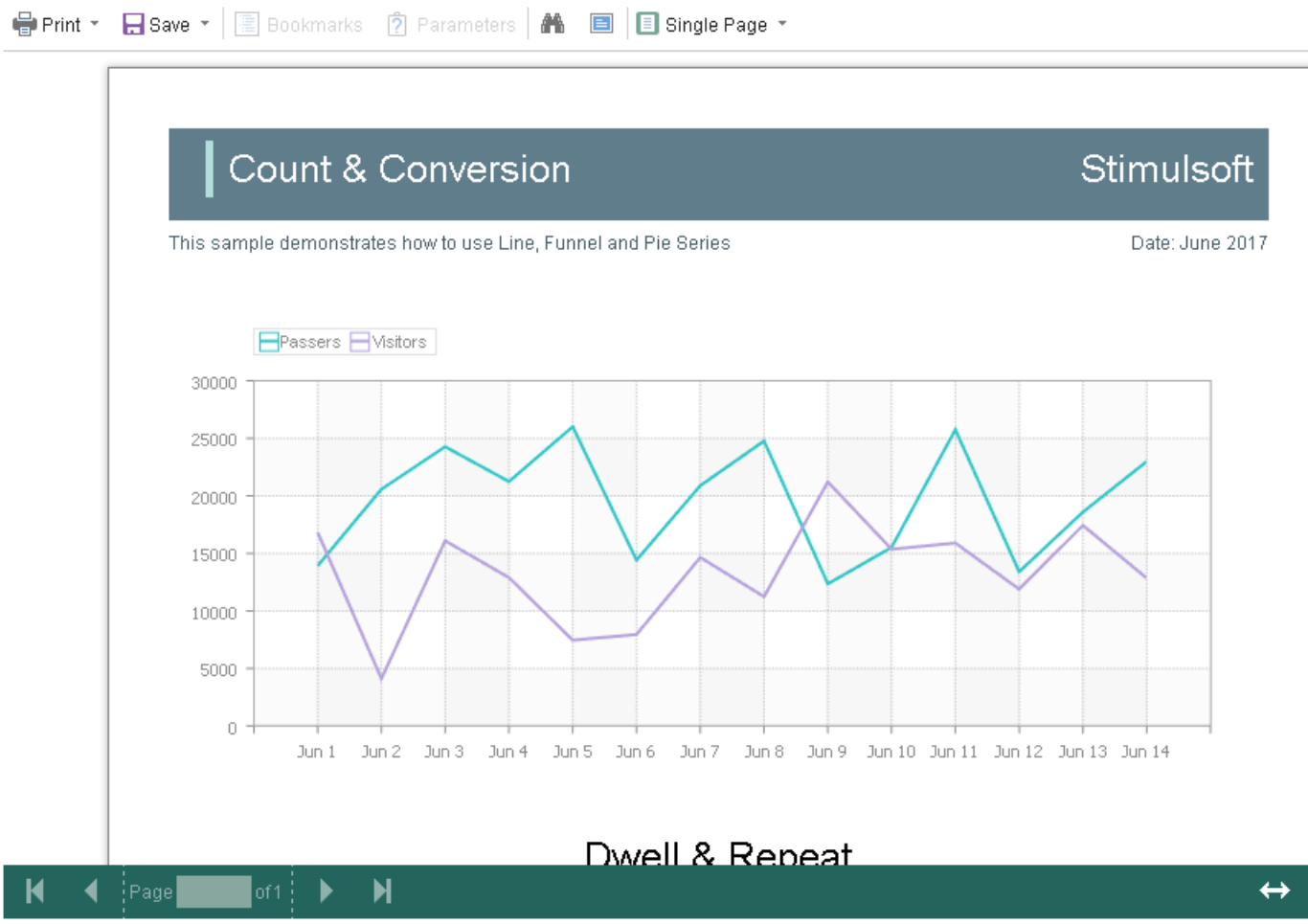


По умолчанию вьювер имеет только верхнюю панель инструментов, на которой находятся все элементы управления отчетом. При необходимости, панель инструментов можно разделить на верхнюю и нижнюю. На верхней панели будут находиться меню печати и экспорта отчета, а также кнопки работы с параметрами и закладками. Нижняя панель инструментов будет содержать элементы переключения между страницами отчета и меню управления масштабом. Для включения указанного режима предназначено свойство **displayMode**, которое может иметь значение **Simple** (стандартный простой режим) и **Separated** (раздельный режим).

© 2003-2026 Stimulsoft

viewer.html

```
...
var options = new Stimulsoft.Viewer.StiViewerOptions();
options.toolbar.displayMode =
Stimulsoft.Viewer.StiToolbarDisplayMode.Simple;
options.appearance.scrollbarsMode = true;
...
```



Дополнительно предусмотрена возможность установки параметров оформления основных элементов вьювера. Например, можно изменить шрифт и цвет надписей панели управления вьювера, установить фон вьювера, задать цвет границ страницы и др. Ниже представлен список доступных свойств, изменяющих оформление вьювера, и их значения по умолчанию.

© 2003-2026 Stimulsoft

viewer.html

```
...
var options = new Stimulsoft.Viewer.StiViewerOptions();

options.appearance.backgroundColor =
Stimulsoft.System.Drawing.Color.white;
options.appearance.pageBorderColor = Stimulsoft.System.Drawing.Color.red;
options.appearance.showPageShadow = false;

options.toolbar.backgroundColor = Stimulsoft.System.Drawing.Color.aqua;
options.toolbar.borderColor = Stimulsoft.System.Drawing.Color.darkGreen;
options.toolbar.fontColor = Stimulsoft.System.Drawing.Color.white;
options.toolbar.fontFamily = "Arial";
...
```

8.2.4 Печать отчета

Информация

Обратите внимание, что возможность печати доступна только для отчетов, и недоступна для дашбордов.

В компоненте **HTML5 Viewer** предусмотрено несколько вариантов печати отчета. Каждый имеет свои особенности, достоинства и недостатки.

Печать в PDF (Print to PDF)
Печать будет осуществляться посредством экспортирования отчета в PDF формат. К достоинствам можно отнести большую точность расположения и печати элементов отчета по сравнению с остальными вариантами печати. Из недостатков можно упомянуть обязательное наличие установленного в браузере плагина для просмотра PDF файлов (современные браузеры имеют встроенное средство просмотра и печати PDF файлов).

Информация

Браузеры Internet Explorer и Edge не поддерживают прямой вывод PDF контента из JavaSctipt кода, поэтому при печати как PDF будет предложено сохранить файл, и только затем его можно будет распечатать.

Печать с просмотром (Print with Preview)

© 2003-2026 Stimulsoft

Печать отчета будет осуществлена в отдельном всплывающем окне браузера в HTML формат. Отчет можно предварительно просмотреть, а затем отправить на принтер или скопировать в другое место в виде текста или HTML-кода. К достоинствам можно отнести кроссбраузерность при печати, отсутствие необходимости установки специальных плагинов. Недостатком является относительно невысокая точность расположения элементов отчета, обусловленная особенностями реализации HTML-форматирования.

Печать без просмотра (Print without Preview)
Печать отчета будет осуществлена непосредственно на принтер без предварительного просмотра. После выбора этого пункта меню отображается системный диалог печати. Так как печать в этом режиме осуществляется в HTML формат, то качество печати аналогично качеству печати отчета с предварительным просмотром.

Информация

Печать отчета осуществляется с использованием встроенных методов текущего браузера, поэтому представление диалогового окна может отличаться в разных операционных системах и браузерах. Также, браузер не позволяет управлять настройками печати из JavaScript кода, поэтому требуемые настройки необходимо будет выполнить в самом диалоговом окне.

Настройки печати отчета

При выборе печати отчета на панели вьювера, отображается меню с выбором варианта печати. Компонент **HTML5 Viewer** имеет возможность принудительно установить требуемый режим печати. Для этого достаточно установить свойство **printDestination** в одно из указанных ниже значений из перечисления **StiPrintDestination**:

- **Default** – при выборе печати будет отображено меню (значение свойства по умолчанию);
- **Pdf** – печать в PDF формат;
- **Direct** – печать в HTML формат непосредственно на принтер, будет отображен системный диалог печати;
- **WithPreview** – печать в HTML формат с предварительным просмотром во всплывающем окне.

© 2003-2026 Stimulsoft

viewer.html

```
...
var options = new Stimulsoft.Viewer.StiViewerOptions();
options.toolbar.printDestination =
Stimulsoft.Viewer.StiPrintDestination.Default;
...
```

Компонент **HTML5 Viewer** имеет возможность полностью отключить печать отчета, если она не требуется. Для этого необходимо установить значение **false** для свойства **showPrintButton**.

viewer.html

```
...
var options = new Stimulsoft.Viewer.StiViewerOptions();
options.toolbar.showPrintButton = false;
...
```

Печать отчета из кода

Так же доступна печать отчета при помощи кода. Для этого можно воспользоваться специальным методом **print()** у объекта отчета.

viewer.html

```
...
var report = new Stimulsoft.Report.StiReport();
report.loadFile("SimpleList.mrt");
report.renderAsync(function(){
   report.print();
});
...
```

При печати отчета есть возможность указать диапазон печати. Для этого предназначен специальный класс **StiPagesRange**, в качестве аргументов конструктора допустимо задать указанные ниже параметры.

› Тип диапазона (доступны следующие значения **Stimulsoft.Report.StiRangeType.All**, **Stimulsoft.Report.StiRangeType.Pages**, **Stimulsoft.Report.StiRangeType.CurrentPage**);

© 2003-2026 Stimulsoft

- Диапазон в строковом представлении (номера страниц, разделенные запятыми или через дефис);
- Текущий номер страницы.

viewer.html

```
...
var pageRange = new
Stimulsoft.Report.StiPagesRange(Stimulsoft.Report.StiRangeType.CurrentPage
, "1,3-8", 5);
report.print(pageRange);
...
```

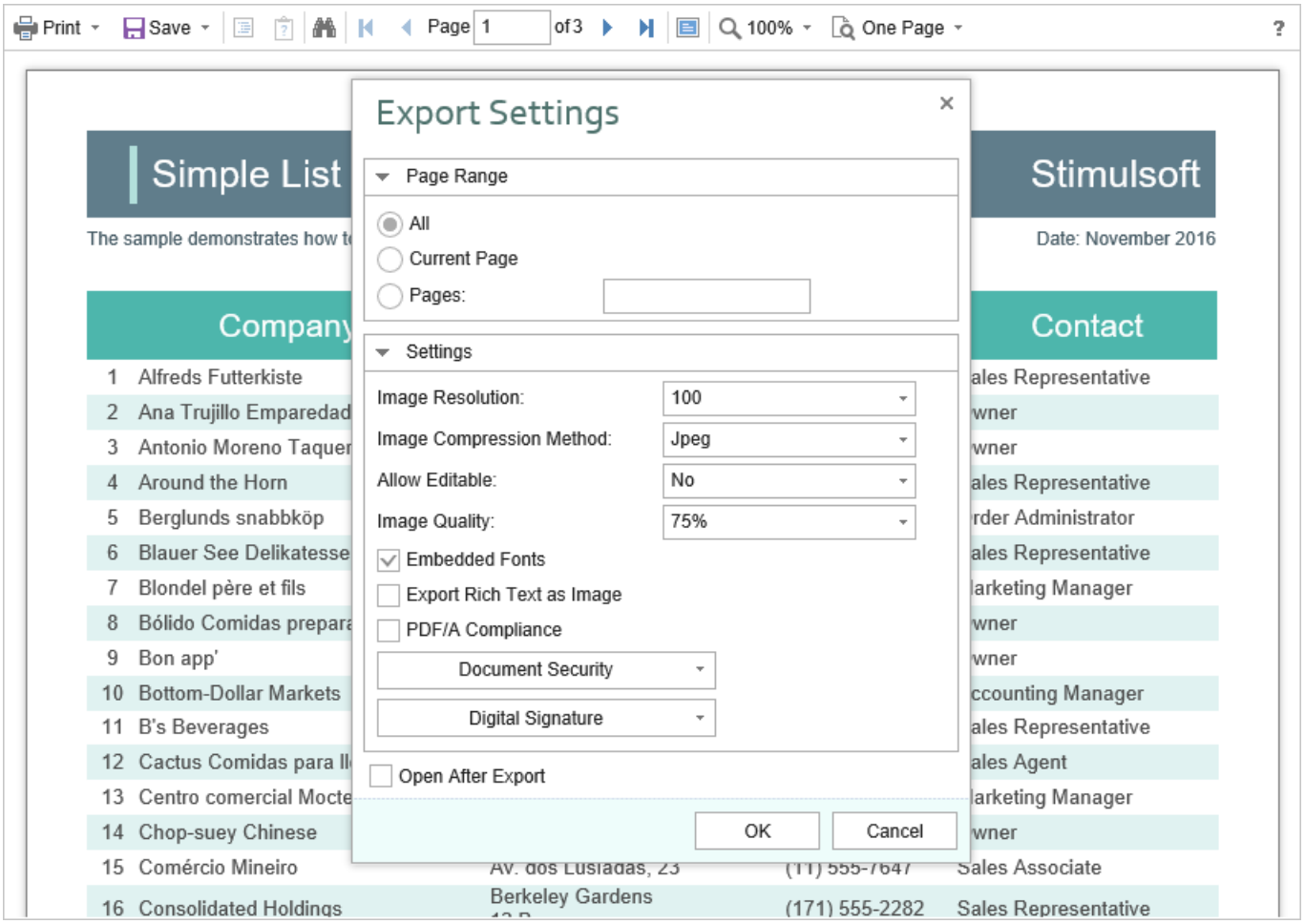
8.2.5 Экспорт отчета

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Компонент **HTML5 Viewer** позволяет экспортировать отображаемый отчет как **PDF**, **PowerPoint**, **HTML**, **Text**, **Word**, **ODT**, **Excel**, **ODC**, **Csv, Json** форматы. Для работы функции экспорта не требуется дополнительных настроек вьювера. Дашборд может быть преобразован в **PDF** и **Excel** форматы.

© 2003-2026 Stimulsoft



События экспорта

Для проведения каких-либо действий перед экспортированием отчета предназначено специальное событие **onBeginExportReport**. В данном событии можно узнать тип экспорта отчета, получить сам отчет, а также получить настройки экспорта отчета и при необходимости изменить их.

viewer.html

```
...
viewer.onBeginExportReport = function (event) {
   switch (event.format) {
      case Stimulsoft.Report.StiExportFormat.Html:
         event.settings.zoom = 2;  // Set zoom to 200%
         break;
   }
}
...
```

© 2003-2026 Stimulsoft

Более подробно работа событий рассмотрена в разделе События вьювера.

Настройки экспорта

Компонент **HTML5 Viewer** содержит 7 различных форматов экспорта, и иногда требуется отключить не используемые форматы. Это позволяет разгрузить интерфейс и упростить использование вьювера. Для отключения не используемых форматов экспортов достаточно установить значение **false** для соответствующих свойств вьювера, представленных в списке ниже.

viewer.html

```
...
var options = new Stimulsoft.Viewer.StiViewerOptions();
options.exports.showExportToDocument = false;
options.exports.showExportToPdf = false;
options.exports.showExportToHtml = false;
options.exports.showExportToHtml5 = false;
options.exports.showExportToWord = false;
options.exports.showExportToExcel = false;
options.exports.showExportToCsv = false;
...
```

Так же, если требуется, можно полностью убрать отображение диалоговых окон экспорта, экспортирование всегда будет осуществляться с настройками по умолчанию. Для этого достаточно установить значение **false** для свойства **showExportDialog**.

viewer.html

```
...
var options = new Stimulsoft.Viewer.StiViewerOptions();
options.exports.showExportDialog = false;
...
```

Экспорт отчета из кода

Так же доступен экспорт отчета при помощи кода. Для этого можно воспользоваться специальным методом **exportDocument()** у объекта отчета.

© 2003-2026 Stimulsoft

viewer.html

```
...
// Create a new report instance
var report = new Stimulsoft.Report.StiReport();
// Load report from url
report.loadFile("../reports/SimpleList.mrt");
// Render report
report.renderAsync(function(){
   report.exportDocument(Stimulsoft.Report.StiExportFormat.PdfFX LL
            mac
 FX
...
```

8.2.6 Режимы отображения

В компоненте **HTML5 Viewer** предусмотрено два режима отображения отчета: с полосами прокрутки и без них. По умолчанию установлен режим просмотра без полос прокрутки. Для включения режима просмотра с полосами прокрутки достаточно установить значение **true** для свойства **scrollbarsMode**.

viewer.html

```
...
var options = new Stimulsoft.Viewer.StiViewerOptions();
options.appearance.scrollbarsMode = true;
...
```

В первом режиме (без полос прокрутки), вьювер отображает страницу или отчет целиком, автоматически растягивая область просмотра. Если заданы размеры по ширине и высоте, то вьювер будет осуществлять обрезку вышедшей за границы страницы. Во втором режиме, в отличие от первого, при выходе страницы за границы размеров вьювера, обрезка осуществляться не будет. Вместо этого появятся полосы прокрутки, при помощи которых можно просмотреть страницу или отчет целиком.

Информация

В режиме просмотра отчета с полосами прокрутки необходимо задать высоту вьювера, иначе будет задана высота по умолчанию, равная **650 пикселей**.

© 2003-2026 Stimulsoft

В компоненте **HTML5 Viewer** предусмотрен режим полноэкранного отображения отчета и дашбоарда. По умолчанию включен стандартный режим просмотра, вьювер имеет заданные в настройках размеры. Для включения полноэкранного режима просмотра достаточно установить значение **true** для свойства **fullScreenMode**.

viewer.html

```
...
var options = new Stimulsoft.Viewer.StiViewerOptions();
options.appearance.fullScreenMode = true;
...
```

Также, для включения либо отключения полноэкранного режима можно воспользоваться соответствующей кнопкой на панели управления вьювера.

В компоненте **HTML5 Viewer** доступно три режима отображения отчета: постраничное отображение, отчет целиком в виде ленты, и табличное отображение страниц отчета. Для управления режимами предназначено свойство **ViewMode**, которое соответственно принимает одно из указанных значений: **SinglePage**, **Continuous**, **MultiplePages**.

viewer.html

```
...
var options = new Stimulsoft.Viewer.StiViewerOptions();
options.toolbar.viewMode = Stimulsoft.Viewer.StiWebViewMode.Continuous;
...
```

В компоненте **HTML5 Viewer** реализована поддержка работы как с обычным компьютером, так и с сенсорными экранами, мобильными устройствами. Для управления режимами интерфейса предназначено свойство **InterfaceType**, которое принимает одно из следующих значений:

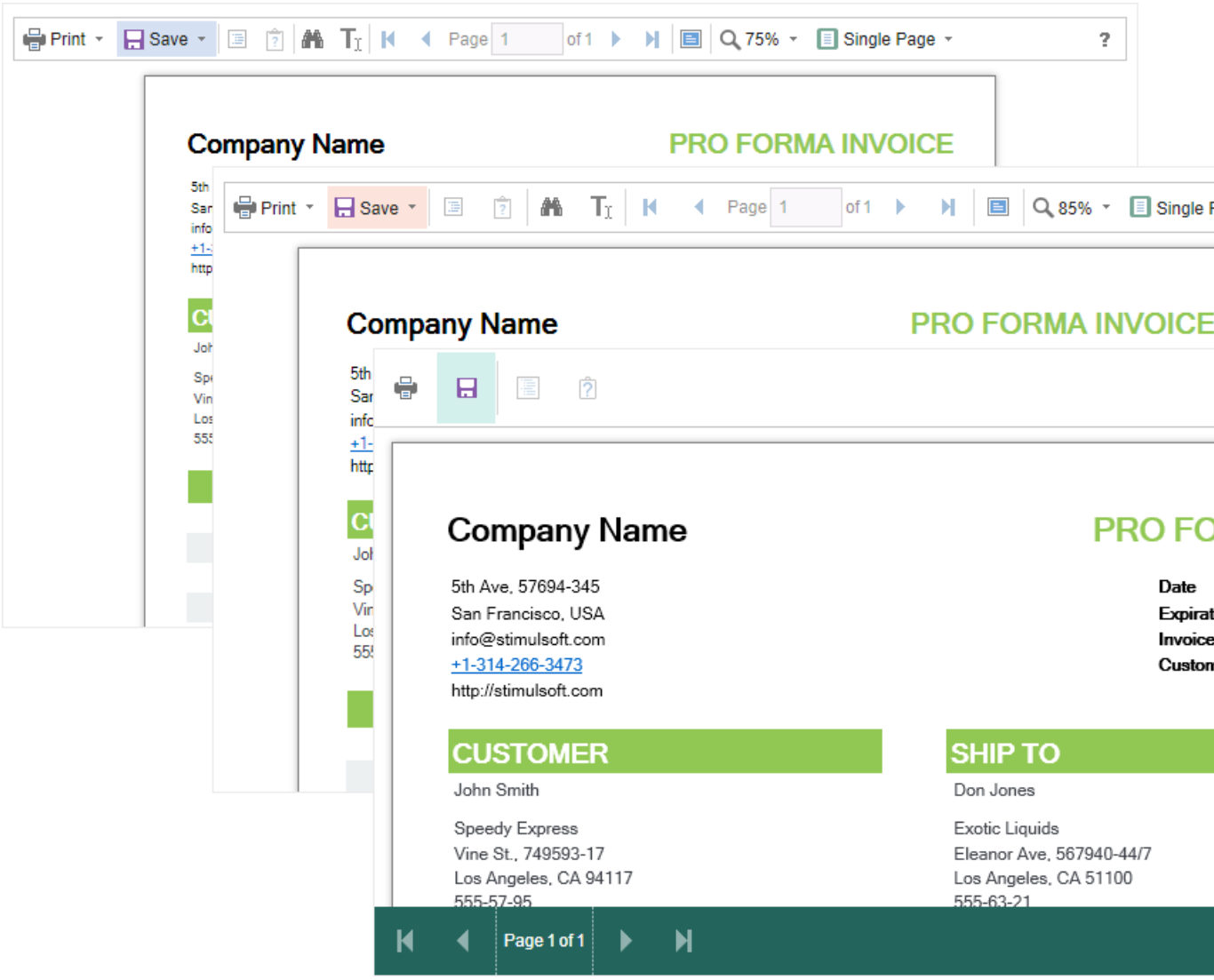
- **Auto** – тип интерфейса вьювера будет выбран автоматически в зависимости от используемого устройства (значение по умолчанию);
- **Mouse** – принудительное использование стандартного интерфейса для управления вьювером при помощи мыши;
- **Touch** – принудительное использование Touch интерфейса для управления вьювером при помощи сенсорного экрана монитора, в этом режиме элементы интерфейса вьювера имеют увеличенные размеры для удобства управления;

© 2003-2026 Stimulsoft

› **Mobile** – принудительное использование Mobile интерфейса для управления вьювером при помощи экрана смартфона, в этом режиме интерфейс вьювера имеет упрощенный вид и адаптирован для управления при помощи мобильного устройства.

viewer.html

```
...
var options = new Stimulsoft.Viewer.StiViewerOptions();
options.appearance.interfaceType =
Stimulsoft.Viewer.StiInterfaceType.Auto;
...
```



© 2003-2026 Stimulsoft

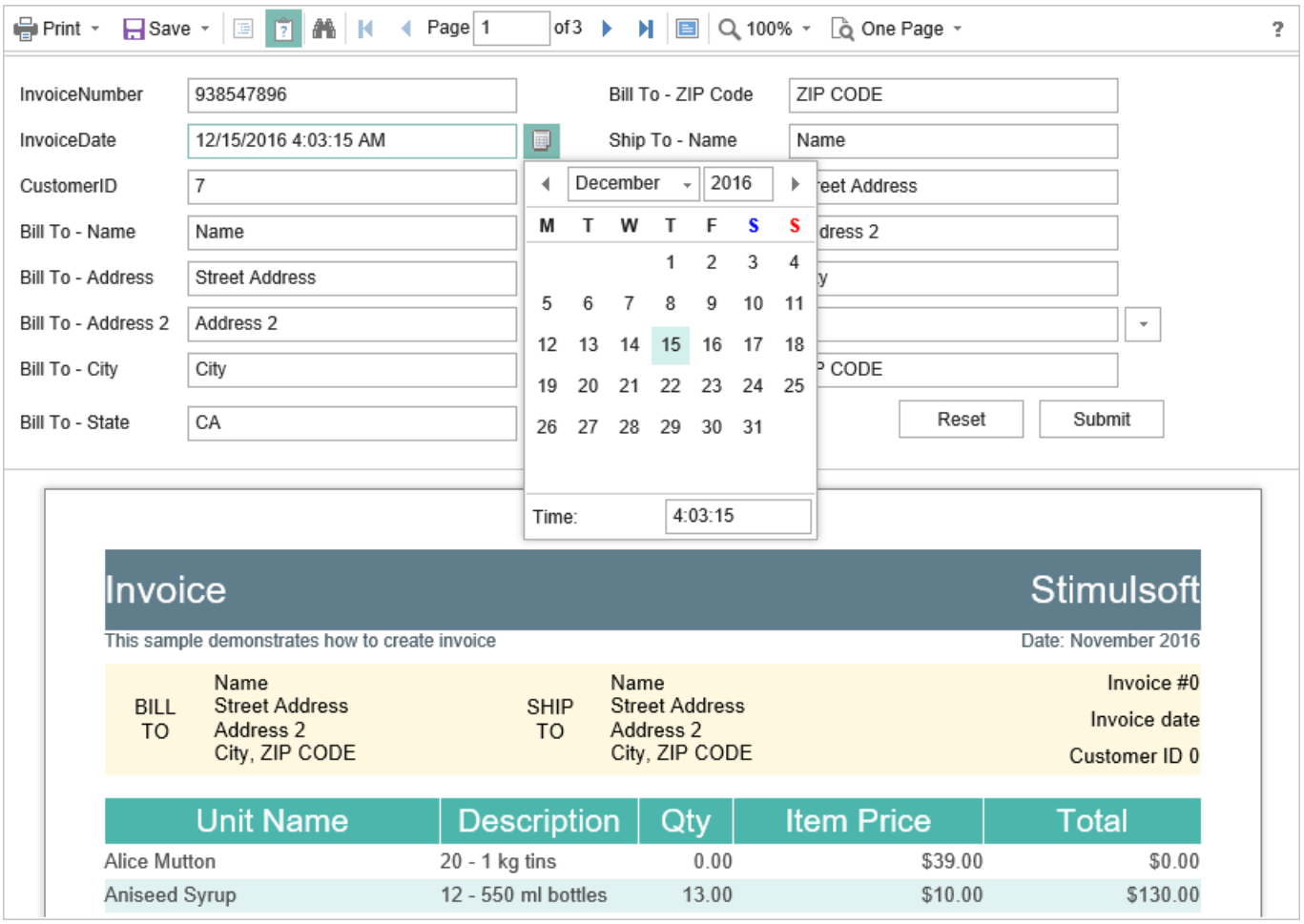
8.2.7 Работа с параметрами

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для работы с параметрами отчета в **HTML5 Viewer** реализована поддержка специальной панели параметров. Для добавления параметра на панель необходимо в отчете определить переменную, запрашиваемую у пользователя. При просмотре отчета во вьювере такая переменная будет автоматически добавлена на панель параметров. Поддерживаются все типы переменных отчета (обычные переменные, дата и время, границы, списки и др.).

© 2003-2026 Stimulsoft



Для выполнения каких-либо действий перед применением параметров предназначено специальное событие **onInteraction**, которое будет вызвано при интерактивных действиях вьювера. При использовании панели параметров, тип действия будет иметь значение **Variables**.

viewer.html

```
...
viewer.onInteraction = function (args) {
   if (args.action == "Variables") {
      var variables = args.variables;
   }
}
...
```

Если работа с параметрами не требуется, то можно полностью отключить

© 2003-2026 Stimulsoft

данную возможность. Для этого предназначено свойство **showParametersButton**, которому необходимо установить значение **false**.

viewer.html

```
...
var options = new Stimulsoft.Viewer.StiViewerOptions();
options.toolbar.showParametersButton = false;
...
```

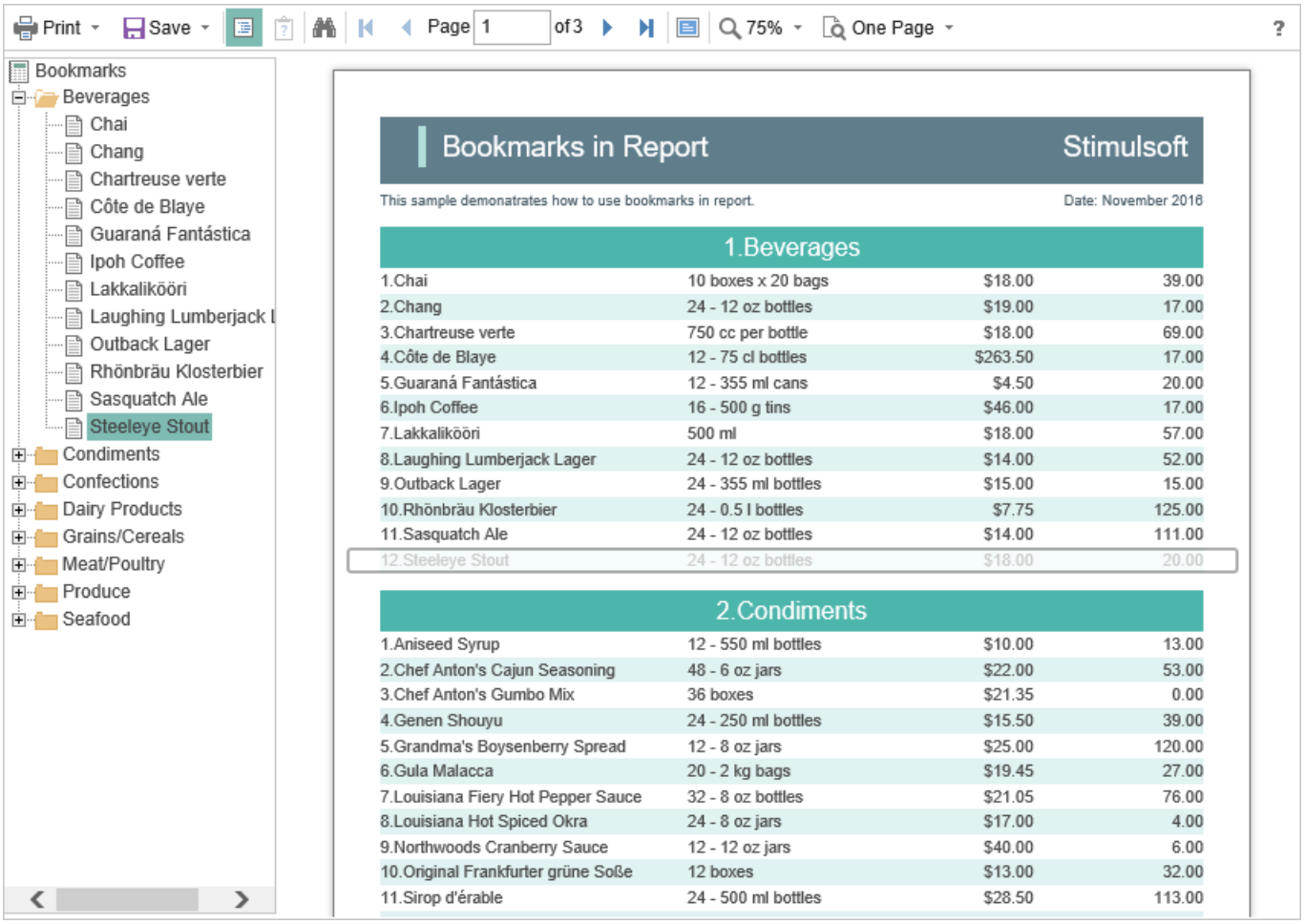
Информация

При такой конфигурации вьювера панель параметров не будет показана даже в том случае, если параметры присутствуют в отображаемом отчете.

8.2.8 Работа с закладками

В компоненте **HTML5 Viewer** реализована поддержка закладок отчета. При отображении такого отчета слева от страницы будет отображена панель с закладками. При выборе закладки отчета вьювер осуществит автоматический переход на нужную страницу, а элемент отчета с закладкой будет подсвечен.

© 2003-2026 Stimulsoft



По умолчанию ширина панели закладок равна 180 пикселей, компонент **HTML5 Viewer** позволяет изменить это значение. Для этого предназначено свойство **bookmarksTreeWidth**, значение которого указывается в пикселях.

viewer.html

```
...
var options = new Stimulsoft.Viewer.StiViewerOptions();
options.appearance.bookmarksTreeWidth = 200;
...
```

Если работа с закладками отчета не требуется, то можно полностью отключить данную возможность. Для этого предназначено свойство **showBookmarksButton**, которое необходимо установить в значение **false**.

© 2003-2026 Stimulsoft

viewer.html

```
...
var options = new Stimulsoft.Viewer.StiViewerOptions();
options.toolbar.showBookmarksButton = true;
...
```

Информация

В этом случае закладки отчета не будут показаны, даже если они присутствуют в отображаемом отчете. Данное свойство не оказывает влияние на печать и экспортирование отчета.

При печати отчета с закладками дерево закладок будет скрыто. Если помимо самого отчета требуется распечатать и закладки, то необходимо установить свойство **bookmarksPrint** в значение **true**.

viewer.html

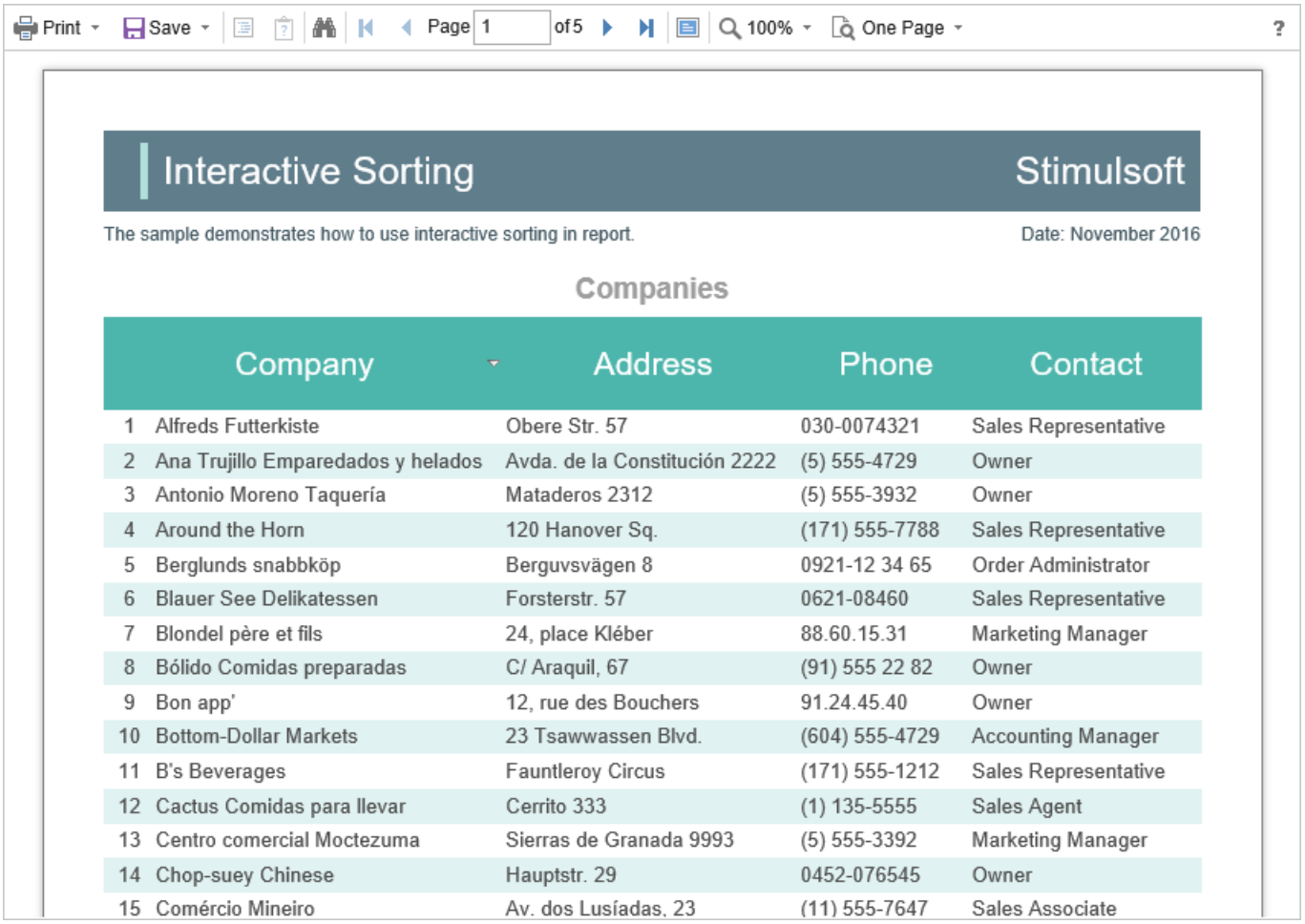
```
...
var options = new Stimulsoft.Viewer.StiViewerOptions();
options.appearance.bookmarksPrint = true;
...
```

8.2.9 Динамическое сворачивание, сортировка и детализация

В компоненте **HTML5 Viewer** реализована поддержка динамической сортировки, сворачивания и детализации отчетов. Динамическая сортировка предоставляет возможность изменять направление сортировки в построенном отчете. Для этого следует щелкнуть по компоненту, у которого была установлена динамическая сортировка. Динамическая сортировка осуществляется в следующих направлениях: **По возрастанию (Ascending)** и **По убыванию (Descending)**. Каждый раз при щелчке по компоненту направление меняется на противоположное.

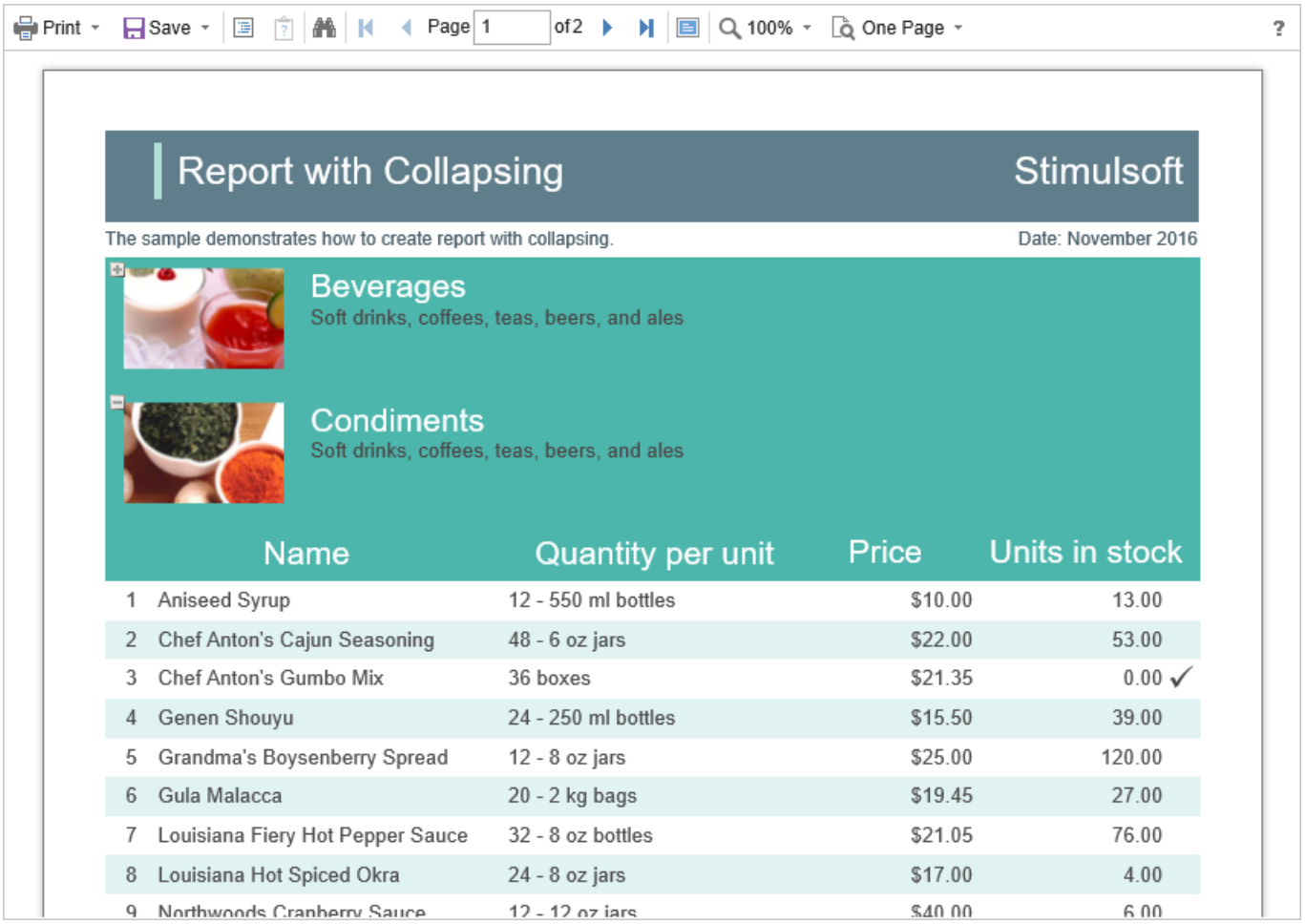
Допускается многоуровневая сортировка в отчете. Для этого необходимо удерживать клавишу **Ctrl** и последовательно нажимать на сортируемые компоненты отчета. Для сброса сортировки можно нажать на любой сортируемый компонент без удержания клавиши **Ctrl**.

© 2003-2026 Stimulsoft



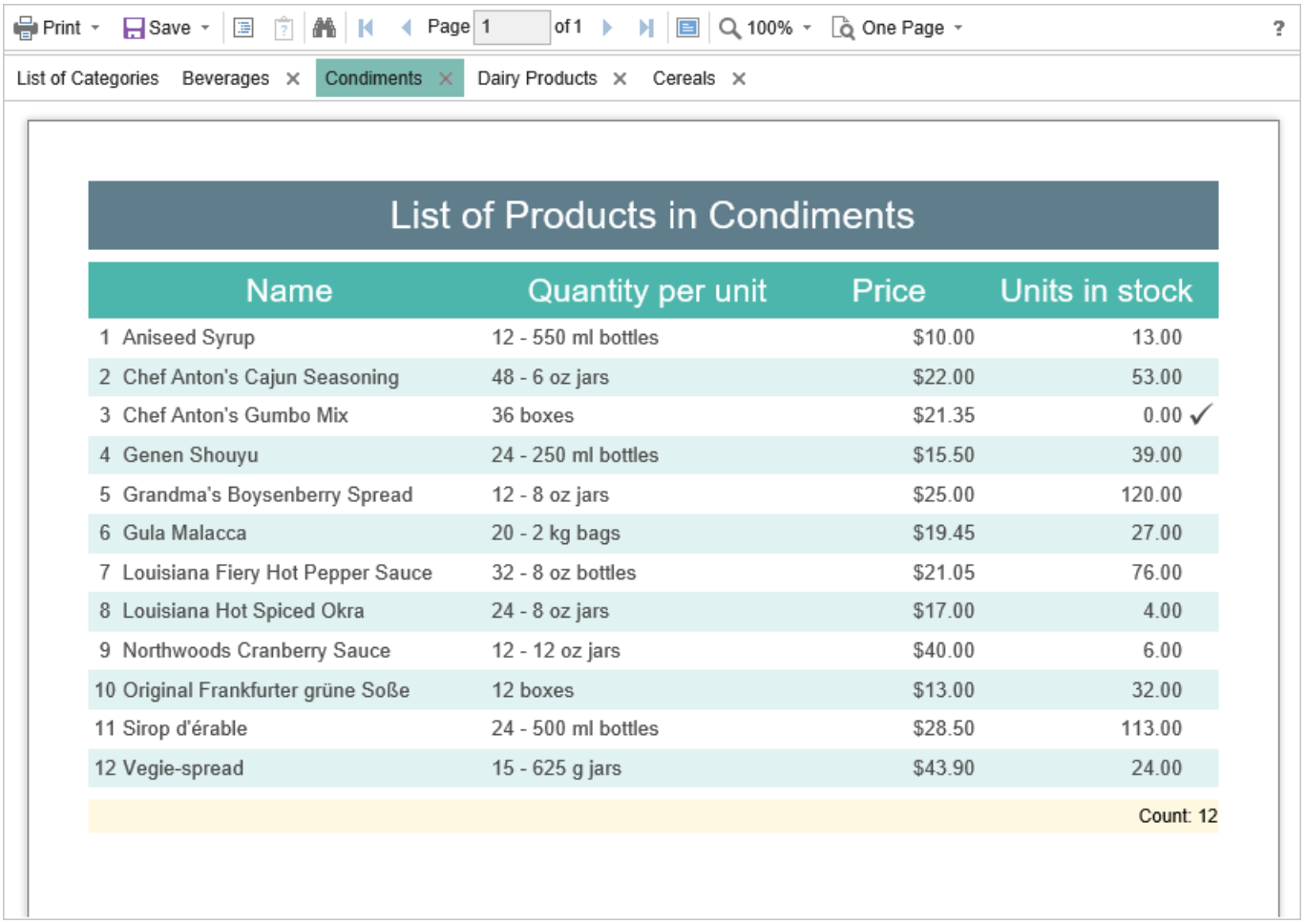
Отчет с динамическим сворачиванием представляет собой интерактивный отчет, в котором сворачивающиеся блоки могут сворачивать/разворачивать свое содержимое при нажатии на заголовок блока. Элементы отчета, которые можно свернуть/развернуть, обозначены специальной иконкой со знаком **[-]** или **[+]**.

© 2003-2026 Stimulsoft



При детализации данных под главной панелью вьювера будет отображена панель детализации с закладками детализированных отчетов. Отображенный в данный момент отчет будет подсвечен.

© 2003-2026 Stimulsoft



Для работы динамической сортировки, сворачивания и детализации отчетов не требуется дополнительных настроек вьювера. Для выполнения каких-либо действий перед сортировкой, сворачиванием или детализацией отчета предназначено специальное событие **onInteraction**, которое будет вызвано при интерактивных действиях вьювера. Для каждого вида интерактивности вьювера предусмотрен определенный тип действия:

- **Sorting** – при использовании сортировки колонок;
- **DrillDown** – при использовании детализации отчета;
- **Collapsing** – при использовании сворачивания блоков отчета.

© 2003-2026 Stimulsoft

viewer.html

```
...
viewer.onInteraction = function (args) {
   switch (args.action) {
      case "Sorting":
        break;

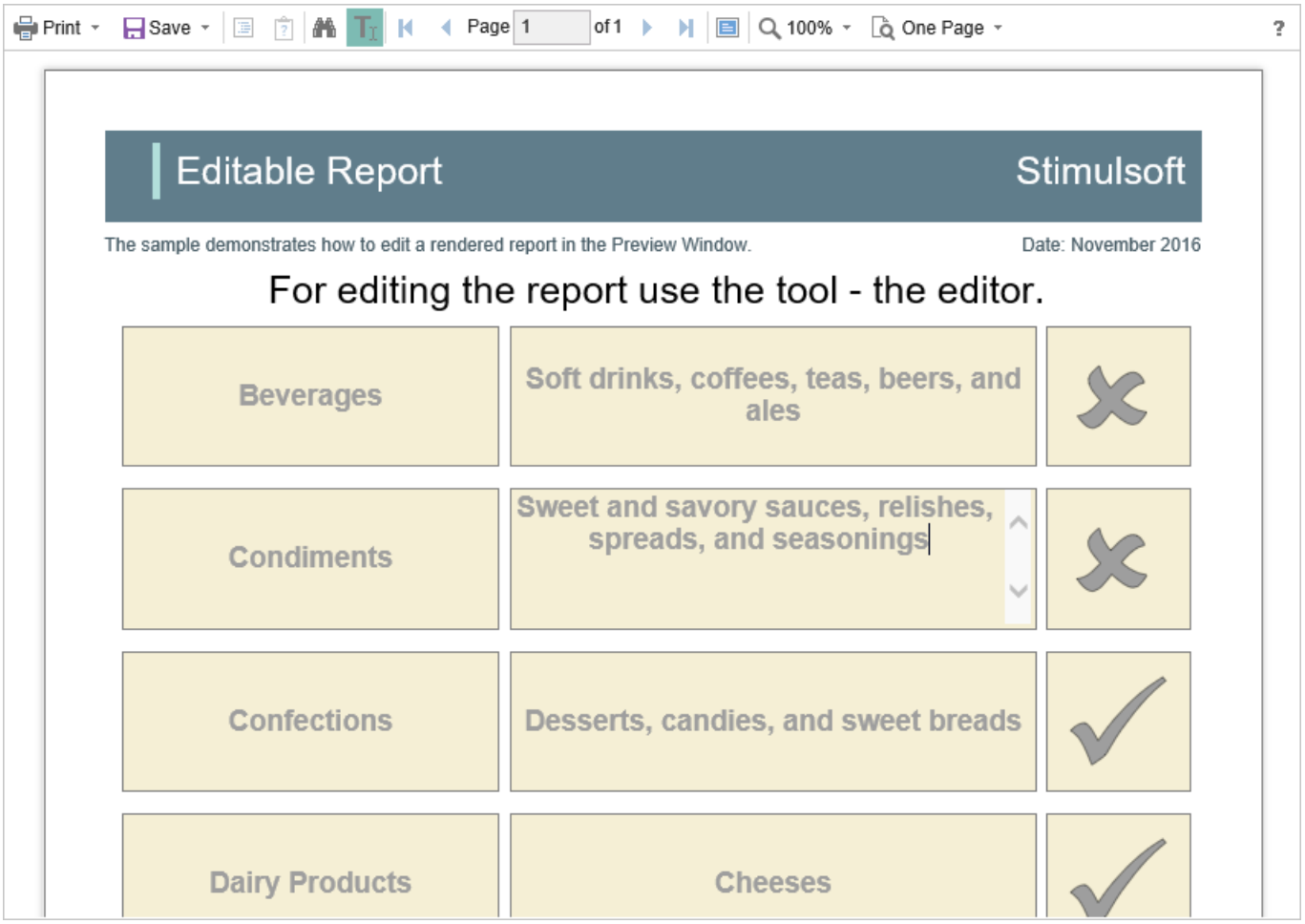
      case "DrillDown":
        break;

      case "Collapsing":
        break;
   }
}
...
```

8.2.10 Редактирование построенного отчета

Компонент **HTML5 Viewer** имеет возможность редактировать элементы построенного отчета, такие как текстовые поля и чекбоксы. Для того, чтобы редактирование стало возможным, в самом шаблоне отчета необходимо пометить нужные компоненты как редактируемые. После отображения отчета во вьювере, для начала редактирования необходимо нажать на соответствующую кнопку на панели вьювера. После завершения редактирования необходимо нажать указанную кнопку еще раз, и все внесенные изменения будут примерены к отчету.

© 2003-2026 Stimulsoft



Для работы режима редактирования отчета не требуется никаких специальных настроек вьювера.

Информация

Отредактированные значения будут применены при печати либо экспортировании отчета, при этом исходный отчет останется не тронутым. После перезагрузки вьювера все значения будут возвращены к исходным.

© 2003-2026 Stimulsoft

8.2.11 Отправка отчета по Email

Информация

Обратите внимание, что возможность **Отправить отчет по Email** доступна только для отчетов, и недоступна для дашбордов.

В компоненте **HTML5 Viewer** предусмотрена возможность отправки отчета по Email. Для активации данной возможности необходимо установить свойство вьювера **showSendEmailButton** в значение **true** и добавить обработчик события **onEmailReport**.

viewer.html

```
...
var options = new Stimulsoft.Viewer.StiViewerOptions();
options.toolbar.showSendEmailButton = true;
...
```

viewer.html

```
...
viewer.onEmailReport = function (args) {
// args.settings -  данные формы отправки Email
// args.settings.email  -  адрес почты
// args.settings.subject  -  тема письма
// args.settings.message  -  текст письма

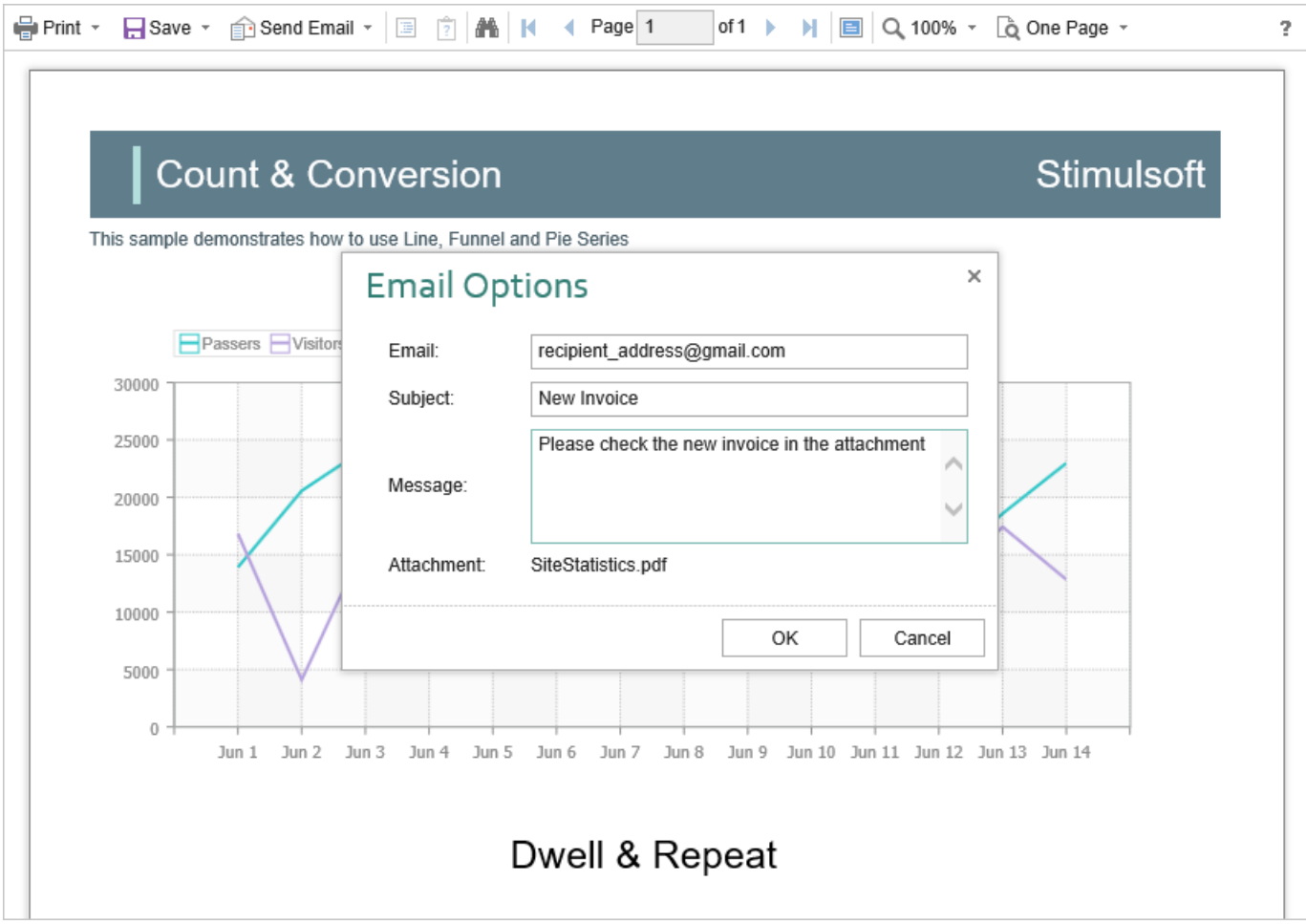
// args.format  -  формат экспорта - Pdf, Html, Html5, Excel2007, Word,
Csv
// args.fileName - имя файла отчета (имя вложения)
// args.data  -  байтовый массив с экспортированным файлом отчета
}
...
```

В событии **onEmailReport** можно узнать необходимые данные для отправки письма, получить тип экспорта отчета, получить сам отчет, а также получить настройки экспорта отчета и при необходимости изменить их.

При отправке отчета по Email отображается меню выбора формата вложения, которое соответствует меню выбора формата экспортирования отчета. После выбора формата отобразиться диалог ввода параметров отправки, таких как

© 2003-2026 Stimulsoft

Email получателя, тема и текст письма.



После подтверждения отправки будет вызвано описанное выше событие **onEmailReport**, в котором можно проверить и скорректировать данные, введенные в этой форме.

© 2003-2026 Stimulsoft

Информация

Чистый JavaScript не имеет встроенных методов работы с электронной почтой, поэтому компонент **StiViewer** предоставляет только интерфейс отправки письма. Для работы с почтой необходимо использовать какой-либо сервер, поддерживающий работу с электронной почтой. На сторону сервера данные для отправки письма можно отправить при помощи AJAX запроса или другим удобным способом.

Компонент **HTML5 Viewer** позволяет установить значения по умолчанию для формы отправки Email. Для этого предназначены свойства **defaultEmailAddress**, **defaultEmailSubject** и **defaultEmailMessage**. По умолчанию данные свойства являются пустыми.

viewer.html

```
...
var options = new Stimulsoft.Viewer.StiViewerOptions();
options.toolbar.showSendEmailButton = true;

options.email.defaultEmailAddress = "recipient_address@gmail.com";
options.email.defaultEmailSubject = "New Invoice";
options.email.defaultEmailMessage = "Please check the new invoice in the
attachment";
...
```

8.2.12 Вызов дизайнера из вьювера

Компонент **HTML5 Viewer** имеет возможность вызвать дизайнер отчетов. Для этого предназначена специальная кнопка **Дизайн (Design)** на панели инструментов вьювера (по умолчанию данная кнопка отключена). Для использования этой возможности необходимо установить свойство **showDesignButton** в значение **true**, а также определить обработчик события **onDesignReport**.

viewer.html

```
...
var options = new Stimulsoft.Viewer.StiViewerOptions();
options.toolbar.showDesignButton = true;
...
```

© 2003-2026 Stimulsoft

viewer.html

```
...
viewer.onDesignReport = function (args) {
window.open("https://www.stimulsoft.com?reportName=" +
args.report.reportName);
}
...
```

Информация

Сам вьювер не запускает дизайнер, он лишь вызывает указанное событие и передает в качестве аргументов просматриваемый отчет. В событии можно выполнить перенаправление на HTML страницу, на которой расположен дизайнер отчетов.

8.2.13 События вьювера

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Компонент **HTML5 Viewer** поддерживает события, которые предоставляют возможность выполнить необходимые операции перед определенными действиями. Ниже представлены события вьювера с описанием и некоторыми примерами их использования:

onPrepareVariables

Асинхронное событие вызывается в начале построения отчета перед заполнением переменных в отчете. Событие возникает сразу после `onPrepareVariables` в `StiReport`. В таблице ниже представлен список аргументов обработчика событий:

Наименование	Описание

© 2003-2026 Stimulsoft

sender	Идентификатор компонента, который инициировал данное событие.
event	Идентификатор текущего события, имеет значение **PrepareVariables**.
report	Объект отчета **StiReport**.
preventDefault	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события. По умолчанию установлено значение **false**.
async	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события до выполнения callback функции. По умолчанию установлено значение **false**.

viewer.html

```
...
viewer.onPrepareVariables = (args, callback) => {
   args.variables[0].value = "Replace value";
}
...
```

onBeginProcessData

Асинхронное событие вызывается перед запросом данных, необходимых для построения отчета. Событие возникает сразу после `onBeginProcessData` в `StiReport`. В таблице ниже представлен список аргументов обработчика событий:

Наименование	Описание
sender	Идентификатор компонента, который инициировал данное событие.
event	Идентификатор текущего события, имеет значение **BeginProcessData**.
report	Объект отчета **StiReport**.

© 2003-2026 Stimulsoft

preventDefault	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события. По умолчанию установлено значение **false**.
async	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события до выполнения callback функции. По умолчанию установлено значение **false**.
command	Идентификатор текущей команды, может иметь значения **TestConnection** (проверка соединения) и **ExecuteQuery** (получения данных из хранилища).
database	Предоставляет возможность указать наименование текущей базы данных.
connection	Имя текущего подключения к источнику данных, заданное в шаблоне отчета.
headers	Предоставляет возможность указать заголовки в запросах.
withCredentials	Предоставляет возможность указать cookies в запросах.
connectionString	Предоставляет возможность указать строку подключения к хранилищу данных.
dataSource	Предоставляет возможность указать наименование источника данных.
queryString	SQL запрос на получение данных. Используется только с командой **ExecuteQuery**.
timeout	Предоставляет возможность указать таймаут запроса.
parameters	Предоставляет возможность указать параметр в запросе.
escapeQueryParameters	Предоставляет возможность использовать экранированную часть запроса. По умолчанию установлено значение **true**.

© 2003-2026 Stimulsoft

pathData	Предоставляет возможность указать путь к файлу данных.
tryParseDateTime	Предоставляет возможность определить возможность преобразования данных в тип DateTime.
relationDirection	Предоставляет возможность изменить направление связи между источниками данных.
pathSchema	Предоставляет возможность указать путь к XSD файлу со схемой данных.
firstRowIsHeader	Предоставляет возможность использовать первую строку как заголовки данных в Excel источниках данных.
collectionName	Наименование коллекции в OData источнике данных.
separator	Предоставляет возможность указать разделитель в CSV источнике данных.
dataType	Предоставляет возможность указать тип данных для GIS источника данных.
codePage	Предоставляет возможность определить кодировку данных в CSV и DBF источниках данных.

© 2003-2026 Stimulsoft

viewer.html

```
...
//Replace connection string
viewer.onBeginProcessData = (args) => {
   if (args.database == "MySQL")
      args.connectionString = "new connection string";
}
...

//Add a some data
viewer.onBeginProcessData = (args, callback) => {
   if (args.database == "MySQL"){
      args.preventDefault = true;
      var result = {
         success: true,
         rows: [
            ["value1", 1, false],
            ["value2", 1, true],
            ["value3", 2, false]
         ],
         columns: [
            "Column1_name",
            "Column2_name",
            "Column3_name"
         ],
         types:[
            "string",
            "int",
            "boolean"
         ]
      }

      // https://github.com/stimulsoft/DataAdapters.JS/
      callback(result);
   }
}
...
```

onEndProcessData

Событие вызывается после получения данных, необходимых для построения отчета. Событие возникает сразу после `onEndProcessData` в `StiReport`. В таблице ниже представлен список аргументов обработчика событий:

Наименование	Описание
sender	Идентификатор компонента, который инициировал данное событие.

© 2003-2026 Stimulsoft

event	Идентификатор текущего события, имеет значение **EndProcessData**.
report	Объект отчета **StiReport**.
command	Предоставляет возможность указать тип команды.
dataSource	Предоставляет возможность указать наименование источника данных.
connection	Имя текущего подключения к источнику данных, заданное в шаблоне отчета.
database	Предоставляет возможность указать наименование текущей базы данных.
result	Результирующий набор данных в определенном JSON формате. Он имеет две коллекции - колонки и строки с их значениями.

viewer.html

```
...
viewer.onEndProcessData = (args) => {
   if (args.command == "ExecuteQuery" && args.dataSource == "Categories")
      args.result.rows.push(rowData) ;
      // https://github.com/stimulsoft/DataAdapters.JS/
}
...
```

onPrintReport

Асинхронное событие вызывается перед печатью отчета. Это событие не актуально для дашбордов. В таблице ниже представлен список аргументов обработчика событий:

Наименование	Описание
sender	Идентификатор компонента, который инициировал данное событие.
event	Идентификатор текущего события, имеет значение **PrintReport**.

© 2003-2026 Stimulsoft

report	Объект отчета **StiReport**.
preventDefault	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события. По умолчанию установлено значение **false**.
async	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события до выполнения callback функции. По умолчанию установлено значение **false**.
printAction	Предоставляет возможность определить метод печати.

viewer.html

```
...
//Remove image before report print
viewer.onPrintReport = (args) => {
   var page = args.report.renderedPages.getByIndex(0);
   var image = page.components.getByName("Image1");
   if (image)
      page.components.remove(image);
}
...
```

onBeginExportReport

Асинхронное событие вызывается перед экспортом отчета, но после установки настроек экспорта, заданных в диалоговом окне. Предоставляет возможность скорректировать настройки экспорта. В таблице ниже представлен список аргументов обработчика событий:

Наименование	Описание
sender	Идентификатор компонента, который инициировал данное событие.
event	Идентификатор текущего события, имеет значение **BeginExportReport**.
report	Объект отчета **StiReport**.

© 2003-2026 Stimulsoft

preventDefault	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события. По умолчанию установлено значение **false**.
async	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события до выполнения callback функции. По умолчанию установлено значение **false**.
action	Этот идентификатор предоставляет возможность определить **StiExportAction**.
settings	Настройки экспорта отчета или дашборда.
format	Формат, в который будет произведен экспорт: › Stimulsoft.Report.StiExportFormat.Pdf, › Stimulsoft.Report.StiExportFormat.Excel2007, › Stimulsoft.Report.StiExportFormat.Word, › Stimulsoft.Report.StiExportFormat.Html, › Stimulsoft.Report.StiExportFormat.Html5, › Stimulsoft.Report.StiExportFormat.Document. › Stimulsoft.Report.StiExportFormat.Text, › Stimulsoft.Report.StiExportFormat.Csv, › Stimulsoft.Report.StiExportFormat.ImageSvg, › Stimulsoft.Report.StiExportFormat.PowerPoint, › Stimulsoft.Report.StiExportFormat.Odt, › Stimulsoft.Report.StiExportFormat.Ods.
formatName	Название выбранного формата экспорта отчета.
fileName	Наименование файла отчета, которое будет использовано после экспорта.
openAfterExport	Флаг указывает на то, что экспортирование отчета будет происходить в новой вкладке браузера (значение `true`), либо по завершению экспорта будет вызвано сохранение файла (значение `false`).

© 2003-2026 Stimulsoft

viewer.html

```
...
viewer.onBeginExportReport = function (args) {
   switch (event.format) {
      case Stimulsoft.Report.StiExportFormat.Html:
      args.settings.zoom = 2;  // Set zoom to 200%
      break;
   }
   console.log("exporting");
}
...
```

onEndExportReport

Асинхронное событие вызывается после экспорта отчета, но перед сохранением файла. В таблице ниже представлен список аргументов обработчика событий:

Наименование	Описание
sender	Идентификатор компонента, который инициировал данное событие.
event	Идентификатор текущего события, имеет значение **EndExportReport**.
fileName	Наименование сэкспортированного файла.
report	Объект отчета **StiReport**.
preventDefault	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события. По умолчанию установлено значение **false**.
async	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события до выполнения callback функции. По умолчанию установлено значение **false**.
action	Этот идентификатор предоставляет возможность определить **StiExportAction**.
format	Формат, в который будет произведен

© 2003-2026 Stimulsoft

	экспорт: › Stimulsoft.Report.StiExportFormat.Pdf, › Stimulsoft.Report.StiExportFormat.Excel2007, › Stimulsoft.Report.StiExportFormat.Word, › Stimulsoft.Report.StiExportFormat.Html, › Stimulsoft.Report.StiExportFormat.Html5, › Stimulsoft.Report.StiExportFormat.Document. › Stimulsoft.Report.StiExportFormat.Text, › Stimulsoft.Report.StiExportFormat.Csv, › Stimulsoft.Report.StiExportFormat.ImageSvg, › Stimulsoft.Report.StiExportFormat.PowerPoint, › Stimulsoft.Report.StiExportFormat.Odt, › Stimulsoft.Report.StiExportFormat.Ods.
formatName	Название выбранного формата экспорта отчета.
fileName	Наименование файла отчета, которое будет использовано после экспорта.
openAfterExport	Флаг указывает на то, что экспортирование отчета будет происходить в новой вкладке браузера (значение `true`), либо по завершению экспорта будет вызвано сохранение файла (значение `false`).
data	Экспортированные данные в виде строки или байтового массива.

viewer.html

```
...
viewer.onEndExportReport = (args) => {
   args.fileName = "SampleFileName.txt";
}
...
```

onInteraction

Асинхронное событие вызывается перед выполнением интерактивности. В таблице ниже представлен список аргументов обработчика событий:

© 2003-2026 Stimulsoft

Наименование	Описание
sender	Идентификатор компонента, который инициировал данное событие.
event	Идентификатор текущего события, имеет значение **Interaction**.
report	Объект отчета **StiReport**.
preventDefault	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события. По умолчанию установлено значение **false**.
async	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события до выполнения callback функции. По умолчанию установлено значение **false**.
action	Идентификатор текущего интерактивного действия, может принимать следующие значения: • `Sorting` - This action happens when using sorting columns; • `DrillDown` - This action happens when using sorting columns; • `Collapsing` - This action happens when using of collapsing report blocks.

viewer.html

```
...
viewer.onInteraction = (args) => {
   if (args.action == "Variables")
      args.variables["Variable1"] = "New Value";
}
...
```

onEmailReport

© 2003-2026 Stimulsoft

Событие вызывается перед отправкой отчета по Email. Это событие не актуально для дашбордов. В таблице ниже представлен список аргументов обработчика событий:

Наименование	Описание
sender	Идентификатор компонента, который инициировал данное событие.
event	Идентификатор текущего события, имеет значение **EmailReport**.
async	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события до выполнения callback функции. По умолчанию установлено значение **false**.
report	Объект отчета **StiReport**.
settings	Предоставляет возможность указать настройки email как объект: › email - адрес получателя; › subject - тема письма; › message - сообщение письма.
format	Формат, в который будет произведен экспорт: › Stimulsoft.Report.StiExportFormat.Pdf, › Stimulsoft.Report.StiExportFormat.Excel2007, › Stimulsoft.Report.StiExportFormat.Word, › Stimulsoft.Report.StiExportFormat.Html, › Stimulsoft.Report.StiExportFormat.Html5, › Stimulsoft.Report.StiExportFormat.Document. › Stimulsoft.Report.StiExportFormat.Text, › Stimulsoft.Report.StiExportFormat.Csv, › Stimulsoft.Report.StiExportFormat.ImageSvg, › Stimulsoft.Report.StiExportFormat.PowerPoint,

© 2003-2026 Stimulsoft

	Stimulsoft.Report.StiExportFormat.Odt, Stimulsoft.Report.StiExportFormat.Ods.
formatName	Название выбранного формата экспорта отчета.
fileName	Наименование файла отчета, которое будет использовано после экспорта.
data	Экспортированные данные в виде строки или байтового массива.

viewer.html

```
...
viewer.onEmailReport = (args, callback) => {
   args.async = true;

   var emailAddress = args.settings.email;
   var emailMessage = args.settings.message;
   var emailSubject = args.settings.subject;
   var emailAttachmentFileName = args.fileName;
   var emailAttachment = args.data;
   sendEmail(emailAddress, emailMessage, emailSubject,
   emailAttachmentFileName, emailAttachment);

   setTimeout(() => {
      callback();
   }, 5000);
}
...
```

onDesignReport

Событие вызывается при нажатии кнопки **Design**. В таблице ниже представлен список аргументов обработчика событий:

Наименование	Описание
sender	Идентификатор компонента, который инициировал данное событие.
event	Идентификатор текущего события, имеет значение **DesignReport**.
report	Объект отчета, просматриваемый во вьювере в данный момент.

© 2003-2026 Stimulsoft

viewer.html

```
...
var viewerOptions = new Stimulsoft.Viewer.StiViewerOptions();
viewerOptions.toolbar.showDesignButton = true;
var viewer = new Stimulsoft.Viewer.StiViewer(viewerOptions, "StiViewer",
false);
viewer.renderHtml("content");

viewer.onDesignReport = (args) => {
   var viewerDiv = document.getElementById("content");
   viewerDiv.innerHTML = "";
   var designerOptions = new Stimulsoft.Designer.StiDesignerOptions();
   designerOptions.appearance.fullScreenMode = true;
   var designer = new Stimulsoft.Designer.StiDesigner(designerOptions,
   "StiDesigner", false);
   designer.renderHtml("content");

   designer.report = args.report;
}
...
```

onShowReport

Асинхронное событие вызывается после построения отчета перед его отображением во вьювере. В таблице ниже представлен список аргументов обработчика событий:

Наименование	Описание
sender	Идентификатор компонента, который инициировал данное событие.
event	Идентификатор текущего события, имеет значение **ShowReport**.
report	Объект отчета **StiReport**.
preventDefault	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события. По умолчанию установлено значение **false**.
async	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события до выполнения callback

© 2003-2026 Stimulsoft

	функции. По умолчанию установлено значение **false**.

viewer.html

```
...
viewer.onShowReport = function (args) {
   console.log("showing");
}
...
```

onOpenReport

Асинхронное событие, которое предоставляет возможность реализовать собственный способ открытия шаблонов. Событие вызывается до диалогового окна выбора файла отчета и перед передачей его во вьювер. В таблице ниже представлен список аргументов обработчика событий:

Наименование	Описание
sender	Идентификатор компонента, который инициировал данное событие.
event	Идентификатор текущего события, имеет значение **OpenReport**.
report	Объект отчета **StiReport**.
preventDefault	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события. По умолчанию установлено значение **true**.
async	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события до выполнения callback функции. По умолчанию установлено значение **false**.

© 2003-2026 Stimulsoft

viewer.html

```
...
viewer.onOpenReport = (args) => {
   args.async = true;
   args.report = anotherReport;
   callback();
}
...
```

onOpenedReport

Асинхронное событие, которое предоставляет возможность модифицировать отчет перед его передачей во вьювере. Событие вызывается после диалогового окна выбора файла отчета, но перед передачей его во вьювер. В таблице ниже представлен список аргументов обработчика событий:

Наименование	Описание
sender	Идентификатор компонента, который инициировал данное событие.
event	Идентификатор текущего события, имеет значение **OpenedReport**.
report	Объект отчета **StiReport**.
preventDefault	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события. По умолчанию установлено значение **false**.
async	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события до выполнения callback функции. По умолчанию установлено значение **false**.

© 2003-2026 Stimulsoft

viewer.html

```
...
viewer.onOpenedReport = (args) => {
   if (args.report.reportAuthor != "Stimulsoft") {
      args.preventDefault = true;
      window.alert("report.reportAuthor == " + args.report.reportAuthor);
   }
}
...
```

8.2.14 Настройки вьювера

Настройка **HTML5 Viewer** выполняется при помощи свойств, находящихся в классе **Stimulsoft.Viewer.StiViewerOptions**. Все свойства разделены на группы для удобства использования. Ниже приведен пример установки свойств вьювера:

viewer.html

```
...
<script type="text/javascript">
   var report = new Stimulsoft.Report.StiReport();
   report.loadFile("SimpleList.mrt");

   var options = new Stimulsoft.Viewer.StiViewerOptions();

   options.width = "1000px";
   options.height = "1000px";

   options.appearance.theme =
   Stimulsoft.Viewer.StiViewerTheme.Office2022WhiteBlue;
   options.appearance.reportDisplayMode =
   Stimulsoft.Report.Export.StiHtmlExportMode.Auto;
   options.appearance.scrollbarsMode = true;
   options.appearance.backgroundColor =
   Stimulsoft.System.Drawing.Color.dodgerBlue;
   options.appearance.showTooltips = false;

   options.toolbar.showPrintButton = false;
   options.toolbar.showDesignButton = false;
   options.toolbar.showAboutButton = false;

   options.exports.showExportToPdf = true;
   options.exports.ShowExportToWord = true;

   var viewer = new Stimulsoft.Viewer.StiViewer(options);
   viewer.report = report;
</script>
...
```

Обратите внимание, что все элементы дашбордов имеют собственные кнопки

© 2003-2026 Stimulsoft

экспорта и полноэкранного режима просмотра. Специальных опций для управления их отображением не предусмотрено, но их можно выключить через свойства элемента. Нижеприведенный код необходимо добавить после загрузки отчета, перед его передачей вьюверу.

Default.aspx.cs

```
...
var dbsElementInteraction =
report.getComponentByName("RegionMap1").dashboardInteraction;
   dbsElementInteraction.showFullScreenButton = false;
   dbsElementInteraction.showSaveButton = false;
...
```

Главная (без группы)

Наименование	Описание
width	Задает ширину компонента в “px” или “%”.
height	Задает высоту компонента в “px” или “%”. Работает только в режиме если свойство options.appearance.**scrollbarsMode** установлено в значение **true**.

Оформление (Appearance)

Наименование	Описание
options.appearance.theme	Задает тему оформления вьювера. Список доступных тем оформления находится в перечислении **StiViewerTheme**. По умолчанию установлено значение **Office2022WhiteBlue**.
options.appearance.iconSet	Предоставляет возможность установить набор иконок: › Stimulsoft.Viewer.StiWebUIIconSet.Auto (значение по умолчанию) - устанавливает набор иконок автоматически. Для тем

© 2003-2026 Stimulsoft

	Office2022 - используется набор иконок в стиле **Monoline**, для тем **Office2013** - используется набор иконок в стиле **Regular**. › Stimulsoft.Viewer.StiWebUIIconSet.Monoline - устанавливает набор иконок в стиле **Monoline**; › Stimulsoft.Viewer.StiWebUIIconSet.Regular - устанавливает набор иконок в стиле **Regular**.
appearance.backgroundColor	Устанавливает цвет фона вьювера. По умолчанию, установлено значение Stimulsoft.System.Drawing.Color.**white**.
appearance.pageBorderColor	Устанавливает цвет границы страниц отчета. По умолчанию, установлено значение Stimulsoft.System.Drawing.Color.**gray**.
appearance.rightToLeft	Устанавливает режим отображения **Right to Left** для элементов управления вьювера. По умолчанию, установлено значение **false**
appearance.fullScreenMode	Устанавливает полноэкранный режим отображения вьювера. Если свойство установлено в значение **true**, то значения свойств **width** и **height** игнорируются. По умолчанию, установлено значение **false**.
appearance.scrollbarsMode	Устанавливает режим отображения отчета с полосами прокрутки. По умолчанию, установлено значение **false**.
appearance.openLinksWindow	Устанавливает целевое окно или фрэйм для открытия гиперссылок из отчета. По умолчанию, установлено значение “**_blank**” (новая вкладка браузера). Может принимать одно из стандартных значений "**_blank**", "**_self**", "**_top**", а так же имя окна либо фрэйма.
appearance.showTooltips	Включает или выключает отображение подсказок при наведении курсора на инструмент вьювера. По умолчанию, установлено значение **true**.

© 2003-2026 Stimulsoft

appearance.showTooltipsHelp	Предоставляет возможность отображать или не отображать ссылку на документацию в подсказках при наведении курсора на инструмент вьювера. По умолчанию, установлено значение **true**.
appearance.showDialogsHelp	Предоставляет возможность отображать или не отображать кнопку вызова справки в различных меню. По умолчанию, установлено значение **true**.
appearance.pageAlignment	Устанавливает выравнивание страниц отчета во вьювере. › Stimulsoft.Viewer.StiContentAlignment.**Left** – выравнивание страниц по левому краю; › Stimulsoft.Viewer.StiContentAlignment.**Center** – выравнивание страниц по центру; › Stimulsoft.Viewer.StiContentAlignment.**Right** – выравнивание страниц по правому краю. По умолчанию, установлено значение Stimulsoft.Viewer.StiContentAlignment.**Center**.
appearance.showPageShadow	Включает или выключает отображение тени страниц отчета. По умолчанию, установлено значение **true**.
appearance.bookmarksPrint	Включает печать закладок в отчете. По умолчанию, установлено значение **false**.
appearance.bookmarksTreeWidth	Устанавливает ширину панели закладок в пикселях. По умолчанию, установлено значение 180.
appearance.parametersPanelMaxHeight	Устанавливает максимальную высоту панели параметров в пикселях. По умолчанию, установлено значение 300.
appearance.parametersPanelColumnsCount	Устанавливает количество колонок на панели параметров. По умолчанию, установлено значение 2.
appearance.parametersPanelSort	Устанавливает или выключает режим

© 2003-2026 Stimulsoft

DataItems	сортировки значений переменной. По умолчанию, опция установлена в значение **true**, т.е. значения переменной сортируются.
appearance.parametersPanelDateFormat	Устанавливает формат даты и времени для переменных, которые отображаются на панели параметров. По умолчанию, установлено значение **String.empty**.
appearance.reportDisplayMode	Устанавливает режим экспорта для отображения страниц отчета. Может принимать одно из следующих значений: › **FromReport** - режим экспорта элементов отчета определяется из настроек шаблона - Div или Table; › **Table** – элементы отчета экспортируются с использованием таблиц HTML (значение по умолчанию); › **Div** – элементы отчета экспортируются с использованием разметки DIV;
appearance.interfaceType	Устанавливает тип интерфейса вьювера. Могут использоваться следующие значения: › Stimulsoft.Viewer.StiInterfaceType.**Auto** – тип интерфейса вьювера будет выбран автоматически в зависимости от используемого устройства (значение по умолчанию); › Stimulsoft.Viewer.StiInterfaceType.**Mouse** – принудительное использование стандартного интерфейса для управления вьювером при помощи мыши; › Stimulsoft.Viewer.StiInterfaceType.**Touch** – принудительное использование Touch интерфейса для управления вьювером при помощи сенсорного экрана монитора, в этом режиме элементы интерфейса вьювера имеют увеличенные размеры для удобства управления;

© 2003-2026 Stimulsoft

	› Stimulsoft.Viewer.StiInterfaceType.**Mobile** – принудительное использование Mobile интерфейса для управления вьювером при помощи экрана смартфона, в этом режиме интерфейс вьювера имеет упрощенный вид и адаптирован для управления при помощи мобильного устройства. По умолчанию, установлено значение Stimulsoft.Viewer. StiInterfaceType.**Auto**.
appearance.allowMobileMode	Включает или выключает возможность отображения отчета или дашборда в мобильном режиме. Если опция установлена в значение **false**, то мобильный режим просмотра не будет использоваться ни при каких обстоятельства. Если опция установлена в значение **true**, то мобильный режим просмотра будет использоваться запуске вьювера на мобильных устройствах. По умолчанию, опция установлена в значение **true**.
appearance.chartRenderType	Устанавливает тип отрисовки диаграмм в отчете: › Stimulsoft.Viewer.StiChartRenderType.**AnimatedVector** – диаграммы будут отрисованы в векторном режиме с анимацией; › Stimulsoft.Viewer.StiChartRenderType.**Vector** – диаграммы будут отрисованы как векторное изображение без анимации. По умолчанию, установлено значение Stimulsoft.Viewer.StiChartRenderType.**AnimatedVector**.
appearance.datePickerFirstDayOfWeek	Предоставляет возможность установить первый день недели для инструмента **Выбор**

© 2003-2026 Stimulsoft

	даты (Date picker). › Stimulsoft.Viewer.StiFirstDayOfWeek.**Auto** - Будет установлен понедельник или воскресенье как первый день недели, в зависимости от культуры браузера. › Stimulsoft.Viewer.StiFirstDayOfWeek.**Monday** - Понедельник будет установлен как первый день недели. › Stimulsoft.Viewer.StiFirstDayOfWeek.**Sunday** - Воскресенье будет установлен как первый день недели.
appearance.datePickerIncludeCurrentDayForRanges	Предоставляет возможность включать или не включать текущий день в диапазон значений элемента **Выбор даты (Date Picker)**. По умолчанию, опция установлена в значение **false**, т.е. текущий день не включается в диапазон значений элемента.
appearance.allowScrollZoom	Предоставляется возможность изменять масштаб вьювера при помощи колеса прокрутки мыши. По умолчанию, опция установлена в значение **true**.
appearance.allowTouchZoom	Предоставляется возможность изменять масштаб вьювера касанием. По умолчанию, опция установлена в значение **true**.
appearance.combineReportPages	Предоставляет возможность объединить обработанные страницы шаблона отчета в один шаблон или представить каждую страницу шаблона отдельной вкладкой во вьювере. По умолчанию, опция установлена в значение **false**, т.е. каждая страница шаблона отчета будет представлена отдельной вкладкой во вьювере.
appearance.allowPropagationEvents	Предоставляет возможность разрешить распространение событий нажатия клавиш, когда вьювер отчетов не в фокусе. По умолчанию свойство имеет значение **true**.

© 2003-2026 Stimulsoft

appearance.dashboardFilterElementItemHeight	Предоставляет возможность изменить высоту флажка в элементе дашборда **Список (List Box)**.

Панель инструментов (Toolbar)

Наименование	Описание
toolbar.visible	Предоставляет возможность отображать или не отображать панель инструментов вьювера. По умолчанию, установлено значение **true**.
toolbar.displayMode	Задает режим отображения панели инструментов вьювера. Может принимать одно из указанных ниже значений перечисления **displayMode**: › **Simple** – простой режим отображения, все элементы управления расположены на одной панели управления (значение по умолчанию); › **Separated** – раздельный режим отображения, панель управления разделена на верхнюю и нижнюю.
toolbar.backgroundColor	Предоставляет возможность изменить цвет панели инструментов. По умолчанию, установлено значение Stimulsoft.System.Drawing.Color.**empty**.
toolbar.borderColor	Предоставляет возможность изменить цвет границы панели инструментов. По умолчанию, установлено значение Stimulsoft.System.Drawing.Color.**empty**.
toolbar.fontColor	Предоставляет возможность изменить цвет шрифта для всех элементов на панели инструментов и во всех меню этой панели. По умолчанию, установлено значение Stimulsoft.System.Drawing.Color.**empty**.

© 2003-2026 Stimulsoft

toolbar.fontFamily	Предоставляет возможность изменить шрифт для всех элементов на панели инструментов и во всех меню этой панели. По умолчанию, установлено значение **"Arial"**.
toolbar.alignment	Устанавливает выравнивание элементов на панели управления. › Stimulsoft.Viewer.StiContentAlignment.**Default** – выравнивание элементов зависит от опции **RightToLeft**; › Stimulsoft.Viewer.StiContentAlignment.**Left** – все элементы будут выровнены по левому краю панели инструментов; › Stimulsoft.Viewer.StiContentAlignment.**Center** – все элементы будут выровнены по центру панели инструментов; › Stimulsoft.Viewer.StiContentAlignment.**Right** – все элементы будут выровнены по правой стороне панели инструментов. По умолчанию, установлено значение Stimulsoft.Viewer.StiContentAlignment.**Default**.
toolbar.showButtonCaptions	Включает или выключает отображение текста кнопок панели инструментов вьювера. По умолчанию, установлено значение **true**.
toolbar.showOpenButton	Включает отображение кнопки **Открыть (Open)** на панели инструментов вьювера при просмотре отчетов или панели индикаторов (дашбордов). По умолчанию свойство имеет значение **true**.
toolbar.showPrintButton	Предоставляет возможность отображать или не отображать кнопку **Печать (Print)** на панели инструментов. По умолчанию, установлено значение **true**.
toolbar.showSaveButton	Включает отображение кнопки **Сохранить (Save)** на панели инструментов вьювера при просмотре отчетов или панели индикаторов

© 2003-2026 Stimulsoft

	(дашбордов). По умолчанию свойство имеет значение **true**.
toolbar.showSendEmailButton	Предоставляет возможность отображать или не отображать кнопку **Отправить по Email (Send Email)** на панели инструментов. По умолчанию, установлено значение **false**. Также, необходимо добавить обработчик события onEmailReport.
toolbar.showFindButton	Предоставляет возможность отображать или не отображать кнопку **Поиск (Find)** на панели инструментов. По умолчанию, установлено значение **true**.
toolbar.showBookmarksButton	Предоставляет возможность отображать или не отображать кнопку **Закладки (Bookmarks)** на панели инструментов. Если эта кнопка не отображается, то и панель закладок в отчете отображаться не будет. По умолчанию, установлено значение **true**.
toolbar.showParametersButton	Предоставляет возможность отображать или не отображать кнопку **Параметры (Parameters)** на панели инструментов. Если эта кнопка не отображается, то и панель параметров в отчете отображаться не будет. По умолчанию, установлено значение **true**.
toolbar.showResourcesButton	Предоставляет возможность отображать или не отображать кнопку **Ресурсы (Resources)** на панели инструментов. Если эта кнопка не отображается, то и панель ресурсов в отчете отображаться не будет. По умолчанию, установлено значение **true**.
toolbar.showEditorButton	Предоставляет возможность отображать или не отображать кнопку **Редактировать (Editor)** на панели инструментов. Если эта кнопка не отображается, то невозможно будет изменить редактируемые элементы. По умолчанию, установлено значение **true**.
toolbar.showFullScreenButton	Включает отображение кнопки

© 2003-2026 Stimulsoft

	Полноэкранный просмотр (Full Screen) на панели инструментов вьювера при просмотре отчетов или панели индикаторов (дашбордов). По умолчанию свойство имеет значение **true**.
toolbar.showRefreshButton	Предоставляет возможность отображать или не отображать кнопку **Обновить (Refresh)** на панели инструментов вьювера при просмотре панели индикаторов (дашбордов). По умолчанию свойство имеет значение **true**.
toolbar.showFirstPageButton	Предоставляет возможность отображать или не отображать кнопку **Первая страница (First Page)** на панели инструментов. По умолчанию, установлено значение **true**.
toolbar.showPreviousPageButton	Предоставляет возможность отображать или не отображать кнопку **Последняя страница (Previous Page)** на панели инструментов. По умолчанию, установлено значение **true**.
toolbar.showCurrentPageControl	Предоставляет возможность отображать или не отображать индикатор текущей страницы на панели инструментов. По умолчанию, установлено значение **true**.
toolbar.showNextPageButton	Предоставляет возможность отображать или не отображать кнопку **Следующая страница (Next Page)** на панели инструментов. По умолчанию, установлено значение **true**.
toolbar.showLastPageButton	Предоставляет возможность отображать или не отображать кнопку **Предыдущая страница (Last Page)** на панели инструментов. По умолчанию, установлено значение **true**.
toolbar.showZoomButton	Предоставляет возможность отображать или не отображать кнопку выбора масштаба на панели инструментов. По умолчанию, установлено значение **true**.
toolbar.showViewModeButton	Предоставляет возможность отображать или

© 2003-2026 Stimulsoft

	не отображать кнопку выбора режимов отображения страниц отчета. По умолчанию, установлено значение **true**.
toolbar.showDesignButton	Включает отображение кнопки **Дизайн (Design)** на панели инструментов вьювера при просмотре отчетов или панели индикаторов (дашбордов). По умолчанию свойство имеет значение **false**.
toolbar.showAboutButton	Предоставляет возможность отображать или не отображать кнопку **О программе (About)** на панели инструментов. По умолчанию, установлено значение **true**.
toolbar.showPinToolbarButton	Предоставляет возможность отображать или не отображать кнопку **Закрепить (Pin)** в мобильном режиме просмотра отчета. По умолчанию, установлено значение **true**.
toolbar.printDestination	Устанавливает режим печати отчета. Может принимать одно из указанных ниже значений перечисления **StiPrintDestination**: › **Default** – будет отображено меню с предоставлением выбора режима печати (значение по умолчанию); › **Pdf** – печать будет осуществлена в формате PDF; › **Direct** – печать будет осуществлена в формате HTML непосредственно на принтер, будет отображен системный диалог печати; › **PopupWindow** – печать будет осуществлена в формате HTML через всплывающее окно предварительного просмотра отчета.
toolbar.viewMode	Устанавливает режим отображения страниц отчета: › Stimulsoft.Viewer.StiWebViewMode.**OnePage** – отображается одна страница, выбранная на

© 2003-2026 Stimulsoft

	панели инструментов. › Stimulsoft.Viewer.StiWebViewMode.**Continuous** – отображаются все страницы отчета в виде ленты. › Stimulsoft.Viewer.StiWebViewMode.**MultiplePages** – отображаются все страницы отчета в виде таблицы. По умолчанию, установлено значение Stimulsoft.Viewer.StiWebViewMode.**OnePage**.
toolbar.zoom()	Предоставляет возможность установить масштаб страниц отчета при загрузке вьювера. По умолчанию, 100 процентов. Максимальное значение 500 процентов. › Stimulsoft.Viewer.StiZoomMode.**PageWidth** – масштаб страниц отчета по ширине страницы. › Stimulsoft.Viewer.StiZoomMode.**PageHeight** – масштаб страниц отчета по высоте страницы. По умолчанию, установлено значение 100 процентов.
toolbar.menuAnimation	Предоставляет возможность включить или выключить анимацию отображения и закрытия различных меню во вьювере. По умолчанию, установлено значение **true**.
toolbar.showMenuMode	Устанавливает режим раскрытия различных меню во вьювере - при наведении или по щелчку. По умолчанию, установлено значение Stimulsoft.Viewer.StiShowMenuMode.**Click**.
toolbar.autoHide	Устанавливает режим автоматического сворачивания панели инструментов при просмотре отчета в мобильном режиме. По умолчанию, установлено значение **false**.

© 2003-2026 Stimulsoft

Экспорт (Exports)

Наименование	Описание
export.storeExportSettings	Предоставляет возможность сохранять настройки экспорта в cookies. По умолчанию, установлено значение **true**.
exports.showExportDialog	Предоставляет возможность отображать или не отображать меню настроек экспорта. Если меню спрятано, экспорт будет выполнен со значениями по умолчанию. По умолчанию, установлено значение **true**.
exports.showExportToDocument	Предоставляет возможность отображать или не отображать элемент **Document File** в меню **Сохранить (Save)**. По умолчанию, установлено значение **true**.
exports.showExportToPdf	Включает отображение пункта меню экспорта **Adobe PDF файл (Adobe PDF File)** при просмотре отчетов, и пункт **Adobe PDF** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
exports.showExportToPowerPoint	Включает отображение пункта меню экспорта **Microsoft PowerPoint 2007/2010 файл (Microsoft PowerPoint 2007/2010 File)**. По умолчанию свойство имеет значение **true**.
exports.showExportToHtml	Предоставляет возможность отображать или не отображать элемент **HTML File** в меню **Настройки экспорта (Export Settings)**. По умолчанию, установлено значение **true**.
exports.showExportToHtml5	Предоставляет возможность отображать или не отображать элемент **HTML5 File** в меню **Настройки экспорта (Export Settings)**. По умолчанию, установлено значение **true**.
exports.showExportToText	Включает отображение пункта меню экспорта **Text файл (Text File)**. По умолчанию свойство имеет значение **true**.
exports.showExportToWord	Предоставляет возможность отображать или

© 2003-2026 Stimulsoft

	не отображать элемент **Microsoft Word 2007/2010 File** в меню **Сохранить (Save)**. По умолчанию, установлено значение **true**.
exports.showExportToOpenDocumentWriter	Включает отображение пункта меню экспорта **OpenDocument Writer файл (OpenDocument Writer File)**. По умолчанию свойство имеет значение **true**.
exports.showExportToExcel	Включает отображение пункта меню экспорта **Microsoft Excel 2007/2010 файл (Microsoft Excel 2007/2010 File)** при просмотре отчетов, и пункт **Microsoft Excel** при просмотре панелей индикаторов. По умолчанию свойство имеет значение **true**.
exports.showExportToOpenDocumentCalc	Включает отображение пункта меню экспорта **OpenDocument Calc файл (OpenDocument Calc File)**. По умолчанию свойство имеет значение **true**.
exports.showExportToCsv	Включает отображение пункта меню экспорта **CSV файл (CSV File)**. По умолчанию свойство имеет значение **true**.
exports.showExportToJson	Включает отображение пункта меню экспорта **JSON файл (JSON File)**. По умолчанию свойство имеет значение **false**.
showExportToImageSvg	Включает отображение пункта меню экспорта **Изображение (Image)**, с возможность экспорта отчета в SVG файл. По умолчанию свойство имеет значение **true**.

Отправить по Email (Email)

Наименование	Описание
email.showEmailDialog	Включает отображение диалогового окна параметров отправки отчета по Email. Если диалоговое окно отключено, то отправка по Email будет производиться с настройками,

© 2003-2026 Stimulsoft

	заданными по умолчанию **onEmailReport**. По умолчанию, установлено значение **true**.
email.showExportDialog	Включает отображение диалогового окна параметров экспорта при отправке Email. Если свойство имеет значение **false**, то экспортирование будет производиться с заданными по умолчанию настройками. По умолчанию, установлено значение **true**.
email.defaultEmailAddress	Устанавливает Email получателя по умолчанию, т.е. адрес на который будет отправлено письмо с вложенным отчетом. По умолчанию, значение не задано.
email.defaultEmailSubject	Устанавливает тему (заголовок) электронного письма по умолчанию. По умолчанию, значение не задано.
email.defaultEmailMessage	Устанавливает сообщение (текст) электронного письма по умолчанию. По умолчанию, значение не задано.

8.3 HTML5 дизайнер

YouTube

Посмотрите видеоуроки по работе с компонентом JS HTML5 Designer. Подписывайтесь на канал Stimulsoft и узнайте первыми о новых видеоуроках. Вопросы и предложения оставляйте в комментариях к видео.

Примеры

Ознакомьтесь с примерами по работе с компонентом JS HTML5 Designer GitHub. Все примеры представляют собой отдельные проекты, сгруппированные в отдельных папках по используемым технологиям.

Компонент **HTML5 Designer (StiDesigner)** предназначен для разработки

© 2003-2026 Stimulsoft

отчетов в окне браузера. При этом Вам не нужен мощный сервер, а на стороне клиента не требуется устанавливать .NET Framework, ActiveX компоненты или какие-либо специальные плагины. Все, что необходимо – это любой современный Web-браузер.

При помощи **HTML5 Designer** можно создавать, редактировать, сохранять, просматривать и печатать отчеты на любом компьютере с любой установленной операционной системой. Так как дизайнер использует только HTML5 и JavaScript технологии, он может быть запущен на устройствах, где нет поддержки Flash или Silverlight – планшеты, смартфоны. Также дизайнер поддерживает Mobile и Touch интерфейсы, которые автоматически включаются при использовании мобильных устройств и мониторов с сенсорным экраном.

Компонент **HTML5 Designer** использует технологию **JavaScript** для выполнения всех действий над редактируемым отчетом, что позволяет использовать практически любую серверную сторону, а на стороне клиента избавиться от перезагрузки всей страницы и повысить скорость работы.

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для использования **HTML5 Designer** в проекте, необходимо установить npm пакет stimulsoft-reports-js:

```
npm install stimulsoft-reports-js
```

Если данный вариант не подходит, можно добавить указанные ниже скрипты вручную в проект.

© 2003-2026 Stimulsoft

designer.html

```
...
<script src="scripts/stimulsoft.report.js"></script>
<script scr="scripts/stimulsoft.viewer.js"></script>
<script src="scripts/stimulsoft.designer.js"></script>

<!-- Stimulsoft Blockly editor for JS Designer -->
<script src="scripts/stimulsoft.blockly.editor.js"></script>
...
```

Для использования **HTML5 Designer** в проекте с возможностью редактирования отчетов и дашбордов , необходимо установить npm пакет stimulsoft-dashboards-js:

```
npm install stimulsoft-dashboards-js
```

Если данный вариант не подходит, можно добавить указанные ниже скрипты вручную в проект.

viewer.html

```
...
<script src="scripts/stimulsoft.report.js"></script>
<script src="scripts/stimulsoft.dashboards.js"></script>
<script scr="scripts/stimulsoft.viewer.js"></script>
<script src="scripts/stimulsoft.designer.js"></script>
...
```

Информация

О вариантах скриптов, которые вы можете использовать в своих проектах, можно ознакомиться в главах Скрипты пакета Reports.JS и Скрипты пакета Dashboards.JS.

© 2003-2026 Stimulsoft

8.3.1 Как это работает

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Компонент **StiDesigner** разработан с использованием только HTML5 и JavaScript технологии, и для его работы не требуется сервер (он необходим только для размещения файлов проекта). При запуске дизайнера отчетов происходят следующие действия:

- JavaScript компонент добавляет на текущую страницу HTML код интерфейса дизайнера;
- Если был присвоен объект отчета, то отчет будет загружен на страницу дизайнера для его редактирования;
- Каждое действие в дизайнере (например, предварительный просмотр отчета, сохранение шаблона отчета, экспорт отчета, применение параметров, сортировка и детализация отчета) вызывает определенное JavaScript событие, в котором можно выполнить необходимые манипуляции с отчетом.

8.3.2 Редактирование отчета

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Компонент **StiDesigner** представляет собой инструмент для создания и редактирования отчетов. Для начала работы с дизайнером отчетов необходимо добавить скрипты, необходимые для компонента, на HTML страницу проекта.

© 2003-2026 Stimulsoft

designer.html

```
...
<script src="scripts/stimulsoft.reports.js" type="text/javascript"></
script>
<script src="scripts/stimulsoft.dashboards.js"></script>
<script scr="scripts/stimulsoft.viewer.js" type="text/javascript"></
script>
<script src="scripts/stimulsoft.designer.js" type="text/javascript"></
script>
...
```

Для редактирования скриптов событий отчета с использованием инструмента Blockly в дизайнере, дополнительно понадобится добавить соответствующий файл скрипта, содержащий визуальную часть редактора:

designer.html

```
...
<script src="scripts/stimulsoft.reports.js" type="text/javascript"></
script>
<script src="scripts/stimulsoft.dashboards.js"></script>
<script scr="scripts/stimulsoft.viewer.js" type="text/javascript"></
script>
<script src="scripts/stimulsoft.designer.js" type="text/javascript"></
script>
<script src="scripts/stimulsoft.blockly.editor.js" type="text/
javascript"></script>
...
```

Информация

Также необходимо подключить скрипты и стили для вьювера отчетов, т.к. этот компонент используется для предварительного просмотра отчетов на соответствующей вкладке дизайнера.

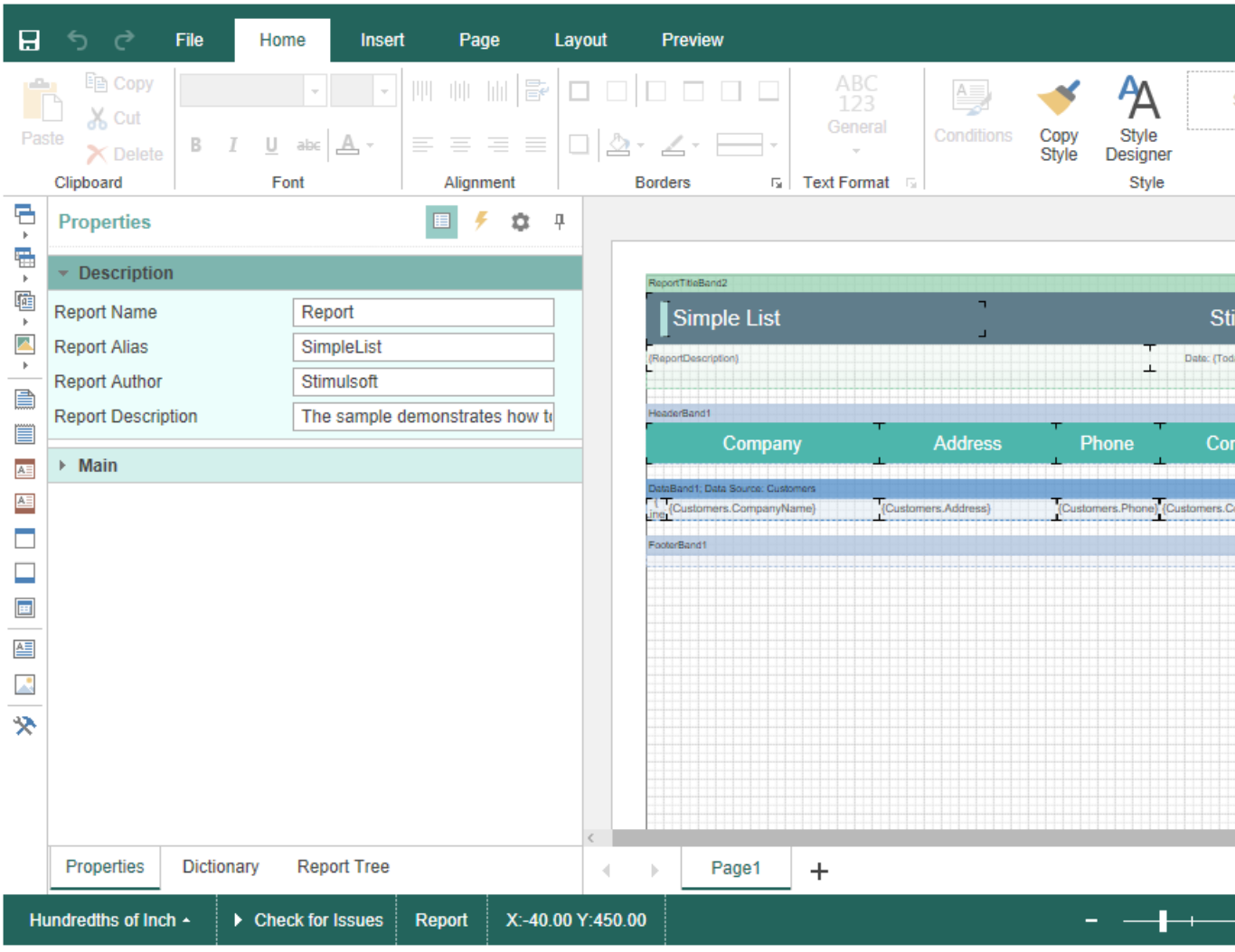
Затем необходимо добавить на HTML страницу JavaScript код загрузки отчета, и присвоить полученный объект дизайнеру. В данном случае, дизайнер будет развернут в текущем DOM элементе в том месте, где расположен скрипт.

© 2003-2026 Stimulsoft

designer.html

```
...
var report = new Stimulsoft.Report.StiReport();
report.loadFile("SimpleList.mrt");
//report.loadFile("Dashboard.mrt");

var designer = new Stimulsoft.Designer.StiDesigner(null, "Designer", false);
designer.report = report;
...
```



Объект **StiDesigner** можно создать при помощи конструктора **Stimulsoft.Designer.StiDesigner()**, который может принимать на вход указанные не обязательные аргументы:

- **options**: представляет собой набор опций, находящихся в классе **Stimulsoft.Designer.StiDesignerOptions**. Все опции разделены на категории.

© 2003-2026 Stimulsoft

Подробное описание категорий и опций находится в разделе Настройки дизайнера.

- **designerId**: идентификатор дизайнера, используется при развертывании компонента как DOM объекта, по умолчанию используется значение "**StiDesigner**"
- **renderAfterCreate**: определяет расположение дизайнера. Если установлено значение **true**, то дизайнер будет отображаться в том же месте DOM дерева, в котором располагается код создания объекта. Если установлено значение **false**, то дизайнер будет располагаться в том месте где вызван метод **renderHtml()**. Например, инициализация дизайнера в заголовке страницы.

designer.html

```
...
<script type="text/javascript">
   var designer = new Stimulsoft.Designer.StiDesigner(null, "StiDesigner",
   false);
</script>
...
```

И последующий вывод дизайнера в текущий элемент DIV.

designer.html

```
...
<div>Page content</div>
<div>
   <script type="text/javascript">
      // Render the report designer in this place
      designer.renderHtml();
   </script>
</div>
...
```

В качестве аргумента метода вывода дизайнера **renderHtml(id)** допускается указать идентификатор элемента HTML страницы, в котором необходимо вывести дизайнер.

© 2003-2026 Stimulsoft

designer.html

```
...
<script type="text/javascript" >
   var designer = new Stimulsoft.Designer.StiDesigner(null, "StiDesigner",
   false);
   designer.renderHtml("content");
</script>
...
```

Указанный элемент должен находиться на HTML странице, на которой используется дизайнер отчетов.

designer.html

```
...
<div id="content"></div>
...
```

8.3.3 Создание нового отчета

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Для запуска дизайнера с новым (пустым) отчетом не требуется никаких действий, при загрузке компонента новый отчет будет создан автоматически. При необходимости, можно создать новый объект отчета и предварительно загрузить для него данные, либо выполнить какие-либо другие необходимые действия.

designer.html

```
...
var report = new Stimulsoft.Report.StiReport();
//var  newDashboard = Stimulsoft.Report.StiReport.createNewDashboard();

report.reportName = "MyNewReport";
//newDashboard.reportName = "MyDashboard";
...
```

© 2003-2026 Stimulsoft

Для хранения данных используется специальный объект **DataSet**, который имеет набор методов для загрузки данных из различных форматов. Для подключения данных к отчету используется метод **regData()**, в аргументах которого указывается подготовленный объект **DataSet**.

designer.html

```
...
report.regData(dataSet);
...
```

Данные добавляются в специальную коллекцию объекта отчета, и используются для его построения. Для того, чтобы отобразить структуру зарегистрированных данных в словаре отчета, требуется синхронизация структуры данных. Для этого используется метод **synchronize()**.

designer.html

```
...
report.dictionary.synchronize();
...
```

Также новый отчет можно создать, используя главное меню дизайнера. Для предварительной загрузки данных для нового отчета, либо выполнения какие-либо других необходимых действий, предназначено событие **onCreateReport**. Данное событие будет вызвано при создании нового пустого отчета из главного меню, либо при создании отчета с использованием мастера.

designer.html

```
...
designer.onCreateReport = function (args) {
   var dataSet = new Stimulsoft.System.Data.DataSet("SimpleDataSet");
   dataSet.readJsonFile("Data/Demo.json");

   args.report.regData(dataSet.dataSetName, "", dataSet);
   args.report.dictionary.synchronize();
}
...
```

© 2003-2026 Stimulsoft

Также можно предварительно создать подключение к источнику данных выбранного типа, и добавить его в коллекцию подключений в шаблоне отчета. Данный способ аналогичен созданию подключения в словаре отчета при помощи интерфейса дизайнера.

designer.html

```
...
var report = new Stimulsoft.Report.StiReport();
//var  newDashboard = Stimulsoft.Report.StiReport.createNewDashboard();

var xmlDataBase = new Stimulsoft.Report.Dictionary.StiXmlDatabase("Demo",
"Demo.xsd", "Demo.xml");

report.loadFile("SimpleList.mrt");
//newDashboard.loadFile("Dashboard.mrt");
report.dictionary.databases.clear();
//newDashboard.dictionary.databases.clear();
report.dictionary.databases.add(xmlDataBase);
//newDashboard.dictionary.databases.add(xmlDataBase);
...
```

Таким образом, можно добавить необходимое количество источников данных различных типов.

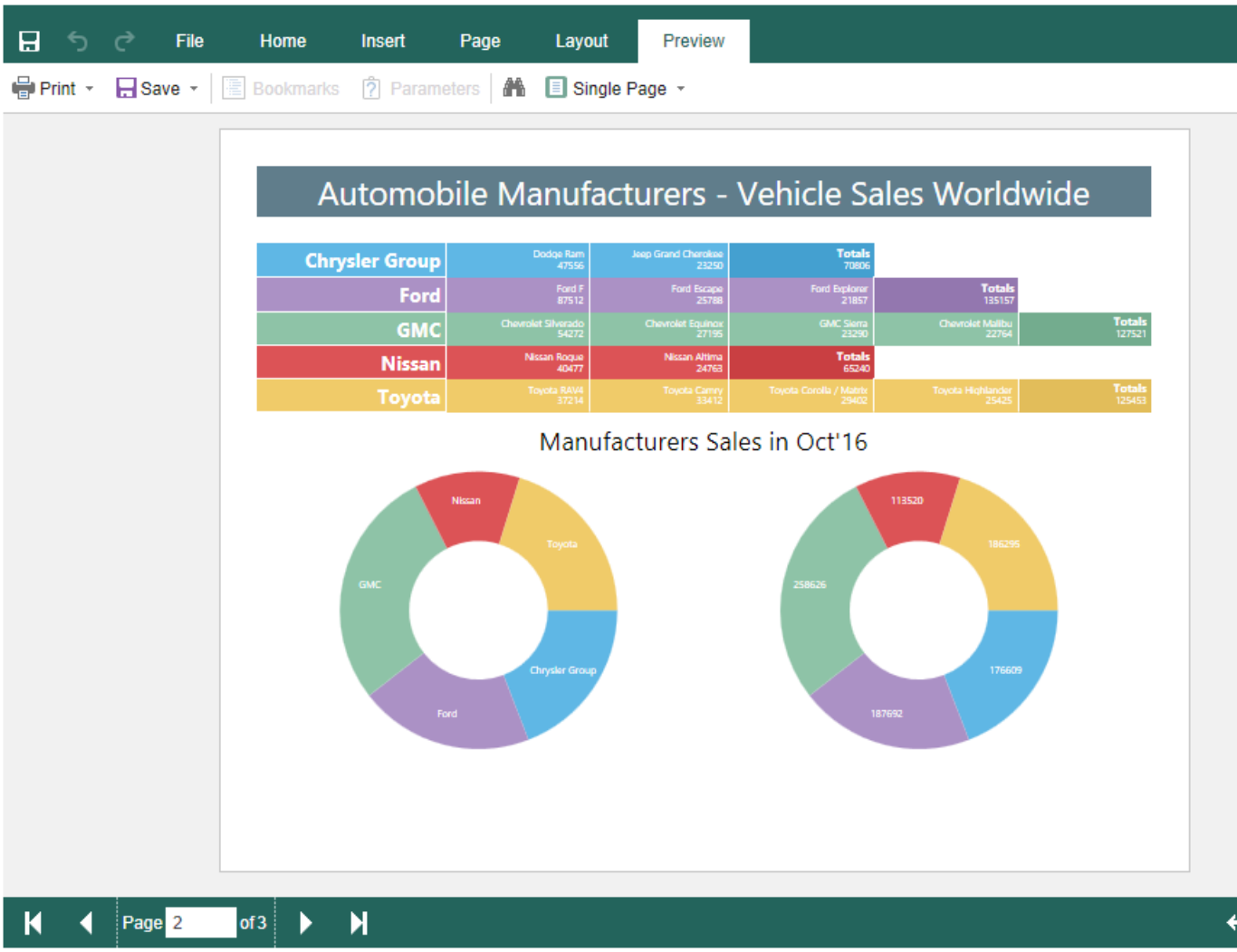
8.3.4 Просмотр

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

У компонента **HTML5 Designer** предусмотрен режим просмотра редактируемого отчета. Для просмотра отчета достаточно перейти на соответствующую вкладку в окне дизайнера. Шаблон отчета будет построен и отображен во встроенном вьювере.

© 2003-2026 Stimulsoft



Перед просмотром отчета предусмотрена возможность выполнить какие-либо необходимые действия, например, подключить данные для отчета. Для этого можно воспользоваться специальным событием **onPreviewReport**, которое будет вызвано перед просмотром отчета. В аргументах события будет находиться отчет, предназначенный для просмотра.

designer.html

```
...
designer.onPreviewReport = function (args) {
   var dataSet = new Stimulsoft.System.Data.DataSet("SimpleDataSet");
   dataSet.readJsonFile("Data/Demo.json");

   args.report.regData(dataSet.dataSetName, "", dataSet);
}
...
```

© 2003-2026 Stimulsoft

8.3.5 Дополнительные возможности предварительного просмотра

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Окно предварительного просмотра отчета компонента **HTML5 Designer** представляет собой полноценный интерактивный **HTML5 Viewer**, который умеет осуществлять печать и экспортирование отчета, поддерживает работу с параметрами отчета. Также поддерживаются интерактивные действия, такие как динамическая сортировка, детализация, сворачивание. Для использования указанных возможностей не требуется каких-либо дополнительных настроек дизайнера отчетов.

designer.html

```
...
designer.viewer.onBeginExportReport = function (event) {
   switch (event.format) {
      case Stimulsoft.Report.StiExportFormat.Html:
         event.settings.zoom = 2;  // Set zoom to 200%
         break;
   }
   console.log("exporting");
}
...
```

Информация

Если какие-либо из указанных дополнительных возможностей предварительного просмотра отчета не требуются (например, экспортирование или печать отчета), можно отключить их, используя соответствующие свойства компонента **HTML5 Designer**.

© 2003-2026 Stimulsoft

8.3.6 Сохранение отчета

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

В компоненте **HTML5 Designer** предусмотрено два варианта сохранения отчета, которые доступны в главном меню и на главной панели дизайнера: **Сохранить (Save)** и **Сохранить как (Save As)**. В свою очередь, каждый из этих вариантов сохранения имеет свои режимы и настройки.

Сохранение отчета и дашборда на стороне сервера

Для сохранения редактируемого отчета на стороне сервера необходимо воспользоваться специальным событием **onSaveReport**, которое будет вызвано при выборе пункта **Сохранить (Save)** в главном меню, либо нажатии на кнопку **Сохранить (Save)** на главной панели дизайнера. В аргументах события будет передан редактируемый отчет, который можно сохранить, например, в JSON строку и затем передать на сторону сервера.

designer.html

```
...
designer.onSaveReport = function (args) {
   args.preventDefault = false;
   var jsonReport = args.report.saveToJsonString();
}
...
```

Информация

Более подробно, об аргументах события `onSaveReport` можно ознакомиться в главе События дизайнера.

© 2003-2026 Stimulsoft

По умолчанию, после сохранения отчета дизайнер продолжает работу без отображения каких-либо сообщений. Если необходимо, после сохранения отчета предусмотрена возможность отобразить диалоговое окно с ошибкой или текстовым сообщением. Для этого предназначена специальная статическая функция **showError()**.

designer.html

```
...
designer.onSaveReport = function (args) {
   args.preventDefault = false;
   Stimulsoft.System.StiError.showError("Some message after saving", true);
}
...
```

Если необходимо, вы можете получить доступ к оригинальному имени отчета или имени отчета из диалога сохранения.

designer.html

```
...
var designer = new Stimulsoft.Designer.StiDesigner(designerOptions,
"StiDesigner", false);
designer.renderHtml("content");

designer.onSaveReport = function (args) {

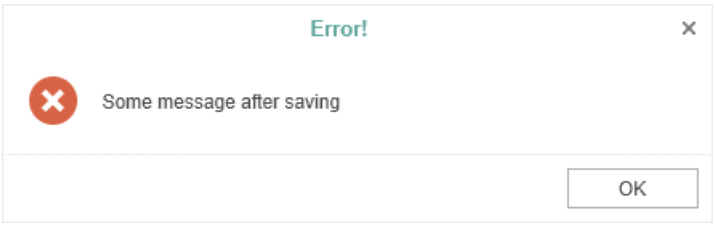
   //a flag to prevent further processing of the event
   args.preventDefault = false;

   //Report name from the designer save dialog
   var reportName = args.fileName;

   //Original report name from properties
   var reportName = args.report.reportName;
}
...
```

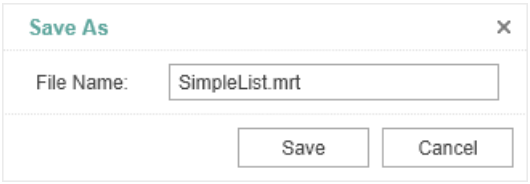
В качестве аргументов функция принимает текст ошибки и флаг, определяющий тип окна. Текст может содержать как сообщение об ошибке сохранения либо предупреждение, так и любое другое сообщение. Если в качестве второго аргумента функции установлен флаг **true**, то будет отображено окно с ошибкой, если установлен флаг **false**, то будет отображено информационное окно.

© 2003-2026 Stimulsoft



Сохранение отчета и дашборда на стороне клиента

Для сохранения редактируемого отчета на стороне клиента в виде файла не требуется дополнительных настроек дизайнера. Достаточно выбрать пункт главного меню **Сохранить как (Save As)**, при нажатии на который отобразится диалог сохранения файла. В данном диалоге можно изменить имя файла отчета, после чего файл будет сохранен на локальном диске компьютера.



Компонент **HTML5 Designer** предоставляет возможность изменить поведение указанного варианта сохранения, для этого в дизайнере предусмотрено специальное событие **onSaveAsReport**. Работа данного события будет аналогична событию **onSaveReport**.

designer.html

```
...
designer.onSaveAsReport = function (args) {
   args.report.repotName = "Report";

   // Save the report template
   var jsonReport = args.report.saveToJsonString();
}
...
```

8.3.7 Использование тем

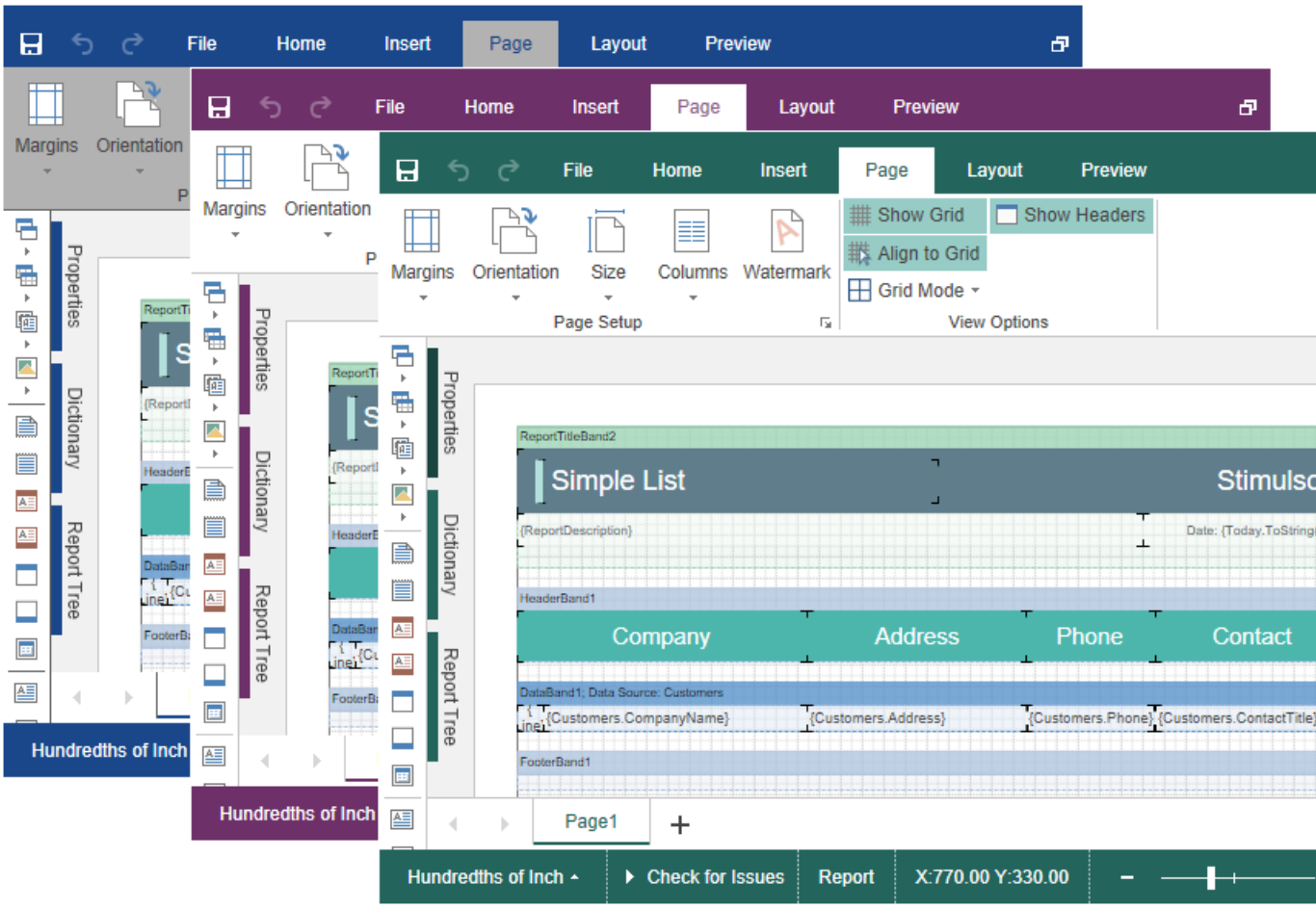
Компонент **HTML5 Designer** имеет возможность смены тем оформления визуальных элементов управления. Для этого можно использовать опцию компонента **theme** или метод **setTheme()**.

© 2003-2026 Stimulsoft

designer.html

```
...
var options = new Stimulsoft.Designer.StiDesignerOptions();
options.appearance.theme =
Stimulsoft.Designer.StiDesignerTheme.Office2022WhiteBlue;
...
designer.setTheme(Stimulsoft.Designer.StiDesignerTheme.Office2022WhiteBlue
);
...
```

На данный момент доступно **2 различные темы** со своими цветовыми акцентами. В результате, доступно **более 50 вариантов** оформления. Это позволяет настроить внешний вид дизайнера практически под любое оформление Web проекта.



© 2003-2026 Stimulsoft

8.3.8 События дизайнера

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Компонент **HTML5 Designer** поддерживает события, которые предоставляют возможность выполнить необходимые операции перед определенными действиями. Ниже представлены события дизайнера с описанием и некоторыми примерами их использования:

onPrepareVariables

Асинхронное событие вызывается в начале построения отчета перед заполнением переменных в отчете. Событие возникает сразу после `onPrepareVariables` в `StiReport`. В таблице ниже представлен список аргументов обработчика событий:

Наименование	Описание
sender	Идентификатор компонента, который инициировал данное событие.
event	Идентификатор текущего события, имеет значение **PrepareVariables**.
report	Объект отчета **StiReport**.
preventDefault	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события. По умолчанию установлено значение **false**.
async	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события до выполнения callback функции. По умолчанию установлено

© 2003-2026 Stimulsoft

	значение **false**.

designer.html

```
...
designer.onPrepareVariables = (args, callback) => {
   args.variables[0].value = "Replace value";
}
...
```

onBeginProcessData

Асинхронное событие вызывается перед запросом данных, необходимых для построения отчета. Событие возникает сразу после `onBeginProcessData` в `StiReport`. В таблице ниже представлен список аргументов обработчика событий:

Наименование	Описание
sender	Идентификатор компонента, который инициировал данное событие.
event	Идентификатор текущего события, имеет значение **BeginProcessData**.
report	Объект отчета **StiReport**.
preventDefault	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события. По умолчанию установлено значение **false**.
async	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события до выполнения callback функции. По умолчанию установлено значение **false**.
command	Идентификатор текущей команды, может иметь значения **TestConnection** (проверка соединения) и **ExecuteQuery** (получения данных из хранилища).
database	Предоставляет возможность указать

© 2003-2026 Stimulsoft

	наименование текущей базы данных.
connection	Имя текущего подключения к источнику данных, заданное в шаблоне отчета.
headers	Предоставляет возможность указать заголовки в запросах.
withCredentials	Предоставляет возможность указать cookies в запросах.
connectionString	Предоставляет возможность указать строку подключения к хранилищу данных.
dataSource	Предоставляет возможность указать наименование источника данных.
queryString	SQL запрос на получение данных. Используется только с командой **ExecuteQuery**.
timeout	Предоставляет возможность указать таймаут запроса.
parameters	Предоставляет возможность указать параметр в запросе.
escapeQueryParameters	Предоставляет возможность использовать экранированную часть запроса. По умолчанию установлено значение **true**.
pathData	Предоставляет возможность указать путь к файлу данных.
tryParseDateTime	Предоставляет возможность определить возможность преобразования данных в тип DateTime.
relationDirection	Предоставляет возможность изменить направление связи между источниками данных.
pathSchema	Предоставляет возможность указать путь к XSD файлу со схемой данных.
firstRowIsHeader	Предоставляет возможность использовать первую строку как заголовки данных в Excel источниках данных.

© 2003-2026 Stimulsoft

collectionName	Наименование коллекции в OData источнике данных.
separator	Предоставляет возможность указать разделитель в CSV источнике данных.
dataType	Предоставляет возможность указать тип данных для GIS источника данных.
codePage	Предоставляет возможность определить кодировку данных в CSV и DBF источниках данных.

designer.html

```
...
//Replace connection string
designer.onBeginProcessData = (args) => {
   if (args.database == "MySQL")
      args.connectionString = "new connection string";
}
...

//Add a some data
designer.onBeginProcessData = (args, callback) => {
   if (args.database == "MySQL"){
      args.preventDefault = true;
      var result = {
         success: true,
         rows: [
            ["value1", 1, false],
            ["value2", 1, true],
            ["value3", 2, false]
         ],
         columns: [
            "Column1_name",
            "Column2_name",
            "Column3_name"
         ],
         types:[
            "string",
            "int",
            "boolean"
         ]
      }

      // https://github.com/stimulsoft/DataAdapters.JS/
      callback(result);
   }
}
...
```

© 2003-2026 Stimulsoft

onEndProcessData

Событие вызывается после получения данных, необходимых для построения отчета. Событие возникает сразу после `onEndProcessData` в `StiReport`. В таблице ниже представлен список аргументов обработчика событий:

Наименование	Описание
sender	Идентификатор компонента, который инициировал данное событие.
event	Идентификатор текущего события, имеет значение **EndProcessData**.
report	Объект отчета **StiReport**.
command	Предоставляет возможность указать тип команды.
dataSource	Предоставляет возможность указать наименование источника данных.
connection	Имя текущего подключения к источнику данных, заданное в шаблоне отчета.
database	Предоставляет возможность указать наименование текущей базы данных.
result	Результирующий набор данных в определенном JSON формате. Он имеет две коллекции - колонки и строки с их значениями.

designer.html

```
...
designer.onEndProcessData = (args) => {
   if (args.command == "ExecuteQuery" && args.dataSource == "Categories")
      args.result.rows.push(rowData) ;
      // https://github.com/stimulsoft/DataAdapters.JS/
}
...
```

onCreateReport

© 2003-2026 Stimulsoft

Асинхронное событие вызывается после создания нового отчета перед присвоением объекта отчета дизайнеру. В таблице ниже представлен список аргументов обработчика событий:

Наименование	Описание
sender	Идентификатор компонента, который инициировал данное событие.
event	Идентификатор текущего события, имеет значение **CreateReport**.
report	Объект отчета **StiReport**.
preventDefault	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события. По умолчанию установлено значение **false**.
async	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события до выполнения callback функции. По умолчанию установлено значение **false**.
isWizardUsed	Этот идентификатор предоставляет возможность использовать мастер создания отчета.

designer.html

```
...
designer.onCreateReport = function (args) {
   var report = args.report;

   var database = new
   Stimulsoft.Report.Dictionary.StiJsonDatabase("DemoData", "http://
   localhost/Demo.json");
   report.dictionary.databases.add(database);
   report.dictionary.synchronize();
}
...
```

onOpenReport

© 2003-2026 Stimulsoft

Асинхронное событие вызывается при нажатии кнопки **Открыть** перед появлением диалогового окна. В таблице ниже представлен список аргументов обработчика событий:

Наименование	Описание
sender	Идентификатор компонента, который инициировал данное событие.
event	Идентификатор текущего события, имеет значение **OpenReport**.
report	Объект отчета **StiReport**.
preventDefault	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события. По умолчанию установлено значение **true**.
async	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события до выполнения callback функции. По умолчанию установлено значение **false**.

designer.html

```
...
//Call custom callback() function for loading template to designer
designer.onOpenReport = (args, callback) => {
   args.async = true;
   args.report = anotherReport;
   callback();
}
...
```

onOpenedReport

Асинхронное событие вызывается при открытии отчета, после выбора файла отчета в диалоговом окне, но перед присвоением его дизайнеру. В таблице ниже представлен список аргументов обработчика событий:

Наименование	Описание

© 2003-2026 Stimulsoft

sender	Идентификатор компонента, который инициировал данное событие.
event	Идентификатор текущего события, имеет значение **OpenedReport**.
report	Объект отчета **StiReport**.
preventDefault	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события. По умолчанию установлено значение **false**.
async	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события до выполнения callback функции. По умолчанию установлено значение **false**.

designer.html

```
...
//Add image to report resource when it has been opening
designer.onOpenedReport = (args, callback) => {
   args.async = true;
   var xhr = new XMLHttpRequest();
   xhr.open('GET', "Url to image");
   xhr.onload = function () {
      var imageData = xhr.response;

      var resource = new
      Stimulsoft.Report.Dictionary.StiResource("ImageName");
      resource.content = imageData;
      args.report.dictionary.resources.add(resource);
      callback();
   };
   xhr.send();
}
...
```

onAssignedReport

Событие вызывается после при присвоении отчета дизайнеру. В таблице ниже представлен список аргументов обработчика событий:

Наименование	Описание

© 2003-2026 Stimulsoft

sender	Идентификатор компонента, который инициировал данное событие.
event	Идентификатор текущего события, имеет значение **AssignedReport**.
report	Объект отчета **StiReport**.

designer.html

```
...
designer.onAssignedReport = (args) => {
   console.log("The report was assigned to the designer")
}
...
```

onSaveReport

Асинхронное событие вызывается перед сохранением отчета. В таблице ниже представлен список аргументов обработчика событий:

Наименование	Описание
sender	Идентификатор компонента, который инициировал данное событие.
event	Идентификатор текущего события, имеет значение **SaveReport**.
report	Объект отчета для сохранения.
preventDefault	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события. По умолчанию установлено значение **true**.
async	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события до выполнения callback функции. По умолчанию установлено значение **false**.
fileName	Имя сохраняемого файла.
autoSave	Предоставляет возможность сохранять

© 2003-2026 Stimulsoft

	автоматически. По умолчанию установлено значение **false**.

designer.html

```
...
//Remove report resources before saving
designer.onSaveReport = (args, callback) => {
   var report = args.report.clone();
   report.dictionary.resources.clear();
   args.report = report;
}
...
```

onSaveAsReport

Асинхронное событие вызывается перед сохранением отчета, при выполнении команды **Сохранить как (Save As)**. В таблице ниже представлен список аргументов обработчика событий:

Наименование	Описание
sender	Идентификатор компонента, который инициировал данное событие.
event	Идентификатор текущего события, имеет значение **SaveAsReport**.
report	Объект отчета для сохранения.
preventDefault	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события. По умолчанию установлено значение **false**.
async	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события до выполнения callback функции. По умолчанию установлено значение **false**.
fileName	Имя сохраняемого файла.
autoSave	Предоставляет возможность сохранять автоматически. По умолчанию установлено

© 2003-2026 Stimulsoft

	значение **false**.

designer.html

```
...
//Stop and redefinition the save method
designer.onSaveAsReport = (args, callback) => {
   args.preventDefault = true;
   var jsonString = args.report.saveToJsonString();
   // save report
}
...
```

onPreviewReport

Асинхронное событие вызывается при переходе на вкладку просмотра отчета. В таблице ниже представлен список аргументов обработчика событий:

Наименование	Описание
sender	Идентификатор компонента, который инициировал данное событие.
event	Идентификатор текущего события, имеет значение **PreviewReport**.
report	Объект отчета для просмотра.
preventDefault	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события. По умолчанию установлено значение **false**.
async	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события до выполнения callback функции. По умолчанию установлено значение **false**.
viewer	Объект **StiViewer**.

© 2003-2026 Stimulsoft

designer.html

```
...
designer.onPreviewReport = function (args) {
   var dataSet = new Stimulsoft.System.Data.DataSet("SimpleDataSet");
   dataSet.readJsonFile("Data/Demo.json");

   args.report.regData(dataSet.dataSetName, "", dataSet);
}
...
```

onCloseReport

Асинхронное событие вызывается после закрытия отчета перед отменой назначенного объекта отчета дизайнеру. В таблице ниже представлен список аргументов обработчика событий:

Наименование	Описание
sender	Идентификатор компонента, который инициировал данное событие.
event	Идентификатор текущего события, имеет значение **CloseReport**.
report	Объект отчета **StiReport**.
preventDefault	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события. По умолчанию установлено значение **false**.
async	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события до выполнения callback функции. По умолчанию установлено значение **false**.

designer.html

```
...
designer.onCloseReport = function (args) {
   console.log("The report was closed")
}
...
```

© 2003-2026 Stimulsoft

onExit

Событие вызывается при нажатии на кнопку **Выход (Exit)** в меню **Файл (File)** дизайнера отчетов. В таблице ниже представлен список аргументов обработчика событий:

Наименование	Описание
sender	Идентификатор компонента, который инициировал данное событие.
event	Идентификатор текущего события, имеет значение **Exit**.

designer.html

```
...
designer.onExit = function (args) {
   console.log(args.event);
}
...
```

8.3.9 Пользовательская настройка дизайнера

Вы можете добавлять пользовательские элементы управления в дизайнер отчетов. В этой главе мы рассмотрим несколько примеров пользовательской настройки дизайнера отчетов.

Добавление кнопки на риббон вкладку **Главная (Home)**.

© 2003-2026 Stimulsoft

designer.html

```
...
var designer = new Stimulsoft.Designer.StiDesigner(designerOptions,
"StiDesigner", false);
designer.renderHtml("content");

//Example add custom button to home toolbar panel
var homePanel = designer.jsObject.options.homePanel;

//Add buttons group to insert panel. Parameters of GroupBlock(groupName,
groupText) method.
var buttonsGroup = designer.jsObject.GroupBlock("buttonsGroup1",
"Group1");
var buttonsGroupTable = designer.jsObject.GroupBlockInnerTable();
buttonsGroup.container.appendChild(buttonsGroupTable);

//Add big button to buttons group. Parameters of BigButton(name,
groupName, caption, imageName, toolTip, arrow, styles) method.
var customBigButton = designer.jsObject.BigButton("customButton1", null,
"Custom Button", " ", "Tooltip for customButton1");
customBigButton.image.src = "https://www.stimulsoft.com/images/blocks/
ultimate-buttons/logo.png";

buttonsGroupTable.addCell(customBigButton).style.padding = "2px";

//customBigButton onclick event
customBigButton.action = function () {
   alert("customButton was pressed!");
}

//Add buttonsGroup and separator to customPanel
homePanel.firstChild.addCell(buttonsGroup);
homePanel.firstChild.addCell(designer.jsObject.GroupBlockSeparator());
...
```

Также, вы можете добавить кнопку на верхнюю панель, возле меню **Файл (File)**.

© 2003-2026 Stimulsoft

designer.html

```
...
var designer = new Stimulsoft.Designer.StiDesigner(designerOptions,
"StiDesigner", false);
designer.renderHtml("content");

var toolBarRow = designer.jsObject.options.toolBar.firstChild.tr[0];

var customButton = designer.jsObject.ToolButton("customButton1", "Custom
Button");

var buttonCell = document.createElement("td");
buttonCell.className = "stiDesignerToolButtonCell";
buttonCell.appendChild(customButton);

//For example insert button to position 3
toolBarRow.insertBefore(buttonCell, toolBarRow.childNodes[3]);

customButton.action = function () {
   alert("Button clicked!");
}
...
```

Ниже представлен пример добавления пользовательской панели в дизайнере отчетов.

© 2003-2026 Stimulsoft

designer.html

```
...
var designer = new Stimulsoft.Designer.StiDesigner(designerOptions,
"StiDesigner", false);
designer.renderHtml("content");

var propertiesPanel = designer.jsObject.options.propertiesPanel;

var customPanel = document.createElement("div");
customPanel.jsObject = designer.jsObject;
customPanel.className = "stiDesignerPropertiesPanelInnerContent";
customPanel.style.top = "35px";
customPanel.style.display = "none";
customPanel.style.background = "gray";

propertiesPanel.containers["Custom"] = customPanel;
propertiesPanel.appendChild(customPanel);

var footerTable = propertiesPanel.footer.firstChild;

var tabButton = designer.jsObject.TabButton("CustomTabButton",
"PropertiesGridTabs", "Custom");
tabButton.style.margin = "0 0 0 3px";

tabButton.action = function () {
   if (!this.isSelected) propertiesPanel.showContainer("Custom");
}

designer.jsObject.loc.Panels.Custom = "Custom Panel Name";
...
```

8.3.10 Настройки

Настройка **HTML5 Designer** выполняется при помощи свойств, находящихся в классе **Stimulsoft.Designer.StiDesignerOptions**. Все свойства разделены на группы для удобства использования. Ниже приведен пример установки свойств дизайнера.

© 2003-2026 Stimulsoft

designer.html

```
...
<script type="text/javascript">
   var report = new Stimulsoft.Report.StiReport();
   report.loadFile("SimpleList.mrt");

   var options = new Stimulsoft.Designer.StiDesignerOptions();
   options.appearance.theme =
   Stimulsoft.Designer.StiDesignerTheme.Office2022WhiteBlue;
   options.viewerOptions.appearance.reportDisplayMode =
   Stimulsoft.Report.Export.StiHtmlExportMode.Auto;
   options.toolbar.showFileMenuExit = false;
   options.toolbar.showFileMenuOptions = false;
   options.bands.showChildBand = false;
   options.components.showPanel = false;
   options.appearance.showReportTree = false;
   options.appearance.showTooltips = false;

   var designer = new Stimulsoft.Designer.StiDesigner(options);
   designer.report = report;
</script>
...
```

Главная (без группы)

Наименование	Описание
Width	Задает ширину компонента в “px” или “%”.
Height	Задает высоту компонента в “px” или “%”.

Оформление (Appearance)

Наименование	Описание
theme	Задает тему оформления дизайнера. Список доступных тем оформления находится в перечислении **StiDesignerTheme**. По умолчанию установлено значение **Office2022WhiteBlue**.
iconSet	Предоставляет возможность установить набор иконок: › Stimulsoft.Designer.StiWebUIIconSet.Auto

© 2003-2026 Stimulsoft

	(значение по умолчанию) - устанавливает набор иконок автоматически. Для тем **Office2022** - используется набор иконок в стиле **Monoline**, для тем **Office2013** - используется набор иконок в стиле **Regular**. › Stimulsoft.Designer.StiWebUIIconSet.Monoline - устанавливает набор иконок в стиле **Monoline**; › Stimulsoft.Designer.StiWebUIIconSet.Regular - устанавливает набор иконок в стиле **Regular**.
defaultUnit	Устанавливает единицы измерения размеров для отчета и всех его компонентов: › Stimulsoft.Report.StiReportUnitType.**Centimeters**; › Stimulsoft.Report.StiReportUnitType.**HundredthsOfInch**; › Stimulsoft.Report.StiReportUnitType.**Inches**; › Stimulsoft.Report.StiReportUnitType.**Millimeters**. По умолчанию, установлено значение Stimulsoft.Report.StiReportUnitType.**Centimeters**.
zoom	Задает масштаб отображения страниц отчета. По умолчанию установлен масштаб 100 процентов. Допускаются значения от 10 до 200 процентов. Может принимать одно из указанных ниже значений перечисления **StiZoomMode**: › **PageWidth** – при запуске дизайнера будет установлен масштаб, необходимый для отображения отчета по ширине страницы; › **PageHeight** – при запуске дизайнера будет установлен масштаб, необходимый для отображения отчета по высоте страницы.

© 2003-2026 Stimulsoft

interfaceType	Устанавливает тип интерфейса дизайнера. Могут использоваться следующие значения: › Stimulsoft.Designer.StiInterfaceType.**Auto** – тип интерфейса дизайнера будет выбран автоматически в зависимости от используемого устройства (значение по умолчанию); › Stimulsoft.Designer.StiInterfaceType.**Mouse** – принудительное использование стандартного интерфейса для управления дизайнером при помощи мыши; › Stimulsoft.Designer.StiInterfaceType.**Touch** – принудительное использование Touch интерфейса для управления дизайнером при помощи сенсорного экрана монитора, в этом режиме элементы интерфейса дизайнера имеют увеличенные размеры для удобства управления;
showAnimation	Предоставляет возможность включить или выключить анимацию отображения и закрытия различных меню в дизайнере. По умолчанию, установлено значение **true**.
showSaveDialog	Включает отображение диалога ввода имени отчета при его сохранении. Имя отчета будет передано в параметрах дизайнера. По умолчанию свойство имеет значение **true**.
showTooltips	Включает отображение подсказок для элементов управления дизайнера при наведении мыши. По умолчанию свойство имеет значение **true**.
showTooltipsHelp	Включает отображение ссылки на онлайн документацию в подсказках для элементов управления дизайнера. По умолчанию свойство имеет значение **true**.
showDialogsHelp	Предоставляет возможность отображать или

© 2003-2026 Stimulsoft

	не отображать кнопку вызова справки в различных меню. По умолчанию, установлено значение **true**.
fullScreenMode	Устанавливает полноэкранный режим отображения дизайнера. Если свойство установлено в значение true, то значения свойств width и height игнорируются. По умолчанию, установлено значение **false**.
maximizeAfterCreating	Предоставляет возможность установить максимальный размер дизайнера отчетов. По умолчанию свойство имеет значение **false**.
showLocalization	Предоставляет возможность отображать или не отображать элемент управления локализацией в дизайнере отчете.
allowChangeWindowTitle	Разрешает использование заголовка окна браузера для отображения имени файла редактируемого отчета. По умолчанию свойство имеет значение **true**.
showPropertiesGrid	Включает отображение панели свойств в дизайнере отчетов. По умолчанию свойство имеет значение **true**.
showReportTree	Включает отображение дерева компонентов отчета. По умолчанию свойство имеет значение **true**.
propertiesGridPosition	Предоставляет возможность определить позицию панели свойств - Слева или Справа.
showSystemFonts	Предоставляет возможность отображать или не отображать системные шрифты в списке шрифтов. По умолчанию свойство имеет значение **true**, т.е. системные шрифты отображаются в списке шрифтов.
datePickerFirstDayOfWeek	Предоставляет возможность установить первый день недели для инструмента **Выбор даты (Date picker)**. › Stimulsoft.Viewer.StiFirstDayOfWeek.**Auto** -

© 2003-2026 Stimulsoft

	Будет установлен понедельник или воскресенье как первый день недели, в зависимости от культуры браузера. › Stimulsoft.Viewer.StiFirstDayOfWeek.**Monday** - Понедельник будет установлен как первый день недели. › Stimulsoft.Viewer.StiFirstDayOfWeek.**Sunday** - Воскресенье будет установлен как первый день недели.
formatForDateControls	Предоставляет возможность переопределить формат для элементов управления дат. По умолчанию, значение для текущей опции не задан и формат для дат определяется из локали браузера.
wizardTypeRunningAfterLoad	Предоставляет возможность вызывать мастер создания отчетов после запуска дизайнера отчетов. Может принимать одно из указанных ниже значений перечисления **Stimulsoft.Designer.StiWizardType**: › **None** - дизайнер отчетов будет запущен без вызова мастера создания отчета; › **StandardReport** - дизайнер отчетов будет запущен с вызовом стандартного мастера создания отчета; › **MasterDetailReport** - дизайнер отчетов будет запущен с вызовом master-detail мастера создания отчета; › **LabelReport** - дизайнер отчетов будет запущен с вызовом мастера создания отчета с этикетками; › **InvoicesReport** - дизайнер отчетов будет запущен с вызовом мастера создания инвойса; › **OrdersReport** - дизайнер отчетов будет запущен с вызовом мастера создания ордера; › **QuotationReport** - дизайнер отчетов будет запущен с вызовом мастера создания квоты.

© 2003-2026 Stimulsoft

defaultRibbonType	Предоставляет возможность установить используемый по умолчанию стиль панели инструментов дизайнера. Может принимать одно из указанных ниже значений перечисления: › **StiDesignerRibbonType.Classic** - Будет установлен классический стиль (значение по умолчанию); › **StiDesignerRibbonType.SingleLine** – Будет установлен компактный однострочный стиль.
allowPropagationEvents	Предоставляет возможность разрешить распространение событий нажатия клавиш, когда дизайнер отчетов не в фокусе. По умолчанию свойство имеет значение **true**.
propertiesPanelViewMode	Предоставляет возможность закрепить или открепить панель свойств, словарь отчета, дерево отчета. Может принимать одно из указанных ниже значений перечисления: › Pinned - панели закреплены (значение по умолчангию); › Unpinned - панели откреплены.

Панель инструментов (Toolbar)

Наименование	Описание
visible	Включает отображение панели инструментов в дизайнере отчетов. По умолчанию свойство имеет значение **true**.
showPageButton	Включает отображение кнопки **Страница (Page)** на панели инструментов дизайнера. По умолчанию свойство имеет значение **true**.
showPreviewButton	Включает отображение кнопки **Предварительный просмотр (Preview)** на панели инструментов дизайнера. По умолчанию свойство имеет значение **true**.
showSaveButton	Включает отображение кнопки **Сохранить**

© 2003-2026 Stimulsoft

	(Save) на панели инструментов дизайнера. По умолчанию свойство имеет значение **true**.
showAboutButton	Включает отображение кнопки **О программе (About)** на панели инструментов дизайнера. По умолчанию свойство имеет значение **false**.
showPublishButton	Включает отображение кнопки **Публикация (Publish)** на панели инструментов дизайнера. По умолчанию свойство имеет значение **true**.
showFileMenu	Включает отображение главного меню дизайнера отчетов. По умолчанию свойство имеет значение **true**.
showFileMenuNew	Включает отображение пункта главного меню **Создать (New)**. По умолчанию свойство имеет значение **true**.
showFileMenuOpen	Включает отображение пункта главного меню **Открыть (Open)**. По умолчанию свойство имеет значение **true**.
showFileMenuSave	Включает отображение пункта главного меню **Сохранить (Save)**. По умолчанию свойство имеет значение **true**.
showFileMenuSaveAs	Включает отображение пункта главного меню **Сохранить как (Save As)**. По умолчанию свойство имеет значение **true**.
showFileMenuClose	Включает отображение пункта главного меню **Закрыть (Close)**. По умолчанию свойство имеет значение **true**.
showFileMenuExit	Включает отображение пункта главного меню **Выход (Exit)**. По умолчанию свойство имеет значение **false**.
showFileMenuReportSetup	Включает отображение пункта главного меню **Свойства отчета (Report Setup)**. По умолчанию свойство имеет значение **true**.
showFileMenuOptions	Включает отображение пункта главного меню **Параметры (Options)**. По умолчанию

© 2003-2026 Stimulsoft

	свойство имеет значение **true**.
showFileMenuInfo	Включает отображение пункта главного меню **Сведения (Info)**. По умолчанию свойство имеет значение **true**.
showFileMenuAbout	Включает отображение пункта главного меню **О программе (About)**. По умолчанию свойство имеет значение **true**.
showSetupToolboxButton	Включает отображение кнопки вызова окна настроек боковой панели компонентов отчета. По умолчанию свойство имеет значение **true**.
showNewPageButton	Включает отображение кнопки **Новая страница (New Page)** на панели инструментов. По умолчанию свойство имеет значение **true**.
showNewDashboardButton	Включает отображение кнопки **Новая панель индикаторов (New Dashboard)** на панели инструментов. По умолчанию свойство имеет значение **true**.

Секции (Bands)

Наименование	Описание
showReportTitleBand	Включает отображение секции **Заголовок отчета (Report Title)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
showReportSummaryBand	Включает отображение секции **Итог отчета (Report Summary)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
showPageHeaderBand	Включает отображение секции **Заголовок страницы (Page Header)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.

© 2003-2026 Stimulsoft

showPageFooterBand	Включает отображение секции **Итог страницы (Page Footer)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
showGroupHeaderBand	Включает отображение секции **Заголовок группы (Group Header)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
showGroupFooterBand	Включает отображение секции **Итог группы (Group Footer)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
showHeaderBand	Включает отображение секции **Заголовок данных (Header)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
showFooterBand	Включает отображение секции **Итог данных (Footer)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
showColumnHeaderBand	Включает отображение секции **Заголовок колонки (Column Header)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
showColumnFooterBand	Включает отображение секции **Итог колонки (Column Footer)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
showDataBand	Включает отображение секции **Данные (Data)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
showHierarchicalBand	Включает отображение секции **Дерево (Hierarchical)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.

© 2003-2026 Stimulsoft

showChildBand	Включает отображение секции **Подчиненный (Child)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
showEmptyBand	Включает отображение секции **Пустые данные (Empty)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
showOverlayBand	Включает отображение секции **Водяной знак (Overlay)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
showTable	Включает отображение компонента **Таблица (Table)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.

Кросс секции (Cross-Bands)

Наименование	Описание
showCrossTab	Включает отображение компонента **Кросс-таблица (Cross-Tab)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
showCrossGroupHeaderBand	Включает отображение секции **Кросс-заголовок группы (Cross Group Header)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
showCrossGroupFooterBand	Включает отображение секции **Кросс-итог группы (Cross Group Footer)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
showCrossHeaderBand	Включает отображение секции **Кросс-заголовок данных (Cross Header)** в меню вставки компонентов дизайнера. По

© 2003-2026 Stimulsoft

	умолчанию свойство имеет значение **true**.
showCrossFooterBand	Включает отображение секции **Кросс-итог данных (Cross Footer)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
showCrossDataBand	Включает отображение секции **Кросс-данные (Cross Data)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.

Компоненты (Components)

Наименование	Описание
showText	Включает отображение компонента **Текст (Text)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
showTextInCells	Включает отображение компонента **Текст в ячейках (Text in Cells)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
showRichText	Включает отображение компонента **Rich текст (Rich Text)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **false**.
showImage	Включает отображение компонента **Картинка (Image)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
showBarCode	Включает отображение компонента **Штрих-код (Bar Code)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
showShape	Включает отображение компонента **Геометрия (Shape)** в меню вставки

© 2003-2026 Stimulsoft

	компонентов дизайнера. По умолчанию свойство имеет значение **true**.
showPanel	Включает отображение компонента **Панель (Panel)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
showClone	Включает отображение компонента **Клон (Clone)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **false**.
showCheckBox	Включает отображение компонента **Флажок (Check Box)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
showSubReport	Включает отображение компонента **Вложенный отчет (Sub Report)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.
showZipCode	Включает отображение компонента **Почтовый индекс (Zip Code)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **false**.
showChart	Включает отображение компонента **Диаграмма (Chart)** в меню вставки компонентов дизайнера. Действует сразу на все типы диаграмм. По умолчанию свойство имеет значение **true**.
showSparkline	Включает отображение компонента **Спарклайн (Sparkline)** в меню вставки компонентов дизайнера. По умолчанию свойство имеет значение **true**.

Элементы дашборда (dashboardElements)

Наименование	Описание

© 2003-2026 Stimulsoft

showTableElement	Включает отображение элемента панели индикаторов **Таблица (Table)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
showChartElement	Включает отображение элемента панели индикаторов **Диаграмма (Chart)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
showGaugeElement	Включает отображение элемента панели индикаторов **Измерительный прибор (Gauge)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
showPivotTableElement	Включает отображение элемента панели индикаторов **Сводная таблица (Pivot)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
showIndicatorElement	Включает отображение элемента панели индикаторов **Индикатор (Indicator)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
showProgressElement	Включает отображение элемента панели индикаторов **Прогресс (Progress)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
showRegionMapElement	Включает отображение элемента панели индикаторов **Региональная карта (Region Map)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
showOnlineMapElement	Включает отображение элемента панели

© 2003-2026 Stimulsoft

	индикаторов **Онлайн карта (Online Map)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
showImageElement	Включает отображение элемента панели индикаторов **Изображение (Image)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
showTextElement	Включает отображение элемента панели индикаторов **Текст (Text)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
showPanelElement	Включает отображение элемента панели индикаторов **Панель (Panel)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
showShapeElement	Включает отображение элемента панели индикаторов **Геометрия (Shape)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
showListBoxElement	Включает отображение элемента панели индикаторов **Список (ListBox)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
showComboBoxElement	Включает отображение элемента панели индикаторов **Выпадающий список (ComboBox)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
showTreeViewElement	Включает отображение элемента панели индикаторов **Иерархический список**

© 2003-2026 Stimulsoft

	(TreeView) на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
showTreeViewBoxElement	Включает отображение элемента панели индикаторов **Выпадающий иерархический список (TreeViewBox)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.
showDatePickerElement	Включает отображение элемента панели индикаторов **Выбор даты (Date Picker)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение **true**.

Словарь данных (Dictionary)

Наименование	Описание
showAdaptersInNewConnectionForm	Включает отображение категории **Объекты (Object)** в окне создания нового соединения. По умолчанию свойство имеет значение **true**.
showDictionaryActions	Устанавливает отображение меню **Действия (Actions)** на панели инструментов словаря данных. По умолчанию свойство имеет значение **true**.
showDictionary	Включает отображение словаря данных отчета. По умолчанию свойство имеет значение **true**.
newReportDictionary	Предоставляет возможность создавать новый словарь данных или объединить с существующим, при создании нового отчета в дизайнере. Может принимать одно из указанных ниже значений: • `StiNewReportDictionary.Auto` -

© 2003-2026 Stimulsoft

	определяет режим создания или объединения словаря данных из сохраненного значения в файлах `cookies` (значение по умолчанию); • `StiNewReportDictionary.DictionaryNew` - устанавливает режим создания нового словаря данных, при создании нового отчета; • `StiNewReportDictionary.DictionaryMerge` - устанавливает режим объединения существующего словаря данных с новым, при создании нового отчета в дизайнере.
useAliases	Предоставляет возможность использовать псевдонимы в словаре данных. Может принимать одно из указанных ниже значений: • `StiUseAliases.Auto` - определяет режим использования псевдонимов из сохраненного значения в файлах cookies (значение по умолчанию); • `StiUseAliases.True` - устанавливает режим использования псевдонимов в словаре данных; • `StiUseAliases.False` - выключает режим использования псевдонимов в словаре данных.
dataSourcesPermissions	Устанавливает доступные действия над источниками данных отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
dataConnectionsPermissions	Устанавливает доступные действия над подключениями к данным отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
dataColumnsPermissions	Устанавливает доступные действия над колонками данных отчета. Может принимать

© 2003-2026 Stimulsoft

	одно или несколько значений из перечисления **StiDesignerPermissions**.
dataRelationsPermissions	Устанавливает доступные действия над связями данных отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
businessObjectsPermissions	Устанавливает доступные действия над бизнес-объектами отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
variablesPermissions	Устанавливает доступные действия над переменными отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
resourcesPermissions	Устанавливает доступные действия над ресурсами в словаре данных отчета. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
dataTransformations	Устанавливает доступные действия над преобразованием данных. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.
userFunctions	Устанавливает доступные действия над пользовательскими функциями. Может принимать одно или несколько значений из перечисления **StiDesignerPermissions**.

В таблице ниже приведены все доступные значения для перечисления **Stimulsoft.Designer.StiDesignerPermissions**, которые можно установить для элементов словаря отчета.

Значение	**Описание**
None	Запрещает любое действие над элементом словаря данных.
Create	Разрешает создавать определенный элемент

© 2003-2026 Stimulsoft

	словаря данных.
Delete	Разрешает удалять определенный элемент словаря данных.
Modify	Разрешает редактировать определенный элемент словаря данных.
View	Разрешает просматривать определенный элемент словаря данных.
ModifyView	Разрешает редактировать и просматривать определенный элемент словаря данных.
All	Разрешает любые действия над элементом словаря данных.

8.4 Engine

Эта глава содержит описание работы с ядром JS генератора отчетов.

- Активация
- Подключение файлов данных
- Подключение SQL данных
- Локализация
- Загрузка и сохранение отчета
- Сохранение построенных отчетов
- Получение доступа к страницам
- Добавление пользовательских функций

8.4.1 Активация

YouTube

Посмотрите видеоуроки по активации JS компонентов. Подписывайтесь на канал Stimulsoft и узнайте первыми о новых видеоуроках. Вопросы и предложения оставляйте в комментариях к видео.

После приобретения продукта Stimulsoft вам необходимо активировать лицензию для используемых компонентов. Это можно выполнить указав лицензионный ключ или загрузив файл с лицензионным ключом. Ниже представлен пример активации компонента **StiDesigner** или **StiViewer**.

© 2003-2026 Stimulsoft

index.html

```
...
function Start() {

   //Activation with using license code
   Stimulsoft.Base.StiLicense.Key = "Your activation code...";

   //Activation with using license file
   Stimulsoft.Base.StiLicense.loadFromFile("license.key");
}
...
```

Получить лицензионный ключ или скачать файл с лицензионным ключом, вы можете в личном кабинете пользователя. Для авторизации в кабинете используйте имя пользователя и пароль, указанные при покупке подписки продукта.

Информация

Обратите внимание, что загрузка файла лицензии из локального хранилища будет невозможна из-за политики безопасности браузеров.

8.4.2 Подключение файлов данных

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Параметры подключения к источникам данных обычно хранятся в самом шаблоне отчета. Но при необходимости можно использовать другие способы подключения данных.

Данные из файлов

Объект **DataSet** используется для хранения данных. В нем есть методы для

© 2003-2026 Stimulsoft

загрузки данных из файлов различных форматов. Метод **regData()** используется для подключения данных в отчете, в аргументах которого указывается подготовленный объект **DataSet**.

Данные могут быть загружены из файлов XML с использованием схемы XSD.

index.html

```
...
var dataSet = new Stimulsoft.System.Data.DataSet("SimpleDataSet");
dataSet.readXmlSchemaFile("Demo.xsd");
dataSet.readXmlFile("Demo.xml");

var report = new Stimulsoft.Report.StiReport();
report.regData(dataSet.dataSetName, "", dataSet);
report.dictionary.synchronize();
...
```

Из JSON файлов:

index.html

```
...
var dataSet = new Stimulsoft.System.Data.DataSet("SimpleDataSet");
dataSet.readJsonFile("Demo.json");

var report = new Stimulsoft.Report.StiReport();
report.regData(dataSet.dataSetName, "", dataSet);
report.dictionary.synchronize();
...
```

Дополнительные методы загрузки данных из файлов:

Объект **DataSet** имеет широкий набор методов для загрузки данных:

- readJsonFile(fileName) – загрузка JSON файла по указанному пути;
- readJson(string) – загрузка JSON данных в виде строки;
- readJson(data) – загрузка JSON данных в виде байтового массива;
- readJson(obj) – использование JavaScript объекта в качестве данных;

- readXmlFile(fileName) – загрузка XML файла по указанному пути;
- readXml(string) – загрузка XML данных в виде строки;
- readXml(data) – загрузка XML данных в виде байтового массива;

© 2003-2026 Stimulsoft

- readXmlSchemaFile(fileName) – загрузка XSD файла по указанному пути;
- readXmlSchema(string) – загрузка XSD схемы данных в виде строки;
- readXmlSchema(data) – загрузка XSD схемы данных в виде байтового массива.

Информация

Загрузка схемы данных не является обязательной. Если вы хотите использовать схему данных, вам следует добавить ее перед загрузкой данных XML.

8.4.3 Подключение SQL данных

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Примеры

Ознакомьтесь с примерами по использованию адаптеров данных.

Поскольку чистый JavaScript не имеет встроенных методов для работы с удаленными базами данных, эта функциональность реализуется с помощью серверного кода. Поэтому продукт Stimulsoft Reports.JS содержит серверные адаптеры данных, реализованные с использованием технологий Node.js, PHP, .NET, .NET Core, Python, Java.. Адаптер базы данных - это программный уровень между СУБД и клиентским скриптом. Адаптер подключается к СУБД и извлекает необходимые данные, конвертируя их в JSON. Сценарий, запущенный на сервере (с использованием адаптера), обеспечивает обмен данными между клиентским приложением JavaScript и серверной частью. Чтобы использовать этот механизм на стороне клиента, достаточно указать URL-адрес хоста адаптера, который обрабатывает запросы к необходимому адаптеру.

© 2003-2026 Stimulsoft

Ссылки на примеры с готовыми адаптерами данных, реализованные для различных платформ: Node.js, PHP, .NET, .NET Core, Python, Java.

Использовать адаптер достаточно просто. Вы должны запусть адаптер и указать адрес адаптера:

index.html

```
...
StiOptions.WebServer.url = "http://localhost:9615";
...
```

При запросе данных из SQL источников данных, JS report engine отправляет POST запрос на URL, указанный в опции:

index.html

```
...
StiOptions.WebServer.url = "https://localhost/handler.php";
...
```

В теле запроса передается JSON объект с параметрами, которые используют указанную ниже структуру:

- **command**: возможны два варианта - "TestConnection" и "ExecuteQuery";
- **connectionString**: строка подключения к базе;
- **queryString**: строка запроса;
- **database**: тип базы данных;
- **timeout**: время ожидания запроса, указанное в источнике данных;
- **parameters**: масив параметров в виде JSON объекта {name, value};
- **escapeQueryParameters**: флаг экранирования параметров перед выполнением запроса.

В ответ JS report engine одидает JSON объект с данными в виде следующей структуры:

- **success**: флаг успешного выполнения команды;
- **notice**: если флаг выполенния команты имеет значение **false**, то данный параметр будет содержать описание ошибки;
- **rows**: масив строк, каждый элемент - это масив из значений, индексом является номер колонки;

© 2003-2026 Stimulsoft

- **columns**: масив имен колонок, индексом является номер колонки;
- **types**: масив типов колонок, индексом является номер колонки. Может принимать значения "string", "number", "int", "boolean", "array", "datetime".

Пример запроса и ответа:

index.html

```
...
request = {
   command: "ExecuteQuery",
   connectionString: "Server=myServerAddress;Database=myDataBase;User
   Id=myUsername;Password=myPassword;",
   queryString: "select * from table1",
   database: "MS SQL"
}

response = {
   success: true,

   rows: [
      ["value1", 1, false],
      ["value2", 1, true]
      ["value3", 2, false]
   ],
   columns: [
      "Column1_name",
      "Column2_name",
      "Column3_name"
   ],
   types:[
      "string",
      "int",
      "boolean"
   ]
}
...
```

Custom Data Base

Также предусмотрена возможность зарегистрировать собственный адаптер данных. Для этого необходимо вызвать функцию:

index.html

```
...
Stimulsoft.Report.Dictionary.StiCustomDatabase.registerCustomDatabase(opti
ons);
...
```

© 2003-2026 Stimulsoft

Опции представляют собой набор свойств и функцию **process()**, которая будет вызываться при запросе данных:

- **serviceName**: имя адаптера которое отобразится в дизанере при создании нового подключения
- **designerCategory**: наименование категории, в которую будет добавлен адаптер в меню **Новый источник данных**. По умолчанию, адаптер отображается в категории `Избранное (Favorites)`, но можно изменить ее, установив опцию в одно из следующих значений: `"Files"`, `"SQL"`, `"NoSQL"`, `"Azure"`, `"Google"`, `"OnlineServices"`, `"REST"`.
- **sampleConnectionString**: пример строки подключения который вставиться в форме настройки нового подключения
- **process**: функция которая вызовется для подготовки и передачи данных в Stimulsoft.Report.Engine

На вход функции **process()** передаются два аргумента: **command** и **callback**.

Аргумент **command** представляет собой JSON объект, в который JS report engine передаст следующие параметры:

- command: действие, которое вызывается в данный момент. Возможные значения: "TestConnection": проверка соединения с базой данных из формы создания нового подключения "RetrieveSchema": извлечение схемы данных, нужно для оптимизации запроса и не передачи только необходимого набора данных. Вызывается после создания подключения "RetrieveData": запрос данных
- connectionString: строка подключения к базе
- queryString: строка SQL запроса
- database: тип базы данных
- timeout: время ожидания запроса, указанное в источнике данных

Аргумент **callback** является функцией, которую нужно вызвать для передачи подготовленных данных в JS report engine. В качестве аргументаcallback функции необходимо передать JSON объект, имеющий указанные ниже параметры:

- **success**: флаг успешного выполнения команды
- **notice**: если флаг выполенния команты имеет значение **false**, то данный параметр должен содержать описание ошибки
- **rows**: масив строк, каждый элемент - это масив из значений, индексом является номер колонки
- **columns**: масив имен колонок, индексом является номер колонки

© 2003-2026 Stimulsoft

› **types**: объект где имя поля это имя колонки а значение тип колонки {Column_Name : "string"}. Тип может принимать значения "string", "number", "int", "boolean", "array", "datetime". Если будет передан масив **columns**, то в **types** можно передать масив типов, индексом должен являться номер колонки. Не работает для "**RetrieveSchema**"

Если **command** = **"RetrieveSchema"**, то помимо типов, в types необходимо передать имена таблиц.

Пример запроса и ответа при получении схемы:

index.html

```
...
request = {
   command: "RetrieveSchema"
}

response = {
   success: true,

   types:{
      Table1: {
         Column1: "string",
         Column2: "number"
      },
      Table2: {
         Column1: "string"
      }
   }
}
...
```

Пример запроса и ответа при получении данных:

© 2003-2026 Stimulsoft

index.html

```
...
request = {
   command: "RetrieveData",
   queryString: "Table1"
}
response = {
   success: true,

   rows: [
      ["value1", 1],
      ["value2", 1]
      ["value3", 2]
   ],
   columns:[
      "Column1",
      "Column2"
   ],
   types:[
      "string",
      "number"
   ]
}
...
```

Пример регистрации адаптера.

Время ожидания запроса

Кроме того, для источников данных SQL, используемых в отчете, вы можете указать **Время ожидания запроса (Query Timeout)** в секундах. Значение этого свойства сохраняется в шаблоне отчета для каждого SQL-соединения отдельно.

Ниже приведен пример кода, который можно использовать для изменения строки подключения для MS SQL, настройки запроса, установки времени ожидания запроса для уже созданного подключения и источников данных в отчете.

© 2003-2026 Stimulsoft

index.html

```
...
var report = new Stimulsoft.Report.StiReport();
report.loadFile("Report.mrt");
report.dictionary.databases.getByName("Connection").connectionString =
"Data Source=server;Integrated Security=True;Initial Catalog=DataBase";
report.dictionary.dataSources.getByName("DataSourceName").sqlCommand =
"select * from Table where Column = 100";
report.dictionary.dataSources.getByName("DataSourceName").commandTimeout =
1000;
...
```

Примеры

Адрес адаптера данных должен быть установлен до кода создания компонента или объекта отчета, поскольку значение этой опции должно быть известно механизму отчетов до его инициализации.

В настоящее время поддерживаются следующие источники данных SQL - MySQL, MS SQL, PostgreSQL, Firebird и Oracle. Адаптеры данных имеют открытый исходный код и могут быть изменены по вашему желанию.

OData источники данных

Вы также можете использовать данные для создания отчетов и информационных панелей, полученные из хранилищ OData. В этом случае вы можете выполнить авторизацию, используя имя пользователя, пароль пользователя или токен. Параметры авторизации указываются в строке подключения к хранилищу OData с помощью символа ";" разделитель.

© 2003-2026 Stimulsoft

viewer.html

```
...
//Authorization using a user account
var oDataDatabase = new
Stimulsoft.Report.Dictionary.StiODataDatabase("OData", "OData", "https://
services.odata.org/V4/Northwind/
Northwind.svc;AddressBearer=adress;UserName=UserName;Password=Password;Cli
ent_Id=Your Client ID", false, null);

//Authorization using a user token
var oDataDatabase = new
Stimulsoft.Report.Dictionary.StiODataDatabase("OData", "OData", "https://
services.odata.org/V4/Northwind/Northwind.svc;Token=Enter your token",
false, null);

report.dictionary.databases.add(oDataDatabase);
report.dictionary.synchronize();

//Query with data filter
var productsDataSource =
report.dictionary.dataSources.getByName("Products");
if (productsDataSource != null) productsDataSource.sqlCommand =
"Products?$filter=ProductID eq 2";
...
```

Также, вы можете указывать пользовательские HTTP headers для источников данных. Это можно выполнить в обработчике события `onBeginProcessData`. Полный пример представлен на нашем сайте:

index.html

```
...
// In `onBeginProcessData` event handler add custom HTTP headers
report.onBeginProcessData = function (args) {
  if (
    args.database === "JSON" &&
    args.command === "GetData" &&
    args.pathData && args.pathData.include("/reports/ProtectedDemo.json")
  ) {
    // Add custom header to pass through backend server protection
    args.headers.push({key: "X-Auth-Token", value: "*YOUR TOKEN*"});
  }
};
...
```

8.4.4 Локализация

Компоненты **HTML5 Viewer** и **HTML5 Designer** поддерживает полную локализацию своего интерфейса. Для локализации интерфейса дизайнера отчетов на требуемый язык, необходимо воспользоваться специальным

© 2003-2026 Stimulsoft

статическим методом **addLocalizationFile()**. В качестве аргументов метода необходимо указать путь к XML-файлу локализации, а также определить, будет ли локализация автоматически загружаться вместе с компонентом.

Указанная функция добавляет локализацию в меню дизайнера:

index.html

```
...
Stimulsoft.Base.Localization.StiLocalization.addLocalizationFile("../
localization/de.xml", true);
...
```

Указанная функция добавляет локализацию в меню дизайнера, но не загружает её автоматически:

index.html

```
...
Stimulsoft.Base.Localization.StiLocalization.addLocalizationFile("../
localization/de.xml", false, "Deutsch");
...
```

После вызова метода, ссылка на локализационный файл будет добавлена во внутреннюю коллекцию. Сам файл будет загружен при первом выборе указанной локализации в меню дизайнера, это экономит память и время запуска компонентов. В данном варианте, в качестве третьего аргумента необходимо указать название локализации для отображения в меню дизайнера, после загрузки локализации это значение будет взято из локализационного файла.

Указанная функция добавляет локализацию в меню дизайнера, и устанавливает её локализацией по-умолчанию:

index.html

```
...
Stimulsoft.Base.Localization.StiLocalization.setLocalizationFile("../
localization/de.xml");
...
```

© 2003-2026 Stimulsoft

При каждой загрузке дизайнера будет выбрана именно эта локализация, даже если ранее в меню была выбрана другая локализация.

Информация

При создании проекта с чистым JavaScript, необходимые локализационные файлы необходимо добавить при помощи указанного выше кода. Если в проекте используется технология Node.js, то достаточно указать папку с локализационными файлами, в этом случае все находящиеся в ней файлы будут добавлены автоматически:

```
Stimulsoft.System.NodeJs.localizationPath = "locales";
```

8.4.5 Загрузка и сохранение отчета

Информация

Поскольку в дашбордах и отчетах используется один и тот же унифицированный формат шаблона - MRT, методы загрузки шаблона и работы с данными, в тексте документации будет использовано слово «отчет».

Для загрузки отчета можно применять один из методов объекта **StiReport** - **loadFile()** и **load()**. Они используются следующим образом:

- loadFile(filePath) – загружает отчет из mrt файла, путь к которому указан в переменной filePath;
- load(str) – загружает отчет из строки, которая содержит **XML** или **JSON**;
- load(data) – загрузка отчета из массива данных типа number[];
- load(xml) – загрузка отчета из XML файла типа **XMLDocument**;
- load(json) – загрузка отчета из **JS** объекта.

Например, используйте приведенный ниже код для загрузки отчета из файла:

index.html

```
...
var report = new Stimulsoft.Report.StiReport();
report.loadFile("SimpleList.mrt");
...
```

© 2003-2026 Stimulsoft

Файл формата **MRT** Stimulsoft Reports - это описание отчетов с разметкой JSON. Вы можете использовать **MRT**-файлы, созданные в других дизайнерах Stimulsoft Reports, с разметкой JSON. Используйте приведенный ниже код, чтобы сохранить отчет в виде строки:

index.html

```
...
var report = new Stimulsoft.Report.StiReport();
var jsonString = report.saveToJsonString();
...
```

Используйте приведенный ниже код, чтобы загрузить отчет из этой строки:

index.html

```
...
var report = new Stimulsoft.Report.StiReport();
report.load(jsonString);
...
```

8.4.6 Сохранение построенных отчетов

Построенный отчет можно сохранить в документ JSON для последующего просмотра (данные хранятся в документе):

index.html

```
...
var report = new Stimulsoft.Report.StiReport();
report.loadFile("SimpleList.mrt");
report.renderAsync(function(){
   var jsonString = report.saveDocumentToJsonString();
});
...
```

8.4.7 Получение доступа к страницам

Получение доступа к страницам отчета

В отчете есть набор страниц, к которым можно получить доступ следующим образом:

© 2003-2026 Stimulsoft

index.html

```
...
var report = new Stimulsoft.Report.StiReport();
report.loadFile("SimpleList.mrt");

for (var index = 0; index < report.pages.count; index++) {
   alert(report.pages.getByIndex(index).name);
}
...
```

Получение доступа к страницам готового отчета

После построения отчета создается набор страниц. Доступ к нему можно получить следующим образом:

index.html

```
...
var report = new Stimulsoft.Report.StiReport();
report.loadFile("SimpleList.mrt");
      K      ^     E         EF
   for (var index = 0; index < report.renderedPages.count; index++) {
      alert(report.renderedPages.getByIndex(index).name);
   }
 FX
...
```

8.4.8 События отчета

Информация

В следствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово «отчет».

Объект **StiReport** поддерживает события, которые предоставляют возможность выполнить необходимые операции перед определенными действиями. Ниже представлены события объекта отчета с описанием и некоторыми примерами их использования:

onPrepareVariables

© 2003-2026 Stimulsoft

Асинхронное событие вызывается в начале построения отчета перед заполнением переменных в отчете. В таблице ниже представлен список аргументов обработчика событий:

Наименование	Описание
sender	Идентификатор компонента, который инициировал данное событие.
event	Идентификатор текущего события, имеет значение **PrepareVariables**.
report	Объект отчета **StiReport**.
preventDefault	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события. По умолчанию установлено значение **false**.
async	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события до выполнения callback функции. По умолчанию установлено значение **false**.

designer.html

```
...
report.onPrepareVariables = (args, callback) => {
   args.variables[0].value = "Replace value";
}
...
```

onBeginProcessData

Асинхронное событие вызывается перед запросом данных, необходимых для построения отчета. В таблице ниже представлен список аргументов обработчика событий:

Наименование	Описание
sender	Идентификатор компонента, который инициировал данное событие.

© 2003-2026 Stimulsoft

event	Идентификатор текущего события, имеет значение **BeginProcessData**.
report	Объект отчета **StiReport**.
preventDefault	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события. По умолчанию установлено значение **false**.
async	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события до выполнения callback функции. По умолчанию установлено значение **false**.
command	Идентификатор текущей команды, может иметь значения **TestConnection** (проверка соединения) и **ExecuteQuery** (получения данных из хранилища).
database	Предоставляет возможность указать наименование текущей базы данных.
connection	Имя текущего подключения к источнику данных, заданное в шаблоне отчета.
headers	Предоставляет возможность указать заголовки в запросах.
withCredentials	Предоставляет возможность указать cookies в запросах.
connectionString	Предоставляет возможность указать строку подключения к хранилищу данных.
dataSource	Предоставляет возможность указать наименование источника данных.
queryString	SQL запрос на получение данных. Используется только с командой **ExecuteQuery**.
timeout	Предоставляет возможность указать таймаут запроса.
parameters	Предоставляет возможность указать

© 2003-2026 Stimulsoft

	параметр в запросе.
escapeQueryParameters	Предоставляет возможность использовать экранированную часть запроса. По умолчанию установлено значение **true**.
pathData	Предоставляет возможность указать путь к файлу данных.
tryParseDateTime	Предоставляет возможность определить возможность преобразования данных в тип DateTime.
relationDirection	Предоставляет возможность изменить направление связи между источниками данных.
pathSchema	Предоставляет возможность указать путь к XSD файлу со схемой данных.
firstRowIsHeader	Предоставляет возможность использовать первую строку как заголовки данных в Excel источниках данных.
collectionName	Наименование коллекции в OData источнике данных.
separator	Предоставляет возможность указать разделитель в CSV источнике данных.
dataType	Предоставляет возможность указать тип данных для GIS источника данных.
codePage	Предоставляет возможность определить кодировку данных в CSV и DBF источниках данных.

© 2003-2026 Stimulsoft

designer.html

```
...
//Replace connection string
report.onBeginProcessData = (args) => {
   if (args.database == "MySQL")
      args.connectionString = "new connection string";
}
...

//Add a some data
report.onBeginProcessData = (args, callback) => {
   if (args.database == "MySQL"){
      args.preventDefault = true;
      var result = {
         success: true,
         rows: [
            ["value1", 1, false],
            ["value2", 1, true],
            ["value3", 2, false]
         ],
         columns: [
            "Column1_name",
            "Column2_name",
            "Column3_name"
         ],
         types:[
            "string",
            "int",
            "boolean"
         ]
      }

      // https://github.com/stimulsoft/DataAdapters.JS/
      callback(result);
   }
}
...
```

onEndProcessData

Событие вызывается после получения данных, необходимых для построения отчета. В таблице ниже представлен список аргументов обработчика событий:

Наименование	Описание
sender	Идентификатор компонента, который инициировал данное событие.
event	Идентификатор текущего события, имеет значение **EndProcessData**.

© 2003-2026 Stimulsoft

report	Объект отчета **StiReport**.
command	Предоставляет возможность указать тип команды.
dataSource	Предоставляет возможность указать наименование источника данных.
connection	Имя текущего подключения к источнику данных, заданное в шаблоне отчета.
database	Предоставляет возможность указать наименование текущей базы данных.
result	Результирующий набор данных в определенном JSON формате. Он имеет две коллекции - колонки и строки с их значениями.

designer.html

```
...
report.onEndProcessData = (args) => {
   if (args.command == "ExecuteQuery" && args.dataSource == "Categories")
      args.result.rows.push(rowData) ;
      // https://github.com/stimulsoft/DataAdapters.JS/
}
...
```

onBeginRender

Событие вызывается в начале построения отчета. Событие не актуально для дашбордов.

onRendering

Событие вызывается в процессе (при создании каждой новой страницы) построения отчета. Событие не актуально для дашбордов.

onEndRender

© 2003-2026 Stimulsoft

Событие вызывается в конце построения отчета. Событие не актуально для дашбордов.

onExportingRender

Событие вызывается перед экспортом отчета или дашборда.

onExportedRender

Событие вызывается после экспорта отчета или дашборда.

onPrinting

Событие возникает при вызове методов `report.print()` или `report.printToPdf()`. Событие не актуально для дашбордов.

viewer.html

```
...
//Remove image before report printing
report.onPrinting = () => {
   var page = report.renderedPages.getByIndex(0);
   var image = page.components.getByName("Image1");
   if (image)
      page.components.remove(image);
}
...
```

onPrinted

Асинхронное событие возникает при вызове методов `report.print()` или `report.printToPdf()` после экспорта отчета в HTML или PDF файл (в зависимости от вызываемого метода). Событие не актуально для дашбордов. В таблице ниже представлен список аргументов обработчика событий:

Наименование	Описание
sender	Идентификатор компонента, который инициировал данное событие.

© 2003-2026 Stimulsoft

event	Идентификатор текущего события, имеет значение **Printed**.
report	Объект отчета **StiReport**.
preventDefault	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события. По умолчанию установлено значение **false**.
async	Этот идентификатор предоставляет возможность остановить дальнейшую обработку события до выполнения callback функции. По умолчанию установлено значение **false**.
data	Экспортированные данные в виде строки или байтового массива.

viewer.html

```
...
//Stop to report print and define a custom print method
report.onPrinted = (args) => {
   args.preventDefault = true;
   var printData = args.data;
   myPrintingMethod(printData);
}
...
```

onRefreshing

Событие возникает после построения отчета, если свойство **report.refreshTime** установлено больше нуля. Также событие возникает при нажатии кнопки **Обновить (Refresh)** при просмотре отчета или дашборда.

8.4.9 Добавление пользовательских функций

При интеграции дизайнера отчетов в пользовательское приложение, присутствует возможность добавления custom функции в словарь данных дизайнера отчетов. После добавления функции, её можно использовать при разработке отчетов и панелей индикаторов (дашбордов). Ниже представлен пример добавления функции расчета итога суммирования.

© 2003-2026 Stimulsoft

index.html

```
...
var myFunc = function (value) {
   if (!Stimulsoft.Data.Extensions.ListExt.isList(value))
   return Stimulsoft.Base.Helpers.StiValueHelper.tryToNumber(value);

   return
   Stimulsoft.Data.Functions.Funcs.skipNulls(Stimulsoft.Data.Extensions.Lis
   tExt.toList(value))
   .tryCastToNumber()
   .sum();
};

Stimulsoft.Report.Dictionary.StiFunctions.addFunction("MyCategory",
"MySum", "MySum", "MySum", "", Number, "Return Description", [Object],
["value"], ["Descriptions"], myFunc);
...
```

9 Reports and Dashboards for PHP

Продукты Stimulsoft Reports.PHP и Stimulsoft Dashboards.PHP представляют собой комбинацию JavaScript и PHP скриптов, а также весь необходимый функционал для их взаимодействия. Построение и экспортирование отчета выполняется при помощи JavaScript ядра на стороне клиента в окне Web браузера, либо на стороне сервера с использованием универсальной платформы Node.js. Серверная сторона PHP содержит все необходимое для работы с файлами отчетов и связи с различными SQL источниками данных. Связь клиента с сервером осуществляется посредством AJAX запросов, передающих и получающих JSON данные в определенном формате. Для удобства использования продукта разработаны специальные события и функции как на стороне JavaScript клиента, включая платформу Node.js, так и на стороне PHP сервера.

В состав продукта входит сам генератор отчетов и ядро анализа данных, а так же компоненты вьювер и дизайнер отчетов.

9.1 Генератор отчетов

Генератор отчетов Stimulsoft Reports.PHP позволяет загружать, выполнять построение и экспортирование отчета в различные форматы без развертывания вьювера и дизайнера. Это позволяет избавиться от необходимости загрузки на сторону клиента емких скриптов.

© 2003-2026 Stimulsoft

Построение и экспортирование отчета выполняется при помощи JavaScript ядра на стороне клиента в окне Web браузера, либо на стороне сервера с использованием универсальной платформы Node.js. Серверная сторона PHP содержит все необходимое для работы с файлами отчетов и связи с различными SQL источниками данных. Связь клиента с сервером осуществляется посредством AJAX запросов, передающих и получающих JSON данные в определенном формате. Для удобства использования продукта разработаны специальные события и функции как на стороне JavaScript клиента, так и на стороне PHP сервера.

Аналитические панели

Аналитические панели Stimulsoft Dashboards.PHP позволяют загружать, выполнять анализ данных и экспортирование дашборда в различные форматы без развертывания вьювера и дизайнера.

Весь анализ данных, за исключением определенных SQL операций, выполняется при помощи JavaScript анализатора данных на стороне клиента в окне Web браузера, либо на стороне сервера с использованием универсальной платформы Node.js. Серверная сторона PHP содержит всё необходимое для работы с файлами дашбордов и связи с различными SQL источниками данных. Связь клиента с сервером осуществляется точно таким же способом, как и в генераторе отчетов, применяются те же события и функции. Серверная сторона PHP является универсальной для дашбордов и отчетов, применяются те же события и функции.

- Использование
- Подключение SQL адаптеров данных
- Уменьшение времени загрузки скриптов
- Работа с переменными отчета
- Активация лицензии
- Подключение пользовательских шрифтов
- Загрузка и сохранение отчета
- Печать отчета из кода
- Обработчик событий PHP
- Экспорт отчета из кода
- Подключение файлов данных
- Printing Report from Code

© 2003-2026 Stimulsoft

9.1.1 Использование

Для использования продукта, достаточно загрузить ZIP архив продукта со страницы Downloads нашего сайта, распаковать его, и скопировать содержимое папки **/PHP** на ваш Web сервер. Данная папка представляет собой Web проект, который содержит все необходимые файлы и ресурсы для работы продукта, а также примеры по работе с вьювером и дизайнером.

Для установки продукта в уже существующий проект, достаточно из папки **/PHP** скопировать папку `/vendor` в корневую директорию проекта, либо воспользоваться менеджером зависимостей Composer, выполнив следующую консольную команду:

console

```
composer require stimulsoft/reports-php
```

Для работы дашбордов понадобится подключить следующий пакет:

console

```
composer require stimulsoft/dashboards-php
```

При работе с продуктом, в большинстве случаев достаточно использовать только PHP код, который обеспечивает работу всех основных возможностей. Для более детальной настройки продукта и использования всех возможностей, необходимо использовать JavaScript код.

Для использования компонентов в Web проекте, достаточно подключить автоматический загрузчик скриптов в начале PHP файла. После этого, можно использовать все доступные PHP классы и функции для работы с отчетами и дашбордами:

© 2003-2026 Stimulsoft

index.php

```
<?php
   require_once 'vendor/autoload.php';
  ...
?>
```

Для работы с генератором отчетов, предназначен класс `StiReport`. При помощи данного класса можно загрузить шаблон либо документ отчета, выполнить построение и экспортирование, обрабатывать запросы и управлять событиями генератора отчетов. Например, требуется загрузить отчет из файла и выполнить его построение, после чего вывести сообщение в окне браузера:

index.php

```
<?php
   require_once 'vendor/autoload.php';

   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->onAfterRender = 'afterRender';
   $report->process();
   $report->loadFile('reports/SimpleList.mrt');
   $report->render();
?>

<html>
   <head>
      <?php
         $report->javascript->renderHtml();
      ?>
      <script type="text/javascript">
         function afterRender() {
            alert('Done!');
         }
      </script>
   </head>
   <body>
      <?php
         $report->renderHtml();
      ?>
   </body>
</html>
```

Полный код примера доступен на GitHub.

© 2003-2026 Stimulsoft

В этом примере последовательно выполняются следующие действия:

- создается экземпляр объекта `StiReport`;
- добавляются необходимые обработчики событий;
- выполняется обработка текущего запроса;
- выполняется загрузка шаблона отчета из файла `SipmleList.mrt`;
- вызывается команда построения отчета;
- в шаблоне HTML файла добавляется код JavaScript события и выполняется вывод необходимого JavaScript и HTML кода компонента.

Метод `$report->process()` выполняет обработку текущего запроса, и, в случае успеха, автоматически возвращает результат на сторону клиента. Более подробно про это рассказано в разделе Обработчик событий.

Метод `$report->javascript->renderHtml()` выполняет вывод кода для подключения необходимых скриптов компонента. Метод `$report->renderHtml()` выполняет вывод JavaScript и HTML кода самого компонента.

Наши продукты Stimulsoft Reports.PHP и Stimulsoft Dashboards.PHP не имеют нативного ядра генератора отчетов на PHP, построение и экспорт отчета выполняется при помощи JavaScript кода на стороне клиента, либо на стороне сервера с использованием платформы Node.js. Поэтому, при использовании PHP кода для работы с компонентами, необходимо вызвать один из специальных методов, который добавит на Web страницу соответствующий JavaScript код для выполнения всех необходимых действий, и HTML код визуальной части компонента.

Наименование	Описание
`getHtml($mode = StiHtmlMode::HtmlScripts)`	Возвращает JavaScript и HTML код, необходимый для работы компонента, со всеми необходимыми действиями над отчетом. Параметр `$mode` позволяет установить варианты возвращаемого кода: • `StiHtmlMode::Scripts` – только необходимый JavaScript код для вставки в блок `<script></script>` на

© 2003-2026 Stimulsoft

	HTML странице; • `StiHtmlMode::HtmlScripts` – необходимый JavaScript и HTML код, для вставки в HTML элемент на странице; • `StiHtmlMode::HtmlPage` – полностью подготовленная HTML страница. **Примечание**: у элементов компонента, например объекта `$report->javascript`, метод `getHtml()` не принимает параметров, поскольку имеет только один вариант вывода кода.
`renderHtml($elementId = null)`	Выводит JavaScript и HTML код, необходимый для работы компонента, со всеми необходимыми действиями над отчетом. Параметр `$elementId` позволяет указать ID элемента HTML страницы, внутри которого будет выведен компонент. По умолчанию, вывод осуществляется в текущем месте страницы.
`printHtml()`	Выводит полностью подготовленную HTML страницу со всеми необходимыми скриптами для работы компонента. Текущий HTML шаблон страницы полностью игнорируется. Данный режим отлично подходит для полноэкранного просмотра либо редактирования отчета.

Таким образом, указанные выше методы позволяют отобразить компонент различными способами в зависимости от требований. Пример упрощённого

© 2003-2026 Stimulsoft

отображения компонента без использования HTML шаблона страницы:

index.php

```
<?php
   require_once 'vendor/autoload.php';

   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->onAfterRender = "alert('Done!');";
   $report->process();
   $report->loadFile('reports/SimpleList.mrt');
   $report->render();
   $report->printHtml();
?>
```

Информация

При использовании платформы Node.js для построения отчета на стороне PHP сервера, указанные методы будут вызваны автоматически внутри обработчика, их явное использование не требуется.

Управление URL для загрузки JavaScript файлов генератора отчетов

По умолчанию все JavaScript файлы продукта загружаются по URL относительно расположения текущего PHP скрипта. В некоторых случаях это поведение требуется изменить, для этого предусмотрено несколько опций. Для использования абсолютного пути для загрузки всех скриптов продукта, необходимо установить опцию `useRelativeUrls` в значение `false`:

index.php

```
<?php
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->javascript->useRelativeUrls = false;
?>
```

© 2003-2026 Stimulsoft

Для корректировки относительного пути предусмотрена опция `relativePath`, для которой необходимо задать строковое значение, которое будет использовано при формировании URL адреса загрузки скриптов. При этом опция `useRelativeUrls` должна быть включена (использовано значение по умолчанию).

index.php

```
<?php
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->javascript->relativePath = '../../';
?>
```

По умолчанию скрипты загружаются как статические файлы. Для использования динамической загрузки скриптов при помощи PHP обработчика, необходимо установить опцию `useStaticUrls` в значение `false`:

index.php

```
<?php
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->javascript->useStaticUrls = false;
?>
```

Информация

Компоненты вьювер отчетов и дизайнер отчетов также имеют свойство `javascript`, при помощи которого можно управлять настройкой скриптов указанным выше способом.

Различные варианты развертывания и оптимизации рассмотрены в разделе Уменьшение времени загрузки скриптов.

© 2003-2026 Stimulsoft

9.1.2 Уменьшение времени загрузки скриптов

Из-за внушительной функциональности продукта, скрипты имеют достаточно большой размер. При первой загрузке Web приложения, либо при отключенном кэшировании браузера, загрузка может занимать некоторое время, особенно при низкоскоростном интернет-соединении. Мы предлагаем два варианта решения этой проблемы: использовать упакованные скрипты, либо использовать частичный функционал, и загружать только то, что требуется. Допускается одновременное использование указанных вариантов.

Упакованные скрипты

Упакованные скрипты имеют такую же структуру, что и обычные скрипты, но имеют окончание `*.pack.js` в имени файла. Такие скрипты содержат блок упакованных данных в виде JavaScript переменной, и компактный распаковщик. При загрузке всех скриптов, распаковщик автоматически распаковывает все загруженные данные, и запускает подготовленный скрипт на выполнение. Распаковка занимает некоторое время, но при определенных обстоятельствах - например, при медленном интернет-соединении - это время намного меньше, чем скорость загрузки обычных скриптов.

Для использования упакованных скриптов всё, что вам понадобится сделать - это установить свойство `$js->usePacked` в значение `true`, например:

index.php

```
<?php
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->javascript->usePacked = true;
?>
```

Частичная загрузка скриптов

Скрипт `stimulsoft.reports.js` содержит в себе весь функционал по построению и экспортированию отчетов. Если для генерации отчетов вам требуются только некоторые из возможностей, предусмотрена загрузка только необходимых частей генератора, содержащих определенный набор возможностей. Например, если в ваших отчетах не используются карты, то их можно не загружать. Это ускорит загрузку Web проекта и уменьшит

© 2003-2026 Stimulsoft

потребление памяти браузером.

Информация

Данная возможность реализована только для ядра генератора отчетов, вьювер и дизайнер нельзя разделить на части, их скрипты будут загружаться целиком одним блоком.

Для использования частичной загрузки скриптов, достаточно установить нужные опции для свойства `javascript` у объекта отчета:

index.php

```
<?php
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->javascript->reportsChart = true;
   $report->javascript->reportsExport = true;
   $report->javascript->reportsImportXlsx = true;
   $report->javascript->reportsMaps = false;
   $report->javascript->blocklyEditor = false;
?>
```

Каждая опция управляет загрузкой скрипта, содержащего определённый функционал. В данной таблице представлен весь набор скриптов, которые можно загрузить раздельно:

Наименование	Описание
`javascript->reportsExport`	Содержит алгоритмы для экспортирования построенного отчета в различные форматы - PDF, HTML, Excel, RichText и другие.
`javascript->reportsChart`	Содержит компоненты для работы со всеми типами диаграмм в отчете.

© 2003-2026 Stimulsoft

`javascript->reportsMaps`	Содержит компоненты для работы с региональными и онлайн картами.
`javascript->blocklyEditor`	Содержит визуальный редактор Blockly для создания скриптов событий в отчете. Сам обработчик событий встроен в движок отчетов.
`javascript->reportsImportXlsx`	Содержит алгоритмы для работы с Excel источниками данных.

Информация

Компоненты вьювер отчетов и дизайнер отчетов также имеют свойство `javascript`, при помощи которого можно управлять настройкой скриптов указанным выше способом.

9.1.3 Активация лицензии

После приобретения продукта вам необходимо активировать лицензию для используемых компонентов. Предусмотрено несколько способов подключения лицензионного ключа.

Активация при помощи строки кода

Для активации при помощи строки, достаточно скопировать зашифрованный текст лицензии из личного кабинета на сайте, и зарегистрировать его при помощи статической функции `setPrimaryKey()`, находящейся в классе `StiLicense`:

index.php

```
<?php
   use Stimulsoft\StiLicense;

   StiLicense::setPrimaryKey('Your activation code...');
?>
```

© 2003-2026 Stimulsoft

Полный код примера доступен на GitHub.

Активация при помощи файла

Для активации при помощи лицензионного файла, достаточно загрузить файл `license.key` из личного кабинета на сайте, и скопировать его в папку PHP проекта, и зарегистрировать его при помощи статической функции `setPrimaryFile()`, находящейся в классе `StiLicense`:

index.php

```
<?php
   use Stimulsoft\StiLicense;

   StiLicense::setPrimaryFile('license.key');
?>
```

Полный код примера доступен на GitHub.

Активация только генератора отчетов

В некоторых случаях может возникнуть необходимость активации генератора отчетов отдельно от остальных компонентов. В этом случае, лицензионный ключ можно зарегистрировать при помощи метода `setKey()` либо `setFile()` свойства `license` у объекта отчета:

index.php

```
<?php
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->license->setKey('Your activation code...');
   $report->license->setFile('license.key');
?>
```

Полный код примера доступен на GitHub.

© 2003-2026 Stimulsoft

Защита от кражи лицензионного ключа

В случае активации лицензии при помощи строки, можно добавлять ее код по определенному условию. Например, требуется добавлять лицензионный ключ только для зарегистрированных пользователей:

index.php

```
<?php
   use Stimulsoft\StiLicense;

   if (!empty($sessionID))
      StiLicense::setPrimaryFile('Your activation code...');
?>
```

Также не будет лишним изменить местоположение и название файла лицензионного ключа, например:

license.php

```
<?php
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->license->setFile('private/a15fc0ef64e6.key');
?>
```

Активация лицензии в одном файле

Если в приложении компоненты используются в нескольких отдельных файлах, например генератор отчетов, вьювер и дизайнер, удобнее применить лицензию только в одном файле вместо активации каждого компонента по отдельности. Для этого необходимо создать отдельный файл, в котором будет применена лицензия, а во всех остальных файлах с компонентами просто подключить подготовленный файл с лицензией при помощи стандартного выражения `require_once`.

© 2003-2026 Stimulsoft

license.php

```
<?php
   use Stimulsoft\StiLicense;

   StiLicense::setPrimaryKey('Your activation code...');
?>
```

license.php

```
<?php
   require_once 'license.php';

   use Stimulsoft\Report\StiReport;

   $report = new StiReport();

   ...
?>
```

9.1.4 Загрузка и сохранение отчета

Информация

Файлы формата Stimulsoft MRT и MDC - это описание отчетов с разметкой XML или JSON. Вы можете использовать MRT и MDC файлы, созданные в других продуктах Stimulsoft.

Загрузка отчета

Отчет может храниться в виде шаблона отчета (MRT файл), либо в виде построенного отчета (MDC документ), предназначенного для последующего просмотра либо экспортирования. Для загрузки отчета при помощи PHP кода можно использовать один из перечисленных ниже методов у объекта `StiReport`. Каждый из методов принимает на вход имя файла отчета либо сам отчет в виде строки:

Наименование	Описание
`loadFile($filePath, $load = false)`	Загружает шаблон отчета из MRT файла на стороне клиента, путь к

© 2003-2026 Stimulsoft

	которому указан в аргументах функции. Если параметр `$load` установлен в значение `true`, файл отчета будет загружен на стороне сервера и передан на сторону клиента в виде упакованной `Base64` строки.
`load($data, $fileName = 'Report')`	Загружает шаблон отчета из XML или JSON строки и передает его на сторону клиента в виде упакованной `Base64` строки. Параметр `$fileName` устанавливает имя файла, которое будет использоваться для последующего сохранения и экспорта отчета.
`loadPacked($data, $fileName = 'Report')`	Загружает и передает на сторону клиента шаблон отчета в виде упакованной `Base64` строки, заданной в параметре `$data`. Параметр `$fileName` устанавливает имя файла, которое будет использоваться для последующего сохранения и экспорта отчета.
`loadDocumentFile($filePath, $load = false)`	Загружает построенный отчет из MDC файла на стороне клиента, путь к которому указан в параметрах функции. Если параметр `$load` установлен в значение `true`, файл документа будет загружен на стороне сервера и передан на сторону клиента в виде упакованной `Base64` строки.
`loadDocument($data, $fileName = 'Report')`	Загружает построенный отчет из XML или JSON строки и передает его на сторону клиента в виде упакованной `Base64` строки.

© 2003-2026 Stimulsoft

	Параметр `$fileName` устанавливает имя файла, которое будет использоваться для последующего сохранения и экспорта отчета.
`loadPackedDocument($data, $fileName = 'Report')`	Загружает и передает на сторону клиента построенный отчет в виде упакованной `Base64` строки, заданной в параметре `$data`. Параметр `$fileName` устанавливает имя файла, которое будет использоваться для последующего сохранения и экспорта отчета.

Пример загрузки отчета из файла на стороне сервера из приватной директории и передача его на сторону клиента в виде упакованной строки для последующего построения:

index.php

```
<?php
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->loadFile('reports/SimpleList.mrt', true);
   $report->render();
?>
```

Сохранение отчета

В режиме построения отчета на стороне JavaScript клиента, генератор отчетов PHP на стороне сервера не имеет доступа к объекту отчета. В этом случае, для сохранения шаблона отчета или документа необходимо воспользоваться событиями и JavaScript функциями. Более подробно про это указано в разделе События генератора отчетов.

Пример сохранения построенного отчета в виде строки для последующего использования:

© 2003-2026 Stimulsoft

index.php

```
<?php
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->onAfterRender = 'afterRender';
   $report->loadFile('reports/SimpleList.mrt', true);
   $report->render();
?>

...

<script>
   function afterRender(args) {
      let reportJson = args.report.saveDocumentToJsonString();
      ...
   }
</script>
```

Полный код примера доступен на GitHub.

В режиме построения отчета на стороне сервера, для сохранения отчета предназначен один из перечисленных ниже методов:

Наименование	Описание
`saveDocument($filePath = null)`	Сохраняет построенный отчет в виде MDC файла, путь к которому указан в аргументах функции. Если параметр `$filePath` не указан, то вместо сохранения файла метод вернет отчет в виде JSON строки.
`savePackedDocument($filePath = null)`	Сохраняет построенный отчет в виде упакованного MDZ файла, путь к которому указан в аргументах функции. Если параметр `$filePath` не указан, то вместо сохранения файла метод вернет отчет в виде упакованной `Base64` строки.

© 2003-2026 Stimulsoft

Пример сохранения построенного отчета в виде файла на стороне сервера:

index.php

```
<?php
   use Stimulsoft\Report\Enums\StiEngineType;
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->engine = StiEngineType::ServerNodeJS;
   $report->loadFile('reports/SimpleList.mrt', true);
   $report->render();
   $report->saveDocument('reports/SimpleList.mdc');
?>
```

Полный код примера доступен на GitHub.

Информация

Генератор отчетов Stimulsoft Reports.PHP и аналитические панели Stimulsoft Dashboards.PHP основаны на платформе JavaScript, и поддерживают сохранение файлов MRT и MDC файлов только в JSON формат. Файлы в формате XML поддерживаются только в режиме загрузки, и при сохранении будут автоматически преобразованы в JSON формат.

Так как для аналитических панелей всегда нужны данные, они не могут быть сохранены в виде MDC документа. Stimulsoft Dashboards.PHP поддерживает сохранение панелей только в виде шаблона с использованием JavaScript события и функций.

При сохранении документа из меню вьювера, файл также сохраняется в JSON формате, и имеет расширение MDC для стандартного документа, MDZ для упакованного документа, MDX для зашифрованного документа.

9.1.5 Построение отчета

Для построения загруженного отчета необходимо вызвать функцию `render()` у объекта отчета. Например, требуется построить отчет перед его экспортированием:

© 2003-2026 Stimulsoft

index.php

```
<?php
   use Stimulsoft\Report\StiReport;
   use Stimulsoft\Export\Enums\StiExportFormat;

   $report = new StiReport();
   $report->loadFile('reports/SimpleList.mrt');
   $report->render();
   $report->exportDocument(StiExportFormat::Pdf);
   $report->printHtml();
?>
```

Полный код примера доступен на GitHub.

Для выполнения каких-либо действий с отчетом перед его построением при помощи JavaScript кода, достаточно определить название JavaScript функции для события `onBeforeRender`, в аргументах функции будут переданы тип действия и сам отчет. Пример регистрации JSON данных перед построением отчета:

index.php

```
<?php
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->onBeforeRender = 'beforeRender';
   $report->loadFile('reports/SimpleList.mrt');
   $report->render();
?>

...

<script>
   function beforeRender(args) {
      let dataSet = new Stimulsoft.System.Data.DataSet("SimpleDataSet");
      dataSet.readJsonFile("Demo.json");

      let report = args.report;
      report.regData(dataSet.dataSetName, "", dataSet);
   }
</script>
```

© 2003-2026 Stimulsoft

Полный код примера доступен на GitHub.

Для выполнения каких-либо действий после построения отчета при помощи JavaScript кода, достаточно определить название JavaScript функции для события `onAfterRender`, в аргументах функции будут переданы тип действия и сам отчет. Пример вывода сообщения после построения отчета:

index.php

```
<?php
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->onAfterRender = 'afterRender';
   $report->loadFile('reports/SimpleList.mrt');
   $report->render();
?>

...

<script>
   function afterRender(args) {
      alert("The report rendering is completed.");
   }
</script>
```

Полный код примера доступен на GitHub.

9.1.6 Построение отчета на стороне сервера

Для построения отчета на стороне сервера используется универсальная платформа Node.js, при помощи которой выполняется необходимый блок JavaScript кода и возвращается подготовленный результат.

Развертывание платформы Node.js

Перед использованием генератора отчета на стороне сервера, необходимо установить саму платформу Node.js и настроить ее работу. Это можно сделать отдельно, используя инструкции по установке Node.js для конкретной операционной системы с официального сайта платформы, а можно сделать автоматически при помощи специальной функции развертывания.

Если платформа уже установлена, достаточно указать путь к директории исполняемых файлов платформы. При необходимости, можно указать путь к рабочей директории, в которой будут установлены необходимые Node.js

© 2003-2026 Stimulsoft

пакеты в поддиректории node_modules:

index.php

```
<?php
   use Stimulsoft\StiNodeJs;

   $nodejs = new StiNodeJs();

   //$nodejs->binDirectory = "C:\\Program Files\\nodejs";
   //$nodejs->binDirectory = "/usr/bin/nodejs";

   //$nodejs->workingDirectory = "";
?>
```

Полный код примера доступен на GitHub.

Если платформа и пакеты не установлены, можно воспользоваться специальными методами для их установки. Эти методы необходимо вызвать только один раз: первый метод для установки самой платформы Node.js, второй метод для установки либо обновления до текущей версии всех необходимых пакетов:

index.php

```
<?php
   use Stimulsoft\StiNodeJs;

   $nodejs = new StiNodeJs();

   $result = $nodejs->installNodeJS();
   if ($result)
      $result = $nodejs->updatePackages();

   $message = $result ? 'The installation was successful.' : $nodejs-
   >error;
?>
```

Полный код примера доступен на GitHub.

Платформа и пакеты будут установлены в указанную рабочую директорию.

© 2003-2026 Stimulsoft

Построение отчета

Для переключения генератора отчетов в режим работы на стороне сервера, необходимо установить свойство отчета engine в значение `StiEngineType::ServerNodeJS`, весь остальной набор событий и методов для работы с отчетом точно такой же, как и при построении отчета на стороне клиента.

После вызова метода `render()` у объекта отчета, построенный отчет можно сохранить при помощи методов `saveDocument()` либо `savePackedDocument()`, подробнее описано в разделе Загрузка и сохранение отчета. Так же, после построения отчета, его можно экспортировать в один из многочисленных форматов при помощи метода `exportDocument()`, подробнее описано в разделе Экспорт отчета из кода.

Пример загрузки и построения шаблона отчета, и последующее сохранение построенного отчета в файл на стороне сервера:

index.php

```
<?php
   use Stimulsoft\Report\Enums\StiEngineType;
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->engine = StiEngineType::ServerNodeJS;
   $report->loadFile('reports/SimpleList.mrt', true);
   $report->render();
   $report->saveDocument('reports/SimpleList.mdc');
?>
```

Полный код примера доступен на GitHub.

Любые ошибки, возникшие при работе с Node.js пакетами, а так же при построении и экспортировании отчета на стороне сервера, можно прочитать в свойствах `report->nodejs->error` и `report->nodejs->errorStack`, эти свойства будут содержать последнюю по очереди ошибку:

© 2003-2026 Stimulsoft

index.php

```
<?php
   use Stimulsoft\Report\Enums\StiEngineType;
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->engine = StiEngineType::ServerNodeJS;
   $report->loadFile('reports/SimpleList.mrt', true);
   $report->render();

   if (!$report) {
      // The main text of the error as a string.
      $error = $report->nodejs->error;

      // The full error text as an array of strings.
      $errorStack = $report->nodejs->errorStack;
   }
?>
```

Полный код примера доступен на GitHub.

9.1.7 Обработчик событий PHP

Генератор отчетов, а также вьювер и дизайнер отчетов, умеют вызывать события на стороне клиента, передавать их на сторону PHP сервера для последующей обработки, и принимать подготовленный ответ. Все действия реализованы в обработчике событий, нет необходимости в использовании каких-либо дополнительных функций для связи клиента с сервером и передачи данных. Для работы с выбранным событием, достаточно добавить имя функции к обработчику, и указанная функция будет автоматически вызвана при возникновении выбранного события. Предусмотрен вызов событий как на стороне JavaScript клиента, так и на стороне PHP сервера. При необходимости, можно добавлять несколько функций любого типа к одному и тому же событию.

Вызов JavaScript события на стороне клиента

Для вызова JavaScript события, необходимо добавить к обработчику имя функции в виде строки. В аргументах события будут переданы все необходимые данные. Список доступных свойств, передаваемых в аргументах для всех событий, можно посмотреть в разделе События генератора отчетов. Пример вывода сообщения с количеством страниц полученного документа после построения отчета:

© 2003-2026 Stimulsoft

index.php

```
<?php
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->onAfterRender = 'afterRender';
   $report->loadFile('reports/SimpleList.mrt');
   $report->render();
?>

...

<script>
   function afterRender(args) {
      let pagesCount = args.report.renderedPages.count;
      alert("The report is rendered, pages: " + pagesCount);
   }
</script>
```

Полный код примера доступен на GitHub.

В данном примере, из аргументов события можно получить JavaScript объект отчета, и прочитать количество построенных страниц документа.

Информация

Более подробно про доступные функции и параметры JavaScript генератора отчетов можно прочитать в документации Stimulsoft Reports.JS и Stimulsoft Dashboards.JS продуктов.

Вызов JavaScript события на стороне Node.js сервера

При использовании Node.js платформы для работы с отчетами нет возможности вызвать JavaScript функцию по имени, т.к. HTML шаблон не используется. В таком случае, для вызова JavaScript события необходимо добавить к обработчику саму функцию в виде строки или строк кода. Аргументы события будут находиться в предварительно заданной переменной args, которую можно использовать в коде события. Пример очистки словаря данных в шаблоне перед построением отчета:

© 2003-2026 Stimulsoft

index.php

```
<?php
   use Stimulsoft\Report\Enums\StiEngineType;
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->engine = StiEngineType::ServerNodeJS;
   $report->onBeforeRender = 'args.report.dictionary.clear();';
   $report->loadFile('reports/SimpleList.mrt', true);
   $report->render();
?>
```

Информация

Точно такой же способ для JavaScript событий можно использовать при отображении вьювера либо дизайнера без HTML шаблона, когда нет возможности заранее определить необходимую JavaScript функцию.

Вызов PHP события на стороне сервера

Для вызова PHP события, необходимо добавить к обработчику имя функции в виде переменной либо саму функцию. В аргументах события будут переданы все необходимые данные. Список доступных свойств, передаваемых в аргументах события, можно посмотреть в разделе События генератора отчетов. Пример корректировки пароля в строке подключения перед запросом данных:

index.php

```
<?php
   use Stimulsoft\Events\StiDataEventArgs;
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->onBeginProcessData = function (StiDataEventArgs $args) {
      args->connectionString = str_replace('Pwd=;', 'Pwd=******;', args-
      >connectionString);
   };

   $report->loadFile('reports/SimpleList.mrt', true);
   $report->render();
?>
```

© 2003-2026 Stimulsoft

Полный код примера доступен на GitHub.

В данном примере, из аргументов события можно получить и изменить все параметры подключения к базе данных.

В событии на стороне PHP сервера предусмотрена возможность вернуть текстовое сообщение об успешном выполнении либо возникшей ошибке обработки события, это сообщение отобразится во вьювере или дизайнере после завершения события. Для отображения окна с ошибкой необходимо вернуть результат функции `StiResult::getError('Error message')`. Для отображения окна с информационным сообщением достаточно вернуть результат функции `StiResult::getSuccess('Info message')` либо просто строку `'Info message'`.

index.php

```
<?php
   $report->onBeginProcessData = function (StiDataEventArgs $args) {
...
      return StiResult::getError('Error message');
      // return StiResult::getSuccess('Info message');
      // return 'Info message';
   };
?>
```

Информация

При возникновении ошибки в самом обработчике событий, например ошибка при подключении к базе данных, ошибка обработки файла и т.п. - будет выведено внутреннее сообщение в любом случае, даже если определено сообщение в самом событии компонента.

© 2003-2026 Stimulsoft

Информация

Диалоговое окно с сообщением будет показано только при использовании компонентов `StiViewer` либо `StiDesigner`. Сам генератор отчетов не содержит визуальных форм, поэтому сообщение обработки события будет выведено в консоли браузера.

Вызов нескольких одинаковый событий

Предусмотрена возможность добавления к обработчику событий неограниченного количества функций, они все будут сгруппированы по типу события, и вызваны последовательно в порядке добавления. Для этого вместо присваивания имени функции необходимо использовать специальный метод события `append()`, в качестве параметра передать имя функции либо саму функцию.

Пример изменения SQL запроса на стороне JavaScript клиента, и строки подключения на стороне PHP сервера:

index.php

```
<?php
   use Stimulsoft\Events\StiDataEventArgs;
   use Stimulsoft\Report\StiReport;

   function beginProcessData(StiDataEventArgs $args) {
      $args->connectionString = str_replace('Pwd=;', 'Pwd=******;', $args-
      >connectionString);
   };

   $report = new StiReport();
   $report->onBeginProcessData->append('beginProcessData');
   $report->onBeginProcessData->append(beginProcessData);
   $report->loadFile('reports/SimpleList.mrt');
   $report->render();
?>

...

<script>
   function beginProcessData(args) {
      args.queryString = args.queryString.replace("TableName", "Products");
   }
</script>
```

© 2003-2026 Stimulsoft

Информация

Некоторые события могут быть вызваны только на стороне JavaScript клиента и не имеют возможности вызова события на стороне PHP сервера, либо наоборот, могут быть вызваны только на стороне PHP сервера. При добавлении функции неподдерживаемого типа к такому событию не будет ошибки, просто добавленная функция не будет вызвана. Все поддерживаемые варианты указаны в разделе События генератора отчетов.

Шифрование данных, передаваемых на сторону PHP сервера

Для исключения кражи передаваемых данных злоумышленниками, мы рекомендуем использовать HTTPS протокол, в большинстве случаев этого достаточно. В дополнении к этому, по умолчанию все передаваемые данные проходят через специальный алгоритм кодирования, и передаются на сервер в зашифрованном виде. Это позволяет скрыть конфиденциальные данные, например логин и пароль в строке подключения, от любопытных пользователей, работающих с вашим приложением.

Однако, если этого не требуется, либо если нужно отобразить исходные данные запроса для отладки приложения, предусмотрена возможность отключения шифрования. Для этого достаточно установить свойство `$encryptData` в значение `false` у обработчика событий, после этого все данные будут передаваться в JSON формате.

index.php

```
<?php
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->handler->encryptData = false;
   $report->process();
?>
```

Передача значений GET параметров в PHP обработчик событий

© 2003-2026 Stimulsoft

Предусмотрена возможность автоматической передачи всех значений параметров GET запроса в обработчик событий, во всех событиях которого можно будет получить их значения. Для включения этой возможности, достаточно установить свойство `passQueryParameters` в значение `true`, после этого все параметры GET запроса будут переданы при каждом запросе к обработчику событий.

index.php

```
<?php
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->handler->passQueryParameters = true;
   $report->process();
?>
```

9.1.8 Работа с данными

Как правило, параметры подключения к источникам данных хранятся в самом шаблоне отчета – это могут быть как файловые источники данных, так и SQL источники данных. При необходимости подключения данных из кода, предусмотрен специальный метод `regData()` у объекта отчета. В качестве параметров метода необходимо указать имя источника данных в отчете, и сами данные. Данные могут быть в формате JSON либо XML в виде строки, а также в виде PHP объекта либо массива.

Пример подключения данных в виде объекта:

index.php

```
<?php
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();

   $data = new stdClass();
   $data->Customers = [
       "CompanyName" => "Centro comercial Moctezuma",
       "Address" => "Sierras de Granada 9993",
       "Phone" => 12355500,
       "ContactTitle" => "Marketing Manager"];

   $report->regData("Demo", $data);
   $report->render();
```

© 2003-2026 Stimulsoft

```
    $report->process();
?>
```

В некоторых случаях может понадобиться синхронизировать структуру словаря данных отчета с передаваемыми данными. Для этого достаточно указать значение `true` в качестве третьего параметра метода `regData()`, в этом случае перед построением отчета будет выполнена синхронизация словаря данных.

Информация

При использовании компонента Дизайнер, можно создать новый объект отчета и передать ему данные со включенной синхронизацией словаря отчета. В этом случае, будет открыт дизайнер с новым отчетом и уже подготовленными для работы данными.

Очистка параметров подключения и словаря данных

Если в загруженном отчете уже настроены параметры подключения к данным, то они имеют приоритет. При необходимости принудительного использования своего набора данных, предусмотрена возможность очистить параметры подключения, для этого достаточно вызвать специальный метод `clearData()` у объекта отчета. По умолчанию, будет выполнена очистка только самих параметров подключения. Если есть необходимость полностью очистить структуру словаря данных (удалить все существующие источники данных), необходимо указать значение `true` в качестве параметра метода.

Пример подключения данных с предварительной очисткой существующего словаря:

index.php

```
<?php
    use Stimulsoft\Report\StiReport;

    $report = new StiReport();

    $report->clearData(true);
```

© 2003-2026 Stimulsoft

```
    $data = '{"Customers": [{"CompanyName": "Centro comercial Moctezuma",
    "Address": "Sierras de Granada 9993", "Phone": 12355500,
    "ContactTitle": "Marketing Manager"}]}';

    $report->regData("Demo", $data, true);
    $report->render();

    $report->process();
?>
```

Событие подготовки к построению отчета

Для подключения данных перед построением отчета, а также для изменения самого отчета, можно воспользоваться специальным событием `onBeforeRender` объекта отчета. Данное событие будет вызвано перед построением отчета после его загрузки. Для подключения данных достаточно вызвать метод `regReportData()` у аргументов события. Параметры метода точно такие же, как и у метода `regData()` объекта отчета.

Пример подключения данных в событии отчета:

index.php

```
<?php
    use Stimulsoft\Report\StiReport;
    use Stimulsoft\Events\StiReportEventArgs;

    $report = new StiReport();

    $report->onBeforeRender = function (StiReportEventArgs $args) {
        $data = '{"Customers": [{"CompanyName": "Centro comercial
    Moctezuma", "Address": "Sierras de Granada 9993", "Phone": 12355500,
    "ContactTitle": "Marketing Manager"}]}';
        $args->regReportData("Demo", $data);
    };

    $report->render();
    $report->process();
?>
```

Информация

Если в отчете были заданы данные при помощи метода regData() у объекта

© 2003-2026 Stimulsoft

отчета – они будут переопределены данными, указанными в событии.

Подключение данных на стороне клиента

При необходимости, можно подключить данные при помощи JavaScript функций генератора отчетов. Для этого можно воспользоваться событием `onBeforeRender` объекта отчета. Данные могут быть загружены напрямую в специальный объект `DataSet`, который используется для их хранения. Он содержит функции для загрузки данных из файлов форматов XML/XSD и JSON. После загрузки файлов, необходимо вызвать функцию `regData()` у объекта отчета для подключения данных, в аргументах функции указывается подготовленный объект `DataSet`.

Пример загрузки данных из файла XML с использованием схемы XSD:

index.php

```
<?php
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->onBeforeRender = 'beforeRender';
   $report->render();
?>

function onBeforeRender(args) {
   let dataSet = new Stimulsoft.System.Data.DataSet("SimpleDataSet");
   dataSet.readXmlSchemaFile("Demo.xsd");
   dataSet.readXmlFile("Demo.xml");

   let report = args.report;
   report.regData(dataSet.dataSetName, "", dataSet);
   report.dictionary.synchronize();
}
```

Информация

Загрузка схемы данных не является обязательной. Если вы хотите использовать схему данных, вам следует добавить ее перед загрузкой данных XML.

© 2003-2026 Stimulsoft

Пример загрузки данных из файла JSON:

index.php

```
<?php
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->onBeforeRender = 'beforeRender';
   $report->render();
?>

function onBeforeRender(args) {
   let dataSet = new Stimulsoft.System.Data.DataSet("SimpleDataSet");
   dataSet.readJsonFile("Demo.json");

   let report = args.report;
   report.regData(dataSet.dataSetName, "", dataSet);
   report.dictionary.synchronize();
}
```

Полный код примера доступен на GitHub.

Кроме указанных функций `readXmlFile()` и `readJsonFile()`, предусмотрены функции `readXml()` и `readJson()`, принимающие на вход данные в виде строки либо объекта.

Информация

Функция `report.dictionary.synchronize()` необходима для синхронизации подключаемых данных со словарем данных шаблона отчета. То есть, при вызове этой функции, словарь отчета будет создан на основе структуры данных, загруженных в `DataSet`. Функция синхронизации не требуется, если словарь создан заранее, и его структура совпадает с подключаемыми данными.

9.1.9 Подключение файлов данных

Как правило, параметры подключения к источникам данных хранятся в самом шаблоне отчета. Для работы с файловыми источниками данных, такими как

© 2003-2026 Stimulsoft

XML, JSON, CSV не требуется никаких дополнительных действий, все алгоритмы находятся в скрипте генератора отчетов.

Событие загрузки данных

Для просмотра и изменения параметров подключения файловых данных перед их загрузкой, необходимо определить событие `onBeginProcessData` у компонента. В аргументах события будет передана информация о подключении к файловому источнику данных - имя и тип подключения в шаблоне отчета, а также путь к файлу данных. Подробное описание доступных значений свойств, передаваемых в аргументах события, находится в разделе События генератора отчетов.

Допускается изменить путь к файлу данных. В этом случае после завершения события, генератор отчетов будет запрашивать файл по новому пути, заданному в аргументах. Например, требуется для указанного соединения изменить путь к JSON файлу данных:

index.php

```
<?php
   use Stimulsoft\Report\StiReport;
   use Stimulsoft\Events\StiDataEventArgs;

   $report = new StiReport();
   $report->onBeginProcessData = function (StiDataEventArgs $args) {
      if ($args->connection == "MyJsonConnection")
         $args->pathData = "Data/Demo.json";
   };

   $report->render();
   $report->process();
?>
```

При необходимости, те же самые действия можно выполнить в событии на стороне JavaScript клиента:

index.php

```
<?php
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
```

© 2003-2026 Stimulsoft

```
   $report->onBeginProcessData = 'beginProcessData';
   $report->render();
?>

function onBeginProcessData(args) {
   if (args.connection == "MyJsonConnection")
      args.pathData = "Data/Demo.json";
}
```

Информация

Для XML источника данных событие `onBeginProcessData` будет вызвано дважды: первый раз для чтения XSD схемы, второй раз для чтения самого XML файла данных.

Событие обработки данных

Для просмотра либо корректировки загруженных данных перед их подключением и построением отчета, необходимо определить событие `onEndProcessData` у компонента. В аргументах события будет передана информация о подключении к файловому источнику данных - имя и тип подключения, сохраненные в шаблоне отчета, а также результат загрузки данных, содержащий загруженные данные в виде строки. Подробное описание доступных значений свойств, передаваемых в аргументах события, доступно в разделе События генератора отчетов.

index.php

```
<?php
   use Stimulsoft\Report\StiReport;
   use Stimulsoft\Events\StiDataEventArgs;

   $report = new StiReport();
   $report->onEndProcessData = function (StiDataEventArgs $args) {
      $data = $args->result->data;
   };

   $report->render();
   $report->process();
?>
```

© 2003-2026 Stimulsoft

При необходимости, просмотр либо корректировку загруженных данных можно выполнить в событии на стороне JavaScript клиента. В аргументах события будет передана информация о подключении к файловому источнику данных - имя и тип подключения, сохраненные в шаблоне отчета, а также подготовленный объект `DataSet`, содержащий таблицы и строки данных, полученных из файлового источника. Подробное описание доступных значений свойств, передаваемых в аргументах события, доступно в разделе [События генератора отчетов].

index.php

```
<?php
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->onEndProcessData = 'onEndProcessData';
   $report->renderHtml();
?>

function onEndProcessData(args) {
   let dataSet = args.dataSet;
}
```

Отключение обработки файловых адаптеров данных на стороне сервера

По умолчанию, загрузка файловых данных осуществляется на стороне сервера. Это необходимо, если требуется контролировать параметры подключения и сами данные в событиях на стороне сервера. Если в этом нет необходимости, обработку файловых данных на стороне сервера можно отключить. В этом случае загрузка данных будет осуществляться при помощи JavaScript, это немного упростит цепочку событий и ускорит загрузку данных.

Для отключения файловых адаптеров данных на стороне сервера, достаточно установить свойство `$allowFileDataAdapters` у обработчика событий в значение `false`, например:

index.php

```
<?php
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
```

© 2003-2026 Stimulsoft

```
    $report->handler->allowFileDataAdapters = false;
?>
```

Использование переменных в файловых данных

Предусмотрена возможность использовать переменные в виде выражений (а также использовать выражения) при указании пути к файловому источнику данных. Переменная или выражение задаются в фигурных скобках. Допустимо использовать несколько выражений в любом месте пути к файлу данных, например:

File Data Source

```
https://localhost/data/{VariableJsonFileName}.json
https://localhost/data/json.php?id={VariableId}
https://localhost/{VariableCategory}/{VariableId}
```

Таким образом, один источник данных может быть преобразован к REST синтаксису, это избавит от необходимости создавать несколько схожих источников данных для получения однотипных данных. А в совокупности с серверной PHP логикой и событиями генератора отчетов, сделать источник данных еще более гибким.

Использование OData

Также можно использовать данные для создания отчетов, полученные из хранилищ OData. В этом случае необходимо выполнить авторизацию, используя имя пользователя, пароль пользователя или токен. Параметры авторизации указываются в строке подключения к хранилищу OData с помощью символа ";" разделитель.

© 2003-2026 Stimulsoft

index.php

```
// Authorization using a user account
var oDataDatabase = new
Stimulsoft.Report.Dictionary.StiODataDatabase("OData", "OData", "https://
services.odata.org/V4/Northwind/
Northwind.svc;AddressBearer=adress;UserName=UserName;Password=Password;Cli
ent_Id=Your Client ID", false, null);

// Authorization using a user token
var oDataDatabase = new
Stimulsoft.Report.Dictionary.StiODataDatabase("OData", "OData", "https://
services.odata.org/V4/Northwind/Northwind.svc;Token=Enter your token",
false, null);

report.dictionary.databases.add(oDataDatabase);
report.dictionary.synchronize();

// Query with data filter
var productsDataSource =
report.dictionary.dataSources.getByName("Products");
if (productsDataSource != null) productsDataSource.sqlCommand =
"Products?$filter=ProductID eq 2";
```

9.1.10 Подключение SQL адаптеров данных

Для построения отчетов, генератор отчетов позволяет использовать данные из различных SQL источников. Поскольку чистый JavaScript не имеет встроенных методов для работы с удаленными базами данных, эта функциональность реализуется с помощью серверного PHP кода. Для работы с SQL источниками данных не требуется никаких дополнительных действий, все адаптеры данных уже подключены и настроены для работы.

Событие создания подключения

При необходимости контролировать все возможные параметры подключения к базе данных, предусмотрено событие `onDatabaseConnect`. В аргументах события будут переданы все необходимые параметры подключения к SQL источнику данных, а также тип и название драйвера используемой базы данных. Все параметры подключения к источнику данных могут быть изменены. Дополнительно, предусмотрена возможность передать в аргументах события уже созданное подключение к базе данных. Подробное описание доступных значений свойств, передаваемых в аргументах события, находится в разделе События генератора отчетов.

Пример создания подключения к MySQL базе данных с указанием приватного

© 2003-2026 Stimulsoft

SSL ключа:

index.php

```
<?php
   use Stimulsoft\Events\StiConnectionEventArgs;
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->onDatabaseConnect = function (StiConnectionEventArgs $args)
   {
      $args->link = mysqli_init();
      mysqli_ssl_set($args->link, null, null, "./private/cert.pem", null,
      null);
      $args->link = mysqli_real_connect(
         $args->link, $args->info->host, $args->info->userId, $args->info-
         >password,
         $args->info->database, $args->info->port, NULL, MYSQLI_CLIENT_SSL);
   };

   $report->render();
?>
```

Событие загрузки данных

При необходимости обработки параметров, используемых для подключения к базе данных, предусмотрено событие `onBeginProcessData`. В аргументах события будут переданы все необходимые параметры подключения к SQL источнику данных, а также параметры SQL запроса. Все параметры подключения к источнику данных могут быть изменены как на стороне JavaScript клиента, так и на стороне PHP сервера. Подробное описание доступных значений свойств, передаваемых в аргументах события, находится в разделе События генератора отчетов.

Пример измерения SQL запроса на стороне JavaScript клиента, и строки подключения к базе данных на стороне PHP сервера:

© 2003-2026 Stimulsoft

index.php

```
<?php
   use Stimulsoft\Events\StiDataEventArgs;
   use Stimulsoft\Report\StiReport;

   function beginProcessData(StiDataEventArgs $args) {
      if ($args->connection == 'MyConnectionName')
         $args->connectionString =
         'Server=localhost;Database=test;uid=root;password=******;';
   };

   $report = new StiReport();
   $report->onBeginProcessData->append(beginProcessData);
   $report->onBeginProcessData->append('beginProcessData');
   $report->render();
?>

...

<script>
   function beginProcessData(args) {
      if (args.dataSource == "MyDataSource")
         args.queryString = "SELECT * FROM ProductsTable";
   }
</script>
```

Полный код примера доступен на GitHub.

Таким образом, в событии `onBeginProcessData` можно узнать тип базы данных, имя подключения и имя источника данных, получить и при необходимости скорректировать строку подключения и SQL запрос. При изменении значений любого свойства в аргументах события на стороне PHP сервера, измененные значения не будут переданы на сторону клиента, поэтому в качестве значений можно использовать конфиденциальные данные, такие как логин и пароль в строке подключения, имена и префиксы таблиц и т.п.

Событие обработки данных

Для просмотра либо корректировки загруженных данных перед их регистрацией и построением отчета, предусмотрено событие `onEndProcessData`. В аргументах события будут переданы все параметры подключения к SQL источнику данных, а также результат выполнения запроса, содержащий названия колонок, типы колонок и строки данных, полученные из SQL источника. Подробное описание доступных

© 2003-2026 Stimulsoft

значений свойств, передаваемых в аргументах события, находится в разделе События генератора отчетов.

Пример результата выполненного SQL запроса, где результат уже содержит подготовленные для отчета данные:

index.php

```
<?php
   use Stimulsoft\Events\StiDataEventArgs;
   use Stimulsoft\Report\StiReport;

   function endProcessData(StiDataEventArgs $args) {
      $args->result->columns = ['id', 'username', 'phone'];
      $args->result->types = ['int', 'string', 'string'];
      $args->result->rows = [
         [1, 'Mario Pontes', '555-6874'],
         [2, 'Helen Bennett', '555-2376']
      ];
   };

   $report = new StiReport();
   $report->onEndProcessData->append(endProcessData);
   $report->onEndProcessData->append('endProcessData');
   $report->render();
?>

...

<script>
   function endProcessData(args) {
      args.result.columns = ["id", "username", "phone"];
      args.result.types = ["int", "string", "string"];
      args.result.rows = [
         [1, "Mario Pontes", "555-6874"],
         [2, "Helen Bennett", "555-2376"]
      ];
   }
</script>
```

Доступные свойства объекта результата выполнения SQL запроса указаны в таблице:

Наименование	Описание
count	Общее количество колонок таблицы SQL источника данных.

© 2003-2026 Stimulsoft

columns	Имена колонок таблицы SQL источника данных.
types	Типы колонок таблицы SQL источника данных, приведенные к известным типам для генератора отчетов.
rows	Строки данных SQL источника данных, представляет собой массив из массивов всех строк таблицы.

Все данные результата выполнения SQL запроса могут быть изменены как на стороне JavaScript клиента, так и на стороне PHP сервера. Количество колонок и типы данных должны совпадать для исключения неверной интерпретации данных генератором отчета.

Использование параметров в SQL запросе

При необходимости, можно использовать параметры в SQL запросе. Для этого, в источнике данных необходимо добавить параметры в специальную коллекцию, для каждого параметра установить необходимый тип и значение по умолчанию. После этого, параметры можно использовать в SQL запросе следующим образом:

SQL Data Source

```
SELECT * FROM @Parameter1 WHERE UserID = @Parameter2
```

Все значения параметров хранятся в самом источнике данных в виде коллекции. Коллекция представляет собой массив из объектов, содержащих имя параметра, его тип и значение. Пример структуры массива параметров на стороне JavaScript клиента:

© 2003-2026 Stimulsoft

index.php

```
args.parameters = [
   {
      name: "ParameterString",
      type: 752,
      typeName: "Text",
      value: "Text value"
   },
   {
      name: "ParameterInt",
      type: 3,
      typeName: "Int32",
      value: 20
   }
];
```

Для доступа к параметрам запроса можно воспользоваться событием `onBeginProcessData`, коллекция параметров будет передана в аргументах события. Допускается изменение значений любых параметров, находящихся в передаваемой коллекции. Пример изменения значения одного и того же параметра на стороне JavaScript клиента и на стороне PHP сервера:

index.php

```
<?php
   use Stimulsoft\Events\StiDataEventArgs;
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->onBeginProcessData->append('
      args.parameters["Parameter1"] = "TableName";
   ');

   $report->onBeginProcessData->append(function (StiDataEventArgs $args) {
      $args->parameters['Parameter1']->value = 'TableName';
   });

   $report->render();
?>
```

При изменении значений параметров запроса, тип нового значения должен соответствовать типу изменяемого параметра. В противном случае, выполнение SQL запроса может вернуть некорректные данные либо вызвать внутреннюю

© 2003-2026 Stimulsoft

ошибку выполнения.

Информация

При изменении значений параметров на стороне PHP сервера, измененные значения не будут переданы на сторону клиента, поэтому в качестве значений можно использовать конфиденциальные данные.

Если в отчете используется несколько источников данных, необходимо выполнять проверку перед присвоением параметров. В противном случае, возникнет ошибка выполнения PHP скрипта, если какой-либо из параметров будет отсутствовать в текущем источнике данных. Например, в отчете создано два источника данных, оба имеют один общий параметр, и один параметр только во втором источнике данных:

index.php

```
$report->onBeginProcessData->append(function (StiDataEventArgs $args) {
   $args->parameters['Parameter1']->value = 'TableName';

   if ($args->dataSource == 'DataSource2')
      $args->parameters['Parameter2']->value = 10;
});
```

Использование переменной отчета в качестве SQL параметра

Предусмотрена возможность использования переменной в качестве SQL параметра. Для этого в редакторе переменной отчета достаточно установить свойство **Allow using as SQL parameter**, после чего её можно использовать в любом SQL запросе. Синтаксис будет точно таким же, как и при использовании параметров в источнике данных.

© 2003-2026 Stimulsoft

Информация

Такая переменная будет передана в коллекции параметров только в том случае, если она используется в запросе. Параметры из коллекции источника данных передаются всегда, даже если в запросе они не использованы.

Экранирование значений параметров

Все значения параметров будут автоматически экранированы для исключения возможности SQL инъекций и нарушения безопасности выполнения запросов. Если вам не требуется экранирование, и вы сами контролируете безопасность значений параметров, автоматическое экранирование можно отключить. Для этого достаточно установить свойство `escapeQueryParameters` в значение `false` у обработчика событий:

index.php

```
<?php
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->handler->escapeQueryParameters = false;
?>
```

После установки указанного свойства, использование параметров является небезопасным, вам нужно строго контролировать значения перед выполнением SQL запросов.

Информация

Экранирование применяется только для параметров SQL запросов, а также для переменных, используемых как параметры. Если переменная используется в виде выражения, т.е. в фигурных скобках, например `{VariableName}`, то экранирование применяться не будет в любом случае. Подробное описание работы переменных находится в разделе Работа с переменными отчета.

© 2003-2026 Stimulsoft

Шифрование данных, передаваемых на сторону клиента

Если включено общее шифрование данных при помощи опции `$encryptData`, то все передаваемые на сторону клиента данные будут зашифрованы, включая данные из SQL источников. Это повышает безопасность, но может замедлить работу на больших объёмах данных. Предусмотрена возможность отдельного отключения шифрования SQL данных, передаваемых на сторону клиента. Для этого достаточно установить свойство `$encryptSqlData` в значение `false` у обработчика событий, после этого все данные будут передаваться в открытом виде в JSON формате. Это не должно в значительной мере повлиять на безопасность, т.к. в незашифрованном виде будут переданы только сами данные, которые в том или ином виде отображаются в отчёте.

index.php

```
<?php
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->handler->encryptSqlData = false;
   $report->process();
?>
```

9.1.11 Работа с переменными отчета

Генератор отчетов позволяет использовать переменные в выражениях, запросах, фильтрах и других элементах отчета. Предусмотрена возможность предварительного просмотра и изменения значений переменных из кода перед построением отчета.

Доступ к значениям переменных из кода

Генератор отчетов позволяет легко получить прямой доступ к переменным в шаблоне отчета через словарь данных. Для этого достаточно определить событие `onBeforeRender` у объекта отчета, в аргументах которого на стороне JavaScript клиента будет передан объект отчета. Подробное описание доступных значений свойств, передаваемых в аргументах события, находится в разделе События генератора отчетов.

Для доступа к переменной отчета, необходимо использовать JavaScript функцию `getByName()` у коллекции переменных в словаре данных отчета. Для

© 2003-2026 Stimulsoft

изменения значения переменной, достаточно присвоить его в свойстве `value`, при этом нет необходимости проверять установленный тип переменной - преобразование типов будет выполнено автоматически.

Пример изменения строкового и целочисленного значение выбранных переменных на стороне JavaScript клиента:

index.php

```
<?php
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->onBeforeRender = 'beforeRender';
   $report->loadFile('reports/Variables.mrt');
   $report->render();
?>

<script>
   function beforeRender(args) {
      let report = args.report;

      let variableString =
      report.dictionary.variables.getByName("VariableString");
      variableString.value = "Text value";

      let variableInt =
      report.dictionary.variables.getByName("VariableInt");
      variableInt.value = "20";
   }
</script>
```

Прямой доступ к переменным в шаблоне отчета на стороне PHP сервера не предусмотрен.

Информация

Значения переменных будут изменены в самом шаблоне отчета, и при его последующем сохранении будут сохранены в файле. Для изменения значений переменных без изменения шаблона, можно воспользоваться событием `onPrepareVariables`, которое вызывается при подготовке значений переменных отчета перед его построением.

© 2003-2026 Stimulsoft

Событие подготовки значений переменных

Генератор отчетов позволяет легко получить доступ к переменным перед построением отчета. Для этого достаточно определить событие onPrepareVariables у объекта отчета. В аргументах события будет передана коллекция переменных отчета с их типами и значениями. Если переменная инициализируется как выражение, то в коллекцию будет передано уже подсчитанное значение выражения. Подробное описание доступных значений свойств, передаваемых в аргументах события, находится в разделе События генератора отчетов.

В событии на стороне JavaScript клиента коллекция переменных представляет собой массив из объектов, содержащих имя переменной, ее тип и значение. Допустимо изменять значения переменных, при этом тип нового значения должен соответствовать типу изменяемой переменной.

index.php

```
<script>
   function prepareVariables(args) {
      args.variables = [
         {
            name: "VariableString",
            type: "String",
            value: "Text value"
         },
         {
            name: "VariableInt",
            type: "Int32",
            value: 20
         }
     ]};
</script>
```

Пример изменения значения переменной на стороне JavaScript клиента:

© 2003-2026 Stimulsoft

index.php

```
<?php
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->onPrepareVariables = 'prepareVariables';
   $report->loadFile('reports/Variables.mrt');
   $report->render();
?>

...

<script>
   function prepareVariables(args) {
      let variables = args.variables;

      variables.find(item => item.name == "VariableString").value = "Text
      value";
      variables.find(item => item.name == "VariableInt").value = 20;
   }
</script>
```

На стороне PHP сервера также допустимо изменять значения переменных, при этом тип нового значения должен соответствовать типу изменяемой переменной. Предусмотрена возможность создания новой переменной отчета, если это необходимо. На сторону клиента будет передана коллекция, содержащая только те переменные, значения которых были изменены, а также новые переменные.

Пример изменения значения переменной на стороне PHP сервера:

index.php

```
<?php
   use Stimulsoft\Events\StiVariablesEventArgs;
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->onPrepareVariables = function (StiVariablesEventArgs $args) {
      $args->variables['VariableString']->value = 'Text value';
      $args->variables['VariableInt']->value = 20;
   };
   $report->loadFile('reports/Variables.mrt');
   $report->render();
?>
```

© 2003-2026 Stimulsoft

Полный код примера доступен на GitHub.

Для изменения значения простой переменной, достаточно заменить значение value в коллекции переменных. Значение должно быть того же типа, что и исходное. Переменные типа DateTime передаются в виде строкового значения в формате "YYYY-MM-dd HH-mm-ss".

index.php

```
$args->variables['VariableString']->value = 'Value from Server-Side';
$args->variables['VariableInt']->value = 123;
$args->variables['VariableDecimal']->value = 123.456;
$args->variables['VariableDateTime']->value = '2021-03-20 22:00:00';
```

Для изменения значения переменной типа `Range`, необходимо использовать вложенные значения `value->from` и `value->to` выбранной переменной в коллекции. Формат каждого из двух значений такой же, как и у простой переменной:

index.php

```
$args->variables['VariableStringRange']->value->from = 'Aaa';
$args->variables['VariableStringRange']->value->to = 'Zzz';
```

Для изменения значения переменной типа `List`, необходимо обратиться к значению списка по его индексу. Допускается установка значений сразу всего списка в виде подготовленного массива:

index.php

```
$args->variables['VariableStringList']->value[0] = 'Test';
$args->variables['VariableStringList']->value = ['1', '2', '2'];
```

© 2003-2026 Stimulsoft

Для создания новой переменной, не определенной в отчете, необходимо присвоить подготовленный ассоциативный массив в формате `['value' => 'New Value']` в коллекцию переменных, используя новое имя переменной. После этого переменную можно использовать для построения отчета, то есть в шаблоне отчета переменная сохранена не будет.

index.php

```
$args->variables['NewVariable'] = ['value' => 'New Value'];
```

Информация

Если переменная используется в SQL запросе в виде выражения, т.е. записана в фигурных скобках, например `{VariableName}`, то ее значение не будет автоматически экранировано. Вам необходимо самим следить за безопасностью значений, либо использовать переменную как параметр запроса, например `@VariableName`. Подробное описание работы параметров находится в разделе Подключение SQL адаптеров данных.

Переменные отчета, передаваемые в URL запросе

Генератор отчетов имеет возможность автоматически присваивать значения переменных, переданных в URL запросе. Для этого необходимо установить свойство `$passQueryParametersToReport` в значение `true` у обработчика событий:

index.php

```
<?php
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->handler->passQueryParametersToReport = true;
   $report->process();
?>
```

Все остальные действия генератор отчетов сделает автоматически. Если

© 2003-2026 Stimulsoft

переменная присутствует в отчете, то ее значение будет изменено на значение из URL запроса. Если такой переменной в отчете нет - она будет создана для построения отчета, то есть в шаблоне отчета новая переменная сохранена не будет. Имена переменных, передаваемых в URL запросе, не чувствительны к регистру.

Информация

Все значения переменных будут присвоены до вызова события `onPrepareVariables`, таким образом в этом событии можно дополнительно контролировать установленные значения, и при необходимости выполнять их корректировку.

9.1.12 Подключение пользовательских шрифтов

В генераторе отчетов доступна возможность загрузки пользовательских шрифтов из файла либо директории. Для этого предназначен статический класс `StiFontCollection`, который содержит необходимые функции по работе со шрифтами.

Для подключения файла шрифта предназначена функция `addFontFile`, в аргументах необходимо указать путь к файлу шрифта, а также при необходимости имя шрифта и его стиль. Если имя шрифта либо стиль не указаны, то будут использоваться параметры из файла шрифта. Пример подключения файла шрифта:

report.php

```
<?php
   use Stimulsoft\Enums\FontStyle;
   use Stimulsoft\StiFontCollection;

   StiFontCollection::addFontFile('Roboto-Black.ttf', 'Roboto',
   FontStyle::Bold);
?>
```

Дополнительно предусмотрена возможность загрузки всех шрифтов, находящихся в одной директории. Для этого достаточно указать необходимую директорию со шрифтами при помощи функции `setFontsFolder`, например:

© 2003-2026 Stimulsoft

report.php

```
<?php
   use Stimulsoft\StiFontCollection;

   StiFontCollection::setFontsFolder('/fonts');
?>
```

9.1.13 Печать отчета из кода

В генераторе отчетов доступна возможность печати отчета из кода. Для этого можно воспользоваться специальным методом `print()` у объекта отчета. Пример вызова диалога печати предварительно построенного отчета:

index.php

```
<?php
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->loadFile('reports/SimpleList.mrt');
   $report->render();
   $report->print();
   $report->printHtml();
?>
```

Полный код примера доступен на GitHub.

Информация

Печать отчета не вызывает его автоматическое построение, поэтому для загруженного шаблона отчета требуется предварительно вызвать метод `render()`, который построит отчет. Для готовых документов (построенных отчетов) указанный метод не требуется.

По умолчанию будут напечатаны все страницы построенного отчета. Предусмотрена возможность указать страницу либо диапазон страниц для печати. Для этого достаточно передать необходимое значение в качестве

© 2003-2026 Stimulsoft

параметра для функции `print()`, это может быть определенная страница отчета либо диапазон страниц. Пример печати указанных страниц отчета:

index.php

```
<?php
   $report->print(5);
   $report->print('1,3-8');
?>
```

9.1.14 Экспорт отчета из кода

В генераторе отчетов доступна возможность экспортирования отчета либо дашборда в различные форматы. В таблице ниже перечислены все доступные форматы экспорта для отчетов и дашбордов:

Формат экспорта	Отчеты	Дашборды
Document (Snapshot)	+	+
Adobe PDF	+	+
XPS (XML Paper Specification)	+	-
Microsoft PowerPoint	+	-
HTML	+	+
HTML5	+	-
Text	+	-
Microsoft Word	+	-
Microsoft Excel	+	+
OpenDocument Writer	+	-
OpenDocument Calc	+	-
RTF (Rich Text Format)	+	-

Формат данных	Отчеты	Дашборды
CSV (Comma Separated Value)	+	+

© 2003-2026 Stimulsoft

JSON (JavaScript Object Notation)	+	+
XML (Extensible Markup Language)	+	+
DBF (dBase/FoxPro)	+	+
DIF	+	+
SYLK (Symbolic Link)	+	+

Формат изображения	Отчеты	Дашборды
PNG (Portable Network Graphics)	+	+
JPEG (Joint Photographic Experts Group)	+	+
GIF (Graphics Interchange)	+	+
TIFF (Tagged Image File Format)	+	+
SVG (Scalable Vector Graphics)	+	+
SVGZ (Compressed SVG)	+	+
PCX (Picture Exchange)	+	+
BMP (Windows Bitmap)	+	+

Для экспортирования необходимо воспользоваться специальным методом `exportDocument()` у объекта отчета:

© 2003-2026 Stimulsoft

index.php

```
<?php
   use Stimulsoft\Export\Enums\StiExportFormat;
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->loadFile('reports/SimpleList.mrt');
   $report->render();
   $report->exportDocument(StiExportFormat::Pdf);
   $report->printHtml();
?>
```

Полный код примера доступен на GitHub.

Информация

Экспортирование отчета не вызывает его автоматическое построение, поэтому для загруженного шаблона отчета требуется предварительно вызвать метод `render()`, который построит отчет. Для готовых документов (построенных отчетов) указанный метод не требуется.

В качестве аргументов метода `exportDocument()` необходимо указать требуемый формат экспорта из перечисления `StiExportFormat`, а так же при необходимости настройки экспортирования. Доступные форматы указаны в таблице ниже:

Наименование	Описание
`StiExportFormat::Document`	Сохранение документа (построенного отчета).
`StiExportFormat::Pdf`	Сохранение в Adobe PDF формат.
`StiExportFormat::Xps`	Сохранение в XPS (XML Paper Specification) формат.
`StiExportFormat::PowerPoint`	Сохранение в Microsoft PowerPoint формат.

© 2003-2026 Stimulsoft

`StiExportFormat::Html`	Сохранение в HTML формат.
`StiExportFormat::Html5`	Сохранение в HTML5 формат, с использованием элементов SVG разметки.
`StiExportFormat::Text`	Сохранение в текстовый формат.
`StiExportFormat::Word`	Сохранение в Microsoft Word формат.
`StiExportFormat::Excel`	Сохранение в Microsoft Excel формат.
`StiExportFormat::Odt`	Сохранение в OpenDocument Writer формат.
`StiExportFormat::Ods`	Сохранение в OpenDocument Calc формат.
`StiExportFormat::Rtf`	Сохранение в RTF (Rich Text Format) формат.
`StiExportFormat::Csv`	Сохранение в CSV (Comma Separated Values) формат данных.
`StiExportFormat::Json`	Сохранение в JSON (JavaScript Object Notation) формат данных.
`StiExportFormat::Xml`	Сохранение в XML (Extensible Markup Language) формат данных.
`StiExportFormat::Dbf`	Сохранение в DBF (dBase/FoxPro) формат данных.
`StiExportFormat::Dif`	Сохранение в DIF формат данных.
`StiExportFormat::Sylk`	Сохранение в SYLK (Symbolic Link) формат данных.
`StiExportFormat::ImagePng`	Сохранение в изображение PNG (Portable Network Graphics) формата.
`StiExportFormat::ImageJpeg`	Сохранение в изображение JPEG (Joint Photographic Experts Group) формата.

© 2003-2026 Stimulsoft

`StiExportFormat::ImageGif`	Сохранение в изображение GIF (Graphics Interchange) формата.
`StiExportFormat::ImageTiff`	Сохранение в изображение TIFF (Tagged Image File Format) формата.
`StiExportFormat::ImageSvg`	Сохранение в изображение SVG (Scalable Vector Graphics) формата.
`StiExportFormat::ImageSvgz`	Сохранение в изображение SVGZ (Compressed SVG) формата.
`StiExportFormat::ImagePcx`	Сохранение в изображение PCX (Picture Exchange) формата.
`StiExportFormat::ImageBmp`	Сохранение в изображение BMP (Windows Bitmap) формата.

После экспорта отчета, полученный поток данных будут передан браузеру на скачивание в виде файла. Имя файла и MIME тип данных будет определён автоматически. Предусмотрена возможность отобразить экспортированный отчет сразу в окне браузера (только для PDF, HTML и изображений), для этого необходимо установить аргумент `openAfterExport` в значение `true`:

index.php

```
<?php
   $report->exportDocument(StiExportFormat::Pdf, null, true);
?>
```

Экспортирование отчета на стороне сервера

Для переключения генератора отчетов в режим работы на стороне сервера, необходимо установить свойство отчета engine в значение `StiEngineType::ServerNodeJS`, весь остальной набор событий и методов для работы с отчетом точно такой же, как и при экспортировании отчета на стороне клиента. При экспорте на стороне сервера, метод `exportDocument()` вернет байтовый поток данных экспортированного отчета.

Пример экспортирования отчета в текстовый формат на стороне сервера:

© 2003-2026 Stimulsoft

index.php

```
<?php
   use Stimulsoft\Export\Enums\StiExportFormat;
   use Stimulsoft\Report\Enums\StiEngineType;
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->engine = StiEngineType::ServerNodeJS;
   $report->loadFile('reports/SimpleList.mrt');
   $report->render();
   $result = $report->exportDocument(StiExportFormat::Text);
?>
```

Полный код примера доступен на GitHub.

Предусмотрена возможность автоматически сохранить экспортированный отчет в виде файла на стороне сервера. Для этого необходимо указать путь сохранения в аргументе `filePath` метода экспорта. В этом случае, метод `exportDocument()` вернет булевый результат экспортирования и сохранения файла.

Пример экспортирования отчета в формат Adobe PDF на стороне сервера и сохранение результата в указанную директорию:

index.php

```
<?php
   use Stimulsoft\Export\Enums\StiExportFormat;
   use Stimulsoft\Report\Enums\StiEngineType;
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->engine = StiEngineType::ServerNodeJS;
   $report->loadFile('reports/SimpleList.mrt');
   $report->render();
   $exportedFilePath = 'reports/SimpleList.pdf';
   $result = $report->exportDocument(StiExportFormat::Pdf, null, false,
   $exportedFilePath);
?>
```

Настройки экспортирования отчета

© 2003-2026 Stimulsoft

При экспорте отчета из кода доступна возможность установить необходимые настройки экспортирования. Для каждого формата экспорта предназначен свой класс настроек, указанный в таблице ниже:

Формат экспорта отчета	Класс настроек
StiExportFormat::Document	Настройки не предусмотрены.
StiExportFormat::Pdf	StiPdfExportSettings()
StiExportFormat::Xps	StiXpsExportSettings()
StiExportFormat::PowerPoint	StiPowerPointExportSettings()
StiExportFormat::Html StiExportFormat::Html5	StiHtmlExportSettings()
StiExportFormat::Text	StiTxtExportSettings()
StiExportFormat::Word	StiWordExportSettings()
StiExportFormat::Excel	StiExcelExportSettings()
StiExportFormat::Odt	StiOdtExportSettings()
StiExportFormat::Ods	StiOdsExportSettings()
StiExportFormat::Rtf	StiRtfExportSettings()
StiExportFormat::Csv StiExportFormat::Json StiExportFormat::Xml StiExportFormat::Dbf StiExportFormat::Dif StiExportFormat::Sylk	StiDataExportSettings()
StiExportFormat::ImagePng StiExportFormat::ImageJpeg StiExportFormat::ImageGif StiExportFormat::ImageTiff StiExportFormat::ImageSvg	StiImageExportSettings()

© 2003-2026 Stimulsoft

StiExportFormat::ImageSvgz StiExportFormat::ImagePcx StiExportFormat::ImageBmp	

Формат экспорта дашборда	Класс настроек
StiExportFormat::Pdf	StiPdfDashboardExportSettings()
StiExportFormat::Html	StiHtmlDashboardExportSettings()
StiExportFormat::Excel	StiExcelDashboardExportSettings()
StiExportFormat::Csv	StiDataDashboardExportSettings()
StiExportFormat::ImageSvg	StiImageDashboardExportSettings()

Для применения настроек экспортирования, достаточно передать подготовленный объект настроек в качестве параметра методу экспорта отчета. Все остальные действия будут выполнены автоматически.

Пример экспортирования отчета в формат Adobe PDF на стороне сервера, в настройках указываются сведения о компании и выключаются встроенные шрифты:

© 2003-2026 Stimulsoft

index.php

```
<?php
   use Stimulsoft\Export\Enums\StiExportFormat;
   use Stimulsoft\Export\StiPdfExportSettings;
   use Stimulsoft\Report\Enums\StiEngineType;
   use Stimulsoft\Report\StiReport;

   $report = new StiReport();
   $report->engine = StiEngineType::ServerNodeJS;
   $report->loadFile('reports/SimpleList.mrt');
   $report->render();

   $settings = new StiPdfExportSettings();
   $settings->creatorString = 'My Company Name';
   $settings->keywordsString = 'SimpleList PHP Report Export';
   $settings->embeddedFonts = false;

   $result = $report->exportDocument(StiExportFormat::Pdf, $settings);
?>
```

Полный код примера доступен на GitHub.

Информация

Класс объекта настроек экспорта должен соответствовать формату экспорта, в противном случае приоритетным будет считаться формат, возвращаемый методом `getExportFormat()` объекта настроек экспорта.

9.1.15 События генератора отчетов

Генератор отчетов поддерживает события, которые предоставляют возможность выполнить необходимые операции перед определенными действиями - как на стороне JavaScript клиента, включая платформу Node.js, так и на стороне PHP сервера.

Для вызова события на стороне JavaScript клиента, необходимо добавить к событию название JavaScript функции, определенной в текущем HTML шаблоне. При отсутствии HTML шаблона, а также при построении либо экспорте отчета на стороне сервера при помощи Node.js платформы, вместо названия функции необходимо к событию добавить саму функцию в виде строки или строк кода. Аргументы события будут переданы в параметрах функции, либо в

© 2003-2026 Stimulsoft

предварительно заданной переменной `args`, которую можно использовать в коде события.

Для вызова события на стороне PHP сервера, необходимо добавить к событию саму PHP функцию, определенную ранее в коде. Допускается присвоение анонимной PHP функции для события. Аргументы события будут переданы в параметрах функции.

Для одного события можно добавлять любое количество функций, как клиентских, так и серверных. В этом случае, необходимо воспользоваться специальным методом `append()` вместо присвоения функции, а саму функцию передать в качестве параметра. Обработчик событий сгруппирует функции по типу и выполнит их вызов в порядке добавления.

Пример различных вариантов добавления функций разного типа к событию:

© 2003-2026 Stimulsoft

index.php

```
<?php
   use Stimulsoft\Events\StiDataEventArgs;
   use Stimulsoft\Events\StiVariablesEventArgs;
   use Stimulsoft\Report\StiReport;

   function prepareVariables(StiVariablesEventArgs $args) {
      $variables = $args->variables;
   };

   $report = new StiReport();
   $report->onPrepareVariables->append(prepareVariables);
   $report->onPrepareVariables->append('prepareVariables');
   $report->onBeginProcessData = function(StiDataEventArgs $args) {
      $args->connectionString =
      'Server=localhost;Database=test;uid=root;password=******;';
   };

   $report->onBeforeRender = 'args.report.dictionary.clear();';
   $report->onAfterRender = 'afterRender';

   $report->loadFile('reports/Variables.mrt');
   $report->render();
?>

...

<script>
   function prepareVariables(args) {
      let variables = args.variables;
   }

   function afterRender(args) {
      alert("The report rendering is completed.");
   }
</script>
```

В зависимости от события, оно может быть вызвано на стороне JavaScript клиента и на стороне PHP сервера одновременно, либо только на стороне JavaScript клиента, либо только на стороне PHP сервера. Это связано с архитектурой компонентов - для построения и экспортирования отчета используется JavaScript ядро генератора, которое работает на стороне клиента, а для работы с данными используется серверный PHP код. Эти две составляющие не имеет прямого доступа друг к другу, поэтому события работают раздельно, а передачу данных обеспечивает обработчик событий. В описании каждого события будет указано, какой именно тип может быть использован. Подробное описание обработчика событий и примеры

© 2003-2026 Stimulsoft

использования находится в разделе Обработчик событий.

Информация

Все события, работающие на стороне JavaScript клиента, в равной степени работают на стороне Node.js сервера, т.к. в этом случае так же используется JavaScript ядро отчетов.

Генератор отчетов поддерживает следующие события:

- `onDatabaseConnect`
- `onBeforeRender`
- `onAfterRender`
- `onBeginProcessData`
- `onEndProcessData`
- `onPrepareVariables`

onDatabaseConnect
[-] JavaScript [+] PHP

Событие вызывается перед подключением к базе данных после получения всех параметров. Подробное описание и примеры использования находятся в разделе и Подключение SQL адаптеров данных. В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне PHP сервера:

Наименование	Описание
`event`	Идентификатор текущего события, для данного события имеет значение `StiEventType::DatabaseConnect`
`sender`	Компонент, который инициировал данное событие, может иметь следующие типы: • `StiViewer`

© 2003-2026 Stimulsoft

	• StiDesigner
database	Тип базы данных, может принимать одно из значений перечисления StiDatabaseType.
driver	Название используемого PHP драйвера базы данных.
info	Параметры подключения к базе данных, полученные из строки подключения.
link	Идентификатор подключения к базе данных. По умолчанию имеет значение null, в этом случае подключение будет создано адаптером данных.

onBeforeRender
[+] JavaScript [+] PHP

Событие вызывается перед построением отчета. Подробное описание и примеры использования находятся в разделе и Построение отчета.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне JavaScript клиента:

Наименование	Описание
event	Идентификатор текущего события, имеет значение "BeforeRender".
sender	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • "Report"
report	Текущий объект отчета.

© 2003-2026 Stimulsoft

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне PHP сервера:

Наименование	Описание
`event`	Идентификатор текущего события, для данного события имеет значение `StiEventType::BeforeRender`
`sender`	Компонент, который инициировал данное событие, может иметь следующие типы: • `StiReport`
`report`	Текущий объект отчета.
`regReportData($name, $data, $synchronize = false)`	Метод, позволяющий передать данные в отчет. В качестве аргументов может принимать следующие значения: `$name` – имя источника данных в отчете; `$data` – данные в виде XML либо JSON строки, либо в виде PHP объекта или массива; `$synchronize` – флаг, указывающий на необходимость синхронизации данных, по умолчанию `false`.

onAfterRender
[+] JavaScript [-] PHP

Событие вызывается после построения отчета. Подробное описание и примеры использования находятся в разделе и Построение отчета.

В таблице ниже представлен список свойств, передаваемых в аргументах

© 2003-2026 Stimulsoft

события на стороне JavaScript клиента:

Наименование	Описание
`event`	Идентификатор текущего события, имеет значение "`AfterRender`".
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • "`Report`"
`report`	Текущий объект отчета.

onBeginProcessData
[+] JavaScript [+] PHP

Событие вызывается перед запросом данных, необходимых для построения отчета. Подробное описание и примеры использования находятся в разделах Подключение файловых данных и Подключение SQL адаптеров данных.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне JavaScript клиента:

Наименование	Описание
`event`	Идентификатор текущего события, имеет значение "`BeginProcessData`".
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • "`Report`" • "`Viewer`" • "`Designer`"

© 2003-2026 Stimulsoft

report	Текущий объект отчета.
command	Идентификатор текущей команды, может принимать следующие значения: • "TestConnection" - выполняется проверка соединения; • "ExecuteQuery" - выполняется запрос данных из указанного SQL источника. • "GetSchema" - выполняется чтение XSD схемы из файлового источника. • "GetData" - выполняется чтение данных из файлового источника.
connection	Имя текущего подключения к источнику данных, заданное в шаблоне отчета.
database	Наименование текущей базы данных. Может принимать следующие значения: • "XML" • "JSON" • "Excel" • "CSV" • "MySQL" • "MS SQL" • "PostgreSQL" • "Firebird" • "Oracle" • "ODBC"
dataSource	Имя текущего источника данных, заданное в шаблоне отчета.

© 2003-2026 Stimulsoft

	Устанавливается только для SQL источников данных.
`connectionString`	Строка подключения к SQL источнику данных.
`queryString`	SQL запрос на получение данных. Используется только с командой `ExecuteQuery`.
`pathData`	Путь к файлу источника данных, заданный в шаблоне отчета. Устанавливается только для XML и JSON источников данных.
`pathSchema`	Путь к файлу схемы данных, заданный в шаблоне отчета. Устанавливается только для XML источника данных.
`parameters`	Коллекция параметров и их значений, заданная в SQL источнике данных.
`preventDefault`	Данный флаг предоставляет возможность остановить дальнейшую обработку события генератором отчетов. По умолчанию установлено значение `false`.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне PHP сервера:

Наименование	Описание
`event`	Идентификатор текущего события, для данного события имеет значение `StiEventType::BeginProcessData`

© 2003-2026 Stimulsoft

`sender`	Компонент, который инициировал данное событие, может иметь следующие типы: • `StiReport` • `StiViewer` • `StiDesigner`
`command`	Идентификатор текущей команды, может принимать следующие значения: • `StiDataCommand::TestConnection` - выполняется проверка соединения; • `StiDataCommand::RetrieveSchema` - выполняется запрос схемы базы данных для NoSQL источников данных; • `StiDataCommand::ExecuteQuery` - выполняется запрос данных из указанного SQL источника. • `StiDataCommand::Execute` - выполняется хранимая процедура указанного SQL источника.
`connection`	Имя текущего подключения к источнику данных, заданное в шаблоне отчета.
`database`	Наименование текущей базы данных. Может принимать следующие значения: • `StiDatabaseType::MySQL` • `StiDatabaseType::MSSQL` • `StiDatabaseType::PostgreSQL` • `StiDatabaseType::Firebird`

© 2003-2026 Stimulsoft

	• StiDatabaseType::Oracle • StiDatabaseType::MongoDB • StiDatabaseType::ODBC
dataSource	Имя текущего источника данных, заданное в шаблоне отчета.
connectionString	Строка подключения к SQL источнику данных.
queryString	SQL запрос на получение данных. Используется только с командой StiDataCommand::ExecuteQuery.
pathData	Путь к файлу источника данных, заданный в шаблоне отчета. Устанавливается только для XML и JSON источников данных.
pathSchema	Путь к файлу схемы данных, заданный в шаблоне отчета. Устанавливается только для XML источника данных.
parameters	Коллекция параметров и их значений, заданная в SQL источнике данных. Значения параметров всегда передаются в первоначальном (не экранированном) виде.

onEndProcessData
[+] JavaScript [+] PHP

Событие вызывается после загрузки данных перед построением отчета. Подробное описание и примеры использования находятся в разделах Подключение файловых данных и Подключение SQL адаптеров данных.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне JavaScript клиента:

© 2003-2026 Stimulsoft

Наименование	Описание
`event`	Идентификатор текущего события, имеет значение "`EndProcessData`".
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • "`Report`" • "`Viewer`" • "`Designer`"
`report`	Текущий объект отчета.
`command`	Идентификатор текущей команды, может принимать следующие значения: • "`ExecuteQuery`" - данные получены из указанного SQL источника. • "`GetData`" - данные получены из файлового источника.
`connection`	Имя текущего подключения к источнику данных, заданное в шаблоне отчета.
`database`	Наименование текущей базы данных. Может принимать следующие значения: • "`XML`" • "`JSON`" • "`Excel`" • "`CSV`" • "`MySQL`" • "`MS SQL`"

© 2003-2026 Stimulsoft

	• "PostgreSQL" • "Firebird" • "Oracle" • "ODBC"
dataSource	Имя текущего источника данных, заданное в шаблоне отчета. Устанавливается только для SQL источников данных.
dataSet	Подготовленный объект Stimulsoft.System.Data.DataSet, содержащий таблицы и строки данных, полученных из файлового источника.
result	Коллекция колонок и их типов, а также строк данных, полученных из SQL источника.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне PHP сервера:

Наименование	Описание
event	Идентификатор текущего события, для данного события имеет значение StiEventType::EndProcessData
sender	Компонент, который инициировал данное событие, может иметь следующие типы: • StiReport • StiViewer • StiDesigner
command	Идентификатор текущей команды, может принимать следующие

© 2003-2026 Stimulsoft

	значения: • `StiDataCommand::TestConnection` - выполняется проверка соединения; • `StiDataCommand::RetrieveSchema` - выполняется запрос схемы базы данных для NoSQL источников данных; • `StiDataCommand::ExecuteQuery` - выполняется запрос данных из указанного SQL источника; • `StiDataCommand::Execute` - выполняется хранимая процедура указанного SQL источника.
`connection`	Имя текущего подключения к источнику данных, заданное в шаблоне отчета.
`database`	Наименование текущей базы данных. Может принимать следующие значения: • `StiDatabaseType::MySQL` • `StiDatabaseType::MSSQL` • `StiDatabaseType::PostgreSQL` • `StiDatabaseType::Firebird` • `StiDatabaseType::Oracle` • `StiDatabaseType::MongoDB` • `StiDatabaseType::ODBC`
`dataSource`	Имя текущего источника данных, заданное в шаблоне отчета. Устанавливается только для SQL источников данных.

© 2003-2026 Stimulsoft

`queryString`	Итоговый SQL запрос со всеми параметрами, который был выполнен для получения данных. Используется только с командой `StiDataCommand::ExecuteQuery`.
`result`	Коллекция колонок и их типов, а также строк данных, полученных из SQL источника, либо строковые данные, полученные из файлового источника.

onPrepareVariables
[+] JavaScript [+] PHP

Событие вызывается перед построением отчета после подготовки переменных отчета. Подробное описание и примеры использования находятся в разделе Работа с переменными отчета.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне JavaScript клиента:

Наименование	Описание
`event`	Идентификатор текущего события, имеет значение `"PrepareVariables"`.
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • `"Report"` • `"Viewer"` • `"Designer"`
`report`	Текущий объект отчета.

© 2003-2026 Stimulsoft

variables	Коллекция переменных отчета и их значений.
preventDefault	Данный флаг предоставляет возможность остановить дальнейшую обработку события генератором отчетов. По умолчанию установлено значение false.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне PHP сервера:

Наименование	Описание
event	Идентификатор текущего события, для данного события имеет значение StiEventType::PrepareVariables.
sender	Компонент, который инициировал данное событие, может иметь следующие типы: • StiReport • StiViewer • StiDesigner
variables	Коллекция переменных отчета и их значений.

9.2 HTML5 вьювер

Вьювер отчетов представляет собой PHP компонент (StiViewer), предназначенный для просмотра, печати и экспортирования отчетов и дашбордов в окне браузера на любом компьютере с любой установленной операционной системой. Вьювер поддерживает множество тем оформления, анимированный интерфейс, закладки, интерактивные отчеты, редактирование элементов отчета на странице, полноэкранный режим, поиск, и другие, необходимые для просмотра отчетов, возможности.

© 2003-2026 Stimulsoft

Вьювер может вывести на просмотр шаблон отчета, построенный отчет, панель индикаторов (дашборд). В случае использования шаблона отчета, вьювер построит его автоматически, используя JavaScript генератор отчетов. Работа генератора более подробно рассмотрена в разделе Генератор отчетов.

Интерфейс вьювера построен с использованием HTML5, что позволяет использовать его практически на любой современной платформе. Компонент использует технологию AJAX для выполнения всех действий (загрузка и построение отчета, подключение к данным, листания страниц и масштабирования, интерактивности в отчетах, печать, экспортирование и др.), что позволяет избавиться от перезагрузки всей страницы и повысить скорость работы, а также использовать компонент в One-Page приложениях. Технология JavaScript, применяемая для построения отчетов, позволяет использовать практически любую серверную сторону невысокой производительности.

Информация

Вследствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово “отчет”.

- Использование
- Активация лицензии
- Отображение отчета и дашборда
- Локализация
- Печать отчета
- Экспорт отчета и дашборда
- Режимы отображения
- Работа с закладками
- Динамическое сворачивание, сортировка и детализация
- Редактирование построенного отчета
- Отправка отчета по Email
- Вызов дизайнера
- Визуальное оформление
- События вьювера

© 2003-2026 Stimulsoft

- Работа с переменными отчета
- Настройки вьювера

9.2.1 Использование

Для использования продукта, достаточно загрузить ZIP архив продукта со страницы Downloads нашего сайта, распаковать его, и скопировать содержимое папки /PHP на ваш Web сервер. Данная папка представляет собой Web проект, который содержит все необходимые файлы и ресурсы для работы продукта, а также примеры по работе с вьювером и дизайнером.

Для установки продукта в уже существующий проект, достаточно из папки **/PHP** скопировать папку `/vendor` в корневую директорию проекта, либо воспользоваться менеджером зависимостей Composer, выполнив следующую консольную команду:

console

```
composer require stimulsoft/reports-php
```

Для работы дашбордов понадобится подключить следующий пакет:

console

```
composer require stimulsoft/dashboards-php
```

При работе с продуктом, в большинстве случаев достаточно использовать только PHP код, который обеспечивает работу всех основных возможностей. Для более детальной настройки продукта и использования всех возможностей, необходимо использовать JavaScript код.

Для использования компонентов в Web проекте, достаточно подключить автоматический загрузчик скриптов в начале PHP файла. После этого, можно использовать все доступные PHP классы и функции для работы с отчетами и дашбордами:

© 2003-2026 Stimulsoft

index.php

```
<?php
   require_once 'vendor/autoload.php';

...
?>
```

Для использования вьювера в Web-проекте, предназначен класс `StiViewer`. При помощи данного класса можно создать объект вьювера, установить необходимые настройки, присвоить отчет для просмотра, обрабатывать запросы и управлять событиями вьювера.

Пример отображения отчета во вьювере на HTML странице:

viewer.php

```
<?php
   require_once 'vendor/autoload.php';

   use Stimulsoft\Report\StiReport;
   use Stimulsoft\Viewer\StiViewer;

   $viewer = new StiViewer();
   $viewer->process();

   $report = new StiReport();
   $report->loadFile('reports/SimpleList.mrt');
   $viewer->report = $report;
?>

<html>
<head>
   <?php
      $viewer->javascript->renderHtml();
   ?>
</head>
<body>
   <?php
      $viewer->renderHtml();
   ?>
</body>
</html>
```

Полный код примера доступен на GitHub.

© 2003-2026 Stimulsoft

В этом примере последовательно выполняются следующие действия:

- создается экземпляр объекта `StiViewer`;
- выполняется обработка текущего запроса;
- создается экземпляр объекта `StiReport`;
- выполняется загрузка шаблона отчета из файла SimpleList.mrt;
- созданный отчет присваивается вьюверу;
- в шаблоне HTML файла выполняется вывод необходимого JavaScript и HTML кода компонента.

Метод `$viewer->process()` выполняет обработку текущего запроса, и, в случае успеха, автоматически возвращает результат на сторону клиента. Более подробно про это рассказано в разделе Обработчик событий.

Метод `$viewer->javascript->renderHtml()` выполняет вывод кода для подключения необходимых скриптов компонента. Метод `$viewer->renderHtml()` выполняет вывод JavaScript и HTML кода самого компонента. Предусмотрено несколько вариантов использования, более подробно про это рассказано в разделе Использование генератора отчетов.

Пример упрощённого отображения вьювера без использования HTML шаблона страницы:

viewer.php

```
<?php
   require_once 'vendor/autoload.php';

   use Stimulsoft\Report\StiReport;
   use Stimulsoft\Viewer\StiViewer;

   $viewer = new StiViewer();
   $viewer->process();

   $report = new StiReport();
   $report->loadFile('reports/SimpleList.mrt');
   $viewer->report = $report;

   $viewer->printHtml();
?>
```

Полный код примера доступен на GitHub.

© 2003-2026 Stimulsoft

9.2.2 Активация лицензии

После приобретения продукта вам необходимо активировать лицензию для используемых компонентов. Предусмотрено несколько способов подключения лицензионного ключа.

Все варианты активации компонентов описаны в разделе Активация лицензии генератора отчетов, имеют одинаковые функции и параметры вызова.

9.2.3 Отображение отчета

Для отображения отчета во вьювере, достаточно создать объект `StiReport`, загрузить в него шаблон отчета, и присвоить полученный объект вьюверу. Все остальные действия будут выполнены автоматически, вьювер построит отчет и отобразит первую страницу:

viewer.php

```
<?php
   require_once 'vendor/autoload.php';

   use Stimulsoft\Report\StiReport;
   use Stimulsoft\Viewer\StiViewer;

   $viewer = new StiViewer();
   $viewer->process();

   $report = new StiReport();
   $report->loadFile('reports/SimpleList.mrt');
   $viewer->report = $report;

   $viewer->printHtml();
?>
```

Полный код примера доступен на GitHub.

Вьювер умеет сам выполнять построение и отображать как шаблоны отчетов, так и документы (построенные отчеты), поэтому отдельный вызов метода построения отчета `$report->render()` не требуется. Подробное описание работы с различными форматами отчетов и документов находится в разделе Загрузка и сохранение отчета.

© 2003-2026 Stimulsoft

9.2.4 Локализация

Вьювер поддерживает полную локализацию своего интерфейса. Для локализации интерфейса на требуемый язык, достаточно установить требуемое название файла для опции `localization` у вьювера:

viewer.php

```
<?php
   use Stimulsoft\Viewer\StiViewer;

   $viewer = new StiViewer();
   $viewer->options->localization = 'de.xml';
?>
```

Полный код примера доступен на GitHub.

Все доступные локализационные XML файлы находятся в ресурсах установленного пакета продукта. При необходимости, локализационный файл можно загрузить из любого другого места, для этого необходимо для опции `localization` указать полный путь к нужному XML файлу:

viewer.php

```
<?php
   use Stimulsoft\Viewer\StiViewer;

   $viewer = new StiViewer();
   $viewer->options->localization = '/resources/loc/de.xml';
?>
```

9.2.5 Печать отчета

Во вьювере предусмотрено несколько вариантов печати отчета. Каждый имеет свои особенности, достоинства и недостатки.

Печать в PDF (Print to PDF)
Печать будет осуществляться посредством экспортирования отчета в PDF формат. К достоинствам можно отнести большую точность расположения и печати элементов отчета по сравнению с остальными вариантами печати. Из недостатков можно упомянуть обязательное наличие установленного в браузере плагина для просмотра PDF файлов (современные браузеры имеют

© 2003-2026 Stimulsoft

встроенное средство просмотра и печати PDF файлов).

Печать с просмотром (Print with Preview)
Печать отчета будет осуществлена в отдельном всплывающем окне браузера в HTML формат. Отчет можно предварительно просмотреть, а затем отправить на принтер или скопировать в другое место в виде текста или HTML-кода. К достоинствам можно отнести кроссбраузерность при печати, отсутствие необходимости установки специальных плагинов. Недостатком является относительно невысокая точность расположения элементов отчета, обусловленная особенностями реализации HTML-форматирования.

Печать без просмотра (Print without Preview)
Печать отчета будет осуществлена непосредственно на принтер без предварительного просмотра. После выбора этого пункта меню отображается системный диалог печати. Так как печать в этом режиме осуществляется в HTML формат, то качество печати аналогично качеству печати отчета с предварительным просмотром.

Информация

Печать отчета осуществляется с использованием встроенных методов используемого браузера, поэтому представление диалогового окна может отличаться в разных операционных системах и браузерах. Также, браузер не позволяет управлять настройками печати из JavaScript кода, поэтому требуемые настройки необходимо будет выполнить в самом диалоговом окне.

Настройки печати отчета
При выборе печати отчета на панели вьювера, отображается меню с выбором варианта печати. Компонент имеет возможность принудительно установить требуемый режим печати. Для этого достаточно установить свойство `printDestination` в одно из указанных ниже значений из перечисления `StiPrintDestination`:

Наименование	Описание
`StiPrintDestination::Default`	При выборе печати будет отображено меню с доступными

© 2003-2026 Stimulsoft

	вариантами печати (значение свойства по умолчанию).
`StiPrintDestination::Pdf`	Печать в PDF формат.
`StiPrintDestination::Direct`	Печать в HTML формат непосредственно на принтер, будет отображен системный диалог печати.
`StiPrintDestination::WithPreview`	Печать в HTML формат с предварительным просмотром во всплывающем окне.

Например, требуется установить режим печати только в PDF формат:

viewer.php

```
<?php
   use Stimulsoft\Viewer\StiViewer;
   use Stimulsoft\Viewer\Enums\StiPrintDestination;

   $viewer = new StiViewer();
   $viewer->options->toolbar->printDestination = StiPrintDestination::Pdf;
?>
```

Вьювер имеет возможность полностью отключить печать отчета, если она не требуется. Для этого необходимо установить значение `false` для свойства `showPrintButton`:

viewer.php

```
<?php
   use Stimulsoft\Viewer\StiViewer;

   $viewer = new StiViewer();
   $viewer->options->toolbar->showPrintButton = false;
?>
```

© 2003-2026 Stimulsoft

Событие печати отчета

При необходимости выполнения каких-либо действий перед печатью отчета, предназначено событие `onPrintReport`. В аргументах события будут переданы тип печати отчета, диапазон страниц и сам отчет, отправленный на печать. Допускается изменение диапазона страниц и значений свойств отчета.

Пример выполнения действий на стороне JavaScript клиента перед печатью отчета:

viewer.php

```
<?php
   use Stimulsoft\Viewer\StiViewer;

   $viewer = new StiViewer();
   $viewer->onPrintReport = 'printReport';
   $viewer->process();
?>

<script>
   function printReport(args) {
      if (args.printAction == 'PrintPdf'){
         args.pageRange.rangeType =
         Stimulsoft.Report.StiRangeType.CurrentPage;
         args.pageRange.currentPage = 1;
      }
   }
</script>
```

Пример выполнения действий на стороне PHP сервера перед печатью отчета:

© 2003-2026 Stimulsoft

viewer.php

```
<?php
   use Stimulsoft\Viewer\StiViewer;
   use Stimulsoft\Events\StiPrintEventArgs;
   use Stimulsoft\Report\Enums\StiRangeType;
   use Stimulsoft\Viewer\Enums\StiPrintAction;

   $viewer = new StiViewer();
   $viewer->onPrintReport = function (StiPrintEventArgs $args) {
      if ($args->printAction == StiPrintAction::PrintPdf){
         $args->pageRange->rangeType = StiRangeType::CurrentPage;
         $args->pageRange->currentPage = 1;
      }
   };
   $viewer->process();
?>
```

Подробное описание доступных значений аргументов находится в разделе События вьювера.

Печать отчета из кода

Предусмотрена возможность распечатать отчет из кода без использования функций вьювера. Подробное описание данного функционала находится в разделе генератора отчетов Печать отчета из кода.

9.2.6 Экспорт отчета

Вьювер позволяет экспортировать отображаемый отчет либо дашборд в различные форматы. Для работы экспорта не требуется дополнительных настроек вьювера. В таблице ниже перечислены все доступные форматы экспорта для отчетов и дашбордов:

Формат экспорта	Отчеты	Дашборды
Document (Snapshot)	+	+
Adobe PDF	+	+
XPS (XML Paper Specification)	+	-
Microsoft PowerPoint	+	-

© 2003-2026 Stimulsoft

HTML	+	+
HTML5	+	-
Text	+	-
Microsoft Word	+	-
Microsoft Excel	+	+
OpenDocument Writer	+	-
OpenDocument Calc	+	-
RTF (Rich Text Format)	+	-

Формат данных	Отчеты	Дашборды
CSV (Comma Separated Value)	+	+
JSON (JavaScript Object Notation)	+	+
XML (Extensible Markup Language)	+	+
DBF (dBase/FoxPro)	+	+
DIF	+	+
SYLK (Symbolic Link)	+	+

Формат изображения	Отчеты	Дашборды
PNG (Portable Network Graphics)	+	+
JPEG (Joint Photographic Experts Group)	+	+
GIF (Graphics Interchange)	+	+
TIFF (Tagged Image File Format)	+	+
SVG (Scalable Vector Graphics)	+	+
SVGZ (Compressed SVG)	+	+

© 2003-2026 Stimulsoft

PCX (Picture Exchange)	+	+
BMP (Windows Bitmap)	+	+

Событие начала экспорта

При необходимости выполнения каких-либо действий перед экспортированием отчета, предназначено событие `onBeginExportReport`. Событие будет вызвано после отображения диалогового окна экспорта с настройками. В аргументах события будут переданы тип экспорта отчета, имя файла экспорта, сам отчет, а также все настройки выбранного экспорта. Допускается изменение значений свойств отчета, настроек экспорта, а также имени файла.

Пример выполнения действий на стороне JavaScript клиента перед экспортированием отчета:

viewer.php

```
<?php
   use Stimulsoft\Viewer\StiViewer;

   $viewer = new StiViewer();
   $viewer->onBeginExportReport = 'beginExportReport';
   $viewer->process();
?>

...

<script>
   function beginExportReport(args) {
      if (args.format == Stimulsoft.Report.StiExportFormat.Pdf) {
         args.settings.creatorString = 'My Company Name';
         args.settings.embeddedFonts = false;
      }
   }
</script>
```

Пример выполнения действий на стороне PHP сервера перед экспортированием отчета:

© 2003-2026 Stimulsoft

viewer.php

```
<?php
   use Stimulsoft\Viewer\StiViewer;
   use Stimulsoft\Events\StiExportEventArgs;
   use Stimulsoft\Export\Enums\StiExportFormat;
   use Stimulsoft\Export\StiPdfExportSettings;

   $viewer = new StiViewer();
   $viewer->onBeginExportReport = function (StiReportEventArgs $args) {
      $args->fileName = "MyExportedFileName.$args->fileExtension";

      if ($args->format == StiExportFormat::Pdf) {
         /** @var StiPdfExportSettings $settings */
         $settings = $args->settings;
         $settings->creatorString = 'My Company Name';
         $settings->embeddedFonts = false;
      }
   };
   $viewer->process();
?>
```

Полный код примера доступен на GitHub.

Подробное описание доступных значений аргументов события находится в разделе События вьювера. Подробное описание доступных значений настроек экспорта находится в разделе Экспортирование отчета из кода.

Событие завершения экспорта

При необходимости выполнения каких-либо действий после экспортирования отчета перед его сохранением, предназначено событие `onEndExportReport`. В аргументах события будут переданы тип экспорта отчета, а также имя и байтовые данные экспортированного файла. Допускается изменение имени и байтовых данных экспортированного файла.

Пример выполнения действий на стороне JavaScript клиента после завершения экспортирования отчета:

© 2003-2026 Stimulsoft

viewer.php

```
<?php
   use Stimulsoft\Viewer\StiViewer;

   $viewer = new StiViewer();
   $viewer->onEndExportReport = 'endExportReport';
   $viewer->process();
?>

...

<script>
   function endExportReport(args) {
      if (args.format == Stimulsoft.Report.StiExportFormat.Html) {
         let fileName = args.fileName;
         let htmlText = args.data;
      }
   }
</script>
```

Пример выполнения действий на стороне PHP сервера после завершения экспортирования отчета:

viewer.php

```
<?php
   use Stimulsoft\Viewer\StiViewer;
   use Stimulsoft\Events\StiExportEventArgs;
   use Stimulsoft\Export\Enums\StiExportFormat;

   $viewer = new StiViewer();
   $viewer->onEndExportReport = function (StiReportEventArgs $args) {
      $fileName = $args->fileName;
      if ($args->format == StiExportFormat::Pdf) {
         $htmlText = base64_decode($args->data);
      }
   };
   $viewer->process();
?>
```

© 2003-2026 Stimulsoft

Информация

Байтовые данные передаются на сторону сервера в кодировке `Base64`, поэтому перед сохранением их необходимо преобразовать в исходный байтовый поток, используя, например, стандартную PHP функцию `base64_decode()`.

Подробное описание доступных значений аргументов находится в разделе События вьювера.

Настройки экспорта

Иногда требуется отключить неиспользуемые форматы экспорта отчета, и оставить только нужные. Это позволяет разгрузить интерфейс и упростить использование вьювера. Для отключения неиспользуемых форматов экспортов достаточно установить значение `false` для соответствующих свойств вьювера, например:

viewer.php

```
<?php
   use Stimulsoft\Viewer\StiViewer;

   $viewer = new StiViewer();
   $viewer->options->exports->showExportToDocument = false;
   $viewer->options->exports->showExportToWord = false;
   $viewer->options->exports->showExportToCsv = false;
?>
```

Также, при необходимости, можно полностью убрать отображение диалоговых окон экспорта, экспортирование всегда будет осуществляться с настройками по умолчанию. В этом случае, настройками можно управлять в событии экспорта. Для отключения диалоговых окон достаточно установить значение `false` для свойства `showExportDialog`:

© 2003-2026 Stimulsoft

viewer.php

```
<?php
   use Stimulsoft\Viewer\StiViewer;

   $viewer = new StiViewer();
   $viewer->options->exports->showExportDialog = false;
?>
```

Полный список доступных опций находится в разделе Настройка вьювера.

Экспортирование отчета из кода

Для экспортирования отчета из кода без помощи вьювера предназначен специальный метод `exportDocument()` у объекта отчета. Подробное описание находится в разделе Экспортирование отчета из кода.

9.2.7 Режимы отображения

Вьювер позволяет настраивать различные режимы отображения интерфейса и страниц отчета, а также управлять отображением на мобильных устройствах.

Полосы прокрутки

Во вьювере предусмотрено два режима отображения отчета: с полосами прокрутки и без них. По умолчанию установлен режим просмотра без полос прокрутки. Для включения режима просмотра с полосами прокрутки достаточно установить значение `true` для свойства `scrollbarsMode`.

viewer.php

```
<?php
   use Stimulsoft\Viewer\StiViewer;

   $viewer = new StiViewer();
   $viewer->options->appearance->scrollbarsMode = true;
?>
```

В первом режиме (без полос прокрутки), вьювер отображает страницу или отчет целиком, автоматически растягивая область просмотра. Если заданы размеры по ширине и высоте, то вьювер будет осуществлять обрезку

© 2003-2026 Stimulsoft

вышедшей за границы страницы. Во втором режиме, в отличие от первого, при выходе страницы за границы размеров вьювера, обрезка осуществляться не будет. Вместо этого появятся полосы прокрутки, при помощи которых можно просмотреть страницу или отчет целиком.

Информация

В режиме просмотра отчета с полосами прокрутки необходимо задать высоту вьювера, иначе будет задана высота по умолчанию, равная 650 пикселей.

Полноэкранный режим

Во вьювере предусмотрен режим полноэкранного отображения отчета и дашборда. По умолчанию включен стандартный режим просмотра, вьювер имеет заданные в настройках размеры. Для включения полноэкранного режима просмотра достаточно установить значение `true` для свойства `fullScreenMode`.

viewer.php

```
<?php
   use Stimulsoft\Viewer\StiViewer;

   $viewer = new StiViewer();
   $viewer->options->appearance->fullScreenMode = true;
?>
```

Также, для включения либо отключения полноэкранного режима можно воспользоваться соответствующей кнопкой на панели управления вьювера.

Отображение страниц отчета

Во вьювере доступно три режима отображения отчета: постраничное отображение, отчет целиком в виде ленты, и табличное отображение страниц отчета. Для управления режимами предназначено свойство `viewMode`, которое соответственно принимает одно из указанных значений:

© 2003-2026 Stimulsoft

Наименование	Описание
`StiWebViewMode::SinglePage`	Отображается одна страница, выбранная на панели инструментов.
`StiWebViewMode::Continuous`	Отображаются все страницы отчета в виде ленты.
`StiWebViewMode::MultiplePages`	Отображаются все страницы отчета в виде таблицы.

Например, требуется установить режим отображения всех страниц в виде ленты:

viewer.php

```
<?php
   use Stimulsoft\Viewer\StiViewer;
   use Stimulsoft\Viewer\Enums\StiWebViewMode;

   $viewer = new StiViewer();
   $viewer->options->toolbar->viewMode = StiWebViewMode::Continuous;
?>
```

Мобильный режим

Во вьювере реализована поддержка работы как с обычным компьютером, так и с сенсорными экранами, мобильными устройствами. Для управления режимами интерфейса предназначено свойство `interfaceType`, которое принимает одно из следующих значений:

Наименование	Описание
`StiInterfaceType::Auto`	Тип интерфейса вьювера будет выбран автоматически в зависимости от используемого устройства (значение по умолчанию).
`StiInterfaceType::Mouse`	Принудительное использование стандартного интерфейса для

© 2003-2026 Stimulsoft

	управления вьювером при помощи мыши.
`StiInterfaceType::Touch`	Принудительное использование `Touch` интерфейса для управления вьювером при помощи сенсорного экрана монитора, в этом режиме элементы интерфейса вьювера имеют увеличенные размеры для удобства управления.
`StiInterfaceType::Mobile`	Принудительное использование `Mobile` интерфейса для управления вьювером при помощи экрана смартфона, в этом режиме интерфейс вьювера имеет упрощенный вид и адаптирован для управления при помощи мобильного устройства.

Например, требуется полностью отключить мобильный режим отображения:

viewer.php

```
<?php
   use Stimulsoft\Viewer\StiViewer;
   use Stimulsoft\Viewer\Enums\StiInterfaceType;

   $viewer = new StiViewer();
   $viewer->options->appearance->interfaceType = StiInterfaceType::Mouse;
?>
```

9.2.8 Работа с переменными отчета

Для работы с переменными отчета во вьювере реализована поддержка специальной панели параметров. Для добавления параметра на панель необходимо в отчете определить переменную, запрашиваемую у пользователя. При просмотре отчета во вьювере такая переменная будет автоматически добавлена на панель параметров. Поддерживаются все типы переменных отчета (обычные переменные, дата и время, диапазон, списки и др.).

© 2003-2026 Stimulsoft

Управление переменными на панели параметров

Для выполнения каких-либо действий перед применением параметров предназначено специальное событие `onInteraction`, которое будет вызвано при интерактивных действиях вьювера. В аргументах события будут переданы тип действия, а также коллекция переменных и их значений, расположенных на панели параметров. Тип действия в данном случае будет иметь строковое значение `"Variables"`.

Пример выполнения действий на стороне JavaScript клиента перед применением параметров отчета:

viewer.php

```
<?php
   use Stimulsoft\Viewer\StiViewer;

   $viewer = new StiViewer();
   $viewer->onInteraction = 'interaction';
   $viewer->process();
?>

...

<script>
   function interaction(args) {
      if (args.action == "Variables") {
         let variables = args.variables;
      }
   }
</script>
```

Коллекция переменных представляет собой объект, содержащий все переменные панели параметров, и их значения, например:

viewer.php

```
var variables = {
   VariableString: "Text value",
   VariableInt: 20
}
```

Допустимо изменять значения переменных, при этом тип нового значения

© 2003-2026 Stimulsoft

должен соответствовать типу изменяемой переменной. Подробное описание доступных значений аргументов находится в разделе События вьювера.

Настройка панели параметров

Если работа с переменными во вьювере не требуется, то можно полностью отключить данную возможность. Для этого предназначено свойство `showParametersButton`, которому необходимо установить значение `false`.

viewer.php

```
<?php
   use Stimulsoft\Viewer\StiViewer;

   $viewer = new StiViewer();
   $viewer->options->toolbar->showParametersButton = false;
?>
```

Информация

При такой конфигурации вьювера панель параметров не будет показана даже в том случае, если параметры присутствуют в отображаемом отчете.

Значения переменных при построении отчета

При необходимости контролирования всех переменных отчета, предназначено специальное событие `onPrepareVariables`, которое будет вызвано перед построением отчета. Подробное описание находится в разделе Работа с переменными отчета в разделе Генератор отчетов.

9.2.9 Работа с закладками

Во вьювере реализована поддержка закладок отчета. При отображении такого отчета слева от страницы будет отображена панель с закладками. При выборе закладки отчета вьювер осуществит автоматический переход на нужную страницу, а элемент отчета с закладкой будет подсвечен.

Настройка закладок

По умолчанию ширина панели закладок равна 180 пикселей, вьювер позволяет

© 2003-2026 Stimulsoft

изменить это значение. Для этого предназначено свойство `bookmarksTreeWidth`, значение которого указывается в пикселях.

viewer.php

```
<?php
   use Stimulsoft\Viewer\StiViewer;

   $viewer = new StiViewer();
   $viewer->options->appearance->bookmarksTreeWidth = 300;
?>
```

Если работа с закладками отчета не требуется, то можно полностью отключить данную возможность. Для этого предназначено свойство `showBookmarksButton`, которое необходимо установить в значение `false`.

viewer.php

```
<?php
   use Stimulsoft\Viewer\StiViewer;

   $viewer = new StiViewer();
   $viewer->options->toolbar->showBookmarksButton = false;
?>
```

Информация

В этом случае закладки отчета не будут показаны, даже если они присутствуют в отображаемом отчете. Данное свойство не оказывает влияние на печать и экспортирование отчета.

При печати отчета с закладками, дерево закладок будет скрыто. Если помимо самого отчета требуется распечатать и закладки, то необходимо установить свойство `bookmarksPrint` в значение `true`.

© 2003-2026 Stimulsoft

viewer.php

```
<?php
   use Stimulsoft\Viewer\StiViewer;

   $viewer = new StiViewer();
   $viewer->options->appearance->bookmarksPrint = true;
?>
```

9.2.10 Динамическое сворачивание, сортировка и детализация

Помимо переменных, значения которых можно задавать на панели параметров, вьювер поддерживает другие виды интерактивности, добавляющей удобства и функциональности при использовании генератора отчетов. Это сортировка, сворачивание и детализация.

Сортировка

Динамическая сортировка предоставляет возможность изменять направление сортировки в построенном отчете. Для этого следует щелкнуть по компоненту, у которого была установлена динамическая сортировка. Динамическая сортировка осуществляется в следующих направлениях: **По возрастанию (Ascending)** и **По убыванию (Descending)**. Каждый раз при щелчке по компоненту направление меняется на противоположное.

Допускается многоуровневая сортировка в отчете. Для этого необходимо удерживать клавишу **Ctrl** и последовательно нажимать на сортируемые компоненты отчета. Для сброса сортировки можно нажать на любой сортируемый компонент без удержания клавиши **Ctrl**.

Сворачивание

Отчет с динамическим сворачиванием представляет собой интерактивный отчет, в котором определенные блоки могут сворачивать/разворачивать свое содержимое при нажатии на заголовок блока. Элементы отчета, которые можно свернуть/развернуть, обозначены специальной иконкой со знаком **[-]** или **[+]**.

Детализация

При детализации данных под главной панелью вьювера будет отображена панель детализации с закладками детализированных отчетов. Отображенный в данный момент отчет будет подсвечен. Предусмотрена возможность закрывать не требуемые в данный момент детализированные страницы.

© 2003-2026 Stimulsoft

Сортировка дашборда
При просмотре дашборда, во многих элемента можно изменять сортировку по полям данных элемента **По возрастанию (Ascending)** и **По убыванию (Descending)**.

Фильтрация дашборда
Фильтрация на дашборде возможна как при помощи специальных элементов фильтрации, так и при помощи других элементов. Фильтрация будет применяться для всех взаимосвязанных элементов на дашборде.

Детализация дашборда
В дашбордах, как и в отчетах, возможно открытие детализирующего дашборда, а также отчета. Кроме этого, для некоторых элементов дашборда возможна детализация данных на уровне элемента.

Событие интерактивности вьювера
Для работы динамической сортировки, сворачивания и детализации отчетов не требуется дополнительных настроек вьювера. Для выполнения каких-либо действий перед сортировкой, сворачиванием или детализацией отчета предназначено специальное событие `onInteraction`, которое будет вызвано при интерактивных действиях вьювера. Подробное описание доступных значений аргументов находится в разделе События вьювера.

Наименование	Описание
`InitVars`	Действие происходит при инициализации переменных отчета, запрашиваемых у пользователя.
`Variables`	Действие происходит при использовании переменных на панели параметров. Подробное описание находится в разделе Работа с переменными отчета.
`Sorting`	Действие происходит при использовании сортировки колонок.

© 2003-2026 Stimulsoft

`DrillDown`	Действие происходит при использовании детализации колонок.
`Collapsing`	Действие происходит при использовании сворачивания блоков отчета.
`DashboardFiltering`	Действие происходит при использовании фильтров внутри элемента дашборда.
`DashboardSorting`	Действие происходит при использовании сортировки внутри элемента дашборда.
`DashboardResetAllFilters`	Действие происходит при сбросе сортировки и фильтров внутри элемента дашборда на значения, заданные в шаблоне.
`DashboardElementDrillDown`	Действие происходит при использовании детализации элемента дашборда.
`DashboardElementDrillUp`	Действие происходит при использовании детализации элемента дашборда.

В аргументах передаются соответствующие коллекции параметров `sortingParameters`, `collapsingParameters` и `drillDownParameters`, содержащие данные в специальном формате, необходимые для текущего интерактивного действия. При необходимости, значения коллекции параметров допускается корректировать, сохраняя структуру и порядок передаваемых значений. Пример передаваемых значений параметров для некоторых интерактивных действий:

© 2003-2026 Stimulsoft

viewer.php

```
<?php
   use Stimulsoft\Viewer\StiViewer;

   $viewer = new StiViewer();
   $viewer->onInteraction = 'interaction';
   $viewer->process();
?>

...

<script>
   function interaction(args) {
      switch (args.action) {
         case "Sorting":
            args.sortingParameters = {
               ComponentName: "Text10;false",
               DataBand: "DataBand1;DESC;CompanyName"
            };
         break;

         case "DrillDown":
            drillDownParameters = [{
               ComponentIndex: "1"
               DrillDownMode: null
               ElementIndex: "6"
               PageGuid: "b916d048d3f446dc97c356d4ff47f48f"
               PageIndex: "0"
               ReportFile: null
            }];
         break;

         case "Collapsing":
            args.collapsingParameters = {
               CollapsingStates: {
                  GroupHeaderBand1: {
                     keys: [1],
                     values: [false]
                  },
               ComponentName: "GroupHeaderBand1"
            };
         break;
      }
   }
</script>
```

Подробное описание доступных значений аргументов находится в разделе [События вьювера](#).

© 2003-2026 Stimulsoft

9.2.11 Редактирование построенного отчета

Вьювер имеет возможность редактировать элементы построенного отчета, такие как текстовые поля и чекбоксы. Для того, чтобы редактирование стало возможным, в самом шаблоне отчета необходимо пометить нужные компоненты как редактируемые. Никаких дополнительных настроек вьювера не требуется. После отображения отчета во вьювере, для начала редактирования необходимо нажать на соответствующую кнопку на панели вьювера. После завершения редактирования, необходимо нажать указанную кнопку еще раз, и все внесенные изменения будут применены к отчету.

Информация

Отредактированные значения будут применены при печати либо экспортировании отчета, при этом исходный отчет останется нетронутым. После перезагрузки вьювера все значения будут возвращены к исходным.

9.2.12 Отправка отчета по Email

Во вьювере предусмотрена возможность отправки отчета по Email. Для активации данной возможности необходимо установить свойство вьювера `showSendEmailButton` в значение `true` и добавить обработчик события `onEmailReport`.

Чистый JavaScript не имеет функций работы с Email, для этой возможности используются функции на стороне PHP сервера. Для отправки Email, в аргументах события на стороне PHP сервера необходимо установить параметры, такие как логин и пароль аккаунта, с которого будет производится отправка, а также настройки сервера - его адрес, порт и другие параметры. Для этого предназначено свойство `$args->settings` в аргументах события. Это свойство представляет собой объект класса `StiEmailSettings`, содержащий все необходимые параметры отправки.

Пример отправки отчета по Email с минимально необходимыми параметрами:

© 2003-2026 Stimulsoft

viewer.php

```
<?php
   use Stimulsoft\Viewer\StiViewer;
   use Stimulsoft\Events\StiEmailEventArgs;
   use Stimulsoft\Report\StiReport;
   use Stimulsoft\StiResult;

   $viewer = new StiViewer();
   $viewer->options->toolbar->showSendEmailButton = true;

   $viewer->onEmailReport = function (StiEmailEventArgs $args) {
      $args->settings->from = 'mail.sender@stimulsoft.com';
      $args->settings->host = 'smtp.stimulsoft.com';
      $args->settings->login = '********';
      $args->settings->password = '********';

      return StiResult::getSuccess('The Email has been sent
      successfully.');
   };

   $viewer->process();
?>
```

Полный код примера доступен на GitHub.

Дополнительно, в аргументах события можно получить и изменить данные для отправки письма (тема письма, текст и имя файла отчета), получить сам отчет, а также получить настройки экспорта отчета и при необходимости изменить их. Подробное описание доступных значений аргументов находится в разделе События вьювера.

При необходимости, можно добавить вызов JavaScript события, в аргументах которого можно узнать необходимые данные для отправки письма, получить тип экспорта отчета, получить сам отчет, а также получить настройки экспорта отчета и при необходимости изменить их.

Пример изменения темы письма перед его отправкой:

© 2003-2026 Stimulsoft

viewer.php

```
<?php
   use Stimulsoft\Viewer\StiViewer;

   $viewer = new StiViewer();
   $viewer->options->toolbar->showSendEmailButton = true;
   $viewer->onEmailReport = 'emailReport';
   $viewer->process();
?>

...

<script>
   function emailReport(args) {
      args.settings.subject = "Invoice: " + args.settings.subject;
   }
</script>
```

При выполнении отправки Email в первую очередь будет вызвано JavaScript событие (если оно задано), затем будет вызвано событие на стороне PHP сервера. Таким образом, можно провести валидацию и корректировку значений непосредственно перед отправкой письма.

Пример проверки и изменения темы письма на стороне PHP сервера перед отправкой:

viewer.php

```
$viewer->onEmailReport = function (StiEmailEventArgs $args) {
   if (strlen($args->settings->subject ?? '') == 0)
      $args->settings->subject = "{$args->formatName} report {$args-
      >settings->attachmentName}";

   return StiResult::getSuccess('The Email has been sent successfully.');
};
```

Пример добавления адресов дополнительных получателей письма с отчетом:

© 2003-2026 Stimulsoft

viewer.php

```
$viewer->onEmailReport = function (StiEmailEventArgs $args) {
   $args->settings->cc[] = 'extra_recipient_one@stimulsoft.com';
   $args->settings->bcc[] = 'hidden_recipient_one@stimulsoft.com';
   $args->settings->bcc[] = 'hidden_recipient_two@stimulsoft.com John
   Smith';

   return StiResult::getSuccess('The Email has been sent successfully.');
};
```

Настройки отправки Email

Вьювер позволяет установить значения по умолчанию для диалога отправки Email во вьювере. Для этого предназначены свойства `defaultEmailAddress`, `defaultEmailSubject` и `defaultEmailMessage`. По умолчанию данные свойства являются пустыми.

viewer.php

```
<?php
   use Stimulsoft\Viewer\StiViewer;

   $viewer = new StiViewer();
   $viewer->options->toolbar->showSendEmailButton = true;
   $viewer->options->email->defaultEmailAddress =
   'recipient_address@stimulsoft.com';
   $viewer->options->email->defaultEmailSubject = 'New Invoice';
   $viewer->options->email->defaultEmailMessage = 'Please check the new
   invoice in the attachment';
?>
```

9.2.13 Вызов дизайнера

Вьювер имеет возможность вызвать дизайнер отчетов. Для этого предназначена специальная кнопка **Дизайн (Design)** на панели инструментов вьювера. По умолчанию данная кнопка отключена. Для использования этой возможности необходимо установить свойство `showDesignButton` в значение `true`, а также определить событие `onDesignReport`:

© 2003-2026 Stimulsoft

viewer.php

```
<?php
   use Stimulsoft\Viewer\StiViewer;

   $viewer = new StiViewer();
   $viewer->options->toolbar->showDesignButton = true;
   $viewer->onDesignReport = 'designReport';
   $viewer->process();
?>

<script>
   function designReport(args) {
      window.open("designer.php?fileName=" + args.fileName);
   }
</script>
```

Полный код примера доступен на GitHub.

Подробное описание доступных значений аргументов находится в разделе События вьювера.

Информация

Сам вьювер не запускает дизайнер, он лишь вызывает указанное событие и передает в качестве аргументов имя файла и просматриваемый отчет. В событии можно выполнить перенаправление на PHP страницу, на которой расположен дизайнер отчетов.

9.2.14 Визуальное оформление

Вьювер имеет возможность смены тем оформления визуальных элементов управления. Для этого достаточно установить свойство `theme` в опциях компонента:

© 2003-2026 Stimulsoft

viewer.php

```
<?php
   use Stimulsoft\Viewer\StiViewer;
   use Stimulsoft\Viewer\Enums\StiViewerTheme;

   $viewer = new StiViewer();
   $viewer->options->appearance->theme =
   StiViewerTheme::Office2022BlackGreen;
?>
```

Полный код примера доступен на GitHub.

На данный момент доступно **8 тем** оформления с различными цветовыми акцентами. В результате, доступно более **60** вариантов оформления. Это позволяет настроить внешний вид вьювера практически под любое оформление Web проекта.

Дополнительные настройки

По умолчанию, вьювер имеет только верхнюю панель инструментов, на которой находятся все элементы управления отчетом. При необходимости, панель инструментов можно разделить на верхнюю и нижнюю. На верхней панели будут находиться меню печати и экспорта отчета, а также кнопки работы с параметрами и закладками. Нижняя панель инструментов будет содержать элементы переключения между страницами отчета и меню управления масштабом. Для включения указанного режима предназначено свойство `displayMode`, которое может иметь следующие значения:

Наименование	Описание
`StiToolbarDisplayMode::Simple`	Простой режим отображения, все элементы управления расположены на одной панели управления (значение по умолчанию).
`StiToolbarDisplayMode::Separated`	Раздельный режим отображения, панель управления разделена на верхнюю - взаимодействие с отчетом, и нижнюю - взаимодействие со страницами.

© 2003-2026 Stimulsoft

viewer.php

```
<?php
   use Stimulsoft\Viewer\StiViewer;
   use Stimulsoft\Viewer\Enums\StiToolbarDisplayMode;

   $viewer = new StiViewer();
   $viewer->options->toolbar->displayMode = StiToolbarDisplayMode::Simple;
   $viewer->options->appearance->scrollbarsMode = true;
?>
```

Дополнительно предусмотрена возможность установки параметров оформления основных элементов вьювера. Например, можно изменить шрифт и цвет надписей панели управления вьювера, установить фон вьювера, задать цвет границ страницы и др. Ниже представлен список доступных свойств, изменяющих оформление вьювера, и их значения по умолчанию:

viewer.php

```
<?php
   use Stimulsoft\Viewer\StiViewer;

   $viewer = new StiViewer();

   $viewer->options->appearance->backgroundColor = 'white';
   $viewer->options->appearance->pageBorderColor = 'red';
   $viewer->options->appearance->showPageShadow = false;

   $viewer->options->toolbar->backgroundColor = 'aqua';
   $viewer->options->toolbar->borderColor = 'darkgreen';
   $viewer->options->toolbar->fontColor = 'white';
   $viewer->options->toolbar->fontFamily = 'Arial';
?>
```

В качестве значения для цвета указывается одна из стандартных констант для HTML цветов, либо код цвета в RGB формате, например '`#ff2020`'.

9.2.15 События вьювера

Вьювер отчетов поддерживает события, которые предоставляют возможность выполнить необходимые операции перед определенными действиями - как на стороне JavaScript клиента, так и на стороне PHP сервера. Подробное описание

© 2003-2026 Stimulsoft

работы событий находится в разделе События генератора отчетов.

Вьювер поддерживает следующие события:

- onDatabaseConnect
- onBeginProcessData
- onEndProcessData
- onPrepareVariables
- onOpenReport
- onOpenedReport
- onPrintReport
- onBeginExportReport
- onEndExportReport
- onInteraction
- onEmailReport
- onDesignReport

onDatabaseConnect
[-] JavaScript [+] PHP

Событие вызывается перед подключением к базе данных после получения всех параметров. Подробное описание и примеры использования находятся в разделе Подключение SQL адаптеров данных. Список аргументов события находится в разделе События генератора отчетов.

onBeginProcessData
[+] JavaScript [+] PHP

Событие вызывается перед запросом данных, необходимых для построения отчета. Подробное описание и примеры использования находятся в разделах Подключение файловых данных и Подключение SQL адаптеров данных. Список аргументов события находится в разделе События генератора отчетов..

© 2003-2026 Stimulsoft

onEndProcessData
[+] JavaScript [+] PHP

Событие вызывается после загрузки данных перед построением отчета. Подробное описание и примеры использования находятся в разделах Подключение файловых данных и Подключение SQL адаптеров данных. Список аргументов события находится в разделе События генератора отчетов..

onPrepareVariables
[+] JavaScript [+] PHP

Событие вызывается перед построением отчета после подготовки переменных отчета. Подробное описание и примеры использования находятся в разделе Работа с переменными отчета. Список аргументов события находится в разделе События генератора отчетов..

onOpenReport
[+] JavaScript [-] PHP

Событие вызывается перед открытием отчета после нажатия кнопки на панели инструментов. В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне JavaScript клиента:

Наименование	Описание
`event`	Идентификатор текущего события, имеет значение "`OpenReport`".
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • "`Viewer`" • "`Designer`"
`report`	Текущий объект отчета, в аргументах данного события будет передано значение `null`.

© 2003-2026 Stimulsoft

preventDefault	Данный флаг предоставляет возможность остановить дальнейшую обработку события вьювером. По умолчанию установлено значение false.

onOpenedReport
[+] JavaScript [+] PHP

Событие вызывается после открытия файла отчета до его передачи вьюверу. В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне JavaScript клиента:

Наименование	Описание
event	Идентификатор текущего события, имеет значение "OpenedReport".
sender	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • "Viewer" • "Designer"
report	Текущий объект отчета.
preventDefault	Данный флаг предоставляет возможность остановить дальнейшую обработку события вьювером. По умолчанию установлено значение false.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне PHP сервера, аргументы имеют тип `StiReportEventArgs`:

Наименование	Описание
event	Идентификатор текущего события,

© 2003-2026 Stimulsoft

	для данного события имеет значение `StiEventType::OpenedReport`
`sender`	Компонент, который инициировал данное событие, может иметь следующие типы: • `StiViewer` • `StiDesigner`
`report`	Текущий объект отчета.

onPrintReport
[+] JavaScript [+] PHP

Событие вызывается перед печатью отчета. Подробное описание и примеры использования находятся в разделе Печать отчета.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне JavaScript клиента:

Наименование	Описание
`event`	Идентификатор текущего события, имеет значение "`PrintReport`".
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • "`Viewer`" • "`Designer`"
`report`	Текущий объект отчета.
`printAction`	Тип печати отчета. Может принимать следующие значения: • "`PrintPdf`" - печать в PDF формат; • "`PrintWithoutPreview`" -

© 2003-2026 Stimulsoft

	печать в HTML формат непосредственно на принтер, будет отображен системный диалог печати; • "PrintWithPreview" - печать в HTML формат с предварительным просмотром во всплывающем окне.
pageRange	Объект, содержащий настройки диапазона страниц для печати.
preventDefault	Данный флаг предоставляет возможность остановить дальнейшую обработку события вьювером. По умолчанию установлено значение false.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне PHP сервера, аргументы имеют тип StiPrintEventArgs:

Наименование	Описание
event	Идентификатор текущего события, для данного события имеет значение StiEventType::PrintReport
sender	Компонент, который инициировал данное событие, может иметь следующие типы: • StiViewer • StiDesigner
report	Текущий объект отчета.
printAction	Тип печати отчета. Может принимать следующие значения: • StiPrintAction::PrintPdf - печать в PDF формат;

© 2003-2026 Stimulsoft

	• `StiPrintAction::PrintWithoutPreview` - печать в HTML формат непосредственно на принтер, будет отображен системный диалог печати; • `StiPrintAction::PrintWithPreview` - печать в HTML формат с предварительным просмотром во всплывающем окне.
`fileName`	Имя файла отчета для сохранения.
`pageRange`	Объект, содержащий настройки диапазона страниц для печати.

onBeginExportReport
[+] JavaScript [+] PHP

Событие вызывается перед экспортированием отчета после диалога настроек экспорта. Подробное описание и примеры использования находятся в разделе Экспорт отчета.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне JavaScript клиента:

Наименование	Описание
`event`	Идентификатор текущего события, имеет значение `"BeginExportReport"`.
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • `"Viewer"` • `"Designer"`
`report`	Текущий объект отчета.

© 2003-2026 Stimulsoft

action	Действие, которое вызвало событие экспорта, может принимать следующие значения перечисления Stimulsoft.Viewer.StiExportAction: • StiExportAction.ExportReport • StiExportAction.SendEmail
format	Выбранный формат экспорта отчета. Может принимать следующие значения перечисления Stimulsoft.Report.StiExportFormat: • StiExportFormat.Document • StiExportFormat.Pdf • StiExportFormat.Xps • StiExportFormat.PowerPoint • StiExportFormat.Html • StiExportFormat.Html5 • StiExportFormat.Text • StiExportFormat.Word • StiExportFormat.Excel • StiExportFormat.Odt • StiExportFormat.Ods • StiExportFormat.Rtf • StiExportFormat.Csv • StiExportFormat.Json • StiExportFormat.Xml • StiExportFormat.Dbf

© 2003-2026 Stimulsoft

	• StiExportFormat.Sylk • StiExportFormat.ImagePng • StiExportFormat.ImageJpeg • StiExportFormat.ImageGif • StiExportFormat.ImageTiff • StiExportFormat.ImageSvg • StiExportFormat.ImageSvgz • StiExportFormat.ImagePcx • StiExportFormat.ImageBmp
formatName	Название выбранного формата экспорта отчета, соответствует названию констант в перечислении форматов.
settings	Настройки выбранного формата экспорта. Тип объекта настроек и список доступных свойств будет зависеть от выбранного типа экспорта.
fileName	Имя файла отчета для сохранения после завершения экспорта.
openAfterExport	Данный флаг предоставляет возможность включить автоматическое открытие экспортированного отчета в новой вкладке браузера вместо его сохранения в файл. Работает только для поддерживаемых браузером форматов. По

© 2003-2026 Stimulsoft

	умолчанию установлено значение `false`.
`preventDefault`	Данный флаг предоставляет возможность остановить дальнейшую обработку события вьювером. По умолчанию установлено значение `false`.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне PHP сервера, аргументы имеют тип `StiExportEventArgs`:

Наименование	Описание
`event`	Идентификатор текущего события, для данного события имеет значение `StiEventType::BeginExportReport`.
`sender`	Компонент, который инициировал данное событие, может иметь следующие типы: • `StiViewer` • `StiDesigner`
`action`	Действие, которое вызвало событие экспорта, может принимать следующие значения: • `StiExportAction::ExportReport` • `StiExportAction::SendEmail`
`format`	Выбранный формат экспорта отчета. Может принимать следующие значения: • `StiExportFormat::Document` • `StiExportFormat::Pdf`

© 2003-2026 Stimulsoft

	• StiExportFormat::Xps • StiExportFormat::PowerPoint • StiExportFormat::Html • StiExportFormat::Html5 • StiExportFormat::Text • StiExportFormat::Word • StiExportFormat::Excel2007 • StiExportFormat::Odt • StiExportFormat::Ods • StiExportFormat::Rtf • StiExportFormat::Csv • StiExportFormat::Json • StiExportFormat::Xml • StiExportFormat::Dbf • StiExportFormat::Sylk • StiExportFormat::ImagePng • StiExportFormat::ImageJpeg • StiExportFormat::ImageGif • StiExportFormat::ImageTiff • StiExportFormat::ImageSvg • StiExportFormat::ImageSvgz • StiExportFormat::ImagePcx • StiExportFormat::ImageBmp

© 2003-2026 Stimulsoft

`formatName`	Название выбранного формата экспорта отчета, соответствует названию констант в перечислении форматов.
`fileName`	Имя файла отчета для сохранения после завершения экспорта.
`settings`	Настройки выбранного формата экспорта. Тип объекта настроек и список доступных свойств будет зависеть от выбранного типа экспорта.
`openAfterExport`	Данный флаг предоставляет возможность включить автоматическое открытие экспортированного отчета в новой вкладке браузера вместо его сохранения в файл. Работает только для поддерживаемых браузером форматов. По умолчанию установлено значение `false`

onEndExportReport
[+] JavaScript [+] PHP

Событие вызывается после экспортирования отчета до его сохранения в виде файла. Подробное описание и примеры использования находятся в разделе Экспорт отчета.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне JavaScript клиента:

Наименование	Описание
`event`	Идентификатор текущего события, имеет значение `"EndExportReport"`.

© 2003-2026 Stimulsoft

sender	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • "Viewer" • "Designer"
report	Текущий объект отчета.
format	Выбранный формат экспорта отчета. Может принимать следующие значения перечисления Stimulsoft.Report.StiExportFormat: • StiExportFormat.Document • StiExportFormat.Pdf • StiExportFormat.Xps • StiExportFormat.PowerPoint • StiExportFormat.Html • StiExportFormat.Html5 • StiExportFormat.Text • StiExportFormat.Word • StiExportFormat.Excel • StiExportFormat.Odt • StiExportFormat.Ods • StiExportFormat.Rtf • StiExportFormat.Csv • StiExportFormat.Json • StiExportFormat.Xml • StiExportFormat.Dbf • StiExportFormat.Sylk

© 2003-2026 Stimulsoft

	• StiExportFormat.ImagePng • StiExportFormat.ImageJpeg • StiExportFormat.ImageGif • StiExportFormat.ImageTiff • StiExportFormat.ImageSvg • StiExportFormat.ImageSvgz • StiExportFormat.ImagePcx • ·StiExportFormat.ImageBmp
formatName	Название выбранного формата экспорта отчета, соответствует названию констант в перечислении форматов.
data	Байтовые данные экспортированного отчета, подготовленные для сохранения в файл.
fileName	Имя файла отчета для сохранения после завершения экспорта.
openAfterExport	Данный флаг предоставляет возможность включить автоматическое открытие экспортированного отчета в новой вкладке браузера вместо его сохранения в файл. Работает только для поддерживаемых браузером форматов. По умолчанию установлено значение false.

© 2003-2026 Stimulsoft

preventDefault	Данный флаг предоставляет возможность остановить дальнейшую обработку события вьювером. По умолчанию установлено значение false.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне PHP сервера, аргументы имеют тип StiExportEventArgs:

Наименование	Описание
	Идентификатор текущего события, для данного события имеет значение StiEventType::EndExportReport.
sender	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • StiViewer • StiDesigner
format	Выбранный формат экспорта отчета. Может принимать следующие значения: • StiExportFormat::Document • StiExportFormat::Pdf • StiExportFormat::Xps • StiExportFormat::PowerPoint • StiExportFormat::Html • StiExportFormat::Html5 • StiExportFormat::Text • StiExportFormat::Word

© 2003-2026 Stimulsoft

	• StiExportFormat::Excel2007 • StiExportFormat::Odt • StiExportFormat::Ods • StiExportFormat::Rtf • StiExportFormat::Csv • StiExportFormat::Json • StiExportFormat::Xml • StiExportFormat::Dbf • StiExportFormat::Sylk • StiExportFormat::ImagePng • StiExportFormat::ImageJpeg • StiExportFormat::ImageGif • StiExportFormat::ImageTiff • StiExportFormat::ImageSvg • StiExportFormat::ImageSvgz • StiExportFormat::ImagePcx • StiExportFormat::ImageBmp
formatName	Название выбранного формата экспорта отчета, соответствует названию констант в перечислении форматов.
data	Байтовые данные экспортированного отчета, подготовленные для сохранения в файл. Данные находятся в формате

© 2003-2026 Stimulsoft

	Base64.
fileName	Имя файла отчета для сохранения после завершения экспорта.
fileExtension	Расширение файла отчета для сохранения после завершения экспорта, соответствует типу выбранного формата.
mimeType	MIME тип для выбранного формата экспортирования.
openAfterExport	Данный флаг предоставляет возможность включить автоматическое открытие экспортированного отчета в новой вкладке браузера вместо его сохранения в файл. Работает только для поддерживаемых браузером форматов. По умолчанию установлено значение false.

onInteraction
[+] JavaScript [-] PHP

Событие вызывается в момент интерактивного действия вьювера (динамическая сортировка, сворачивание, детализация, применение параметров) до обработки значений генератором отчетов. Подробное описание и примеры использования находятся в разделе Динамическая сортировка, сворачивание и детализация.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне JavaScript клиента:

Наименование	Описание
event	Идентификатор текущего события, имеет значение "Interaction".

© 2003-2026 Stimulsoft

`sender`	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • `"Viewer"` • `"Designer"`
`report`	Текущий объект отчета.
`action`	Идентификатор текущего интерактивного действия, может принимать следующие значения: • `"InitVars"` - действие происходит при инициализации переменных отчета, запрашиваемых у пользователя; • `"Variables"` - действие происходит при применении значений переменных, запрашиваемых у пользователя; • `"Sorting"` - действие происходит при использовании сортировки колонок; • `"DrillDown"` - действие происходит при детализации отчета; • `"Collapsing"` - действие происходит при использовании сворачивания блоков отчета; • `"DashboardFiltering"` - действие происходит при использовании фильтров внутри элемента дашборда; • `"DashboardSorting"` - действие происходит при

© 2003-2026 Stimulsoft

	использовании сортировки внутри элемента дашборда; • "DashboardResetAllFilters" - действие происходит при сбросе сортировки и фильтров внутри элемента дашборда на значения, заданные в шаблоне; • "DashboardElementDrillDown" - действие происходит при использовании детализации элемента дашборда; • "DashboardElementDrillUp" - действие происходит при использовании детализации элемента дашборда.
variables	Коллекция переменных отчета и их значений, заданных на панели параметров.
sortingParameters	Коллекция параметров, необходимых для динамической сортировки отчета.
collapsingParameters	Коллекция параметров, необходимых для динамического сворачивания элементов отчета.
drillDownParameters	Коллекция параметров, необходимых для детализации отчета.
filteringParameters	Коллекция параметров, необходимых для сортировки, фильтрации и детализации элементов дашборда.
preventDefault	Данный флаг предоставляет возможность остановить дальнейшую обработку события. По умолчанию установлено

© 2003-2026 Stimulsoft

	значение `false`.

onEmailReport
[+] JavaScript [+] PHP

Событие вызывается после экспортирования отчета перед его отправкой по Email. Подробное описание и примеры использования находятся в разделе Отправка отчета по Email.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне JavaScript клиента:

Наименование	Описание
`event`	Идентификатор текущего события, имеет значение "`EmailReport`".
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • "`Viewer`" • "`Designer`"
`report`	Текущий объект отчета.
`format`	Выбранный формат экспорта отчета. Может принимать следующие значения перечисления `Stimulsoft.Report.StiExportFormat`: • `StiExportFormat.Document` • `StiExportFormat.Pdf` • `StiExportFormat.Xps` • `StiExportFormat.PowerPoint`

© 2003-2026 Stimulsoft

	• StiExportFormat.Html • StiExportFormat.Html5 • StiExportFormat.Text • StiExportFormat.Word • StiExportFormat.Excel • StiExportFormat.Odt • StiExportFormat.Ods • StiExportFormat.Rtf • StiExportFormat.Csv • StiExportFormat.Json • StiExportFormat.Xml • StiExportFormat.Dbf • StiExportFormat.Sylk • StiExportFormat.ImagePng • StiExportFormat.ImageJpeg • StiExportFormat.ImageGif • StiExportFormat.ImageTiff • StiExportFormat.ImageSvg • StiExportFormat.ImageSvgz • StiExportFormat.ImagePcx • ·StiExportFormat.ImageBmp
formatName	Название выбранного формата экспорта отчета, соответствует названию констант в перечислении форматов.

© 2003-2026 Stimulsoft

`data`	Байтовые данные экспортированного отчета, подготовленные для отправки по Email.
`fileName`	Имя файла отчета для отправки по Email.
`settings`	Объект, содержащий параметры, заполненные в диалоге вьювера отправки отчета по Email. Описание всех параметров находится в отдельной таблице ниже.

В таблице ниже представлен список параметров отправки Email на стороне JavaScript клиента:

Наименование	Описание
`email`	Email адрес, на который будет отправлен экспортированный отчет.
`subject`	Тема письма.
`message`	Текст письма.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне PHP сервера, аргументы имеют тип `StiEmailEventArgs`:

Наименование	Описание
	Идентификатор текущего события, для данного события имеет значение `StiEventType::EmailReport`.
`sender`	Идентификатор компонента, который инициировал данное

© 2003-2026 Stimulsoft

	событие, может принимать следующие значения: • StiViewer • StiDesigner
format	Выбранный формат экспорта отчета. Может принимать следующие значения: • StiExportFormat::Document • StiExportFormat::Pdf • StiExportFormat::Xps • StiExportFormat::PowerPoint • StiExportFormat::Html • StiExportFormat::Html5 • StiExportFormat::Text • StiExportFormat::Word • StiExportFormat::Excel2007 • StiExportFormat::Odt • StiExportFormat::Ods • StiExportFormat::Rtf • StiExportFormat::Csv • StiExportFormat::Json • StiExportFormat::Xml • StiExportFormat::Dbf • StiExportFormat::Sylk • StiExportFormat::ImagePng • StiExportFormat::ImageJpeg • StiExportFormat::ImageG

© 2003-2026 Stimulsoft

	if • StiExportFormat::ImageTiff • StiExportFormat::ImageSvg • StiExportFormat::ImageSvgz • StiExportFormat::ImagePcx • StiExportFormat::ImageBmp
formatName	Название выбранного формата экспорта отчета, соответствует названию констант в перечислении форматов.
data	Байтовые данные экспортированного отчета, подготовленные для отправки по Email. Данные находятся в формате Base64.
fileName	Имя файла отчета для отправки по Email.
settings	Объект, содержащий параметры отправки Email на стороне сервера. Описание всех параметров находится в отдельной таблице ниже.

В таблице ниже представлен список параметров отправки Email на стороне PHP сервера, объект параметров имеет тип StiEmailSettings:

Наименование	Описание
from	Email адрес отправителя письма.
name	Имя и фамилия отправителя

© 2003-2026 Stimulsoft

	письма.
to	Email адрес, на который будет отправлен экспортированный отчет, передается из диалогового окна вьювера.
subject	Тема письма, передается из диалогового окна вьювера.
message	Текст письма, передается из диалогового окна вьювера.
attachmentName	Имя отчета во вложении, по умолчанию используется имя файла отчета.
charset	Кодировка, используемая для текста письма, по умолчанию используется "UTF-8".
host	Адрес SMTP сервера. Является обязательным для заполнения.
port	Порт SMTP сервера, по умолчанию используется 465.
secure	Тип шифрования соединения с почтовым сервером, может использоваться "ssl" (по умолчанию) либо "tls" шифрование.
login	Логин для подключения к почтовому серверу. Является обязательным для заполнения.
password	Пароль для подключения к почтовому серверу. Является обязательным для заполнения.
cc	Массив CC (Carbon Copy) адресов вторичных получателей письма.
bcc	Массив BCC (Blind Carbon Copy) адресов скрытых получателей письма.

© 2003-2026 Stimulsoft

onDesignReport
[+] JavaScript [-] PHP

Событие вызывается при нажатии на кнопку **Дизайн (Design)** на панели вьювера. Подробное описание и примеры использования находятся в разделе Вызов дизайнера из вьювера.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне JavaScript клиента:

Наименование	Описание
`event`	Идентификатор текущего события, имеет значение "`DesignReport`".
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • "`Viewer`"
`report`	Текущий объект отчета.
`fileName`	Имя файла текущего отчета.

9.2.16 Настройки вьювера

Настройка вьювера выполняется при помощи изменения значений свойств, находящихся в главном контейнере свойств `options` у компонента. Все свойства разделены на группы для удобства использования.

Пример установки некоторых свойств вьювера:

© 2003-2026 Stimulsoft

viewer.php

```
<?php
   use Stimulsoft\Viewer\StiViewer;
   use Stimulsoft\Viewer\Enums\StiToolbarDisplayMode;
   use Stimulsoft\Viewer\Enums\StiViewerTheme;
   use Stimulsoft\Viewer\Enums\StiHtmlExportMode;

   $viewer = new StiViewer();
   $viewer->options->appearance->theme =
   StiViewerTheme::Office2022WhiteGreen;
   $viewer->options->appearance->reportDisplayMode =
   StiHtmlExportMode::FromReport;
   $viewer->options->width = '1000px';
   $viewer->options->height = '1000px';
   $viewer->options->toolbar->displayMode =
   StiToolbarDisplayMode::Separated;
   $viewer->options->toolbar->zoom = 50;
   $viewer->options->appearance->fullScreenMode = true;
   $viewer->options->appearance->scrollbarsMode = true;
   $viewer->options->exports->ShowExportToWord = false;
   $viewer->options->exports->showExportToCsv = false;
?>
```

Полный код примера доступен на GitHub.

Главная (без группы)

Наименование	Описание
width	Задает ширину компонента в "px" или "%". По умолчанию установлено значение "100%".
height	Задает высоту компонента в "px" или "%". По умолчанию установлено значение "100%" для стандартного режима, и "650px" для режима отображения с полосами прокрутки.
localization	Устанавливает выбранную локализацию компонента. По умолчанию установлена

© 2003-2026 Stimulsoft

	английская локализация, которая встроена в сам компонент.

Оформление (Appearance)

Наименование	Описание
`theme`	Задает тему оформления вьювера. Список доступных тем оформления находится в перечислении `StiViewerTheme`. По умолчанию установлено значение `StiViewerTheme::Office2022WhiteBlue`.
`iconSet`	Предоставляет возможность установить набор иконок: • `StiWebUIIconSet::Auto` (значение по умолчанию) - устанавливает набор иконок автоматически. Для тем `Office2022` - используется набор иконок в стиле `Monoline`, для тем `Office2013` - используется набор иконок в стиле `Regular`. • `StiWebUIIconSet::Monoline` - устанавливает набор иконок в стиле `Monoline`; • `StiWebUIIconSet::Regular` - устанавливает набор иконок в стиле `Regular`.
`backgroundColor`	Устанавливает цвет фона вьювера. По умолчанию, установлено значение `'white'`.
`pageBorderColor`	Устанавливает цвет границы страниц отчета. По умолчанию,

© 2003-2026 Stimulsoft

	установлено значение 'gray'.
rightToLeft	Устанавливает режим отображения **Right to Left** для элементов управления вьювера. По умолчанию, установлено значение false
fullScreenMode	Устанавливает полноэкранный режим отображения вьювера. Если свойство установлено в значение true, то значения свойств **width** и **height** игнорируются. По умолчанию, установлено значение false.
scrollbarsMode	Устанавливает режим отображения отчета с полосами прокрутки. По умолчанию, установлено значение false.
openLinksWindow	Устанавливает целевое окно или фрэйм для открытия гиперссылок из отчета. По умолчанию, установлено значение '_blank' (новая вкладка браузера). Может принимать одно из стандартных значений '_blank', '_self', '_top', а так же имя окна либо фрэйма.
openExportedReportWindow	Устанавливает целевое окно или фрэйм для открытия экспортированного отчета. По умолчанию, установлено значение '_blank' (новая вкладка браузера). Может принимать одно из стандартных значений '_blank', '_self', '_top', а так же имя окна либо фрэйма.
showTooltips	Включает или выключает отображение подсказок при

© 2003-2026 Stimulsoft

	наведении курсора на инструмент вьювера. По умолчанию, установлено значение `true`.
`showTooltipsHelp`	Предоставляет возможность отображать или не отображать ссылку на документацию в подсказках при наведении курсора на инструмент вьювера. По умолчанию, установлено значение `true`.
`showDialogsHelp`	Предоставляет возможность отображать или не отображать кнопку вызова справки в различных меню. По умолчанию, установлено значение `true`.
`pageAlignment`	Устанавливает выравнивание страниц отчета во вьювере. • `StiContentAlignmen::Default` – выравнивание страниц определяется из настроек шаблона; • `StiContentAlignment::Left` – выравнивание страниц по левому краю; • `StiContentAlignment::Center` – выравнивание страниц по центру (значение по умолчанию); • `StiContentAlignment::Right` – выравнивание страниц по правому краю.
`showPageShadow`	Включает или выключает отображение тени страниц отчета. По умолчанию, установлено значение `false`.
`bookmarksPrint`	Включает печать закладок в отчете. По умолчанию,

© 2003-2026 Stimulsoft

	установлено значение `false`.
`bookmarksTreeWidth`	Устанавливает ширину панели закладок в пикселях. По умолчанию, установлено значение 180.
`parametersPanelPosition`	Устанавливает расположение панели параметров во вьювере: • `StiParametersPanelPosition::FromReport` – расположение панели определяется из настроек шаблона (значение по умолчанию); • `StiParametersPanelPosition::Top` - панель располагается вверху над страницей отчета; • `StiParametersPanelPosition::Left` - панель располагается слева от страницы отчета.
`parametersPanelMaxHeight`	Устанавливает максимальную высоту панели параметров в пикселях. По умолчанию, установлено значение 300.
`parametersPanelColumnsCount`	Устанавливает количество колонок на панели параметров. По умолчанию, установлено значение 2.
`minParametersCountForMultiColumns`	Устанавливает минимальное количество переменных на панели параметров для режима отображения в несколько колонок. По умолчанию, установлено значение 5.
`parametersPanelDateFormat`	Устанавливает формат даты и времени для переменных, которые

© 2003-2026 Stimulsoft

	отображаются на панели параметров. По умолчанию, значение не задано.
`parametersPanelSortDataItems`	Устанавливает или выключает режим сортировки значений переменной. По умолчанию, опция установлена в значение `false`, т.е. значения переменной отображаются в исходном порядке.
`interfaceType`	Устанавливает тип интерфейса вьювера. Могут использоваться следующие значения: • `StiInterfaceType::Auto` – тип интерфейса вьювера будет выбран автоматически в зависимости от используемого устройства (значение по умолчанию); • `StiInterfaceType::Mouse` – принудительное использование стандартного интерфейса для управления вьювером при помощи мыши; • `StiInterfaceType::Touch` – принудительное использование Touch интерфейса для управления вьювером при помощи сенсорного экрана монитора, в этом режиме элементы интерфейса вьювера имеют увеличенные размеры для удобства управления; • `StiInterfaceType::Mobile` – принудительное использование Mobile интерфейса для управления вьювером при помощи экрана

© 2003-2026 Stimulsoft

	смартфона, в этом режиме интерфейс вьювера имеет упрощенный вид и адаптирован для управления при помощи мобильного устройства.
`allowMobileMode`	Включает или выключает возможность отображения отчета или дашборда в мобильном режиме. Если опция установлена в значение `false`, то мобильный режим просмотра не будет использоваться ни при каких обстоятельства. Если опция установлена в значение `true`, то мобильный режим просмотра будет использоваться запуске вьювера на мобильных устройствах. По умолчанию, опция установлена в значение `true`.
`chartRenderType`	Устанавливает тип отрисовки диаграмм в отчете: • `StiChartRenderType::AnimatedVector` – диаграммы будут отрисованы в векторном режиме с анимацией (значение по умолчанию); • `StiChartRenderType::Vector` – диаграммы будут отрисованы как векторное изображение без анимации.
`reportDisplayMode`	Устанавливает режим экспорта для отображения страниц отчета. Может принимать одно из следующих значений: • `StiHtmlExportMode::FromReport` - режим экспорта

© 2003-2026 Stimulsoft

	элементов отчета определяется из настроек шаблона - Div или Table (значение по умолчанию); • `StiHtmlExportMode::Table` – элементы отчета экспортируются с использованием таблиц HTML; • `StiHtmlExportMode::Div` – элементы отчета экспортируются с использованием разметки DIV;
`datePickerFirstDayOfWeek`	Предоставляет возможность установить первый день недели для инструмента **Выбор даты (Date picker)**. • `StiFirstDayOfWeek::Auto` - Будет установлен понедельник или воскресенье как первый день недели, в зависимости от культуры браузера. • `StiFirstDayOfWeek::Monday` - Понедельник будет установлен как первый день недели. • `StiFirstDayOfWeek::Sunday` - Воскресенье будет установлен как первый день недели.
`datePickerIncludeCurrentDayForRanges`	Предоставляет возможность включать или не включать текущий день в диапазон значений элемента **Выбор даты (Date Picker)**. По умолчанию, опция установлена в значение `false`, т.е. текущий день не включается в диапазон значений элемента.

© 2003-2026 Stimulsoft

appearance.allowScrollZoom	Предоставляется возможность изменять масштаб вьювера при помощи колеса прокрутки мыши. По умолчанию, опция установлена в значение true.
allowTouchZoom	Предоставляется возможность изменять масштаб вьювера касанием. По умолчанию, опция установлена в значение true.
combineReportPages	Предоставляет возможность объединить обработанные страницы шаблона отчета в один шаблон или представить каждую страницу шаблона отдельной вкладкой во вьювере. По умолчанию, опция установлена в значение false, т.е. каждая страница шаблона отчета будет представлена отдельной вкладкой во вьювере.
allowPropagationEvents	Предоставляет возможность разрешить распространение событий нажатия клавиш, когда вьювер отчетов не в фокусе. По умолчанию свойство имеет значение true.
dashboardFilterElementItemHeight	Предоставляет возможность изменить высоту флажка в элементе дашборда **Список (List Box)**.

Панель инструментов (Toolbar)

Наименование	Описание
visible	Предоставляет возможность отображать или не отображать

© 2003-2026 Stimulsoft

	панель инструментов вьювера. По умолчанию, установлено значение `true`.
`displayMode`	Задает режим отображения панели инструментов вьювера. Может принимать одно из указанных ниже значений перечисления `displayMode`: • `StiToolbarDisplayMode::Simple` – простой режим отображения, все элементы управления расположены на одной панели управления (значение по умолчанию); • `StiToolbarDisplayMode::Separated` – раздельный режим отображения, панель управления разделена на верхнюю и нижнюю.
`backgroundColor`	Предоставляет возможность изменить цвет панели инструментов. По умолчанию, установлено значение `'transparent'`.
`borderColor`	Предоставляет возможность изменить цвет границы панели инструментов. По умолчанию, установлено значение `'transparent'`.
`fontColor`	Предоставляет возможность изменить цвет шрифта для всех элементов на панели инструментов и во всех меню этой панели. По умолчанию, установлено значение `'transparent'`.
`fontFamily`	Предоставляет возможность

© 2003-2026 Stimulsoft

	изменить шрифт для всех элементов на панели инструментов и во всех меню этой панели. По умолчанию, установлено значение `'Arial'`.
`alignment`	Устанавливает выравнивание элементов на панели управления. • `StiContentAlignment::Default` – выравнивание элементов зависит от опции **RightToLeft** (значение по умолчанию); • `StiContentAlignment::Left` – все элементы будут выровнены по левому краю панели инструментов; • `StiContentAlignment::Center` – все элементы будут выровнены по центру панели инструментов; • `StiContentAlignment::Right` – все элементы будут выровнены по правой стороне панели инструментов.
`showButtonCaptions`	Включает или выключает отображение текста кнопок панели инструментов вьювера. По умолчанию, установлено значение `true`.
`showPrintButton`	Предоставляет возможность отображать или не отображать кнопку **Печать (Print)** на панели инструментов. По умолчанию, установлено значение `true`.
`showOpenButton`	Включает отображение кнопки **Открыть (Open)** на панели инструментов вьювера при просмотре отчетов или панели

© 2003-2026 Stimulsoft

	индикаторов (дашбордов). По умолчанию свойство имеет значение `true`.
`showSaveButton`	Включает отображение кнопки **Сохранить (Save)** на панели инструментов вьювера при просмотре отчетов или панели индикаторов (дашбордов). По умолчанию свойство имеет значение `true`.
`showSendEmailButton`	Предоставляет возможность отображать или не отображать кнопку **Отправить по Email (Send Email)** на панели инструментов. По умолчанию, установлено значение `false`. Также, необходимо добавить обработчик события onEmailReport.
`showFindButton`	Предоставляет возможность отображать или не отображать кнопку **Поиск (Find)** на панели инструментов. По умолчанию, установлено значение `true`.
`showSignatureButton`	Предоставляет возможность отображать или не отображать кнопку **Подпись (Signature)** на панели инструментов. По умолчанию, установлено значение `true`.
`showBookmarksButton`	Предоставляет возможность отображать или не отображать кнопку **Закладки (Bookmarks)** на панели инструментов. Если эта кнопка не отображается, то и панель закладок в отчете отображаться не будет. По умолчанию, установлено

© 2003-2026 Stimulsoft

	значение `true`.
`showParametersButton`	Предоставляет возможность отображать или не отображать кнопку **Параметры (Parameters)** на панели инструментов. Если эта кнопка не отображается, то и панель параметров в отчете отображаться не будет. По умолчанию, установлено значение `true`.
`showResourcesButton`	Предоставляет возможность отображать или не отображать кнопку **Ресурсы (Resources)** на панели инструментов. Если эта кнопка не отображается, то и панель ресурсов в отчете отображаться не будет. По умолчанию, установлено значение `true`.
`showEditorButton`	Предоставляет возможность отображать или не отображать кнопку **Редактировать (Editor)** на панели инструментов. Если эта кнопка не отображается, то невозможно будет изменить редактируемые элементы. По умолчанию, установлено значение `true`.
`showFullScreenButton`	Включает отображение кнопки **Полноэкранный просмотр (Full Screen)** на панели инструментов вьювера при просмотре отчетов или панели индикаторов (дашбордов). По умолчанию свойство имеет значение `true`.
`showRefreshButton`	Предоставляет возможность отображать или не отображать

© 2003-2026 Stimulsoft

	кнопку **Обновить (Refresh)** на панели инструментов вьювера при просмотре панели индикаторов (дашбордов). По умолчанию свойство имеет значение `true`.
`showFirstPageButton`	Предоставляет возможность отображать или не отображать кнопку **Первая страница (First Page)** на панели инструментов. По умолчанию, установлено значение `true`.
`showPreviousPageButton`	Предоставляет возможность отображать или не отображать кнопку **Последняя страница (Previous Page)** на панели инструментов. По умолчанию, установлено значение `true`.
`showCurrentPageControl`	Предоставляет возможность отображать или не отображать индикатор текущей страницы на панели инструментов. По умолчанию, установлено значение `true`.
`showNextPageButton`	Предоставляет возможность отображать или не отображать кнопку **Следующая страница (Next Page)** на панели инструментов. По умолчанию, установлено значение `true`.
`showLastPageButton`	Предоставляет возможность отображать или не отображать кнопку **Предыдущая страница (Last Page)** на панели инструментов. По умолчанию, установлено значение `true`.
`showZoomButton`	Предоставляет возможность отображать или не отображать

© 2003-2026 Stimulsoft

	кнопку выбора масштаба на панели инструментов. По умолчанию, установлено значение `true`.
`showViewModeButton`	Предоставляет возможность отображать или не отображать кнопку выбора режимов отображения страниц отчета. По умолчанию, установлено значение `true`.
`showDesignButton`	Включает отображение кнопки **Дизайн (Design)** на панели инструментов вьювера при просмотре отчетов или панели индикаторов (дашбордов). По умолчанию свойство имеет значение `false`.
`showAboutButton`	Предоставляет возможность отображать или не отображать кнопку **О программе (About)** на панели инструментов. По умолчанию, установлено значение `true`.
`showPinToolbarButton`	Предоставляет возможность отображать или не отображать кнопку **Закрепить (Pin)** в мобильном режиме просмотра отчета. По умолчанию, установлено значение `true`.
`printDestination`	Устанавливает режим печати отчета. Может принимать одно из указанных ниже значений перечисления: • `StiPrintDestination::Default` – будет отображено меню с предоставлением выбора режима печати

© 2003-2026 Stimulsoft

	(значение по умолчанию); • `StiPrintDestination::Pdf` – печать будет осуществлена в формате PDF; • `StiPrintDestination::Direct` – печать будет осуществлена в формате HTML непосредственно на принтер, будет отображен системный диалог печати; • `StiPrintDestination::PopupWindow` – печать будет осуществлена в формате HTML через всплывающее окно предварительного просмотра отчета.
`viewMode`	Устанавливает режим отображения страниц отчета: • `StiWebViewMode::SinglePage` – отображается одна страница, выбранная на панели инструментов (значение по умолчанию); • `StiWebViewMode::Continuous` – отображаются все страницы отчета в виде ленты; • `StiWebViewMode::MultiplePages` – отображаются все страницы отчета в виде таблицы.
`zoom`	Предоставляет возможность установить масштаб страниц отчета при загрузке вьювера. По умолчанию, установлено значение 100 процентов. Максимальное значение 500 процентов. Дополнительно, можно установить одно из следующих значений для

© 2003-2026 Stimulsoft

	масштаба: • `StiZoomMode::PageWidth` – масштаб страниц отчета по ширине страницы. • `StiZoomMode::PageHeight` – масштаб страниц отчета по высоте страницы.
`menuAnimation`	Предоставляет возможность включить или выключить анимацию отображения и закрытия различных меню во вьювере. По умолчанию, установлено значение `true`.
`showMenuMode`	Устанавливает режим раскрытия различных меню во вьювере. Может принимать одно из указанных ниже значений: • `StiShowMenuMode::Click` – режим раскрытия меню по щелчку (значение по умолчанию); • `StiShowMenuMode::Hover` – режим раскрытия меню при наведении курсора.
`autoHide`	Устанавливает режим автоматического сворачивания панели инструментов при просмотре отчета в мобильном режиме. По умолчанию, установлено значение `false`.

Экспорт (Exports)

Наименование	Описание
`storeExportSettings`	Предоставляет возможность сохранять настройки экспорта в

© 2003-2026 Stimulsoft

	cookies. По умолчанию, установлено значение `true`.
`showExportDialog`	Включает отображение меню настроек экспорта. Если меню спрятано, экспорт будет выполнен со значениями по умолчанию. По умолчанию, установлено значение `true`.
`showExportToDocument`	Включает отображение пункта меню **Document File** в меню **Сохранить (Save)**. По умолчанию, установлено значение `true`.
`showExportToPdf`	Включает отображение пункта меню экспорта **Adobe PDF**. По умолчанию свойство имеет значение `true`.
`showExportToXps`	Включает отображение пункта меню экспорта **XPS файл**. По умолчанию свойство имеет значение `true`.
`showExportToPowerPoint`	Включает отображение пункта меню экспорта **Microsoft PowerPoint**. По умолчанию свойство имеет значение `true`.
`showExportToHtml`	Включает отображение пункта меню экспорта **HTML File**. По умолчанию свойство имеет значение `true`.
`showExportToHtml5`	Включает отображение пункта меню экспорта **HTML File**. По умолчанию свойство имеет значение `true`.
`showExportToText`	Включает отображение пункта меню экспорта **Text файл**. По умолчанию свойство имеет

© 2003-2026 Stimulsoft

	значение `true`.
`showExportToWord`	Включает отображение пункта меню экспорта **Microsoft Word**. По умолчанию свойство имеет значение `true`.
`showExportToOpenDocumentWriter`	Включает отображение пункта меню экспорта **OpenDocument Writer**. По умолчанию свойство имеет значение `true`.
`showExportToExcel`	Включает отображение пункта меню экспорта **Microsoft Excel**. По умолчанию свойство имеет значение `true`.
`showExportToOpenDocumentCalc`	Включает отображение пункта меню экспорта **OpenDocument Calc**. По умолчанию свойство имеет значение `true`.
`showExportToRtf`	Включает отображение пункта меню экспорта **RTF**. По умолчанию свойство имеет значение `true`.
`showExportToCsv`	Включает отображение типа **CSV** в настройках пункта меню экспорта **Данные (Data)**. По умолчанию свойство имеет значение `true`.
`showExportToJson`	Включает отображение типа **JSON** в настройках пункта меню экспорта **Данные (Data)**. По умолчанию свойство имеет значение `true`.
`showExportToXml`	Включает отображение типа **XML** в настройках пункта меню экспорта **Данные (Data)**. По умолчанию свойство имеет значение `true`.

© 2003-2026 Stimulsoft

showExportToDbf	Включает отображение типа **DBF** в настройках пункта меню экспорта **Данные (Data)**. По умолчанию свойство имеет значение true.
showExportToDif	Включает отображение типа **DIF** в настройках пункта меню экспорта **Данные (Data)**. По умолчанию свойство имеет значение true.
showExportToSylk	Включает отображение типа **SYLK** в настройках пункта меню экспорта **Данные (Data)**. По умолчанию свойство имеет значение true.
showExportToImagePng	Включает отображение типа **PNG** в настройках пункта меню экспорта **Изображение (Image)**. По умолчанию свойство имеет значение true.
showExportToImageJpeg	Включает отображение типа **JPEG** в настройках пункта меню экспорта **Изображение (Image)**. По умолчанию свойство имеет значение true.
showExportToImageSvg	Включает отображение типа **SVG** в настройках пункта меню экспорта **Изображение (Image)**. По умолчанию свойство имеет значение true.
showExportToImageSvgz	Включает отображение типа **SVGZ** в настройках пункта меню экспорта **Изображение (Image)**. По умолчанию свойство имеет значение true.
showExportToImagePcx	Включает отображение типа **PCX** в

© 2003-2026 Stimulsoft

	настройках пункта меню экспорта **Изображение (Image)**. По умолчанию свойство имеет значение `true`.
`showExportToImageBmp`	Включает отображение типа **BMP** в настройках пункта меню экспорта **Изображение (Image)**. По умолчанию свойство имеет значение `true`.
`showExportToImageGif`	Включает отображение типа **GIF** в настройках пункта меню экспорта **Изображение (Image)**. По умолчанию свойство имеет значение `true`.
`showExportToImageTiff`	Включает отображение типа **TIFF** в настройках пункта меню экспорта **Изображение (Image)**. По умолчанию свойство имеет значение `true`.
`showExportDataOnly`	Включает отображение опции **Экспортировать только данные (Export Data Only)** в настройках пункта меню экспорта **Данные (Data)**. По умолчанию свойство имеет значение `true`.

Отправить по Email (Email)

Наименование	Описание
`showEmailDialog`	Включает отображение диалогового окна параметров отправки отчета по Email. Если диалоговое окно отключено, то отправка по Email будет производиться с настройками,

© 2003-2026 Stimulsoft

	заданными по умолчанию `onEmailReport`. По умолчанию, установлено значение `true`.
`showExportDialog`	Включает отображение диалогового окна параметров экспорта при отправке Email. Если свойство имеет значение `false`, то экспортирование будет производиться с заданными по умолчанию настройками. По умолчанию, установлено значение `true`.
`defaultEmailAddress`	Устанавливает Email получателя по умолчанию, т.е. адрес на который будет отправлено письмо с вложенным отчетом. По умолчанию, значение не задано.
`defaultEmailSubject`	Устанавливает тему (заголовок) электронного письма по умолчанию. По умолчанию, значение не задано.
`defaultEmailMessage`	Устанавливает сообщение (текст) электронного письма по умолчанию. По умолчанию, значение не задано.

9.3 HTML5 дизайнер

Дизайнер отчетов представляет собой PHP компонент `StiDesigner`, предназначенный для разработки отчетов и дашбордов в окне браузера на любом компьютере с любой установленной операционной системой. Дизайнер имеет современный интерфейс и различные темы оформления, доступно для использования множество профессионально подготовленных шаблонов отчетов и дашбордов, мастера создания отчетов, большое количество компонентов для построения отчета и дашборда практически любой сложности.

Дизайнер имеет удобный режим предварительного просмотра отчета во

© 2003-2026 Stimulsoft

встроенном вьювере, что значительно ускоряет разработку отчета. Работа встроенного вьювера практически не отличается от работы отдельного компонента, более подробно рассмотрена в разделе Вьювер отчетов.

Интерфейс дизайнера построен с использованием HTML5, что позволяет использовать его практически на любой современной платформе. Компонент использует технологию AJAX для выполнения всех действий (загрузка и построение отчета, подключение к данным, работа с компонентами и их настройками, предварительный просмотр отчета, печать, экспортирование и др.), что позволяет избавиться от перезагрузки всей страницы и повысить скорость работы, а также использовать компонент в One-Page приложениях. Технология JavaScript, применяемая для построения отчетов, позволяет использовать практически любую серверную сторону невысокой производительности.

Информация

Вследствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово “отчет”.

- Использование
- Активация лицензии
- Создание и редактирование отчета
- Сохранение отчета
- Локализация
- Просмотр
- Визуальное оформление
- События дизайнера
- Настройки

9.3.1 Использование

Для использования продукта, достаточно загрузить ZIP архив продукта со страницы Downloads нашего сайта, распаковать его, и скопировать содержимое папки /PHP на ваш Web сервер. Данная папка представляет собой Web проект, который содержит все необходимые файлы и ресурсы для работы продукта, а

© 2003-2026 Stimulsoft

также примеры по работе с вьювером и дизайнером.

Для установки продукта в уже существующий проект, достаточно из папки **/PHP** скопировать папку `/vendor` в корневую директорию проекта, либо воспользоваться менеджером зависимостей Composer, выполнив следующую консольную команду:

console

```
composer require stimulsoft/reports-php
```

Для работы дашбордов понадобится подключить следующий пакет:

console

```
composer require stimulsoft/dashboards-php
```

При работе с продуктом, в большинстве случаев достаточно использовать только PHP код, который обеспечивает работу всех основных возможностей. Для более детальной настройки продукта и использования всех возможностей, необходимо использовать **JavaScript** код.

Для использования компонентов в Web проекте, достаточно подключить автоматический загрузчик скриптов в начале PHP файла. После этого, можно использовать все доступные PHP классы и функции для работы с отчетами и дашбордами:

designer.php

```
<?php
   require_once 'vendor/autoload.php';

...
?>
```

© 2003-2026 Stimulsoft

Для использования дизайнера в Web-проекте, предназначен класс `StiDesigner`. При помощи данного класса можно создать объект дизайнера, установить необходимые настройки, присвоить отчет для редактирования, обрабатывать запросы и управлять событиями дизайнера.

Пример редактирования отчета в дизайнере на HTML странице:

designer.php

```
<?php
   require_once 'vendor/autoload.php';

   use Stimulsoft\Report\StiReport;
   use Stimulsoft\Designer\StiDesigner;

   $designer = new StiDesigner();
   $designer->process();

   $report = new StiReport();
   $report->loadFile('reports/SimpleList.mrt');
   $designer->report = $report;
?>

<html>
<head>
   <?php
      $designer->javascript->renderHtml();
   ?>
</head>
<body>
   <?php
      $designer->renderHtml();
   ?>
</body>
</html>
```

Полный код примера доступен на GitHub.

В этом примере последовательно выполняются следующие действия:

- создается экземпляр объекта StiDesigner;
- выполняется обработка текущего запроса;
- создается экземпляр объекта StiReport;
- выполняется загрузка шаблона отчета из файла SipmleList.mrt;
- созданный отчет присваивается дизайнеру;
- в шаблоне HTML файла выполняется вывод необходимого JavaScript и HTML кода компонента.

© 2003-2026 Stimulsoft

Метод `$designer->process()` выполняет обработку текущего запроса, и, в случае успеха, автоматически возвращает результат на сторону клиента. Более подробно про это рассказано в разделе Обработчик событий.

Метод `$designer->javascript->renderHtml()` выполняет вывод кода для подключения необходимых скриптов компонента. Метод `$designer->renderHtml()` выполняет вывод JavaScript и HTML кода самого компонента. Предусмотрено несколько вариантов использования, более подробно про это рассказано в разделе Использование генератора отчетов.

Пример упрощённого отображения дизайнера без использования HTML шаблона страницы:

designer.php

```
<?php
   require_once 'vendor/autoload.php';

   use Stimulsoft\Report\StiReport;
   use Stimulsoft\Designer\StiDesigner;

   $designer = new StiDesigner();
   $designer->process();

   $report = new StiReport();
   $report->loadFile('reports/SimpleList.mrt');
   $designer->report = $report;

   $designer->printHtml();
?>
```

Полный код примера доступен на GitHub.

9.3.2 Активация лицензии

После приобретения продукта вам необходимо активировать лицензию для используемых компонентов. Предусмотрено несколько способов подключения лицензионного ключа.

Все варианты активации компонентов описаны в разделе Активация лицензии генератора отчетов, имеют одинаковые функции и параметры вызова.

© 2003-2026 Stimulsoft

9.3.3 Создание и редактирование отчета

Для запуска дизайнера без отчета не требуется никаких действий. После загрузки компонента, отобразится главное меню дизайнера. При необходимости запуска дизайнера с новым (пустым) отчетом, можно создать новый объект отчета `StiReport` и присвоить его дизайнеру.

Для редактирования отчета в дизайнере, достаточно создать объект `StiReport`, загрузить в него шаблон отчета, и присвоить полученный объект дизайнеру. Все остальные действия будут выполнены автоматически, дизайнер отобразит первую страницу шаблона:

designer.php

```
<?php
   require_once 'vendor/autoload.php';

   use Stimulsoft\Report\StiReport;
   use Stimulsoft\Designer\StiDesigner;

   $designer = new StiDesigner();
   $designer->process();

   $report = new StiReport();
   $report->loadFile('reports/SimpleList.mrt');
   $designer->report = $report;

   $designer->printHtml();
?>
```

Полный код примера доступен на GitHub.

Дизайнер умеет работать со стандартными, упакованными и зашифрованными шаблонами. Подробное описание работы с различными форматами отчетов находится в разделе Загрузка и сохранение отчета.

Событие создания отчета

Новый отчет можно создать, используя главное меню дизайнера. Для выполнения какие-либо необходимых действий с новым отчетом, предназначено событие `onCreateReport`. Данное событие будет вызвано при

© 2003-2026 Stimulsoft

создании нового пустого отчета из главного меню, либо при создании отчета с использованием мастера. В аргументах события будет передан созданный в дизайнере объект отчета. При необходимости, его можно изменить, подключить данные, либо загрузить заранее подготовленный шаблон отчета. После завершения события, этот отчет будет автоматически загружен в дизайнер для редактирования.

Пример подключения данных и синхронизации словаря данных при создании нового отчета:

designer.php

```
<?php
   use Stimulsoft\Report\StiReport;
   use Stimulsoft\Designer\StiDesigner;

   $designer = new StiDesigner();
   $designer->onCreateReport = 'createReport';
   $designer->process();

   $report = new StiReport();
   $report->loadFile('reports/SimpleList.mrt');
   $designer->report = $report;
?>

...

<script>
   function createReport(args) {
      let dataSet = new Stimulsoft.System.Data.DataSet("SimpleDataSet");
      dataSet.readJsonFile("Data/Demo.json");

      args.report.regData(dataSet.dataSetName, "", dataSet);
      args.report.dictionary.synchronize();
   }
</script>
```

Полный код примера доступен на GitHub.

При необходимости, можно контролировать процесс создания нового отчета на стороне PHP сервера. Пример изменения параметров нового отчета на стороне PHP сервера:

© 2003-2026 Stimulsoft

designer.php

```
<?php
   use Stimulsoft\Designer\StiDesigner;
   use Stimulsoft\Events\StiReportEventArgs;

   $designer = new StiDesigner();
   $designer->onCreateReport = function (StiReportEventArgs $args) {
      $args->report->ReportAlias = 'New Report Alias';
   };

   $designer->process();
   $designer->printHtml();
?>
```

Пример загрузки предварительно подготовленного шаблона с настроенными подключениями к данным:

designer.php

```
<?php
   use Stimulsoft\Designer\StiDesigner;
   use Stimulsoft\Events\StiReportEventArgs;

   $designer = new StiDesigner();
   $designer->onCreateReport = function (StiReportEventArgs $args) {
      $reportJson = file_get_contents('reports/NewTemplateWithData.mrt');
      $args->setReportJson($reportJson);
   };

   $designer->process();
   $designer->printHtml();
?>
```

Подробное описание доступных значений аргументов находится в разделе События дизайнера.

9.3.4 Сохранение отчета

В дизайнере предусмотрено два варианта сохранения отчета, которые доступны в главном меню и на главной панели дизайнера: **Сохранить (Save)** и **Сохранить как (Save As)**. В свою очередь, каждый из этих вариантов сохранения имеет свои режимы и настройки.

© 2003-2026 Stimulsoft

Сохранение отчета на стороне JavaScript клиента

При нажатии кнопки **Сохранить (Save)**, сохранение файла шаблона отчета осуществляется средствами браузера, для этого не требуется каких-либо настроек. При необходимости сохранения отчета своими методами, предназначено событие `onSaveReport`. В аргументах события будут переданы имя файла отчета и сам отчет. Отчет можно сохранить, например, в JSON строку и передать на сторону сервера своими методами.

Пример сохранения отчета в JSON строку на стороне JavaScript клиента для дальнейшего использования в приложении:

designer.php

```
<?php
   use Stimulsoft\Report\StiReport;
   use Stimulsoft\Designer\StiDesigner;

   $designer = new StiDesigner();
   $designer->onSaveReport = 'saveReport';
   $designer->process();

   $report = new StiReport();
   $report->loadFile('reports/SimpleList.mrt');
   $designer->report = $report;
?>

...

<script>
   function saveReport(args) {
      let fileName = args.fileName;
      let report = args.report;
      let reportJson = args.report.saveToJsonString();
      ...
   }
</script>
```

При необходимости, после сохранения отчета можно отобразить диалоговое окно с ошибкой или текстовым сообщением. Для этого предназначена специальная функция `StiError.showError()`. Вы сами определяете необходимость отображения сообщения об ошибке.

© 2003-2026 Stimulsoft

designer.php

```
<script>
   function saveReport(args) {
      let report = args.report;

      // Error message
      Stimulsoft.System.StiError.showError("An error occurred while saving
      the report.");

      // Info message
      Stimulsoft.System.StiError.showError("The report was saved
      successfully.", true, true);
   }
</script>
```

При нажатии кнопки **Сохранить как (Save As)**, будет отображено диалоговое окно с запросом имени файла отчета. После этого сохранение файла шаблона отчета будет осуществляться средствами браузера. При необходимости сохранения отчета своими методами, предназначено событие `onSaveAsReport`. В аргументах события будут переданы имя файла отчета и сам отчет.

Работа данного события не отличается от работы события `onSaveReport`, за исключением того, что после завершения события произойдет автоматическое сохранение шаблона отчета на компьютере средствами браузера. Для предотвращения этого действия, достаточно установить свойство `preventDefault` в значение `true` в аргументах события, в этом случае автоматическое сохранение не будет выполнено.

designer.php

```
<script>
   function saveReport(args) {
      args.preventDefault = true;
   }
</script>
```

При необходимости, вы можете получить доступ к оригинальному имени отчета или имени отчета из диалога сохранения следующим образом:

© 2003-2026 Stimulsoft

designer.php

```
<script>
   function saveReport(args) {
      // Report name from the designer save dialog
      let reportName1 = args.fileName;

      // Original report name from properties
      let reportName2 = args.report.reportName;
   }
</script>
```

Подробное описание доступных значений аргументов находится в разделе События дизайнера.

Сохранение отчета на стороне PHP сервера

Для сохранения отчета на стороне PHP сервера, достаточно определить событие `onSaveReport`, в аргументах события будут переданы имя файла отчета и сам отчет в виде объекта. Для сохранения отчета можно воспользоваться стандартными PHP функциями.

Пример сохранения редактируемого отчета в виде файла в заданной директории:

© 2003-2026 Stimulsoft

designer.php

```
<?php
   use Stimulsoft\Report\StiReport;
   use Stimulsoft\Designer\StiDesigner;
   use Stimulsoft\Events\StiReportEventArgs;

   $designer = new StiDesigner();
   $designer->onSaveReport = function (StiReportEventArgs $args) {
      $reportJson = $args->getReportJson();
      file_put_contents('reports/' . $reportFileName, $reportJson);
   };

   $designer->process();

   $report = new StiReport();
   $report->loadFile('reports/SimpleList.mrt');
   $designer->report = $report;

   $designer->printHtml();
?>
```

Полный код примера доступен на GitHub.

Подробное описание доступных значений аргументов находится в разделе События дизайнера.

Информация

Аналогичным образом реализована работа события `onSaveAsReport` на стороне PHP сервера, все аргументы события имеют те же имена и значения.

9.3.5 Локализация

Дизайнер поддерживает полную локализацию своего интерфейса. Для локализации интерфейса на требуемый язык, достаточно установить требуемое название файла для опции `localization` у дизайнера:

© 2003-2026 Stimulsoft

designer.php

```
<?php
   use Stimulsoft\Designer\StiDesigner;

   $designer = new StiDesigner();
   $designer->options->localization = 'de.xml';
   $designer->printHtml();
?>
```

Полный код примера доступен на GitHub.

Все доступные локализационные XML файлы находятся в ресурсах установленного пакета продукта. При необходимости, локализационный файл можно загрузить из любого другого места, для этого необходимо для опции `localization` указать полный путь к нужному XML файлу:

designer.php

```
<?php
   use Stimulsoft\Designer\StiDesigner;

   $designer = new StiDesigner();
   $designer->options->localization = '/resources/loc/de.xml';
   $designer->printHtml();
?>
```

Дизайнер имеет возможность выбрать необходимую локализацию интерфейса при помощи специального меню на панели инструментов. По умолчанию, в данное меню добавлена английская (встроенная) локализация, а также заданная при помощи свойства `localization`. Для добавления дополнительных локализаций в меню, предназначен специальный метод `addLocalization()` в опциях дизайнера. В качестве параметра указывается локализационный файл либо полный путь к этому файлу.

Пример добавления дополнительных локализаций, находящихся в ресурсах компонента:

© 2003-2026 Stimulsoft

designer.php

```
<?php
   use Stimulsoft\Designer\StiDesigner;

   $designer = new StiDesigner();
   $designer->options->localization = 'de.xml';
   $designer->options->addLocalization('fr.xml');
   $designer->options->addLocalization('es.xml');
   $designer->options->addLocalization('pt.xml');
   $designer->printHtml();
?>
```

Полный код примера доступен на GitHub.

9.3.6 Просмотр

В дизайнере предусмотрен режим предварительного просмотра редактируемого отчета. Для этого достаточно перейти на соответствующую вкладку в окне дизайнера. Шаблон отчета будет построен и отображен во встроенном вьювере.

Событие предварительного просмотра

Перед просмотром отчета предусмотрена возможность выполнить какие-либо необходимые действия, например, подключить данные для отчета. Для этого предназначено специальное событие `onPreviewReport`, которое будет вызвано перед просмотром отчета. В аргументах события будет передан отчет, предназначенный для просмотра.

Пример подключения данных на стороне JavaScript клиента для предварительного просмотра отчета:

© 2003-2026 Stimulsoft

designer.php

```
<?php
   use Stimulsoft\Report\StiReport;
   use Stimulsoft\Designer\StiDesigner;

   $designer = new StiDesigner();
   $designer->onPreviewReport = 'previewReport';
   $designer->process();

   $report = new StiReport();
   $report->loadFile('reports/SimpleList.mrt');
   $designer->report = $report;
?>

...

<script>
   function previewReport(args) {
      let dataSet = new Stimulsoft.System.Data.DataSet("SimpleDataSet");
      dataSet.readJsonFile("Data/Demo.json");

      args.report.regData(dataSet.dataSetName, "", dataSet);
   }
</script>
```

Пример изменения свойств отчета на стороне PHP сервера перед предварительным просмотром отчета:

designer.php

```
<?php
   use Stimulsoft\Report\StiReport;
   use Stimulsoft\Designer\StiDesigner;

   $designer = new StiDesigner();
   $designer->onPreviewReport = function (StiReportEventArgs $args) {
      $args->report->ReportDescription = 'This is a report description from
      the PHP server-side.';
   };

   $designer->process();

   $report = new StiReport();
   $report->loadFile('reports/SimpleList.mrt');
   $designer->report = $report;

   $designer->printHtml();
?>
```

© 2003-2026 Stimulsoft

Подробное описание доступных значений аргументов находится в разделе События дизайнера.

Дополнительные возможности

Окно предварительного просмотра отчета в дизайнере представляет собой полноценный интерактивный вьювер, который умеет осуществлять печать и экспортирование отчета, поддерживает работу с параметрами отчета. Поддерживаются все доступные интерактивные действия, такие как динамическая сортировка, детализация, сворачивание. Для использования указанных возможностей не требуется каких-либо дополнительных настроек дизайнера отчетов.

9.3.7 Визуальное оформление

Дизайнер имеет возможность смены тем оформления визуальных элементов управления. Для этого достаточно установить свойство `theme` в опциях компонента:

designer.php

```
<?php
   use Stimulsoft\Designer\StiDesigner;
   use Stimulsoft\Designer\Enums\StiDesignerTheme;

   $designer = new StiDesigner();
   $designer->options->appearance->theme =
   StiDesignerTheme::Office2022BlackGreen;
?>
```

На данный момент доступно **2 различные темы** со своими цветовыми акцентами. В результате, доступно **более 50 вариантов** оформления. Это позволяет настроить внешний вид дизайнера практически под любое оформление Web проекта.

9.3.8 Добавление пользовательских функций

При интеграции дизайнера отчетов в пользовательское приложение, присутствует возможность добавления собственных JavaScript функции в словарь данных дизайнера отчетов. После добавления функции, её можно

© 2003-2026 Stimulsoft

использовать при разработке отчетов и панелей индикаторов (дашбордов). Ниже представлен пример добавления функции расчета итога суммирования:

designer.php

```
<?php
   use Stimulsoft\Designer\StiDesigner;
   use Stimulsoft\Enums\Types;
   use Stimulsoft\Report\StiFunctions;

   StiFunctions::addFunction(
      "MyCategory", "MySum", "MySum", "MySum", "", Types::string,
      "Return Description", [Types::int], ["value"], ["Descriptions"],
      "myFunc");


   $designer = new StiDesigner();
   $designer->printHtml();
?>

...

<script>
   function myFunc (value) {
      if (!Stimulsoft.Data.Extensions.ListExt.isList(value))
         return Stimulsoft.Base.Helpers.StiValueHelper.tryToNumber(value);

      return Stimulsoft.Data.Functions.Funcs
         .skipNulls(Stimulsoft.Data.Extensions.ListExt.toList(value))
         .tryCastToNumber()
         .sum();
   };
</script>
```

В случае использования дизайнера без HTML шаблона, предусмотрена возможность передать код функции вместо её имени. Остальные параметры функции останутся такими же:

designer.php

```
<?php
   use Stimulsoft\Designer\StiDesigner;
   use Stimulsoft\Enums\Types;
   use Stimulsoft\Report\StiFunctions;

   StiFunctions::addFunction(
      "MyCategory", "MySum", "MySum", "MySum", "", Types::string,
      "Return Description", [Types::int], ["value"], ["Descriptions"],
```

© 2003-2026 Stimulsoft

```
    "
      function myFunc (value) {
        if (!Stimulsoft.Data.Extensions.ListExt.isList(value))
          return
          Stimulsoft.Base.Helpers.StiValueHelper.tryToNumber(value);

        return Stimulsoft.Data.Functions.Funcs
          .skipNulls(Stimulsoft.Data.Extensions.ListExt.toList(value))
          .tryCastToNumber()
          .sum();
      };
    ");

  $designer = new StiDesigner();
  $designer->printHtml();
?>
```

9.3.9 События дизайнера

Дизайнер отчетов поддерживает события, которые предоставляют возможность выполнить необходимые операции перед определенными действиями - как на стороне JavaScript клиента, так и на стороне PHP сервера. Подробное описание работы событий находится в разделе События генератора отчетов.

Дизайнер поддерживает следующие события:

- onDatabaseConnect
- onBeginProcessData
- onEndProcessData
- onPrepareVariables
- onCreateReport
- onOpenReport
- onOpenedReport
- onSaveReport
- onSaveAsReport
- onPreviewReport
- onCloseReport
- onExit

© 2003-2026 Stimulsoft

onDatabaseConnect
[-] JavaScript [+] PHP

Событие вызывается перед подключением к базе данных после получения всех параметров. Подробное описание и примеры использования находятся в разделе Подключение SQL адаптеров данных. Список аргументов события находится в разделе События генератора отчетов.

onBeginProcessData
[+] JavaScript [+] PHP

Событие вызывается перед запросом данных, необходимых для построения отчета. Подробное описание и примеры использования находятся в разделах Подключение файловых данных и Подключение SQL адаптеров данных. Список аргументов события находится в разделе События генератора отчетов.

onEndProcessData
[+] JavaScript [+] PHP

Событие вызывается после загрузки данных перед построением отчета. Подробное описание и примеры использования находятся в разделах Подключение файловых данных и Подключение SQL адаптеров данных.. Список аргументов события находится в разделе События генератора отчетов.

onPrepareVariables
[+] JavaScript [+] PHP

Событие вызывается перед построением отчета после подготовки переменных отчета. Подробное описание и примеры использования находятся в разделе Работа с переменными отчета.. Список аргументов события находится в разделе События генератора отчетов..

onCreateReport
[+] JavaScript [+] PHP

© 2003-2026 Stimulsoft

Событие вызывается после создания нового отчета в дизайнере. Подробное описание и примеры использования находятся в разделе Создание и редактирование отчета.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне JavaScript клиента:

Наименование	Описание
`event`	Идентификатор текущего события, имеет значение "`CreateReport`".
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • "`Designer`"
`report`	Текущий объект отчета.
`isWizardUsed`	Флаг указывает на то, что создание нового отчета осуществляется при помощи мастера (значение `true`), либо создается пустой отчет (значение `false`).

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне PHP сервера, аргументы имеют тип `StiReportEventArgs`:

Наименование	Описание
`event`	Идентификатор текущего события, для данного события имеет значение `StiEventType::CreateReport`
`sender`	Компонент, который инициировал данное событие, может иметь следующие типы: • `StiDesigner`

© 2003-2026 Stimulsoft

report	Текущий объект отчета.
isWizardUsed	Флаг указывает на то, что создание нового отчета осуществляется при помощи мастера (значение true), либо создается пустой отчет (значение false).

onOpenReport
[+] JavaScript [-] PHP

Событие вызывается перед открытием отчета из меню дизайнера.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне JavaScript клиента:

Наименование	Описание
event	Идентификатор текущего события, имеет значение "OpenReport".
sender	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • "Designer"
preventDefault	Данный флаг предоставляет возможность остановить дальнейшую обработку события дизайнером. По умолчанию установлено значение true.

onOpenedReport
[+] JavaScript [+] PHP

Событие вызывается после открытия отчета из меню дизайнера до его передачи в сам дизайнер.

© 2003-2026 Stimulsoft

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне JavaScript клиента:

Наименование	Описание
`event`	Идентификатор текущего события, имеет значение `"OpenedReport"`.
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • `"Designer"`
`report`	Текущий объект отчета.
`preventDefault`	Данный флаг предоставляет возможность остановить дальнейшую обработку события вьювером. По умолчанию установлено значение `true`.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне PHP сервера, аргументы имеют тип `StiReportEventArgs`:

Наименование	Описание
`event`	Идентификатор текущего события, для данного события имеет значение `StiEventType::OpenedReport`
`sender`	Компонент, который инициировал данное событие, может иметь следующие типы: • `StiDesigner`
`report`	Текущий объект отчета.

© 2003-2026 Stimulsoft

onSaveReport
[+] JavaScript [+] PHP

Событие вызывается при сохранении отчета в дизайнере. Подробное описание и примеры использования находятся в разделе Сохранение отчета.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне JavaScript клиента:

Наименование	Описание
`event`	Идентификатор текущего события, имеет значение "`SaveReport`".
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • "`Designer`"
`report`	Текущий объект отчета.
`fileName`	Имя файла отчета для сохранения.
`autoSave`	Данный флаг на то, что сохранение отчета осуществляется автоматически (значение `true`), либо при нажатии на кнопку сохранения (значение `false`).
`preventDefault`	Данный флаг предоставляет возможность остановить дальнейшую обработку события дизайнером. По умолчанию установлено значение `true`.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне PHP сервера, аргументы имеют тип `StiReportEventArgs`:

Наименование	Описание

© 2003-2026 Stimulsoft

`event`	Идентификатор текущего события, для данного события имеет значение `StiEventType::SaveReport`.
`sender`	Компонент, который инициировал данное событие, может иметь следующие типы: • `StiDesigner`
`report`	Текущий объект отчета.
`fileName`	Имя файла отчета для сохранения.
`autoSave`	Данный флаг на то, что сохранение отчета осуществляется автоматически (значение `true`), либо при нажатии на кнопку сохранения (значение `false`).

onSaveAsReport
[+] JavaScript [+] PHP

Событие вызывается при сохранении отчета в дизайнере с предварительным вводом имени файла. Подробное описание и примеры использования находятся в разделе Сохранение отчета.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне JavaScript клиента:

Наименование	Описание
`event`	Идентификатор текущего события, имеет значение "`SaveAsReport`".
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • `"Designer"`

© 2003-2026 Stimulsoft

`report`	Текущий объект отчета.
`fileName`	Имя файла отчета для сохранения.
`preventDefault`	Данный флаг предоставляет возможность остановить дальнейшую обработку события дизайнером. По умолчанию установлено значение `false`.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне PHP сервера, аргументы имеют тип `StiReportEventArgs`:

Наименование	Описание
`event`	Идентификатор текущего события, для данного события имеет значение `StiEventType::SaveAsReport`.
`sender`	Компонент, который инициировал данное событие, может иметь следующие типы: • `StiDesigner`
`report`	Текущий объект отчета.
`fileName`	Имя файла отчета для сохранения.

onPreviewReport
[+] JavaScript [+] PHP

Событие вызывается при переходе на вкладку просмотра отчета. Подробное описание и примеры использования находятся в разделе Просмотр.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне JavaScript клиента:

Наименование	Описание

© 2003-2026 Stimulsoft

event	Идентификатор текущего события, имеет значение "PreviewReport".
sender	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • "Designer"
report	Текущий объект отчета.
viewer	Текущий объект встроенного в дизайнер компонента StiViewer.
preventDefault	Данный флаг предоставляет возможность остановить дальнейшую обработку события дизайнером. По умолчанию установлено значение true.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне PHP сервера, аргументы имеют тип StiReportEventArgs:

Наименование	Описание
event	Идентификатор текущего события, для данного события имеет значение StiEventType::PreviewReport.
sender	Компонент, который инициировал данное событие, может иметь следующие типы: • StiDesigner
report	Текущий объект отчета.

onCloseReport
[+] JavaScript [+] PHP

© 2003-2026 Stimulsoft

Событие вызывается после закрытия отчета из меню дизайнера до его отмены в дизайнере.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне JavaScript клиента:

Наименование	Описание
`event`	Идентификатор текущего события, имеет значение `"CloseReport"`.
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • `"Designer"`
`report`	Текущий объект отчета.
`preventDefault`	Данный флаг предоставляет возможность остановить дальнейшую обработку события вьювером. По умолчанию установлено значение `true`.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне PHP сервера, аргументы имеют тип `StiReportEventArgs`:

Наименование	Описание
`event`	Идентификатор текущего события, для данного события имеет значение `StiEventType::CloseReport`
`sender`	Компонент, который инициировал данное событие, может иметь следующие типы: • `StiDesigner`
`report`	Текущий объект отчета.

© 2003-2026 Stimulsoft

onExit
[+] JavaScript [-] PHP

Событие вызывается при нажатии на кнопку **Выход (Exit)** в главном меню дизайнера.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне JavaScript клиента:

Наименование	Описание
`event`	Идентификатор текущего события, имеет значение "`Exit`".
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • "`Designer`"

9.3.10 Настройки

Настройка дизайнера выполняется при помощи изменения значений свойств, находящихся в главном контейнере свойств `options` у компонента. Все свойства разделены на группы для удобства использования.

Пример установки некоторых свойств дизайнера:

© 2003-2026 Stimulsoft

designer.php

```
<?php
   use Stimulsoft\Designer\StiDesigner;
   use Stimulsoft\Designer\Enums\StiDesignerTheme;

   $designer = new StiDesigner();
   $designer->options->appearance->theme =
   StiDesignerTheme::Office2022WhiteGreen;
   $designer->options->toolbar->showFileMenuExit = false;
   $designer->options->toolbar->showFileMenuOptions = false;
   $designer->options->bands->showChildBand = false;
   $designer->options->components->showPanel = false;
   $designer->options->appearance->showReportTree = false;
   $designer->options->appearance->showTooltips = false;
?>
```

Главная (без группы)

Наименование	Описание
width	Задает ширину компонента в "px" или "%". По умолчанию установлено значение "100%".
height	Задает высоту компонента в "px" или "%". По умолчанию установлено значение "800px".
localization	Устанавливает выбранную локализацию компонента. По умолчанию установлена английская локализация, которая встроена в сам компонент.

Оформление (Appearance)

Наименование	Описание
theme	Задает тему оформления дизайнера. Список доступных тем оформления находится в

© 2003-2026 Stimulsoft

	перечислении `StiDesignerTheme`. По умолчанию установлено значение `StiDesignerTheme::Office2022WhiteBlue`.
`iconSet`	Предоставляет возможность установить набор иконок: • `StiWebUIIconSet::Auto` (значение по умолчанию) - устанавливает набор иконок автоматически. Для тем `Office2022` - используется набор иконок в стиле `Monoline`, для тем `Office2013` - используется набор иконок в стиле `Regular`. • `StiWebUIIconSet::Monoline` - устанавливает набор иконок в стиле `Monoline`; • `StiWebUIIconSet::Regular` - устанавливает набор иконок в стиле `Regular`.
`defaultUnit`	Устанавливает единицы измерения размеров для отчета и всех его компонентов: • `StiReportUnitType::Centimeters` (значение по уомлчанию); • `StiReportUnitType::HundredthsOfInch`; • `StiReportUnitType::Inches`; • `StiReportUnitType::Millimeters`.
`zoom`	Задает масштаб отображения страниц отчета. По умолчанию установлен масштаб 100

© 2003-2026 Stimulsoft

	процентов. Допускаются значения от 10 до 200 процентов. Может принимать одно из указанных ниже значений перечисления `StiZoomMode`: • `StiZoomMode::PageWidth` – масштаб страниц отчета по ширине страницы; • `StiZoomMode::PageHeight` – масштаб страниц отчета по высоте страницы.
`interfaceType`	Устанавливает тип интерфейса дизайнера. Могут использоваться следующие значения: • `StiInterfaceType::Auto` – тип интерфейса дизайнера будет выбран автоматически в зависимости от используемого устройства (значение по умолчанию); • `StiInterfaceType::Mouse` – принудительное использование стандартного интерфейса для управления дизайнером при помощи мыши; • `StiInterfaceType::Touch` – принудительное использование Touch интерфейса для управления дизайнером при помощи сенсорного экрана монитора, в этом режиме элементы интерфейса дизайнера имеют увеличенные размеры для удобства управления.
`showAnimation`	Предоставляет возможность

© 2003-2026 Stimulsoft

	включить или выключить анимацию отображения и закрытия различных меню в дизайнере. По умолчанию, установлено значение `true`.
`showSaveDialog`	Включает отображение диалога ввода имени отчета при его сохранении. Имя отчета будет передано в параметрах дизайнера. По умолчанию свойство имеет значение `true`.
`showTooltips`	Включает отображение подсказок для элементов управления дизайнера при наведении мыши. По умолчанию свойство имеет значение `true`.
`showTooltipsHelp`	Включает отображение ссылки на онлайн документацию в подсказках для элементов управления дизайнера. По умолчанию свойство имеет значение `true`.
`showDialogsHelp`	Предоставляет возможность отображать или не отображать кнопку вызова справки в различных меню. По умолчанию, установлено значение `true`.
`fullScreenMode`	Устанавливает полноэкранный режим отображения дизайнера. Если свойство установлено в значение true, то значения свойств width и height игнорируются. По умолчанию, установлено значение `false`.
`maximizeAfterCreating`	Предоставляет возможность установить максимальный размер дизайнера отчетов. По умолчанию

© 2003-2026 Stimulsoft

	свойство имеет значение `false`.
`showLocalization`	Предоставляет возможность отображать или не отображать элемент управления локализацией в дизайнере отчете. По умолчанию свойство имеет значение `true`.
`allowChangeWindowTitle`	Разрешает использование заголовка окна браузера для отображения имени файла редактируемого отчета. По умолчанию свойство имеет значение `true`.
`showPropertiesGrid`	Включает отображение панели свойств в дизайнере отчетов. По умолчанию свойство имеет значение `true`.
`showReportTree`	Включает отображение дерева компонентов отчета. По умолчанию свойство имеет значение `true`.
`propertiesGridPosition`	Предоставляет возможность определить позицию панели свойств. Могут использоваться следующие значения: • `StiPropertiesGridPosition::Left` – панель свойств будет отображаться слева (значение по умолчанию); • `StiPropertiesGridPosition::Right` – панель свойств будет отображаться справа.
`showSystemFonts`	Предоставляет возможность отображать или не отображать системные шрифты в списке шрифтов. По умолчанию свойство имеет значение `true`, т.е. системные шрифты отображаются

© 2003-2026 Stimulsoft

	в списке шрифтов.
datePickerFirstDayOfWeek	Предоставляет возможность установить первый день недели для инструмента **Выбор даты (Date picker)**. • StiFirstDayOfWeek::Auto - Будет установлен понедельник или воскресенье как первый день недели, в зависимости от культуры браузера (значение по умолчанию); • StiFirstDayOfWeek::Monday - Понедельник будет установлен как первый день недели; • StiFirstDayOfWeek::Sunday - Воскресенье будет установлен как первый день недели.
formatForDateControls	Предоставляет возможность переопределить формат для элементов управления дат. По умолчанию, значение для текущей опции не задан и формат для дат определяется из локали браузера.
undoMaxLevel	Устанавливает максимальную глубину отмены изменений отчета при его редактировании. Влияет на потребление памяти. По умолчанию установлено 6 изменений.
wizardTypeRunningAfterLoad	Предоставляет возможность вызывать мастер создания отчетов после запуска дизайнера отчетов. Может принимать одно из указанных ниже значений перечисления:

© 2003-2026 Stimulsoft

	• `StiWizardType::None` - дизайнер отчетов будет запущен без вызова мастера создания отчета (значение по умолчанию); • `StiWizardType::StandardReport` - дизайнер отчетов будет запущен с вызовом стандартного мастера создания отчета; • `StiWizardType::MasterDetailReport` - дизайнер отчетов будет запущен с вызовом master-detail мастера создания отчета; • `StiWizardType::LabelReport` - дизайнер отчетов будет запущен с вызовом мастера создания отчета с этикетками; • `StiWizardType::InvoicesReport` - дизайнер отчетов будет запущен с вызовом мастера создания инвойса; • `StiWizardType::OrdersReport` - дизайнер отчетов будет запущен с вызовом мастера создания ордера; • `StiWizardType::QuotationReport` - дизайнер отчетов будет запущен с вызовом мастера создания квоты.
`allowWordWrapTextEditors`	Включает или выключает перенос строк в редакторах текста в дизайнере. По умолчанию свойство имеет значение `true`.
`allowLoadingCustomFontsToClientSide`	Включает или выключает передачу пользовательских шрифтов на сторону клиента. По умолчанию

© 2003-2026 Stimulsoft

	свойство имеет значение `false`.
`enableShortCutKeys`	Включает или выключает обработку сочетаний клавиш. По умолчанию свойство имеет значение `true`.
`defaultRibbonType`	Предоставляет возможность установить используемый по умолчанию стиль панели инструментов дизайнера. Может принимать одно из указанных ниже значений перечисления: • `StiDesignerRibbonType::Classic` - Будет установлен классический стиль (значение по умолчанию); • `StiDesignerRibbonType::SingleLine` – Будет установлен компактный однострочный стиль.
`allowPropagationEvents`	Предоставляет возможность разрешить распространение событий нажатия клавиш, когда дизайнер отчетов не в фокусе. По умолчанию свойство имеет значение `true`.
`propertiesPanelViewMode`	Предоставляет возможность закрепить или открепить панель свойств, словарь отчета, дерево отчета. Может принимать одно из указанных ниже значений перечисления: › `Pinned` - панели закреплены (значение по умолчангию); › `Unpinned` - панели откреплены.

Панель инструментов (Toolbar)

© 2003-2026 Stimulsoft

Наименование	Описание
visible	Включает отображение панели инструментов в дизайнере отчетов. По умолчанию свойство имеет значение true.
showPreviewButton	Включает или выключает отображение кнопки **Просмотр (Preview)** на панели инструментов дизайнера. По умолчанию свойство имеет значение true.
showSaveButton	Включает отображение кнопки **Сохранить (Save)** на панели инструментов дизайнера. По умолчанию свойство имеет значение false.
showAboutButton	Включает отображение кнопки **О программе (About)** на панели инструментов дизайнера. По умолчанию свойство имеет значение false.
showFileMenu	Включает отображение главного меню дизайнера отчетов. По умолчанию свойство имеет значение true.
showFileMenuNew	Включает отображение пункта главного меню **Создать (New)**. По умолчанию свойство имеет значение true.
showFileMenuOpen	Включает отображение пункта главного меню **Открыть (Open)**. По умолчанию свойство имеет значение true.
showFileMenuSave	Включает отображение пункта главного меню **Сохранить (Save)**. По умолчанию свойство имеет

© 2003-2026 Stimulsoft

	значение `true`.
`showFileMenuSaveAs`	Включает отображение пункта главного меню **Сохранить как (Save As)**. По умолчанию свойство имеет значение `true`.
`showFileMenuClose`	Включает отображение пункта главного меню **Закрыть (Close)**. По умолчанию свойство имеет значение `true`.
`showFileMenuExit`	Включает отображение пункта главного меню **Выход (Exit)**. По умолчанию свойство имеет значение `false`.
`showFileMenuReportSetup`	Включает отображение пункта главного меню **Свойства отчета (Report Setup)**. По умолчанию свойство имеет значение `true`.
`showFileMenuOptions`	Включает отображение пункта главного меню **Параметры (Options)**. По умолчанию свойство имеет значение `true`.
`showFileMenuInfo`	Включает отображение пункта главного меню **Сведения (Info)**. По умолчанию свойство имеет значение `true`.
`showFileMenuAbout`	Включает отображение пункта главного меню **О программе (About)**. По умолчанию свойство имеет значение `true`.
`showFileMenuNewReport`	Включает или выключает отображение пункта главного меню **Новая страница (New Page)**. По умолчанию свойство имеет значение `true`.
`showFileMenuNewDashboard`	Включает или выключает отображение пункта главного

© 2003-2026 Stimulsoft

	меню **Новая панель индикаторов (New Dashboard)**. По умолчанию свойство имеет значение `true`.
`showSetupToolboxButton`	Включает или выключает отображение кнопки вызова окна настроек боковой панели компонентов отчета. По умолчанию свойство имеет значение `true`.
`showNewPageButton`	Включает или выключает отображение кнопки **Новая страница (New Page)** на панели инструментов. По умолчанию свойство имеет значение `true`.
`showNewDashboardButton`	Включает или выключает отображение кнопки **Новая панель индикаторов (New Dashboard)** на панели инструментов. По умолчанию свойство имеет значение `true`.

Секции (Bands)

Наименование	Описание
`showReportTitleBand`	Включает отображение секции **Заголовок отчета (Report Title)** в списке меню **Секции**. По умолчанию свойство имеет значение `true`.
`showReportSummaryBand`	Включает отображение секции **Итог отчета (Report Summary)** в списке меню **Секции**. По умолчанию свойство имеет значение `true`.
`showPageHeaderBand`	Включает отображение секции

© 2003-2026 Stimulsoft

	Заголовок страницы (Page Header) в списке меню **Секции**. По умолчанию свойство имеет значение `true`.
`showPageFooterBand`	Включает отображение секции **Итог страницы (Page Footer)** в списке меню **Секции**. По умолчанию свойство имеет значение `true`.
`showGroupHeaderBand`	Включает отображение секции **Заголовок группы (Group Header)** в списке меню **Секции**. По умолчанию свойство имеет значение `true`.
`showGroupFooterBand`	Включает отображение секции **Итог группы (Group Footer)** в списке меню **Секции**. По умолчанию свойство имеет значение `true`.
`showHeaderBand`	Включает отображение секции **Заголовок данных (Header)** в списке меню **Секции**. По умолчанию свойство имеет значение `true`.
`showFooterBand`	Включает отображение секции **Итог данных (Footer)** в списке меню **Секции**. По умолчанию свойство имеет значение `true`.
`showColumnHeaderBand`	Включает отображение секции **Заголовок колонки (Column Header)** в списке меню **Секции**. По умолчанию свойство имеет значение `true`.
`showColumnFooterBand`	Включает отображение секции **Итог колонки (Column Footer)** в списке меню **Секции**. По

© 2003-2026 Stimulsoft

	умолчанию свойство имеет значение `true`.
`showDataBand`	Включает отображение секции **Данные (Data)** в списке меню **Секции**. По умолчанию свойство имеет значение `true`.
`showHierarchicalBand`	Включает отображение секции **Дерево (Hierarchical)** в списке меню **Секции**. По умолчанию свойство имеет значение `true`.
`showChildBand`	Включает отображение секции **Подчиненный (Child)** в списке меню **Секции**. По умолчанию свойство имеет значение `true`.
`showEmptyBand`	Включает отображение секции **Пустые данные (Empty)** в списке меню **Секции**. По умолчанию свойство имеет значение `true`.
`showOverlayBand`	Включает отображение секции **Водяной знак (Overlay)** в списке меню **Секции**. По умолчанию свойство имеет значение `true`.
`showTable`	Включает отображение компонента **Таблица (Table)** в списке меню **Секции**. По умолчанию свойство имеет значение `true`.
`showTableOfContents`	Включает или выключает отображение компонента **Оглавление (Table of Contents)** в списке меню **Секции**. По умолчанию свойство имеет значение `true`.

Кросс секции (Cross-Bands)

© 2003-2026 Stimulsoft

Наименование	Описание
showCrossTab	Включает отображение компонента **Кросс-таблица (Cross-Tab)** в списке меню **Кросс**. По умолчанию свойство имеет значение true.
showCrossGroupHeaderBand	Включает отображение секции **Кросс-заголовок группы (Cross Group Header)** в списке меню **Кросс**. По умолчанию свойство имеет значение true.
showCrossGroupFooterBand	Включает отображение секции **Кросс-итог группы (Cross Group Footer)** в списке меню **Кросс**. По умолчанию свойство имеет значение true.
showCrossHeaderBand	Включает отображение секции **Кросс-заголовок данных (Cross Header)** в списке меню **Кросс**. По умолчанию свойство имеет значение true.
showCrossFooterBand	Включает отображение секции **Кросс-итог данных (Cross Footer)** в списке меню **Кросс**. По умолчанию свойство имеет значение true.
showCrossDataBand	Включает отображение секции **Кросс- данные (Cross Data)**в списке меню **Кросс**. По умолчанию свойство имеет значение true.

Компоненты (Components)

© 2003-2026 Stimulsoft

Наименование	Описание
showText	Включает отображение компонента **Текст (Text)** в списке меню **Компоненты**. По умолчанию свойство имеет значение true.
showTextInCells	Включает отображение компонента **Текст в ячейках (Text in Cells)** в списке меню **Компоненты**. По умолчанию свойство имеет значение true.
showRichText	Включает отображение компонента **Rich текст (Rich Text)** в списке меню **Компоненты**. По умолчанию свойство имеет значение false.
showImage	Включает отображение компонента **Картинка (Image)** в списке меню **Компоненты**. По умолчанию свойство имеет значение true.
showBarCode	Включает отображение компонента **Штрих-код (Bar Code)** на инструментарии или вкладке **Вставить** в дизайнере. По умолчанию свойство имеет значение true.
showShape	Включает отображение компонента **Геометрия (Shape)** на инструментарии или вкладке **Вставить** в дизайнере. По умолчанию свойство имеет значение true.
showPanel	Включает отображение компонента **Панель (Panel)** в

© 2003-2026 Stimulsoft

	списке меню **Компоненты**. По умолчанию свойство имеет значение `true`.
`showClone`	Включает отображение компонента **Клон (Clone)** в списке меню **Компоненты**. По умолчанию свойство имеет значение `false`.
`showCheckBox`	Включает отображение компонента **Флажок (Check Box)** в списке меню **Компоненты**. По умолчанию свойство имеет значение `true`.
`showSubReport`	Включает отображение компонента **Вложенный отчет (Sub Report)** в списке меню **Компоненты**. По умолчанию свойство имеет значение `true`.
`showZipCode`	Включает отображение компонента **Почтовый индекс (Zip Code)** в списке меню **Компоненты**. По умолчанию свойство имеет значение `false`.
`showChart`	Включает отображение компонента **Диаграмма (Chart)** на инструментарии или вкладке **Вставить** в дизайнере. Действует сразу на все типы диаграмм. По умолчанию свойство имеет значение `true`.
`showGauge`	Включает или выключает отображение компонента **Измерительный прибор (Gauge)** на инструментарии или вкладке **Вставить** в дизайнере. По умолчанию свойство имеет значение `true`.

© 2003-2026 Stimulsoft

showSparkline	Включает отображение компонента **Спарклайн (Sparkline)** в списке меню **Компоненты**. По умолчанию свойство имеет значение true.
showMathFormula	Включает или выключает отображение компонента **Математическая формула (Math Formula)** в списке меню **Компоненты**. По умолчанию свойство имеет значение false.
showMap	Включает или выключает отображение компонента **Карта (Map)** на инструментарии или вкладке **Вставить** в дизайнере. По умолчанию свойство имеет значение true.
showElectronicSignature	Включает или выключает отображение компонента **Электронная подпись (Electronic Signature)** на инструментарии или вкладке Вставить в дизайнере. По умолчанию свойство имеет значение true.
showPdfDigitalSignature	Включает или выключает отображение компонента **Цифровая подпись PDF (PDF Digital Signature)** на инструментарии или вкладке Вставить в дизайнере. По умолчанию свойство имеет значение true.
showHorizontalLinePrimitive	Включает или выключает отображение компонента **Горизонтальная линия (Horizontal Line)** на

© 2003-2026 Stimulsoft

	инструментарии или вкладке Вставить в дизайнере. По умолчанию свойство имеет значение `true`.
`showVerticalLinePrimitive`	Включает или выключает отображение компонента **Вертикальная линия (Vertical Line)** на инструментарии или вкладке Вставить в дизайнере. По умолчанию свойство имеет значение `true`.
`showRectanglePrimitive`	Включает или выключает отображение компонента **Прямоугольник (Rectangle)** на инструментарии или вкладке Вставить в дизайнере. По умолчанию свойство имеет значение `true`.
`showRoundedRectanglePrimitive`	Включает или выключает отображение компонента **Закругленный прямоугольник (Rounded Rectangle)** на инструментарии или вкладке Вставить в дизайнере. По умолчанию свойство имеет значение `true`.

Элементы дашборда (Dashboard Elements)

Наименование	Описание
`showTableElement`	Включает отображение элемента панели индикаторов **Таблица (Table)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию

© 2003-2026 Stimulsoft

	свойство имеет значение `true`.
`showCardsElement`	Включает отображение элемента панели индикаторов **Карточки (Cards)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `true`.
`showChartElement`	Включает отображение элемента панели индикаторов **Диаграмма (Chart)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `true`.
`showGaugeElement`	Включает отображение элемента панели индикаторов **Измерительный прибор (Gauge)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `true`.
`showPivotTableElement`	Включает отображение элемента панели индикаторов **Сводная таблица (Pivot)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `true`.
`showIndicatorElement`	Включает отображение элемента панели индикаторов **Индикатор (Indicator)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `true`.
`showProgressElement`	Включает отображение элемента панели индикаторов **Прогресс (Progress)** на инструментарии или вкладке **Вставить (Insert)** в

© 2003-2026 Stimulsoft

	дизайнере. По умолчанию свойство имеет значение `true`.
`showRegionMapElement`	Включает отображение элемента панели индикаторов **Региональная карта (Region Map)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `true`.
`showOnlineMapElement`	Включает отображение элемента панели индикаторов **Онлайн карта (Online Map)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `true`.
`showImageElement`	Включает отображение элемента панели индикаторов **Изображение (Image)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `true`.
`showWebContentElement`	Включает отображение элемента панели индикаторов **Веб-контент (Web Content)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `true`.
`showTextElement`	Включает отображение элемента панели индикаторов **Текст (Text)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `true`.
`showPanelElement`	Включает отображение элемента панели индикаторов **Панель (Panel)** на инструментарии или

© 2003-2026 Stimulsoft

	вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `true`.
`showShapeElement`	Включает отображение элемента панели индикаторов **Геометрия (Shape)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `true`.
`showButtonElement`	Включает отображение элемента панели индикаторов **Кнопка (Button)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `true`.
`showListBoxElement`	Включает отображение элемента панели индикаторов **Список (ListBox)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `true`.
`showComboBoxElement`	Включает отображение элемента панели индикаторов **Выпадающий список (ComboBox)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `true`.
`showTreeViewElement`	Включает отображение элемента панели индикаторов **Иерархический список (TreeView)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `true`.
`showTreeViewBoxElement`	Включает отображение элемента панели индикаторов

© 2003-2026 Stimulsoft

	Выпадающий иерархический список (TreeViewBox) на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `true`.
`showDatePickerElement`	Включает отображение элемента панели индикаторов **Выбор даты (Date Picker)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `true`.
`showNumberBoxElement`	Включает отображение элемента панели индикаторов **Числовое поле (Number Box)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `true`.

Словарь данных (Dictionary)

Наименование	Описание
`showAdaptersInNewConnectionForm`	Включает отображение категории **Объекты (Object)** в окне создания нового соединения. По умолчанию свойство имеет значение `true`.
`showDictionary`	Включает отображение словаря данных отчета. По умолчанию свойство имеет значение `true`.
`newReportDictionary`	Предоставляет возможность создавать новый словарь данных или объединить с существующим, при создании нового отчета в дизайнере. Может принимать одно

© 2003-2026 Stimulsoft

	из указанных ниже значений: • `StiNewReportDictionary::Auto` - определяет режим создания или объединения словаря данных из сохраненного значения в файлах `cookies` (значение по умолчанию); • `StiNewReportDictionary::DictionaryNew` - устанавливает режим создания нового словаря данных, при создании нового отчета; • `StiNewReportDictionary::DictionaryMerge` - устанавливает режим объединения существующего словаря данных с новым, при создании нового отчета в дизайнере.
`useAliases`	Предоставляет возможность использовать псевдонимы в словаре данных. Может принимать одно из указанных ниже значений: • `StiUseAliases::Auto` - определяет режим использования псевдонимов из сохраненного значения в файлах cookies (значение по умолчанию); • `StiUseAliases::True` - устанавливает режим использования псевдонимов в словаре данных; • `StiUseAliases::False` - выключает режим использования псевдонимов в словаре данных.

© 2003-2026 Stimulsoft

showDictionaryContextMenuProperties	Устанавливает видимость пункта **Свойства (Properties)** в контекстном меню словаря данных. По умолчанию свойство имеет значение `true`.
showDictionaryActions	Устанавливает отображение меню **Действия (Actions)** на панели инструментов словаря данных. По умолчанию свойство имеет значение **true**.
dataSourcesPermissions	Устанавливает доступные действия над источниками данных отчета. Может принимать одно или несколько значений из перечисления `StiDesignerPermissions`.
dataConnectionsPermissions	Устанавливает доступные действия над подключениями к данным отчета. Может принимать одно или несколько значений из перечисления `StiDesignerPermissions`.
dataColumnsPermissions	Устанавливает доступные действия над колонками данных отчета. Может принимать одно или несколько значений из перечисления `StiDesignerPermissions`.
dataRelationsPermissions	Устанавливает доступные действия над связями данных отчета. Может принимать одно или несколько значений из перечисления `StiDesignerPermissions`.
businessObjectsPermissions	Устанавливает доступные действия над бизнес-объектами отчета. Может принимать одно или несколько значений из

© 2003-2026 Stimulsoft

	перечисления StiDesignerPermissions.
variablesPermissions	Устанавливает доступные действия над переменными отчета. Может принимать одно или несколько значений из перечисления StiDesignerPermissions.
resourcesPermissions	Устанавливает доступные действия над ресурсами в словаре данных отчета. Может принимать одно или несколько значений из перечисления StiDesignerPermissions.
dataTransformations	Устанавливает доступные действия над преобразованием данных. Может принимать одно или несколько значений из перечисления StiDesignerPermissions.
userFunctions	Устанавливает доступные действия над пользовательскими функциями. Может принимать одно или несколько значений из перечисления StiDesignerPermissions.

В таблице ниже приведены все доступные значения для перечисления StiDesignerPermissions, которые можно установить для элементов словаря отчета.

Значение	Описание
StiDesignerPermissions::None	Запрещает любое действие над элементом словаря данных.
StiDesignerPermissions::Create	Разрешает создавать определенный элемент словаря

© 2003-2026 Stimulsoft

	данных.
StiDesignerPermissions::Delete	Разрешает удалять определенный элемент словаря данных.
StiDesignerPermissions::Modify	Разрешает редактировать определенный элемент словаря данных.
StiDesignerPermissions::View	Разрешает просматривать определенный элемент словаря данных.
StiDesignerPermissions::ModifyView	Разрешает редактировать и просматривать определенный элемент словаря данных.
StiDesignerPermissions::All	Разрешает любые действия над элементом словаря данных (установлено по умолчанию для всех свойств, использующих данное перечисление).

Предусмотрена настройка встроенного компонента `StiViewer`, используемого для предварительного просмотра отчета. Для доступа ко всем его настройкам предназначено свойство `viewerOptions`, которое представляет собой объект опций вьювера. Все его свойства описаны в разделе Настройки вьювера. Пример установки режима отображения отчета и отключения некоторых форматов экспорта:

designer.php

```
<?php
   use Stimulsoft\Designer\StiDesigner;
   use Stimulsoft\Viewer\Enums\StiHtmlExportMode;

   $designer = new StiDesigner();
   $designer->options->viewerOptions->appearance->reportDisplayMode =
   StiHtmlExportMode::FromReport;
   $designer->options->viewerOptions->exports->ShowExportToWord = false;
   $designer->options->viewerOptions->exports->showExportToCsv = false;
?>
```

© 2003-2026 Stimulsoft

10 Reports and Dashboards for Python

Продукты Stimulsoft Reports.PYTHON и Stimulsoft Dashboards.PYTHON представляют собой комбинацию JavaScript и Python скриптов, а также весь необходимый функционал для их взаимодействия. Построение и экспортирование отчета выполняется при помощи JavaScript ядра на стороне клиента в окне Web браузера, либо на стороне сервера с использованием универсальной платформы Node.js. Серверная сторона Python содержит все необходимое для работы с файлами отчетов и связи с различными SQL источниками данных. Связь клиента с сервером осуществляется посредством AJAX запросов, передающих и получающих JSON данные в определенном формате. Для удобства использования продукта разработаны специальные события и функции как на стороне JavaScript клиента, включая платформу Node.js, так и на стороне Python сервера.

В состав продукта входит сам генератор отчетов и ядро анализа данных, а так же компоненты вьювер и дизайнер отчетов.

10.1 Генератор отчетов

Генератор отчетов Stimulsoft Reports.PYTHON позволяет загружать, выполнять построение и экспортирование отчета в различные форматы без развертывания вьювера и дизайнера. Это позволяет избавиться от необходимости загрузки на сторону клиента большого количества скриптов и увеличивает скорость и простоту работы.

Построение и экспортирование отчета выполняется при помощи JavaScript ядра на стороне клиента в окне Web браузера, либо на стороне сервера с использованием универсальной платформы Node.js. Серверная сторона Python содержит все необходимое для работы с файлами отчетов и связи с различными SQL источниками данных. Связь клиента с сервером осуществляется посредством AJAX запросов, передающих и получающих JSON данные в определенном формате. Для удобства использования продукта разработаны специальные события и функции как на стороне JavaScript клиента, так и на стороне Python сервера.

Аналитические панели

Аналитические панели Stimulsoft Dashboards.PYTHON позволяют загружать, выполнять анализ данных и экспортирование дашборда в различные форматы без развертывания вьювера и дизайнера.

© 2003-2026 Stimulsoft

Весь анализ данных, за исключением определенных SQL операций, выполняется при помощи JavaScript анализатора данных на стороне клиента в окне Web браузера, либо на стороне сервера с использованием универсальной платформы Node.js. Серверная сторона Python содержит все необходимое для работы с файлами дашбордов и связи с различными SQL источниками данных. Связь клиента с сервером осуществляется точно таким же способом, как и в генераторе отчетов, применяются те же события и функции. Серверная сторона Python является универсальной для дашбордов и отчетов, применяются те же события и функции.

- Использование
- Подключение файлов данных
- Уменьшение времени загрузки скриптов
- Подключение SQL адаптеров данных
- Активация лицензии
- Работа с переменными отчета
- Загрузка и сохранение отчета
- Подключение пользовательских шрифтов
- Построение отчета
- Печать отчета из кода
- Построение отчета на стороне сервера
- Экспорт отчета из кода
- Обработчик событий
- События генератора отчетов
- Работа с данными

10.1.1 Использование

Для использования генератора отчетов достаточно установить пакет stimulsoft-reports либо stimulsoft-dashboards при помощи установщика пакетов, выполнив одну из указанных ниже команд:

console

```
python -m pip install stimulsoft-reports
```

© 2003-2026 Stimulsoft

console

```
python -m pip install stimulsoft-dashboards
```

Информация

Для Linux систем может потребоваться установка дополнительного пакета для работы с ODBC, это можно сделать, например выполнив следующую команду:

```
sudo apt-get install unixodbc.
```

Будет установлена последняя доступная версия генератора отчетов в текущее рабочее пространство, после этого можно использовать классы и функции по работе с отчетами.

По умолчанию устанавливается минимальный набор драйверов данных. Необходимые драйверы можно дополнительно установить вручную, либо воспользоваться командой для установки всех доступных драйверов:

console

```
python -m pip install stimulsoft-reports[ext]
```

console

```
python -m pip install stimulsoft-dashboards[ext]
```

© 2003-2026 Stimulsoft

Информация

Для примеров кода использован **Flask** фреймворк, как один из самых популярных и простых для понимания кода. Допускается использовать любой Web фреймворк, т.к. все классы и функции для работы с генератором отчетов являются универсальными.

Для работы с генератором отчетов предназначен класс `StiReport`. При помощи данного класса, можно загрузить шаблон либо документ отчета, выполнить построение и экспортирование, обрабатывать запросы и управлять событиями генератора отчетов.

Например, требуется загрузить отчет из файла и выполнить его построение, после чего вывести сообщение в окне браузера:

app.py

```
from flask import Flask, render_template, url_for, request
from stimulsoft_reports.report import StiReport

app = Flask(__name__)

@app.route('/report', methods = ['GET', 'POST'])
def report():
    report = StiReport()
    report.onAfterRender += 'afterRender'

    if report.processRequest(request):
        return report.getFrameworkResponse()

    report.loadFile(url_for('static', filename='reports/SimpleList.mrt'))
    report.render()

    js = report.javascript.getHtml()
    html = report.getHtml()
    return render_template('report.html', reportJavaScript = js,
reportHtml = html)
```

© 2003-2026 Stimulsoft

report.html

```
<!DOCTYPE html>
<html>

<head>
   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
   <title>Render and Export a Report</title>

   {{ reportJavaScript|safe }}

    <script type="text/javascript">
       function afterRender() {
           alert('Done!');
       }
   </script>
</head>

<body>
   {{ reportHtml|safe }}
</body>

</html>
```

Полный код примера доступен на GitHub.

В этом примере последовательно выполняются следующие действия:

- создается экземпляр объекта `StiReport`;
- добавляются необходимые обработчики событий;
- выполняется обработка текущего запроса;
- выполняется загрузка шаблона отчета из файла `SipmleList.mrt`, расположенного в директории статических файлов;
- вызывается команда построения отчета;
- в шаблоне HTML файла добавляется код JavaScript события и выполняется вывод необходимого JavaScript и HTML кода компонента.

Метод `report.processRequest(request)` выполняет обработку текущего запроса, и возвращает результат работы. В случае успешной обработки, вместо шаблона страницы необходимо вернуть результат его выполнения. Более подробно про это рассказано в разделе Обработчик событий.

Метод `report.javascript.getHtml()` выполняет генерацию кода для подключения необходимых скриптов компонента. Метод `report.getHtml()` выполняет генерацию JavaScript и HTML кода самого компонента.

© 2003-2026 Stimulsoft

Сгенерированный HTML код передается в качестве параметров в HTML шаблон `report.html`, в котором выводится в необходимых местах.

Наши продукты Stimulsoft Reports.PYTHON и Stimulsoft Dashboards.PYTHON не имеют нативного ядра генератора отчетов на Python, построение и экспорт отчета выполняется при помощи JavaScript кода на стороне клиента, либо на стороне сервера с использованием платформы Node.js. Поэтому, при использовании Python кода для работы с компонентами, необходимо вызвать один из специальных методов, который добавит на Web страницу соответствующий JavaScript код для выполнения всех необходимых действий, и HTML код визуальной части компонента.

Наименование	Описание
`getHtml(mode = StiHtmlMode.HtmlScripts)`	Возвращает JavaScript и HTML код, необходимый для работы компонента, со всеми необходимыми действиями над отчетом. Параметр `mode` позволяет установить варианты возвращаемого кода: • `StiHtmlMode.Scripts` – только необходимый JavaScript код для вставки в блок `<script></script>` на HTML странице; • `StiHtmlMode.HtmlScripts` – необходимый JavaScript и HTML код, для вставки в HTML элемент на странице; • `StiHtmlMode.HtmlPage` – полностью подготовленная HTML страница. **Примечание:** у элементов компонента, например объекта `report.javascript`, метод `getHtml()` не принимает параметров, поскольку имеет только один вариант вывода кода.

© 2003-2026 Stimulsoft

`renderHtml(elementId: str = None)`	Выводит JavaScript и HTML код, необходимый для работы компонента, со всеми необходимыми действиями над отчетом. Параметр `elementId` позволяет указать ID элемента HTML страницы, внутри которого будет выведен компонент. По умолчанию, вывод осуществляется в текущем месте страницы.
`printHtml()`	Выводит полностью подготовленную HTML страницу со всеми необходимыми скриптами для работы компонента. Текущий HTML шаблон страницы полностью игнорируется. Данный режим отлично подходит для полноэкранного просмотра либо редактирования отчета.

Таким образом, указанные выше методы позволяют отобразить компонент различными способами в зависимости от требований. Пример упрощённого отображения компонента без использования HTML шаблона страницы:

app.py

```
from flask import Flask, url_for, request
from stimulsoft_reports.report import StiReport

app = Flask(__name__)

@app.route('/report', methods = ['GET', 'POST'])
def report():
    report = StiReport()
    report.onAfterRender += "alert('Done!');"
    if report.processRequest(request):
        return report.getFrameworkResponse()

    report.loadFile(url_for('static', filename='reports/SimpleList.mrt'))
    report.render()

    return report.getFrameworkResponse()
```

© 2003-2026 Stimulsoft

Информация

При использовании платформы Node.js для построения отчета на стороне Python сервера, указанные методы будут вызваны автоматически внутри обработчика, их явное использование не требуется.

Поддержка фреймворков

В данный момент реализована встроенная поддержка для трех популярных фреймворков Python - это **Flask**, **Django** и **Tornado**. Также доступны универсальные функции для работы в любых других фреймворках. Для примеров кода использован **Flask** фреймворк, как один из самых популярных и простых для понимания кода, развертывание продукта для него было рассмотрено выше. Ниже представлены примеры этого же кода для остальных фреймворков. Все они очень похожи и отличаются только функциями, используемыми для конкретного фреймворка.

Django

app.py

```
from django.shortcuts import render
from django.templatetags.static import static
from stimulsoft_reports.report import StiReport

def report(request):
    report = StiReport()
    report.onAfterRender += 'afterRender'

    if report.processRequest(request):
        return report.getFrameworkResponse()

    report.loadFile(static('reports/SimpleList.mrt'))
    report.render()

    js = report.javascript.getHtml()
    html = report.getHtml()
    return render(request, 'report.html', {'reportJavaScript': js,
'reportHtml': html})
```

© 2003-2026 Stimulsoft

report.html

```
<!DOCTYPE html>
<html>

<head>
   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
   <title>Render and Export a Report</title>

   {{ reportJavaScript|safe }}

    <script type="text/javascript">
       function afterRender() {
           alert('Done!');
       }
   </script>
</head>

<body>
   {{ reportHtml|safe }}
</body>

</html>
```

Tornado

© 2003-2026 Stimulsoft

app.py

```
import asyncio, os
from tornado.web import Application, RequestHandler, url
from stimulsoft_reports.report import StiReport

class ReportHandler(RequestHandler):
    def get(self):
        report = StiReport()
        report.onAfterRender += 'afterRender'

        if report.processRequest(request):
            return report.getFrameworkResponse(self)

        report.loadFile(self.static_url('reports/SimpleList.mrt'))
        report.render()

        js = report.javascript.getHtml()
        html = report.getHtml()
        self.render('report.html', reportJavaScript = js, reportHtml =
html)

    def post(self):
        handler = StiHandler()
        if handler.processRequest(self.request):
            return handler.getFrameworkResponse(self)
```

report.html

```
<!DOCTYPE html>
<html>

<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>Render and Export a Report</title>

  {% raw reportJavaScript %}
   <script type="text/javascript">
     function afterRender() {
         alert('Done!');
     }
  </script>
</head>

<body>
  {% raw reportHtml %}
</body>

</html>
```

© 2003-2026 Stimulsoft

Универсальные функции

При использовании любого другого фреймворка, вместо метода `getFrameworkResponse()` необходимо использовать метод `getResponse()`, который вернёт объект `StiResponse`, содержащий все необходимые параметры для формирования ответа:

app.py

```
from stimulsoft_reports.report import StiReport

def report():
    report = StiReport()
    report.onAfterRender += 'afterRender'
    query = 'query string'
    body = 'post data'
    if report.processRequest(None, query, body):
        response = report.getResponse()
        data = response.data
        contentType = response.contentType
        mimetype = response.mimetype

    report.loadFile(url_for('static', filename='reports/SimpleList.mrt'))
    report.render()

    js = report.javascript.getHtml()
    html = report.getHtml()
```

Информация

В большинстве случаев для работы с продуктом достаточно использовать только Python код, который обеспечивает взаимодействие со всеми основными возможностями. Для более детальной настройки продукта и использования всех возможностей JS генератора, необходимо использовать JavaScript код. Вариант развертывания продукта при помощи только JavaScript кода описан в разделе Reports and Dashboards for JS, в этом случае использование Python кода требуется только для подключения адаптеров данных.

Различные варианты развертывания и оптимизации рассмотрены в разделе Уменьшение времени загрузки скриптов.

© 2003-2026 Stimulsoft

10.1.2 Уменьшение времени загрузки скриптов

Из-за внушительной функциональности продукта, скрипты имеют достаточно большой размер. При первой загрузке Web приложения, либо при отключенном кэшировании браузера, загрузка может занимать некоторое время, особенно при низкоскоростном интернет-соединении. Мы предлагаем два варианта решения этой проблемы: использовать упакованные скрипты, либо использовать частичный функционал, и загружать только то, что требуется. Допускается одновременное использование указанных вариантов.

Упакованные скрипты

Упакованные скрипты имеют такую же структуру, что и обычные скрипты, но имеют окончание `*.pack.js` в имени файла. Такие скрипты содержат блок упакованных данных в виде JavaScript переменной, и компактный распаковщик. При загрузке всех скриптов, распаковщик автоматически распаковывает все загруженные данные, и запускает подготовленный скрипт на выполнение. Распаковка занимает некоторое время, но при определенных обстоятельствах - например, при медленном интернет-соединении - это время намного меньше, чем скорость загрузки обычных скриптов.

Для использования упакованных скриптов все, что вам понадобится сделать - это установить свойство `javascript.usePacked` у объекта отчета в значение `True`, например:

app.py

```
report = StiReport()
report.javascript.usePacked = True
```

Частичная загрузка скриптов

При развертывании генератора отчетов, по умолчанию загружается только один файл со скриптами `stimulsoft.reports.js`, который содержит в себе весь функционал по построению и экспортированию отчетов. Если для генерации отчетов вам требуются только некоторые из возможностей, предусмотрена загрузка только необходимых частей генератора, содержащих определенный набор возможностей. Например, если в ваших отчетах не используются карты, то их можно не загружать. Это ускорит загрузку Web

© 2003-2026 Stimulsoft

проекта и уменьшит потребление памяти браузером.

Информация

Данная возможность реализована только для движка генератора отчетов, вьювер и дизайнер нельзя разделить на части, их скрипты будут загружаться целиком одним блоком.

Для использования частичной загрузки скриптов, достаточно установить нужные опции для свойства `javascript` у объекта отчета:

app.py

```
from stimulsoft_reports.report import StiReport

report = StiReport()

report.javascript.reportsChart = True
report.javascript.reportsExport = True
report.javascript.reportsImportXlsx = True
report.javascript.reportsMaps = False
report.javascript.blocklyEditor = False
```

Каждая опция управляет загрузкой скрипта, содержащего определенный функционал. В данной таблице представлен весь набор скриптов, которые можно загрузить раздельно:

Наименование	Описание
`javascript.reportsExport`	Содержит алгоритмы для экспортирования построенного отчета в различные форматы - PDF, HTML, Excel, RichText и другие.
`javascript.reportsChart`	Содержит компоненты для работы со всеми типами диаграмм в отчете.
`javascript.reportsMaps`	Содержит компоненты для работы

© 2003-2026 Stimulsoft

	с региональными и онлайн картами.
`javascript.blocklyEditor`	Содержит визуальный редактор Blockly для создания скриптов событий в отчете. Сам обработчик событий встроен в движок отчетов.
`javascript.reportsImportXlsx`	Содержит алгоритмы для работы с Excel источниками данных.

Информация

Компоненты вьювер отчетов и дизайнер отчетов также имеют свойство `javascript`, при помощи которого можно управлять настройкой скриптов указанным выше способом.

10.1.3 Активация лицензии

После приобретения продукта вам необходимо активировать лицензию для используемых компонентов. Предусмотрено несколько способов подключения лицензионного ключа.

Активация при помощи строки кода

Для активации при помощи строки, достаточно скопировать зашифрованный текст лицензии из личного кабинета на сайте, и зарегистрировать его при помощи статической функции `setKey()`, находящейся в классе `StiLicense`:

app.py

```
from stimulsoft_reports import StiLicense

StiLicense.setKey('Your activation code...')
```

Полный код примера доступен на GitHub.

Активация при помощи файла

© 2003-2026 Stimulsoft

Для активации при помощи лицензионного файла, достаточно загрузить файл `license.key` из личного кабинета на сайте, и скопировать его в папку Web-проекта, например в папку `static` со статическими файлами. Затем, достаточно зарегистрировать его при помощи статической функции `setKey()`, находящейся в классе `StiLicense`:

app.py

```
from stimulsoft_reports import StiLicense

StiLicense.setFile(url_for('static', filename='license.key'))
```

Полный код примера доступен на GitHub.

Активация только генератора отчетов

В некоторых случаях может возникнуть необходимость активации генератора отчетов отдельно от остальных компонентов. В этом случае, лицензионный ключ можно зарегистрировать при помощи метода `setKey()` либо `setFile()` свойства `license` у объекта отчета:

app.py

```
from stimulsoft_reports.report import StiReport

report = StiReport()
report.license.setKey('Your activation code...')
# report.license.setFile(url_for('static', filename='license.key'))
```

Информация

Компоненты вьювер отчетов и дизайнер отчетов также имеют свойство `license`, при помощи которого можно управлять лицензионным ключом указанным выше способом.

Защита от кражи лицензионного ключа

© 2003-2026 Stimulsoft

В случае активации лицензии при помощи строки, можно добавлять ее код по определенному условию. Например, в переменной `sessionID` хранится информация о текущей сессии клиента, требуется добавлять лицензионный ключ только для авторизованных пользователей:

app.py

```
from stimulsoft_reports.report import StiReport

report = StiReport()

if sessionID != None:
    report.license.setKey('Your activation code...')
```

Также не будет лишним изменить местоположение и название файла лицензионного ключа, например:

app.py

```
from stimulsoft_reports.report import StiReport

report = StiReport()
report.license.setFile(url_for('private', filename='a15fc0ef64e6.key'))
```

10.1.4 Загрузка и сохранение отчета

Информация

Файл формата Stimulsoft MRT и MDC - это описание отчетов с разметкой XML или JSON. Вы можете использовать MRT и MDC файлы, созданные в других дизайнерах Stimulsoft.

Загрузка отчета

Отчет может храниться в виде шаблона отчета (MRT файл), либо в виде построенного отчета (MDC документ), предназначенного для последующего просмотра либо экспортирования. Для загрузки отчета при помощи Python кода можно использовать один из перечисленных ниже методов у объекта `StiReport`. Каждый из методов принимает на вход имя файла отчета либо сам отчет в виде строки:

© 2003-2026 Stimulsoft

Наименование	Описание
`loadFile(filePath: str, load: bool = False)`	Загружает шаблон отчета из MRT файла на стороне клиента, путь к которому указан в аргументах функции. Если параметр `load` установлен в значение `True`, файл отчета будет загружен на стороне сервера и передан на сторону клиента в виде упакованной `Base64` строки.
`load(data: str, fileName: str = 'Report')`	Загружает шаблон отчета из XML или JSON строки и передает его на сторону клиента в виде упакованной `Base64` строки. Параметр `fileName` устанавливает имя файла, которое будет использоваться для последующего сохранения и экспорта отчета.
`loadPacked(data: str, fileName: str = 'Report')`	Загружает и передает на сторону клиента шаблон отчета в виде упакованной `Base64` строки, заданной в параметре `data`. Параметр `fileName` устанавливает имя файла, которое будет использоваться для последующего сохранения и экспорта отчета.
`loadDocumentFile(filePath: str, load: bool = False)`	Загружает построенный отчет из MDC файла на стороне клиента, путь к которому указан в параметрах функции. Если

© 2003-2026 Stimulsoft

	параметр `load` установлен в значение `True`, файл документа будет загружен на стороне сервера и передан на сторону клиента в виде упакованной `Base64` строки.
`loadDocument(data: str, fileName: str = 'Report')`	Загружает построенный отчет из XML или JSON строки и передает его на сторону клиента в виде упакованной `Base64` строки. Параметр `fileName` устанавливает имя файла, которое будет использоваться для последующего сохранения и экспорта отчета.
`loadPackedDocument(data: str, fileName: str = 'Report')`	Загружает и передает на сторону клиента построенный отчет в виде упакованной `Base64` строки, заданной в параметре `data`. Параметр `fileName` устанавливает имя файла, которое будет использоваться для последующего сохранения и экспорта отчета.

Пример загрузки отчета из файла на стороне сервера из приватной директории и передача его на сторону клиента в виде упакованной строки для последующего построения:

app.py

```
from stimulsoft_reports.report import StiReport

report = StiReport()
report.loadFile(url_for('static', filename='reports/SimpleList.mrt'),
True)
report.render()
```

Сохранение отчета

В режиме построения отчета на стороне JavaScript клиента, генератор отчетов

© 2003-2026 Stimulsoft

Python на стороне сервера не имеет доступа к объекту отчета. В этом случае, для сохранения шаблона отчета или документа необходимо воспользоваться событиями и JavaScript функциями. Более подробно про это указано в разделе События генератора отчетов.

Пример сохранения построенного отчета в виде строки для последующего использования:

app.py

```
from stimulsoft_reports.report import StiReport

report = StiReport()
report.onAfterRender += 'afterRender'
report.loadFile(url_for('static', filename='reports/SimpleList.mrt'),
True)
report.render()
```

report.html

```
<script>
    function afterRender(args) {
        let json = args.report.saveDocumentToJsonString();
    }
</script>
```

Полный код примера доступен на GitHub.

В режиме построения отчета на стороне сервера, для сохранения отчета предназначен один из перечисленных ниже методов:

Наименование	Описание
`saveDocument(filePath: str = None)`	Сохраняет построенный отчет в виде MDC файла, путь к которому указан в аргументах функции. Если параметр `filePath` не указан, то вместо сохранения файла метод вернет отчет в виде JSON строки.
`savePackedDocument(filePath: str`	Сохраняет построенный отчет в

© 2003-2026 Stimulsoft

= None)	виде упакованного MDZ файла, путь к которому указан в аргументах функции. Если параметр `filePath` не указан, то вместо сохранения файла метод вернет отчет в виде упакованной `Base64` строки.

Например, требуется сохранить построенный отчет в виде файла на стороне сервера:

app.py

```
from stimulsoft_reports.report import StiReport
from stimulsoft_reports.report.enums import StiEngineType

report = StiReport()
report.engine = StiEngineType.SERVER_NODE_JS
report.loadFile(url_for('static', filename='reports/SimpleList.mrt'),
True)
report.render()
report.saveDocument(url_for('static', filename='reports/SimpleList.mdc'))
```

Полный код примера доступен на GitHub.

© 2003-2026 Stimulsoft

Информация

Генератор отчетов Stimulsoft Reports.PYTHON и аналитические панели Stimulsoft Dashboards.PYTHON поддерживают сохранение MRT- и MDC-файлов только в JSON формат. MRT-файлы в формате XML поддерживаются только в режиме загрузки, и при сохранении будут автоматически преобразованы в JSON формат.

Так как для аналитических панелей всегда нужны данные, они не могут быть сохранены в виде документа. Stimulsoft Dashboards.PYTHON поддерживает сохранение панелей только в виде шаблона с использованием JavaScript события и функций.

При сохранении документа из меню вьювера, файл также сохраняется в JSON формате, и имеет расширение MDC для стандартного документа, MDZ для упакованного документа, MDX для зашифрованного документа.

10.1.5 Построение отчета

Для построения загруженного отчета необходимо вызвать функцию `render()` у объекта отчета `StiReport`. Например, требуется построить отчет перед его экспортированием:

app.py

```
from stimulsoft_reports.report import StiReport
from stimulsoft_reports.report.enums import StiExportFormat

report = StiReport()
report.loadFile(url_for('static', filename='reports/SimpleList.mrt'))
report.render()
report.exportDocument(StiExportFormat.PDF)
```

Полный код примера доступен на GitHub.

Если требуется выполнить какие-либо действия с отчетом перед его построением при помощи JavaScript кода, достаточно определить название JavaScript функции для события `onBeforeRender`, в аргументах функции будут переданы тип действия и сам отчет. Пример регистрации JSON данных перед

© 2003-2026 Stimulsoft

построением отчета:

app.py

```
from stimulsoft_reports.report import StiReport

report = StiReport()
report.onBeforeRender += 'beforeRender'
report.loadFile(url_for('static', filename='reports/SimpleList.mrt'))
report.render()
```

render.html

```
<script>
   function onBeforeRender(args) {
      var dataSet = new Stimulsoft.System.Data.DataSet("SimpleDataSet");
      dataSet.readJsonFile("Demo.json");

      var report = args.report;
      report.regData(dataSet.dataSetName, "", dataSet);
   }
</script>
```

Для выполнения каких-либо действий после построения отчета при помощи JavaScript кода, достаточно определить название JavaScript функции для события `onAfterRender`, в аргументах функции будут переданы тип действия и сам отчет. Пример вывода сообщения после построения отчета:

app.py

```
from stimulsoft_reports.report import StiReport

report = StiReport()
report.onAfterRender += 'afterRender'
report.loadFile(url_for('static', filename='reports/SimpleList.mrt'))
report.render()
```

© 2003-2026 Stimulsoft

render.html

```
<script>
   function afterRender(args) {
      alert("The report rendering is completed.");
   }
</script>
```

Полный код примера доступен на GitHub.

10.1.6 Построение отчета на стороне сервера

Для построения отчета на стороне сервера используется универсальная платформа **Node.js**, при помощи которой выполняется необходимый блок JavaScript кода и возвращается подготовленный результат.

Развертывание платформы Node.js

Перед использованием генератора отчета на стороне сервера, необходимо установить саму платформу Node.js и настроить ее работу. Это можно сделать отдельно, используя инструкции по установке Node.js для конкретной операционной системы с официального сайта платформы, а можно сделать автоматически при помощи специальной функции развертывания.

Если платформа уже установлена, она будет определена автоматически, никаких настроек не потребуется. Предусмотрена возможность указать путь к директории исполняемых файлов платформы, если это необходимо. Также, можно указать путь к рабочей директории, в которой будут установлены необходимые Node.js пакеты.

app.py

```
from stimulsoft_reports import StiNodeJs

nodejs = StiNodeJs()

# nodejs.binDirectory = 'C:\\Program Files\\nodejs'
# nodejs.binDirectory = '/usr/bin/nodejs'

# nodejs.workingDirectory = ''
```

© 2003-2026 Stimulsoft

Полный код примера доступен на GitHub.

Если платформа и пакеты не установлены, можно воспользоваться специальными методами для их установки. Эти методы необходимо вызвать только один раз: первый метод для установки самой платформы Node.js, второй метод для установки либо обновления до текущей версии всех необходимых пакетов:

app.py

```
...

result = nodejs.installNodeJS()
if result:
    result = nodejs.updatePackages()

message = 'The installation was successful.' if result else nodejs.error
```

Полный код примера доступен на GitHub.

Платформа и пакеты будут установлены в указанную рабочую директорию.

Построение отчета

Для переключения генератора отчетов в режим работы на стороне сервера, необходимо установить свойство отчета `engine` в значение `StiEngineType.SERVER_NODE_JS`, весь остальной набор событий и методов для работы с отчетом точно такой же, как и при построении отчета на стороне клиента.

После вызова метода `render()` у объекта отчета, построенный отчет можно сохранить при помощи методов `saveDocument()` либо `savePackedDocument()`, подробнее описано в разделе Загрузка и сохранение отчета. Так же, после построения отчета, его можно экспортировать в один из многочисленных форматов при помощи метода `exportDocument()`, подробнее описано в разделе Экспорт отчета из кода.

Пример загрузки и построения шаблона отчета, и последующее сохранение построенного отчета в файл на стороне сервера:

© 2003-2026 Stimulsoft

app.py

```
from stimulsoft_reports.report import StiReport
from stimulsoft_reports.report.enums import StiEngineType

report = StiReport()
report.engine = StiEngineType.SERVER_NODE_JS
report.loadFile(url_for('static', filename='reports/SimpleList.mrt'),
True)
report.render()
report.saveDocument(url_for('static', filename='reports/SimpleList.mdc'))
```

Полный код примера доступен на GitHub.

Любые ошибки, возникшие при работе с Node.js пакетами, а так же при построении и экспортировании отчета на стороне сервера, можно прочитать в свойствах `report.nodejs.error` и `report.nodejs.errorStack`, эти свойства будут содержать последнюю по очереди ошибку:

app.py

```
from stimulsoft_reports.report import StiReport
from stimulsoft_reports.report.enums import StiEngineType

report = StiReport()
report.engine = StiEngineType.SERVER_NODE_JS
report.loadFile(url_for('static', filename='reports/SimpleList.mrt'),
True)
result = report.render()
if not result:
    # The main text of the error as a string.
    error = report.nodejs.error
    # The full error text as an array of strings.
    errorStack = report.nodejs.errorStack
```

Полный код примера доступен на GitHub.

10.1.7 Обработчик событий

Генератор отчетов, а также вьювер и дизайнер отчетов, умеют вызывать события на стороне клиента, передавать их на сторону Python сервера для последующей обработки, и принимать подготовленный ответ. Все действия реализованы в обработчике событий, нет необходимости в использовании

© 2003-2026 Stimulsoft

каких-либо дополнительных функций для связи клиента с сервером и передачи данных. Для работы с выбранным событием, достаточно добавить имя функции к обработчику, и указанная функция будет автоматически вызвана при возникновении выбранного события. Предусмотрен вызов событий как на стороне JavaScript клиента, так и на стороне Python сервера. При необходимости, можно добавлять несколько функций любого типа к одному и тому же событию.

Вызов JavaScript события на стороне клиента

Для вызова JavaScript события, необходимо добавить к обработчику имя функции в виде строки. В аргументах события будут переданы все необходимые данные, необходимые для работы определённого события. Список доступных аргументов можно посмотреть в разделе Список событий генератора отчетов.

Пример вывода сообщения с количеством страниц полученного документа после построения отчета:

app.py

```
from stimulsoft_reports.report import StiReport

report = StiReport()
report.onAfterRender += 'afterRender'
report.loadFile(url_for('static', filename='reports/SimpleList.mrt'))
report.render()
```

report.html

```
<script>
   function afterRender(args) {
      let pagesCount = args.report.renderedPages.count;
      alert("The report is rendered, pages: " + pagesCount);
   }
</script>
```

В данном примере, из аргументов события можно получить JavaScript объект отчета, и прочитать количество построенных страниц документа.

© 2003-2026 Stimulsoft

Информация

Более подробно про доступные функции и параметры JavaScript генератора отчетов можно прочитать в документации Stimulsoft Reports.JS и Stimulsoft Dashboards.JS продуктов.

Вызов JavaScript события на стороне Node.js сервера

При использовании Node.js платформы для работы с отчетами нет возможности вызвать JavaScript функцию по имени, т.к. HTML шаблон не используется. В таком случае, для вызова JavaScript события необходимо добавить к обработчику саму функцию в виде строки или строк кода. Аргументы события будут находиться в предварительно заданной переменной `args`, которую можно использовать в коде события.

Пример очистки словаря данных в шаблоне перед построением отчета:

app.py

```
from stimulsoft_reports.report import StiReport
from stimulsoft_reports.report.enums import StiEngineType

report = StiReport()
report.engine = StiEngineType.SERVER_NODE_JS
report.onBeforeRender += 'args.report.dictionary.clear();'
report.loadFile(url_for('static', filename='reports/SimpleList.mrt'))
report.render()
```

Информация

Точно такой же способ для JavaScript событий можно использовать при отображении вьювера либо дизайнера без HTML шаблона, когда нет возможности заранее определить необходимую JavaScript функцию.

Вызов Python события на стороне сервера

Для вызова Python события, необходимо добавить к обработчику имя функции

© 2003-2026 Stimulsoft

в виде переменной. В аргументах события будут переданы все необходимые данные, необходимые для работы определённого события. Список доступных аргументов можно посмотреть в разделе Список событий генератора отчетов.

Пример корректировки пароля в строке подключения перед запросом данных:

app.py

```
from stimulsoft_reports.events import StiDataEventArgs

def beginProcessData(args: StiDataEventArgs):
    args.connectionString = args.connectionString.replace('Pwd=;',
'Pwd=******;')

report = StiReport()
report.onBeginProcessData += beginProcessData
report.loadFile(url_for('static', filename='reports/SimpleList.mrt'))
report.render()
```

В данном примере, из аргументов события можно получить и изменить все параметры подключения к базе данных.

В событии на стороне Python сервера предусмотрена возможность вернуть текстовое сообщение об успешном выполнении либо возникшей ошибке обработки события, это сообщение отобразится во вьювере или дизайнере после завершения события. Для отображения окна с ошибкой необходимо вернуть один из перечисленных в таблице результатов выполнения события:

Наименование	Описание
`StiResult.getError('Error message')`	Отображает окно с пиктограммой красного цвета с восклицательным знаком и указанным сообщением об ошибке.
`StiResult.getSuccess('Info message')`	Отображает окно с пиктограммой голубого цвета с иконкой «i» и указанным информационным сообщением.
`False`	Отображает окно с пиктограммой красного цвета с восклицательным

© 2003-2026 Stimulsoft

	знаком и стандартным сообщением об ошибке, с указанием названия события.
`'Info message'`	Работает аналогично вызову метода `StiResult.getSuccess('Info message')`
`None`	Обычная работа события без вывода сообщений. При возникновении внутренней ошибки обработчика событий, будет выведено соответствующее сообщение о возникшей ошибке.

Пример вывода различных сообщений результата работы события:

app.py

```
from stimulsoft_reports import StiResult
from stimulsoft_reports.designer import StiDesigner
from stimulsoft_reports.events import StiReportEventArgs
from stimulsoft_reports.report import StiReport

def saveReport(args: StiReportEventArgs):
    #StiResult.getError('An error occurred while saving.')
    #StiResult.getSuccess('The report was successfully saved.')
    return 'The report was successfully saved.'

designer = StiDesigner()
designer.onSaveReport += saveReport
```

Информация

При возникновении ошибки в самом обработчике событий, например неудача при подключении к базе данных, ошибка обработки файла и т.п. - будет выведено внутреннее сообщение в любом случае, даже если определено сообщение в событии компонента.

© 2003-2026 Stimulsoft

Информация

Диалоговое окно с сообщением будет показано только при использовании компонентов StiViewer либо StiDesigner. Сам генератор отчетов не содержит визуальных форм, поэтому сообщение обработки события будет выведено в консоли браузера.

Вызов нескольких одинаковый событий

Предусмотрена возможность добавления к обработчику событий неограниченного количества функций, они все будут сгруппированы по типу события, и вызваны последовательно в порядке добавления. Например, требуется изменить SQL запрос на стороне JavaScript клиента, а строку подключения на стороне Python сервера:

Пример изменения SQL запроса на стороне JavaScript клиента, и строки подключения на стороне PHP сервера:

app.py

```
from stimulsoft_reports.events import StiDataEventArgs
from stimulsoft_reports.report import StiReport

def beginProcessData(args: StiDataEventArgs):
    args.connectionString = args.connectionString.replace('Pwd=;',
'Pwd=;')

report = StiReport()
report.onBeginProcessData += beginProcessData
report.onBeginProcessData += 'beginProcessData'
report.loadFile(url_for('static', filename='reports/SimpleList.mrt'))
report.render()
```

viewer.html

```
<script>
  function beginProcessData(args) {
    args.queryString = args.queryString.replace('TableName', 'Products')
  }
</script>
```

© 2003-2026 Stimulsoft

Информация

Некоторые события могут быть вызваны только на стороне JavaScript клиента и не имеют возможности вызова события на стороне Python сервера. При добавлении функции к такому событию не будет ошибки, просто добавленная функция не будет вызвана. Все поддерживаемые варианты указаны в разделе Список событий генератора отчетов.

Шифрование данных, передаваемых на сторону Python сервера

Для исключения кражи передаваемых данных злоумышленниками, мы рекомендуем использовать HTTPS протокол, в большинстве случаев этого достаточно. В дополнении к этому, по умолчанию все передаваемые данные проходят через специальный алгоритм кодирования, и передаются на сервер в зашифрованном виде. Это позволяет скрыть конфиденциальные данные, например логин и пароль в строке подключения, от любопытных пользователей, работающих с вашим приложением.

Однако, если этого не требуется, либо если нужно отобразить исходные данные запроса для отладки приложения, предусмотрена возможность отключения шифрования. Для этого достаточно установить свойство `encryptData` в значение `False` у обработчика событий, после этого все данные будут передаваться в JSON формате.

app.py

```
from stimulsoft_reports.report import StiReport

report = StiReport()
report.handler.encryptData = False
```

10.1.8 Работа с данными

Как правило, параметры подключения к источникам данных хранятся в самом шаблоне отчета – это могут быть как файловые источники данных, так и SQL источники данных. При необходимости подключения данных из кода, предусмотрен специальный метод `regData()` у объекта отчета. В качестве параметров метода необходимо указать имя источника данных в отчете, и сами данные. Данные могут быть в формате JSON либо XML в виде строки, а также в

© 2003-2026 Stimulsoft

виде PHP объекта либо массива.

Пример подключения данных в виде объекта:

app.py

```
from stimulsoft_reports.report import StiReport

report = StiReport()
report.loadFile(url_for('static', filename='reports/SimpleList.mrt'))

data = { "Customers": {
    "CompanyName": "Chop-suey Chinese",
    "Address": "Emilii Plater Street 49",
    "Phone": 555123000,
    "ContactTitle": "Sales Representative"}
}

report.regData("Demo", data)
report.render()
```

В некоторых случаях может понадобиться синхронизировать структуру словаря данных отчета с передаваемыми данными. Для этого достаточно указать значение True в качестве третьего параметра метода `regData()`, в этом случае перед построением отчета будет выполнена синхронизация словаря данных.

Информация

При использовании компонента Дизайнер, можно создать новый объект отчета и передать ему данные со включенной синхронизацией словаря отчета. В этом случае, будет открыт дизайнер с новым отчетом и уже подготовленными для работы данными.

Очистка параметров подключения и словаря данных

Если в загруженном отчете уже настроены параметры подключения к данным, то они имеют приоритет. При необходимости принудительного использования своего набора данных, предусмотрена возможность очистить параметры подключения, для этого достаточно вызвать специальный метод `clearData()` у

© 2003-2026 Stimulsoft

объекта отчета. По умолчанию, будет выполнена очистка только самих параметров подключения. Если есть необходимость полностью очистить структуру словаря данных (удалить все существующие источники данных), необходимо указать значение true в качестве параметра метода.

Пример подключения данных с предварительной очисткой существующего словаря:

app.py

```
from stimulsoft_reports.report import StiReport

report = StiReport()
report.loadFile(url_for('static', filename='reports/SimpleList.mrt'))

report.clearData(True)

data = '{"Customers": [{"CompanyName": "Centro comercial Moctezuma",
"Address": "Sierras de Granada 9993", "Phone": 12355500, "ContactTitle":
"Marketing Manager"}]}'

report.regData("Demo", data, True)
report.render()
```

Событие подготовки к построению отчета

Для подключения данных перед построением отчета, а также для изменения самого отчета, можно воспользоваться специальным событием `onBeforeRender` объекта отчета. Данное событие будет вызвано перед построением отчета после его загрузки. Для подключения данных достаточно вызвать метод `regReportData()` у аргументов события. Параметры метода точно такие же, как и у метода `regData()` объекта отчета.

Пример подключения данных в событии отчета:

app.py

```
from stimulsoft_reports.report import StiReport
from stimulsoft_reports.events import StiReportEventArgs

def beforeRender(args: StiReportEventArgs):
    data = '{"Customers": [{"CompanyName": "Centro comercial Moctezuma",
"Address": "Sierras de Granada 9993", "Phone": 12355500, "ContactTitle":
"Marketing Manager"}]}'
```

© 2003-2026 Stimulsoft

```
    args.regReportData("Demo", data)


report = StiReport()
report.onBeforeRender += beforeRender
report.loadFile(url_for('static', filename='reports/SimpleList.mrt'))
report.render()
```

Информация

Если в отчете были заданы данные при помощи метода `regData()` у объекта отчета – они будут переопределены данными, указанными в событии.

Подключение данных на стороне клиента

При необходимости, можно подключить данные при помощи JavaScript функций генератора отчетов. Для этого можно воспользоваться событием `onBeforeRender` объекта отчета. Данные могут быть загружены напрямую в специальный объект `DataSet`, который используется для их хранения. Он содержит функции для загрузки данных из файлов форматов XML/XSD и JSON. После загрузки файлов, необходимо вызвать функцию `regData()` у объекта отчета для подключения данных, в аргументах функции указывается подготовленный объект `DataSet`.

Пример загрузки данных из файла XML с использованием схемы XSD:

app.py

```
from stimulsoft_reports.report import StiReport

report = StiReport()
report.onBeforeRender += 'beforeRender'
report.loadFile(url_for('static', filename='reports/SimpleList.mrt'))
report.render()
```

report.html

```
<script>
  function beforeRender(args) {
```

© 2003-2026 Stimulsoft

```
    let dataSet = new Stimulsoft.System.Data.DataSet("SimpleDataSet");

    dataSet.readXmlSchemaFile("Demo.xsd");
    dataSet.readXmlFile("Demo.xml");

    //dataSet.readJsonFile("Demo.json");

    let report = args.report;
    report.regData(dataSet.dataSetName, "", dataSet);
    report.dictionary.synchronize();
  }
</script>
```

Полный код примера доступен на GitHub.

Информация

Загрузка схемы данных не является обязательной. Если вы хотите использовать схему данных, вам следует добавить ее перед загрузкой данных XML.

Пример загрузки данных из файла JSON:

app.py

```
from stimulsoft_reports.report import StiReport

report = StiReport()
report.onBeforeRender += 'beforeRender'
report.loadFile(url_for('static', filename='reports/SimpleList.mrt'))
report.render()
```

report.html

```
<script>
  function beforeRender(args) {
    let dataSet = new Stimulsoft.System.Data.DataSet("SimpleDataSet");
    dataSet.readJsonFile("Demo.json");

    let report = args.report;
    report.regData(dataSet.dataSetName, "", dataSet);
    report.dictionary.synchronize();
```

© 2003-2026 Stimulsoft

```
    }
</script>
```

Полный код примера доступен на GitHub.

Кроме указанных функций `readXmlFile()` и `readJsonFile()`, предусмотрены функции `readXml()` и `readJson()`, принимающие на вход данные в виде строки либо объекта.

Информация

Функция `report.dictionary.synchronize()` необходима для синхронизации подключаемых данных со словарем данных шаблона отчета. То есть, при вызове этой функции, словарь отчета будет создан на основе структуры данных, загруженных в `DataSet`. Функция синхронизации не требуется, если словарь создан заранее, и его структура совпадает с подключаемыми данными.

10.1.9 Подключение файлов данных

Как правило, параметры подключения к источникам данных хранятся в самом шаблоне отчета. Для работы с файловыми источниками данных, такими как XML, JSON, CSV не требуется никаких дополнительных действий, все алгоритмы находятся в скрипте генератора отчетов.

Событие загрузки данных

Для просмотра и изменения параметров подключения файловых данных перед их загрузкой, необходимо определить событие `onBeginProcessData` у компонента. В аргументах события будет передана информация о подключении к файловому источнику данных - имя и тип подключения в шаблоне отчета, а также путь к файлу данных. Подробное описание доступных значений свойств, передаваемых в аргументах события, находится в разделе События генератора отчетов.

Допускается изменить путь к файлу данных. В этом случае после завершения события, генератор отчетов будет запрашивать файл по новому пути, заданному в аргументах. Например, требуется для указанного соединения изменить путь к

© 2003-2026 Stimulsoft

JSON файлу данных:

app.py

```
from stimulsoft_reports.report import StiReport
from stimulsoft_reports.events import StiDataEventArgs

def beginProcessData(args: StiDataEventArgs):
    if args.connection == 'MyJsonConnection':
        args->pathData = 'Data/Demo.json'


report = StiReport()
report.onBeginProcessData += beginProcessData
report.loadFile(url_for('static', filename='reports/SimpleList.mrt'))
report.render()
```

При необходимости, те же самые действия можно выполнить в событии на стороне JavaScript клиента:

report.html

```
<script>
  function beginProcessData(args) {
    if (args.connection == "MyJsonConnection")
      args.pathData = "Data/Demo.json";
  }
</script>
```

Информация

Для XML источника данных событие `onBeginProcessData` будет вызвано дважды: первый раз для чтения XSD схемы, второй раз для чтения самого XML файла данных.

Событие обработки данных

Для просмотра либо корректировки загруженных данных перед их подключением и построением отчета, необходимо определить событие `onEndProcessData` у компонента. В аргументах события будет передана

© 2003-2026 Stimulsoft

информация о подключении к файловому источнику данных - имя и тип подключения, сохраненные в шаблоне отчета, а также результат загрузки данных, содержащий загруженные данные в виде строки. Подробное описание доступных значений свойств, передаваемых в аргументах события, доступно в разделе События генератора отчетов.

app.py

```
from stimulsoft_reports.report import StiReport
from stimulsoft_reports.events import StiDataEventArgs

def endProcessData(args: StiDataEventArgs):
    data = args.result.data


report = StiReport()
report.onEndProcessData += endProcessData
report.loadFile(url_for('static', filename='reports/SimpleList.mrt'))
report.render()
```

При необходимости, просмотр либо корректировку загруженных данных можно выполнить в событии на стороне JavaScript клиента. В аргументах события будет передана информация о подключении к файловому источнику данных - имя и тип подключения, сохраненные в шаблоне отчета, а также подготовленный объект `DataSet`, содержащий таблицы и строки данных, полученных из файлового источника. Подробное описание доступных значений аргументов находится в разделе События генератора отчетов.

app.py

```
from stimulsoft_reports.report import StiReport

report = StiReport()
report.onEndProcessData += 'endProcessData'
report.loadFile(url_for('static', filename='reports/SimpleList.mrt'))
report.render()
```

© 2003-2026 Stimulsoft

report.html

```
<script>
   function endProcessData(args) {
      let dataSet = args.dataSet;
   }
</script>
```

Информация

Компоненты вьювер отчетов и дизайнер отчетов также имеют свойство указанные выше события, при помощи которых можно управлять загрузкой данных указанным выше способом.

Отключение обработки файловых адаптеров данных на стороне сервера

По умолчанию, загрузка файловых данных осуществляется на стороне сервера. Это необходимо, если требуется контролировать параметры подключения и сами данные в событиях на стороне сервера. Если в этом нет необходимости, обработку файловых данных на стороне сервера можно отключить. В этом случае загрузка данных будет осуществляться при помощи JavaScript, это немного упростит цепочку событий и ускорит загрузку данных.

Для отключения файловых адаптеров данных на стороне сервера, достаточно установить свойство `allowFileDataAdapters` у обработчика событий в значение `False`, например:

index.php

```
from stimulsoft_reports.report import StiReport

report = StiReport()
report.handler.allowFileDataAdapters = False
```

Использование переменных в файловых данных

Предусмотрена возможность использовать переменные в виде выражений (а также использовать выражения) при указании пути к файловому источнику данных. Переменная или выражение задаются в фигурных скобках. Допустимо

© 2003-2026 Stimulsoft

использовать несколько выражений в любом месте пути к файлу данных, например:

File Data Source

```
https://localhost/data/{VariableJsonFileName}.json
https://localhost/data/json?id={VariableId}
https://localhost/{VariableCategory}/{VariableId}
```

Таким образом, один источник данных может быть преобразован к REST синтаксису, это избавит от необходимости создавать несколько схожих источников данных для получения однотипных данных. А в совокупности с серверной Python логикой и событиями генератора отчетов, сделать источник данных еще более гибким.

Использование OData

Также можно использовать данные для создания отчетов, полученные из хранилищ OData. В этом случае необходимо выполнить авторизацию, используя имя пользователя, пароль пользователя или токен. Параметры авторизации указываются в строке подключения к хранилищу OData с помощью символа ";" разделитель.

© 2003-2026 Stimulsoft

report.html

```
<script>
   function beginProcessData(args) {
      let report = args.report;

      // Authorization using a user account
      let oDataDatabase = new
      Stimulsoft.Report.Dictionary.StiODataDatabase("OData", "OData",
      "https://services.odata.org/V4/Northwind/
      Northwind.svc;AddressBearer=adress;UserName=UserName;Password=Passwor
      d;Client_Id=Your Client ID", false, null);

      // Authorization using a user token
      let oDataDatabase = new
      Stimulsoft.Report.Dictionary.StiODataDatabase("OData", "OData",
      "https://services.odata.org/V4/Northwind/Northwind.svc;Token=Enter
      your token", false, null);

      report.dictionary.databases.add(oDataDatabase);
      report.dictionary.synchronize();

      // Query with data filter
      var productsDataSource =
      report.dictionary.dataSources.getByName("Products");
      if (productsDataSource != null) productsDataSource.sqlCommand =
      "Products?filter=ProductID eq 2";
   }
</script>
```

10.1.10 Подключение SQL адаптеров данных

Для построения отчетов, генератор отчетов позволяет использовать данные из различных SQL источников. Поскольку чистый JavaScript не имеет встроенных методов для работы с удаленными базами данных, эта функциональность реализуется с помощью серверного Python кода. Для работы с SQL источниками данных не требуется никаких дополнительных действий, все адаптеры данных уже подключены и настроены для работы.

Событие загрузки данных

При необходимости обработки параметров, используемых для подключения к базе данных, предусмотрено событие `onBeginProcessData`. В аргументах события будут переданы все необходимые параметры подключения к SQL источнику данных, а также параметры SQL запроса. Все параметры подключения к источнику данных могут быть изменены как на стороне JavaScript клиента, так и на стороне PHP сервера. Подробное описание доступных значений свойств, передаваемых в аргументах события, находится в разделе События генератора отчетов.

© 2003-2026 Stimulsoft

Пример измерения SQL запроса на стороне JavaScript клиента, и строки подключения к базе данных на стороне PHP сервера:

app.py

```
from stimulsoft_reports.events import StiDataEventArgs
from stimulsoft_reports.report import StiReport

def beginProcessData(args: StiDataEventArgs):
    if args.connection == 'MyConnectionName':
        args.connectionString =
'Server=localhost;Database=test;uid=root;password=******;'

report = StiReport()
report.onBeginProcessData += beginProcessData
report.onBeginProcessData += 'beginProcessData'
report.loadFile(url_for('static', filename='reports/SimpleList.mrt'))
report.render()
```

report.html

```
<script>
  function beginProcessData(args) {
    if (args.dataSource == "MyDataSource")
      args.queryString = "SELECT * FROM ProductsTable";
  }
</script>
```

Таким образом, в событии `onBeginProcessData` можно узнать тип базы данных, имя подключения и имя источника данных, а также узнать и при необходимости скорректировать строку подключения и SQL запрос на получение данных, установить значения параметров запроса. При изменении значений аргументов на стороне сервера, измененные значения не будут переданы на сторону клиента, поэтому здесь можно использовать конфиденциальные данные, такие как логин и пароль в строке подключения, имена и префиксы таблиц и т.п.

Событие обработки данных

Для просмотра либо корректировки загруженных данные перед их регистрацией и построением отчета, предусмотрено событие

© 2003-2026 Stimulsoft

`onEndProcessData`. В аргументах события будут переданы все необходимые параметры подключения к SQL источнику данных, а также результат выполнения запроса, содержащий названия колонок, типы колонок и строки данных, полученные из SQL источника. Подробное описание доступных свойств, передаваемых в аргументах события, находится в разделе События генератора отчетов.

Пример результата выполнения SQL запроса, где результат уже содержит подготовленные для отчета данные:

app.py

```
from stimulsoft_reports.events import StiDataEventArgs
from stimulsoft_reports.report import StiReport

def endProcessData(args: StiDataEventArgs):
    args.result = {
        columns: ['id', 'username', 'phone'],
        types: ['int', 'string', 'string'],
        rows: [
            [1, 'Mario Pontes', '555-6874'],
            [2, 'Helen Bennett', '555-2376']
        ]
    }

report = StiReport()
report.onEndProcessData += endProcessData
report.onEndProcessData += 'endProcessData'
report.loadFile(url_for('static', filename='reports/SimpleList.mrt'))
report.render()
```

report.html

```
<script>
  function endProcessData(args) {
    args.result = {
      columns: ["id", "username", "phone"],
      types: ["int", "string", "string"],
      rows: [
                  [1, "Mario Pontes", "555-6874"],
                  [2, "Helen Bennett", "555-2376"]
      ]
    };
  }
</script>
```

© 2003-2026 Stimulsoft

Доступные свойства объекта данных указаны в таблице:

Наименование	Описание
`count`	Общее количество колонок таблицы SQL источника данных.
`columns`	Имена колонок таблицы SQL источника данных.
`types`	Типы колонок таблицы SQL источника данных, приведенные к известным типам для генератора отчетов.
`rows`	Строки данных SQL источника данных, представляет собой массив из массивов всех строк таблицы.

Все данные результата выполнения SQL запроса могут быть изменены как на стороне JavaScript клиента, так и на стороне PHP сервера. Количество колонок и типы данных должны совпадать для исключения неверной интерпретации данных генератором отчета.

Использование параметров в SQL запросе

При необходимости, можно использовать параметры в SQL запросе. Для этого, в источнике данных необходимо добавить параметры в специальную коллекцию, для каждого параметра установить необходимый тип и значение по умолчанию. После этого, параметры можно использовать в SQL запросе следующим образом:

SQL Data Source

```
SELECT * FROM @Parameter1 WHERE UserID = @Parameter2
```

Все значения параметров хранятся в самом источнике данных в виде коллекции. Коллекция представляет собой массив из объектов, содержащих имя параметра, его тип и значение. Пример структуры массива параметров на

© 2003-2026 Stimulsoft

стороне JavaScript клиента:

app.py

```
args.parameters = [
   {
     name: "ParameterString",
     type: 752,
     typeName: "Text",
     value: "Text value"
  },
   {
     name: "ParameterInt",
     type: 3,
     typeName: "Int32",
     value: 20
   }
];
```

Для доступа к параметрам запроса можно воспользоваться событием `onBeginProcessData`, коллекция параметров будет передана в аргументах события. Допускается изменение значений любых параметров, находящихся в передаваемой коллекции. Пример изменения значения одного и того же параметра на стороне JavaScript клиента и на стороне PHP сервера:

app.py

```
from stimulsoft_reports.events import StiDataEventArgs
from stimulsoft_reports.report import StiReport

def beginProcessData(args: StiDataEventArgs):
    if args.dataSource == 'DataSourceWithParams':
        args.parameters['Parameter1'].value = 'TableName'
        args.parameters['Parameter2'].value = 10


report = StiReport()
report.onBeginProcessData += beginProcessData
report.onBeginProcessData += 'beginProcessData'
report.loadFile(url_for('static', filename='reports/
SimpleListSQLParameters.mrt'))
report.render()
```

report.html

© 2003-2026 Stimulsoft

```
<script>
   function beginProcessData(args) {
      if (args.dataSource == "DataSourceWithParams") {
         args.parameters['Parameter1'].value = "TableName";
         args.parameters['Parameter2'].value = 10;
      }
   }
</script>
```

При изменении значений параметров запроса, тип нового значения должен соответствовать типу изменяемого параметра. В противном случае, выполнение SQL запроса может вернуть некорректные данные либо вызвать внутреннюю ошибку выполнения.

Если в отчете используется несколько источников данных, необходимо выполнять проверку перед присвоением параметров. В противном случае, возникнет ошибка выполнения PHP скрипта, если какой-либо из параметров будет отсутствовать в текущем источнике данных.

Информация

При изменении значений параметров на стороне Python сервера, измененные значения не будут переданы на сторону клиента, поэтому в качестве значений можно использовать конфиденциальные данные.

Использование переменной отчета в качестве SQL параметра

Предусмотрена возможность использования переменной в качестве SQL параметра. Для этого в редакторе переменной отчета достаточно установить свойство **Allow using as SQL parameter**, после чего ее можно использовать в любом SQL запросе. Синтаксис будет точно таким же, как и при использовании параметров в источнике данных.

© 2003-2026 Stimulsoft

Информация

Такая переменная будет передана в коллекции параметров только в том случае, если она используется в запросе. Параметры из коллекции источника данных передаются всегда, даже если в запросе они не использованы.

Экранирование значений параметров

Все значения параметров будут автоматически экранированы для исключения возможности SQL инъекций и нарушения безопасности выполнения запросов. Если вам не требуется экранирование, и вы сами контролируете безопасность значений параметров, автоматическое экранирование можно отключить. Для этого достаточно установить свойство `escapeQueryParameters` в значение `False` у обработчика событий:

app.py

```
from stimulsoft_reports.report import StiReport

report = StiReport()
report.handler.escapeQueryParameters = False
```

После установки указанного свойства, использование параметров является небезопасным, вам нужно строго контролировать значения перед выполнением SQL запросов.

Информация

Экранирование применяется только для параметров SQL запросов, а также для переменных, используемых как параметры. Если переменная используется в виде выражения, т.е. в фигурных скобках, например `{VariableName}`, то экранирование применяться не будет в любом случае. Подробное описание работы переменных находится в разделе Работа с переменными отчета.

© 2003-2026 Stimulsoft

Шифрование данных, передаваемых на сторону клиента

Если включено общее шифрование данных при помощи опции `encryptData`, то все передаваемые на сторону клиента данные будут зашифрованы, включая данные из SQL источников. Это повышает безопасность, но может замедлить работу на больших объёмах данных. Предусмотрена возможность отдельного отключения шифрования SQL данных, передаваемых на сторону клиента. Для этого достаточно установить свойство `encryptSqlData` в значение `False` у обработчика событий, после этого все данные будут передаваться в открытом виде в JSON формате. Это не должно в значительной мере повлиять на безопасность, т.к. в незашифрованном виде будут переданы только сами данные, которые в том или ином виде отображаются в отчёте.

app.py

```
from stimulsoft_reports.report import StiReport

report = StiReport()
report.handler.encryptSqlData = False
```

10.1.11 Работа с переменными отчета

Генератор отчетов позволяет использовать переменные в выражениях, запросах, фильтрах и других элементах отчета. Предусмотрена возможность предварительного просмотра и изменения значений переменных из кода перед построением отчета.

Значения переменных шаблона отчета

Генератор отчетов позволяет легко получить прямой доступ к переменным в шаблоне отчета через словарь данных. Для этого достаточно определить событие `onBeforeRender` у объекта отчета, в аргументах которого на стороне клиента будет передан JavaScript объект отчета. Подробное описание доступных значений аргументов находится в разделе События генератора отчетов..

Для доступа к переменной отчета, необходимо использовать JavaScript функцию `getByName()` у коллекции переменных в словаре данных отчета. Для изменения значения переменной, достаточно присвоить его в свойстве `value`, при этом нет необходимости проверять установленный тип переменной - преобразование типов будет выполнено автоматически. Например, требуется изменить строковое и целочисленное значение указанных переменных:

© 2003-2026 Stimulsoft

Пример изменения строкового и целочисленного значение выбранных переменных на стороне JavaScript клиента:

app.py

```
from stimulsoft_reports.report import StiReport

report = StiReport()
report.onBeforeRender += 'beforeRender'
report.loadFile(url_for('static', filename='reports/SimpleList.mrt'))
report.render()
```

report.html

```
<script>
   function beforeRender(args) {
      let report = args.report;

      let variableString =
      report.dictionary.variables.getByName("VariableString");
      variableString.value = "Text value";

      let variableInt =
      report.dictionary.variables.getByName("VariableInt");
      variableInt.value = "20";
   }
</script>
```

Прямой доступ к переменным в шаблоне отчета на стороне Python сервера не предусмотрен.Прямой доступ к переменным в шаблоне отчета на стороне Python сервера не предусмотрен.

Информация

Значения переменных будут изменены в самом шаблоне отчета, и при его последующем сохранении будут сохранены в файле. Для изменения значений переменных без изменения шаблона, можно воспользоваться событием `onPrepareVariables`, которое вызывается при подготовке значений переменных отчета перед его построением.

© 2003-2026 Stimulsoft

Значения переменных при построении отчета

Генератор отчетов позволяет легко получить доступ к переменным перед построением отчета. Для этого достаточно определить событие `onPrepareVariables` у объекта отчета. В аргументах события будет передана коллекция переменных отчета с их типами и значениями. Если переменная инициализируется как выражение, то в коллекцию будет передано уже подсчитанное значение выражения. Подробное описание доступных значений свойств, передаваемых в аргументах события, находится в разделе События генератора отчетов.

В событии на стороне JavaScript клиента коллекция переменных представляет собой массив из объектов, содержащих имя переменной, ее тип и значение. Допустимо изменять значения переменных, при этом тип нового значения должен соответствовать типу изменяемой переменной.

report.html

```
<script>
   function prepareVariables(args) {
      args.variables = [
      {
         name: "VariableString",
         type: "String",
         value: "Text value"
      },
      {
         name: "VariableInt",
         type: "Int32",
         value: 20
      }
   ];
</script>
```

Пример изменения значения переменной на стороне JavaScript клиента:

© 2003-2026 Stimulsoft

app.py

```
from stimulsoft_reports.report import StiReport

report = StiReport()
report.onPrepareVariables += 'prepareVariables'
report.loadFile(url_for('static', filename='reports/Variables.mrt'))
report.render()
```

В событии на стороне Python сервера коллекция переменных представляет собой словарь из объектов типа `StiVariable`, в каждом объекте хранится имя переменной, тип переменной и значение. Ключ словаря соответствует имени переменной отчета.

report.html

```
<script>
   function prepareVariables(args) {
      let variables = args.variables;

      variables.find(item => item.name == "VariableString").value = "Text
      value";
      variables.find(item => item.name == "VariableInt").value = 20;
   }
</script>
```

На стороне Python сервера также допустимо изменять значения переменных, при этом тип нового значения должен соответствовать типу изменяемой переменной. Предусмотрена возможность создания новой переменной отчета, если это необходимо. На сторону клиента будет передана коллекция, содержащая только те переменные, значения которых были изменены, а также новые переменные.

Пример изменения значения переменной на стороне PHP сервера:

app.py

```
from stimulsoft_reports.report import StiReport
from stimulsoft_reports.events import StiVariablesEventArgs
```

© 2003-2026 Stimulsoft

```
def prepareVariables(args: StiVariablesEventArgs):
    args.variables['VariableString'].value = 'Text value'
    args.variables['VariableInt'].value = 20

report = StiReport()
report.onPrepareVariables += prepareVariables
report.loadFile(url_for('static', filename='reports/Variables.mrt'))
report.render()
```

Полный код примера доступен на GitHub.

app.py

```
from stimulsoft_reports.events import StiVariablesEventArgs

def prepareVariables(args: StiVariablesEventArgs):
    args.variables['VariableString'].value = 'Value from Server-Side'
    args.variables['VariableInt'].value = 123
    args.variables['VariableDecimal'].value = 123.456
```

Переменные типа `DateTime` передаются в виде строкового значения в формате `"YYYY-MM-dd HH-mm-ss"`, либо в виде объекта типа `datetime`, например:

app.py

```
from datetime import datetime
from stimulsoft_reports.events import StiVariablesEventArgs

def prepareVariables(args: StiVariablesEventArgs):
    args.variables['VariableDateTime'].value = '2021-03-20 22:00:00'
    args.variables['VariableDateTime'].value = datetime.today()
```

Для изменения значения переменной типа `Range`, необходимо использовать вложенные значения `value['from']` и `value['to']` выбранной переменной в коллекции. Формат каждого из двух значений такой же, как и у простой переменной, например:

© 2003-2026 Stimulsoft

app.py

```
from stimulsoft_reports.events import StiVariablesEventArgs

def prepareVariables(args: StiVariablesEventArgs):
    args.variables['VariableStringRange'].value['from'] = 'Aaa'
    args.variables['VariableStringRange'].value['to'] = 'Zzz'
```

Для изменения значения переменной типа `List`, необходимо обратиться к значению списка по его индексу. Допускается установка значений сразу всего списка в виде подготовленного массива, например:

app.py

```
from stimulsoft_reports.events import StiVariablesEventArgs

def prepareVariables(args: StiVariablesEventArgs):
    args.variables['VariableStringList'].value[0] = 'Test'
    args.variables['VariableIntList'].value = [1, 22, 333]
```

Для создания новой переменной, не определенной в отчете, необходимо добавить в словарь переменных новый объект типа `StiVariable`, используя новое имя переменной. После этого переменную можно использовать для построения отчета, то есть в шаблоне отчета переменная сохранена не будет.

app.py

```
from stimulsoft_reports.report import StiVariable
from stimulsoft_reports.report.enums import StiVariableType

def prepareVariables(args: StiVariablesEventArgs):
    args.variables['NewVarInt'] = StiVariable('NewVarInt',
StiVariableType.INT, 10)
```

После обработки события `onPrepareVariables`, на сторону клиента будет передана новая коллекция, содержащая только те переменные, значения которых были изменены, а также новые переменные.

© 2003-2026 Stimulsoft

Информация

Если переменная используется в SQL запросе в виде выражения, т.е. записана в фигурных скобках, например `{VariableName}`, то ее значение не будет автоматически экранировано. Вам необходимо самим следить за безопасностью значений, либо использовать переменную как параметр запроса, например **@VariableName**. Подробное описание работы параметров находится в разделе Подключение SQL адаптеров данных.

Переменные отчета, передаваемые в URL запросе

Генератор отчетов имеет возможность автоматически присваивать значения переменных, переданных в URL запросе. Для этого необходимо установить свойство `passQueryParametersToReport` в значение `True` у обработчика событий:

app.py

```
@app.route('/report', methods = ['GET', 'POST'])
def report():
    report = StiReport()
    report.handler.passQueryParametersToReport = True
    if report.processRequest(request):
        return report.getFrameworkResponse()
```

Все остальные действия генератор отчетов сделает автоматически. Если переменная присутствует в отчете, то ее значение будет изменено на значение из URL запроса. Если такой переменной в отчете нет - она будет создана для построения отчета, то есть в шаблоне отчета новая переменная сохранена не будет. Имена переменных, передаваемых в URL запросе, не чувствительны к регистру.

© 2003-2026 Stimulsoft

Информация

Все значения переменных будут присвоены до вызова события `onPrepareVariables`, таким образом в этом событии можно дополнительно контролировать установленные значения, и при необходимости выполнять их корректировку.

10.1.12 Подключение пользовательских шрифтов

В генераторе отчетов доступна возможность загрузки пользовательских шрифтов из файла либо директории. Для этого предназначен статический класс `StiFontCollection`, который содержит необходимые функции по работе со шрифтами.

Для подключения файла шрифта предназначена функция `addFontFile`, в аргументах необходимо указать путь к файлу шрифта, а также при необходимости имя шрифта и его стиль. Если имя шрифта либо стиль не указаны, то будут использоваться параметры из файла шрифта. Пример подключения файла шрифта:

app.py

```
from stimulsoft_reports import StiFontCollection
from stimulsoft_reports.enums import FontStyle

StiFontCollection.addFontFile('Roboto.ttf', 'Roboto');
StiFontCollection.addFontFile('Roboto-Black.ttf', 'Roboto Black',
FontStyle.BOLD)
```

Дополнительно предусмотрена возможность загрузки всех шрифтов, находящихся в одной директории. Для этого достаточно указать необходимую директорию со шрифтами при помощи функции `setFontsFolder`, например:

app.py

```
from stimulsoft_reports import StiFontCollection

StiFontCollection.setFontsFolder('/fonts')
```

© 2003-2026 Stimulsoft

10.1.13 Печать отчета из кода

В генераторе отчетов доступна возможность печати отчета из кода. Для этого можно воспользоваться специальным методом `print()` у объекта отчета:

app.py

```
report = StiReport()
report.loadFile(url_for('static', filename='reports/SimpleList.mrt'))
report.render()
report.print()
```

Информация

Печать отчета не вызывает его автоматическое построение, поэтому для загруженного шаблона отчета требуется предварительно вызвать метод `render()`, который построит отчет. Для готовых документов (построенных отчетов) указанный метод не требуется.

По умолчанию будут напечатаны все страницы построенного отчета. Предусмотрена возможность указать страницу либо диапазон страниц для печати. Для этого достаточно передать необходимое значение в качестве параметра для функции `print()`, например:

app.py

```
report = StiReport()
report.loadFile(url_for('static', filename='reports/SimpleList.mrt'))

report.render()
report.print(5)
report.print('1,3-8')
```

10.1.14 Экспорт отчета из кода

В генераторе отчетов доступна возможность экспортирования построенного отчета либо дашборда в различные форматы. В таблице ниже перечислены все

© 2003-2026 Stimulsoft

доступные форматы экспорта для отчетов и дашбордов:

Формат экспорта	Отчеты	Дашборды
Document (Snapshot)	+	+
Adobe PDF	+	+
XPS (XML Paper Specification)	+	-
Microsoft PowerPoint	+	-
HTML	+	+
HTML5	+	-
Text	+	-
Microsoft Word	+	-
Microsoft Excel	+	+
OpenDocument Writer	+	-
OpenDocument Calc	+	-
RTF (Rich Text Format)	+	-

Формат данных	Отчеты	Дашборды
CSV (Comma Separated Value)	+	+
JSON (JavaScript Object Notation)	+	+
XML (Extensible Markup Language)	+	+
DBF (dBase/FoxPro)	+	+
DIF	+	+
SYLK (Symbolic Link)	+	+

Формат изображения	Отчеты	Дашборды
PNG (Portable Network Graphics)	+	+

© 2003-2026 Stimulsoft

JPEG (Joint Photographic Experts Group)	+	+
GIF (Graphics Interchange)	+	+
TIFF (Tagged Image File Format)	+	+
SVG (Scalable Vector Graphics)	+	+
SVGZ (Compressed SVG)	+	+
PCX (Picture Exchange)	+	+
BMP (Windows Bitmap)	+	+

Для экспортирования отчета необходимо воспользоваться методом `exportDocument()` у объекта отчета:

app.py

```
from stimulsoft_reports.report import StiReport
from stimulsoft_reports.report.enums import StiExportFormat

report = StiReport()
report.loadFile(url_for('static', filename='reports/SimpleList.mrt'))
report.render()
report.exportDocument(StiExportFormat.PDF)
```

Полный код примера доступен на GitHub.

Информация

Экспортирование отчета не вызывает его автоматическое построение, поэтому для загруженного шаблона отчета требуется предварительно вызвать метод `render()`, который построит отчет. Для готовых документов (построенных отчетов) вызов указанной функции не требуется.

© 2003-2026 Stimulsoft

В качестве аргументов функции `exportDocument()` необходимо указать требуемый формат экспорта из перечисления `StiExportFormat`, а так же при необходимости настройки экспортирования. Доступны следующие варианты формата:

Формат экспорта	Описание
`StiExportFormat.DOCUMENT`	Сохранение документа (построенного отчета).
`StiExportFormat.PDF`	Сохранение в Adobe PDF формат.
`StiExportFormat.XPS`	Сохранение в XPS формат.
`StiExportFormat.POWERPOINT`	Сохранение в Microsoft PowerPoint формат.
`StiExportFormat.HTML`	Сохранение в HTML формат.
`StiExportFormat.HTML5`	Сохранение в HTML5 формат, с использованием элементов SVG разметки.
`StiExportFormat.TEXT`	Сохранение в текстовый формат.
`StiExportFormat.WORD`	Сохранение в Microsoft Word формат.
`StiExportFormat.EXCEL`	Сохранение в Microsoft Excel формат.
`StiExportFormat.ODT`	Сохранение в OpenDocument Text формат.
`StiExportFormat.ODS`	Сохранение в OpenDocument Spreadsheet формат.
`StiExportFormat.RTF`	Сохранение в RichText формат.
`StiExportFormat.CSV`	Сохранение в CSV формат данных.
`StiExportFormat.JSON`	Сохранение в JSON формат данных.
`StiExportFormat.XML`	Сохранение в XML формат данных.
`StiExportFormat.DBF`	Сохранение в DBF формат данных.

© 2003-2026 Stimulsoft

StiExportFormat.DIF	Сохранение в DIF формат данных.
StiExportFormat.SYLK	Сохранение в SYLK формат данных.
StiExportFormat.IMAGE_PNG	Сохранение в изображение PNG формата.
StiExportFormat.IMAGE_JPEG	Сохранение в изображение JPEG формата.
StiExportFormat.IMAGE_GIF	Сохранение в изображение GIF формата.
StiExportFormat.IMAGE_TIFF	Сохранение в изображение TIFF формата.
StiExportFormat.IMAGE_SVG	Сохранение в изображение SVG формата.
StiExportFormat.IMAGE_SVGZ	Сохранение в изображение SVGZ формата.
StiExportFormat.IMAGE_PCX	Сохранение в изображение PCX формата.
StiExportFormat.IMAGE_BMP	Сохранение в изображение BMP формата.

После экспорта отчета, полученный поток данных будут передан браузеру на скачивание в виде файла. Имя файла и MIME тип данных будет определен автоматически. Предусмотрена возможность отобразить экспортированный отчет сразу в окне браузера (только для PDF, HTML и изображений), для этого необходимо установить аргумент `openAfterExport` в значение `True`:

app.py

```
report.exportDocument(StiExportFormat.PDF, openAfterExport = True)
```

Экспортирование отчета на стороне сервера

Для переключения генератора отчетов в режим работы на стороне сервера, необходимо установить свойство отчета `engine` в

© 2003-2026 Stimulsoft

значение `StiEngineType.SERVER_NODE_JS`, весь остальной набор событий и методов для работы с отчетом точно такой же, как и при экспортировании отчета на стороне клиента. При экспорте на стороне сервера, метод `exportDocument()` вернет байтовый поток данных экспортированного отчета.

Пример экспортирования отчета в PDF формат на стороне сервера:

app.py

```
from stimulsoft_reports.report import StiReport
from stimulsoft_reports.report.enums import StiEngineType,
StiExportFormat

report = StiReport()
report.engine = StiEngineType.SERVER_NODE_JS
report.loadFile(url_for('static', filename='reports/SimpleList.mrt'))
report.render()
bytes = report.exportDocument(StiExportFormat.PDF)
```

Полный код примера доступен на GitHub.

Предусмотрена возможность автоматически сохранить экспортированный отчет в виде файла на стороне сервера. Для этого необходимо указать путь сохранения в аргументе `filePath` метода экспорта. В этом случае, метод `exportDocument()` вернет булевый результат экспортирования и сохранения файла.

Пример экспортирования отчета в формат PDF на стороне сервера и сохранение результата в указанную директорию:

app.py

```
exportedFilePath = url_for('static', filename='reports/SimpleList.pdf')
bytes = report.exportDocument(StiExportFormat.PDF, filePath =
exportedFilePath)
```

Настройки экспортирования отчета

При экспорте отчета из кода доступна возможность установить необходимые

© 2003-2026 Stimulsoft

настройки экспортирования. Для каждого формата экспорта предназначен свой класс настроек, указанный в таблице ниже:

Формат экспорта отчета	Класс настроек
StiExportFormat.DOCUMENT	Настройки не предусмотрены.
StiExportFormat.PDF	StiPdfExportSettings()
StiExportFormat.XPS	StiXpsExportSettings()
StiExportFormat.POWERPOINT	StiPowerPointExportSettings()
StiExportFormat.HTML StiExportFormat.HTML5	StiHtmlExportSettings()
StiExportFormat.TEXT	StiTxtExportSettings()
StiExportFormat.WORD	StiWordExportSettings()
StiExportFormat.EXCEL	StiExcelExportSettings()
StiExportFormat.ODT	StiOdtExportSettings()
StiExportFormat.ODS	StiOdsExportSettings()
StiExportFormat.RTF	StiRtfExportSettings()
StiExportFormat.CSV StiExportFormat.JSON StiExportFormat.XML StiExportFormat.DBF StiExportFormat.DIF StiExportFormat.SYLK	StiDataExportSettings()
StiExportFormat.IMAGE_PNG StiExportFormat.IMAGE_JPEG StiExportFormat.IMAGE_GIF StiExportFormat.IMAGE_TIFF StiExportFormat.IMAGE_SVG StiExportFormat.IMAGE_SVGZ	StiImageExportSettings()

© 2003-2026 Stimulsoft

StiExportFormat.IMAGE_PCX StiExportFormat.IMAGE_BMP	

Формат экспорта дашборда	Класс настроек
StiExportFormat.PDF	StiPdfDashboardExportSettings()
StiExportFormat.HTML	StiHtmlDashboardExportSettings()
StiExportFormat.EXCEL	StiExcelDashboardExportSettings()
StiExportFormat.CSV	StiDataDashboardExportSettings()
StiExportFormat.IMAGE_SVG	StiImageDashboardExportSettings()

Для применения настроек экспортирования, необходимо передать подготовленный объект настроек в качестве параметра методу экспорта отчета. Все остальные действия будут выполнены автоматически.

Пример экспортирования отчета в формат PDF на стороне сервера, в настройках указываются сведения о компании и разрешается редактирование:

app.py

```
from stimulsoft_reports.report import StiReport
from stimulsoft_reports.report.enums import StiEngineType, StiExportFormat
from stimulsoft_reports.export import StiPdfExportSettings
from stimulsoft_reports.export.enums import StiPdfAllowEditable

report = StiReport()
report.engine = StiEngineType.SERVER_NODE_JS
report.loadFile(url_for('static', filename='reports/SimpleList.mrt'))
report.render()

settings = StiPdfExportSettings()
settings.creatorString = 'Stimulsoft'
settings.allowEditable = StiPdfAllowEditable.YES

bytes = report.exportDocument(StiExportFormat.PDF, settings)
```

© 2003-2026 Stimulsoft

Полный код примера доступен на GitHub.

Информация

Класс объекта настроек экспорта должен соответствовать формату экспорта, в противном случае приоритетным будет считаться формат, возвращаемый методом `getExportFormat()` объекта настроек экспорта.

10.1.15 События генератора отчетов

Генератор отчетов поддерживает события, которые предоставляют возможность выполнить необходимые операции перед определенными действиями - как на стороне JavaScript клиента, включая платформу Node.js, так и на стороне Python сервера.

Для вызова события на стороне JavaScript клиента, необходимо добавить к событию название определенной в HTML шаблоне JavaScript функции в виде строки. При отсутствии HTML шаблона, а также при построении либо экспорте отчета на стороне сервера при помощи Node.js платформы, вместо названия функции необходимо к событию добавить саму функцию в виде строки или строк кода. Аргументы события будут находиться в предварительно заданной переменной `args`, которую можно использовать в коде события.

Для вызова события на стороне Python сервера, необходимо добавить к событию саму Python функцию, определенную ранее в коде. Аргументы события будут переданы в параметрах функции.

Для одного события можно добавлять любое количество функций, как клиентских, так и серверных. Пример различных вариантов добавления функций разного типа к событию:

© 2003-2026 Stimulsoft

app.py

```
def prepareVariables(args: StiVariablesEventArgs):
    variables = args.variables

report = StiReport()
report.onPrepareVariables += prepareVariables
report.onPrepareVariables += 'prepareVariables'
report.onBeforeRender += 'args.report.dictionary.clear();'
report.onAfterRender += 'afterRender'
report.loadFile(url_for('static', filename='reports/SimpleList.mrt'))
report.render()
```

report.html

```
<script>
    function prepareVariables(args) {
        let variables = args.variables;
    }

    function afterRender(args) {
        alert("The report rendering is completed.");
    }
</script>
```

В зависимости от события, оно может быть вызвано на стороне JavaScript клиента и на стороне PHP сервера одновременно, либо только на стороне JavaScript клиента, либо только на стороне Python сервера. Это связано с архитектурой компонентов - для построения и экспортирования отчета используется JavaScript ядро генератора, которое работает на стороне клиента, а для работы с данными используется серверный Python код. Эти две составляющие не имеет прямого доступа друг к другу, поэтому события работают раздельно, а передачу данных обеспечивает обработчик событий. В описании каждого события будет указано, какой именно тип может быть использован. Подробное описание обработчика событий и примеры использования находится в разделе Обработчик событий.

© 2003-2026 Stimulsoft

Информация

Все события, работающие на стороне JavaScript клиента, в равной степени работают на стороне Node.js сервера, т.к. в этом случае так же используется JavaScript ядро отчетов.

Некоторые аргументы событий принимают значения из перечислений, которые находятся в определенных пространствах имен. Все используемые в событиях генератора отчетов перечисления указаны в блоке кода ниже:

app.py

```
from stimulsoft_reports.enums import StiDataCommand, StiDatabaseType,
StiEventType
```

Генератор отчетов поддерживает следующие события:

- onDatabaseConnect
- onBeforeRender
- onAfterRender
- onBeginProcessData
- onEndProcessData
- onPrepareVariables

onDatabaseConnect
[-] JavaScript [+] Python

Событие вызывается перед подключением к базе данных после получения всех параметров. Подробное описание и примеры использования находятся в разделе и Подключение SQL адаптеров данных. В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне Python сервера:

© 2003-2026 Stimulsoft

Наименование	Описание
event	Идентификатор текущего события, для данного события имеет значение StiEventType.DATABASE_CONNECT
sender	Компонент, который инициировал данное событие, может иметь следующие типы: • StiViewer • StiDesigner
database	Тип базы данных, может принимать одно из значений перечисления StiDatabaseType.
driver	Название используемого Python драйвера базы данных.
info	Параметры подключения к базе данных, полученные из строки подключения.
link	Идентификатор подключения к базе данных. По умолчанию имеет значение None, в этом случае подключение будет создано адаптером данных.

onBeforeRender
[+] JavaScript [+] Python

Событие вызывается перед построением отчета. Подробное описание и примеры использования находятся в разделе и Построение отчета.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне JavaScript клиента:

Наименование	Описание
event	Идентификатор текущего события,

© 2003-2026 Stimulsoft

	имеет значение "BeforeRender".
sender	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • "Report"
report	Текущий объект отчета.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне Python сервера:

Наименование	Описание
event	Идентификатор текущего события, для данного события имеет значение StiEventType.BEFORE_RENDER
sender	Компонент, который инициировал данное событие, может иметь следующие типы: • StiReport
report	Текущий объект отчета.
regReportData(name, data, synchronize = False)	Метод, позволяющий передать данные в отчет. В качестве аргументов может принимать следующие значения: name – имя источника данных в отчете; data – данные в виде XML либо JSON строки, либо в виде Python объекта или словаря; synchronize – флаг, указывающий на необходимость синхронизации данных, по умолчанию False.

© 2003-2026 Stimulsoft

onAfterRender
[+] JavaScript [-] Python

Событие вызывается после построения отчета. Подробное описание и примеры использования находятся в разделе и Построение отчета.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне JavaScript клиента:

Наименование	Описание
`event`	Идентификатор текущего события, имеет значение "`AfterRender`".
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • "`Report`"
`report`	Текущий объект отчета.

onBeginProcessData
[+] JavaScript [+] Python

Событие вызывается перед запросом данных, необходимых для построения отчета. Подробное описание и примеры использования находятся в разделах Подключение файловых данных и Подключение SQL адаптеров данных.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне JavaScript клиента:

Наименование	Описание
`event`	Идентификатор текущего события, имеет значение "`BeginProcessData`".
`sender`	Идентификатор компонента,

© 2003-2026 Stimulsoft

	который инициировал данное событие, может принимать следующие значения: • "Report" • "Viewer" • "Designer"
report	Текущий объект отчета.
command	Идентификатор текущей команды, может принимать следующие значения: • "TestConnection" - выполняется проверка соединения; • "ExecuteQuery" - выполняется запрос данных из указанного SQL источника; • "GetSchema" - выполняется чтение XSD схемы из файлового источника; • "GetData" - выполняется чтение данных из файлового источника.
connection	Имя текущего подключения к источнику данных, заданное в шаблоне отчета.
database	Наименование текущей базы данных. Может принимать следующие значения: • "XML" • "JSON" • "Excel" • "CSV" • "MySQL" • "MS SQL" • "PostgreSQL"

© 2003-2026 Stimulsoft

	• "Firebird" • "Oracle" • "MongoDB" • "ODBC"
dataSource	Имя текущего источника данных, заданное в шаблоне отчета. Устанавливается только для SQL источников данных.
connectionString	Строка подключения к SQL источнику данных.
queryString	SQL запрос на получение данных. Используется только с командой ExecuteQuery.
pathData	Путь к файлу источника данных, заданный в шаблоне отчета. Устанавливается только для XML и JSON источников данных.
pathSchema	Путь к файлу схемы данных, заданный в шаблоне отчета. Устанавливается только для XML источника данных.
parameters	Коллекция параметров и их значений, заданная в SQL источнике данных.
preventDefault	Данный флаг предоставляет возможность остановить дальнейшую обработку события генератором отчетов. По умолчанию установлено значение false.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне Python сервера:

© 2003-2026 Stimulsoft

Наименование	Описание
`event`	Идентификатор текущего события, для данного события имеет значение `StiEventType.BEGIN_PROCESS_DATA`
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • `StiHandler` • `StiReport` • `StiViewer` • `StiDesigner`
`command`	Идентификатор текущей команды. Может принимать следующие значения: • `StiDataCommand.TEST_CONNECTION` - выполняется проверка соединения; • `StiDataCommand.RETRIEVE_SCHEMA` - выполняется запрос схемы базы данных для NoSQL источников данных; • `StiDataCommand.EXECUTE_QUERY` - выполняется запрос данных из указанного SQL или NoSQL источника; • `StiDataCommand.EXECUTE` - выполняется хранимая процедура указанного SQL источника.
`connection`	Имя текущего подключения к источнику данных, заданное в шаблоне отчета.
`database`	Наименование текущей базы

© 2003-2026 Stimulsoft

	данных. Может принимать следующие значения: • StiDatabaseType.XML • StiDatabaseType.JSON • StiDatabaseType.CSV • StiDatabaseType.MYSQL • StiDatabaseType.MSSQL • StiDatabaseType.POSTGRESQL • StiDatabaseType.FIREBIRD • StiDatabaseType.ORACLE • StiDatabaseType.MONGODB • StiDatabaseType.ODBC
dataSource	Имя текущего источника данных, заданное в шаблоне отчета.
connectionString	Строка подключения к SQL источнику данных.
queryString	SQL запрос на получение данных. Используется только с командой StiDataCommand.EXECUTE_QUERY.
pathData	Путь к файлу источника данных, заданный в шаблоне отчета.
pathSchema	Путь к файлу схемы данных, заданный в шаблоне отчета. Устанавливается только для XML источника данных.
parameters	Коллекция параметров и их значений, заданная в SQL источнике данных. Значения параметров всегда передаются в первоначальном (не экранированном) виде.

© 2003-2026 Stimulsoft

onEndProcessData
[+] JavaScript [+] Python

Событие вызывается после загрузки данных перед построением отчета. Подробное описание и примеры использования находятся в разделах Подключение файловых данных и Подключение SQL адаптеров данных.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне JavaScript клиента:

Наименование	Описание
`event`	Идентификатор текущего события, имеет значение `"EndProcessData"`.
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • `"Report"` • `"Viewer"` • `"Designer"`
`report`	Текущий объект отчета.
`command`	Идентификатор текущей команды, может принимать следующие значения: • `"ExecuteQuery"` - данные получены из указанного SQL источника. • `"GetData"` - данные получены из файлового источника.
`database`	Наименование текущей базы данных. Может принимать следующие значения: • `"XML"`

© 2003-2026 Stimulsoft

	• "JSON" • "Excel" • "CSV" • "MySQL" • "MS SQL" • "PostgreSQL" • "Firebird" • "Oracle" • "MongoDB" • "ODBC"
connection	Имя текущего подключения к источнику данных, заданное в шаблоне отчета.
dataSource	Имя текущего источника данных, заданное в шаблоне отчета. Устанавливается только для SQL источников данных.
dataSet	Подготовленный объект Stimulsoft.System.Data.DataSet, содержащий таблицы и строки данных, полученных из файлового источника.
result	Коллекция колонок и их типов, а также строк данных, полученных из SQL источника.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне Python сервера:

Наименование	Описание
event	Идентификатор текущего события, для данного события имеет значение

© 2003-2026 Stimulsoft

	StiEventType.END_PROCESS_DATA
sender	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • StiHandler • StiReport • StiViewer • StiDesigner
command	Идентификатор текущей команды, может принимать следующие значения: • StiDataCommand.TEST_CONNECTION - выполняется проверка соединения; • StiDataCommand.RETRIEVE_SCHEMA - выполняется запрос схемы базы данных для NoSQL источников данных; • StiDataCommand.EXECUTE_QUERY - выполняется запрос данных из указанного SQL или NoSQL источника; • StiDataCommand.EXECUTE - выполняется хранимая процедура указанного SQL источника.
connection	Имя текущего подключения к источнику данных, заданное в шаблоне отчета.
database	Наименование текущей базы данных. Может принимать следующие значения: • StiDatabaseType.XML • StiDatabaseType.JSON

© 2003-2026 Stimulsoft

	• StiDatabaseType.CSV • StiDatabaseType.MYSQL • StiDatabaseType.MSSQL • StiDatabaseType.POSTGRESQL • StiDatabaseType.FIREBIRD • StiDatabaseType.ORACLE • StiDatabaseType.MONGODB • StiDatabaseType.ODBC
dataSource	Имя текущего источника данных, заданное в шаблоне отчета. Устанавливается только для SQL источников данных.
queryString	Итоговый SQL запрос со всеми параметрами, который был использован на получение данных. Используется только с командой StiDataCommand.EXECUTE_QUERY
result	Коллекция колонок и их типов, а также строк данных, полученных из SQL источника, либо строковые данные, полученные из файлового источника.

onPrepareVariables
[+] JavaScript [+] Python

Событие вызывается перед построением отчета после подготовки переменных отчета. Подробное описание и примеры использования находятся в разделе Работа с переменными отчета.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне JavaScript клиента:

© 2003-2026 Stimulsoft

Наименование	Описание
`event`	Идентификатор текущего события, имеет значение `"PrepareVariables"`.
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • `"Report"` • `"Viewer"` • `"Designer"`
`report`	Текущий объект отчета.
`variables`	Коллекция переменных отчета и их значений.
`preventDefault`	Данный флаг предоставляет возможность остановить дальнейшую обработку события генератором отчетов. По умолчанию установлено значение `false`.

В таблице ниже представлен список свойств, передаваемых в аргументах события на стороне Python сервера:

Наименование	Описание
`event`	Идентификатор текущего события, для данного события имеет значение `StiEventType.PREPARE_VARIABLES`
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • `StiHandler`

© 2003-2026 Stimulsoft

	• StiReport • StiViewer • StiDesigner
variables	Коллекция переменных отчета и их значений.

10.2 HTML5 вьювер

Вьювер отчетов представляет собой Python компонент `StiViewer`, предназначенный для просмотра, печати и экспортирования отчетов в окне браузера на любом компьютере с любой установленной операционной системой. Вьювер поддерживает множество тем оформления, анимированный интерфейс, закладки, интерактивные отчеты, редактирование элементов отчета на странице, полноэкранный режим, поиск, и другие, необходимые для просмотра отчетов, возможности.

Вьювер может вывести на просмотр как шаблон отчета, как уже построенный отчет. В случае использования шаблона отчета, вьювер построит его автоматически, используя JavaScript генератор отчетов. Реализована полная поддержка работы с интерактивными аналитическими панелями индикаторов (дашбордами). Работа генератора более подробно рассмотрена в разделе Генератор отчетов.

Интерфейс вьювера построен с использованием HTML5, что позволяет использовать его практически на любой современной платформе. Компонент использует технологию AJAX для выполнения всех действий (загрузка и построение отчета, подключение к данным, листания страниц и масштабирования, интерактивности в отчетах, печать, экспортирование и др.), что позволяет избавиться от перезагрузки всей страницы и повысить скорость работы, а также использовать компонент в One-Page приложениях. Технология JavaScript, применяемая для построения отчетов, позволяет использовать практически любую серверную сторону невысокой производительности.

© 2003-2026 Stimulsoft

Информация

Вследствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово “отчет”.

- Развертывание
- Активация лицензии
- Отображение отчета и дашборда
- Локализация
- Печать отчета
- Экспорт отчета и дашборда
- Режимы отображения
- Работа с переменными отчета
- Работа с закладками
- Динамическое сворачивание, сортировка и детализация
- Редактирование построенного отчета
- Отправка отчета по Email
- Вызов дизайнера
- Визуальное оформление
- События вьювера
- Настройки вьювера

10.2.1 Использование

Примеры

Полный код примера можно найти на GitHub по данной ссылке.

Для использования вьювера достаточно установить пакет `stimulsoft-reports` либо `stimulsoft-dashboards` при помощи установщика пакетов, выполнив следующую команду:

console

© 2003-2026 Stimulsoft

```
python -m pip install stimulsoft-reports
```

console

```
python -m pip install stimulsoft-dashboards
```

По умолчанию устанавливается минимальный набор драйверов данных. Необходимые драйверы можно дополнительно установить вручную, либо воспользоваться командой для установки всех доступных драйверов:

console

```
python -m pip install stimulsoft-reports[ext]
```

console

```
python -m pip install stimulsoft-dashboards[ext]
```

Будет установлена последняя доступная версия продукта в текущее рабочее пространство, после этого можно использовать классы и функции по работе с отчетами и дашбордами.

Информация

Для примеров кода в документации использован **Flask** фреймворк, как один из самых популярных и простых для понимания кода. Можно использовать любой Web-фреймворк, все классы и функции для работы с компонентами являются универсальными.

Для добавления вьювера в Web-проект, предназначен класс `StiViewer`. При помощи данного класса можно создать объект вьювера, установить

© 2003-2026 Stimulsoft

необходимые настройки, обработать запрос и вернуть результат его выполнения, получить подготовленный JavaScript и HTML код компонента. Пример отображения вьювера на HTML странице:

app.py

```
from flask import Flask, render_template, url_for, request
from stimulsoft_reports.viewer import StiViewer

app = Flask(__name__)

@app.route('/viewer', methods = ['GET', 'POST'])
def viewer():
    viewer = StiViewer()
    viewer.options.appearance.fullScreenMode = True

    if viewer.processRequest(request):
        return viewer.getFrameworkResponse()

    js = viewer.javascript.getHtml()
    html = viewer.getHtml()
    return render_template('viewer.html', viewerJavaScript = js,
viewerHtml = html)
```

viewer.html

```
<!DOCTYPE html>
<html>

<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>Showing a Report in the Viewer</title>

  {{ viewerJavaScript|safe }}
</head>

<body>
  {{ viewerHtml|safe }}
</body>

</html>
```

В этом примере создается экземпляр объекта `StiViewer` и задаются необходимые настройки.

Специальная функция компонента `processRequest(request)` выполняет

© 2003-2026 Stimulsoft

обработку текущего запроса. Если функция возвращает `True`, значит запрос успешно обработан, и необходимо вернуть результат его выполнения. Более подробно про это рассказано в разделе Обработчик событий.

Далее, необходимо сгенерировать подготовленный JavaScript и HTML код, необходимый для работы вьювера. Функция `viewer.javascript.getHtml()` генерирует HTML код для подключения необходимых скриптов и ресурсов, функция `viewer.getHtml()` генерирует HTML код самого компонента. Сгенерированный код передается в качестве параметров в HTML шаблон viewer.html, и выводится в заданных местах.

Информация

Наши продукты Stimulsoft Reports.PYTHON и Stimulsoft Dashboards.PYTHON не имеют нативного ядра генератора отчетов на Python, построение и экспорт отчета выполняется на стороне клиента при помощи JavaScript кода. При использовании Python кода для работы с компонентами, необходимо вызвать функцию `getHtml()`, которая вернет весь необходимый JavaScript и HTML код для работы генератора отчетов и компонентов.

Предусмотрено упрощенное развертывание вьювера без использования HTML шаблона страницы. Например, этот же пример можно реализовать, использовав только Python код:

© 2003-2026 Stimulsoft

app.py

```
from flask import Flask, url_for, request
from stimulsoft_reports.viewer import StiViewer

app = Flask(__name__)

@app.route('/viewer', methods = ['GET', 'POST'])
def viewer():
    viewer = StiViewer()
    viewer.options.appearance.fullScreenMode = True

    if viewer.processRequest(request):
        return viewer.getFrameworkResponse()

    # Here is the code for working with the report

    return viewer.getFrameworkResponse()
```

В данном случае, после вызова обработчика событий и действий с отчетом, сразу возвращается финальный результат запроса. Т.е. вьювер вернет полностью готовую HTML страницу со всеми необходимыми скриптами и кодом. Также, предусмотрены другие варианты работы, более подробно про это рассказано в разделе Обработчик событий.

Информация

В большинстве случаев для работы с продуктом достаточно использовать только Python код, который обеспечивает взаимодействие со всеми основными возможностями компонентов. Для детальной настройки продукта и использования всех возможностей JS генератора, необходимо использовать JavaScript код. Вариант развертывания продукта при помощи только JavaScript кода описан в разделе Reports and Dashboards for JS, в этом случае использование Python кода требуется только для подключения адаптеров данных.

Для формирования ответа используется функция `getFrameworkResponse()`, которая возвращает готовый объект, подходящий для используемого в данный момент Web-фреймворка. Текущий фреймворк определяется в момент обработки запроса. Наши компоненты поддерживают работу с такими

© 2003-2026 Stimulsoft

популярными фреймворками, как **Django**, **Flask**, **Tornado**. Для обработки запроса и формирования ответа в Web-проектах, не использующих указанные фреймворки, необходимо передать текущие GET и POST данные в функцию обработки запроса, а для формирования ответа воспользоваться функцией `getResponse()`, которая вернет все необходимые данные. Например, обработку событий вьювера можно реализовать указанным ниже способом:

app.py

```
from flask import Flask, make_response, render_template, url_for, request
from stimulsoft_reports.viewer import StiViewer

app = Flask(__name__)

@app.route('/viewer', methods = ['GET', 'POST'])
def viewer():
    viewer = StiViewer()
    viewer.options.appearance.fullScreenMode = True

    query = request.args.to_dict()
    body = request.get_data(False)
    if viewer.processRequest(None, query, body):
        viewerResponse = viewer.getResponse()
        response = make_response(viewerResponse.data)
        response.mimetype = viewerResponse.mimetype
        response.headers.add('Access-Control-Allow-Origin',
request.origin)
        return response

    js = viewer.javascript.getHtml()
    html = viewer.getHtml()
    return render_template('viewer.html', viewerJavaScript = js,
viewerHtml = html)
```

10.2.2 Активация лицензии

После приобретения продукта вам необходимо активировать лицензию для используемых компонентов. Предусмотрено несколько способов подключения лицензионного ключа.

Все варианты активации компонентов описаны в разделе Активация лицензии генератора отчетов, имеют одинаковые функции и параметры вызова.

© 2003-2026 Stimulsoft

10.2.3 Отображение отчета

Примечание

При присвоении отчета вьювер компоненту, происходит автоматическое построение отчета. Вызов метода `report.render()` необходим только в том случае, если требуется провести какие-либо действия с построенным отчетом перед его отображением во вьювере.

Для отображения отчета во вьювере, достаточно создать объект `StiReport`, загрузить в него шаблон отчета, и присвоить полученный объект вьюверу. Все остальные действия будут выполнены автоматически, вьювер построит отчет и отобразит первую страницу:

app.py

```
from flask import Flask, url_for, request
from stimulsoft_reports.report import StiReport
from stimulsoft_reports.viewer import StiViewer

app = Flask(__name__)

@app.route('/viewer', methods = ['GET', 'POST'])
def viewer():
    viewer = StiViewer()
    viewer.options.appearance.fullScreenMode = True

    if viewer.processRequest(request):
        return viewer.getFrameworkResponse()

    report = StiReport()
    report.loadFile(url_for('static', filename='reports/SimpleList.mrt'))
    viewer.report = report

    return viewer.getFrameworkResponse()
```

Вьювер сам выполняет построение отчета и умеет отображать как шаблоны отчетов, так и документы (построенные отчеты). Подробное описание работы с различными форматами отчетов и документов находится в разделе Загрузка и сохранение отчета.

© 2003-2026 Stimulsoft

10.2.4 Локализация

Вьювер поддерживает полную локализацию своего интерфейса. Для локализации интерфейса на требуемый язык, достаточно установить требуемое название файла для опции `localization` у вьювера:

app.py

```
from stimulsoft_reports.viewer import StiViewer

viewer = StiViewer()
viewer.options.localization = 'de.xml'
```

Все доступные локализационные XML файлы находятся в ресурсах установленного пакета продукта. При необходимости, локализационный файл можно загрузить из любого другого места, для этого необходимо для опции `localization` указать полный путь к нужному XML файлу:

app.py

```
from stimulsoft_reports.viewer import StiViewer

viewer = StiViewer()
viewer.options.localization = '/resources/loc/de.xml'
```

Если файл доступен для чтения из Python приложения, локализация будет загружена во вьювер. Иначе будет использована встроенная английская локализация интерфейса.

10.2.5 Печать отчета

Во вьювере предусмотрено несколько вариантов печати отчета. Каждый имеет свои особенности, достоинства и недостатки.

Печать в PDF (Print to PDF)

Печать будет осуществляться посредством экспортирования отчета в PDF формат. К достоинствам можно отнести большую точность расположения и печати элементов отчета по сравнению с остальными вариантами печати. Из недостатков можно упомянуть обязательное наличие установленного в

© 2003-2026 Stimulsoft

браузере плагина для просмотра PDF файлов (современные браузеры имеют встроенное средство просмотра и печати PDF файлов).

Печать с просмотром (Print with Preview)
Печать отчета будет осуществлена в отдельном всплывающем окне браузера в HTML формат. Отчет можно предварительно просмотреть, а затем отправить на принтер или скопировать в другое место в виде текста или HTML-кода. К достоинствам можно отнести кроссбраузерность при печати, отсутствие необходимости установки специальных плагинов. Недостатком является относительно невысокая точность расположения элементов отчета, обусловленная особенностями реализации HTML-форматирования.

Печать без просмотра (Print without Preview)
Печать отчета будет осуществлена непосредственно на принтер без предварительного просмотра. После выбора этого пункта меню отображается системный диалог печати. Так как печать в этом режиме осуществляется в HTML формат, то качество печати аналогично качеству печати отчета с предварительным просмотром.

Информация

Печать отчета осуществляется с использованием встроенных методов используемого браузера, поэтому представление диалогового окна может отличаться в разных операционных системах и браузерах. Также, браузер не позволяет управлять настройками печати из JavaScript кода, поэтому требуемые настройки необходимо будет выполнить в самом диалоговом окне.

Настройки печати отчета
При выборе печати отчета на панели вьювера, отображается меню с выбором варианта печати. Компонент имеет возможность принудительно установить требуемый режим печати. Для этого достаточно установить свойство `printDestination` в одно из указанных ниже значений из перечисления `StiPrintDestination`:

Наименование	Описание

© 2003-2026 Stimulsoft

`StiPrintDestination.DEFAULT`	При выборе печати будет отображено меню с доступными вариантами печати (значение свойства по умолчанию).
`StiPrintDestination.PDF`	Печать в PDF формат.
`StiPrintDestination.DIRECT`	Печать в HTML формат непосредственно на принтер, будет отображен системный диалог печати.
`StiPrintDestination.WITH_PREVIEW`	Печать в HTML формат с предварительным просмотром во всплывающем окне.

Например, требуется установить режим печати только в PDF формат:

app.py

```
from stimulsoft_reports.viewer import StiViewer
from stimulsoft_reports.viewer.enums import StiPrintDestination

viewer = StiViewer()
viewer.options.toolbar.printDestination = StiPrintDestination.PDF
```

Вьювер имеет возможность полностью отключить печать отчета, если она не требуется. Для этого необходимо установить значение `False` для свойства `showPrintButton`:

app.py

```
from stimulsoft_reports.viewer import StiViewer

viewer = StiViewer()
viewer.options.toolbar.showPrintButton = False
```

Событие печати отчета

© 2003-2026 Stimulsoft

При необходимости выполнения каких-либо действий перед печатью отчета, предназначено событие `onPrintReport`. В аргументах события будут переданы тип печати отчета и сам отчет, отправленный на печать.

app.py

```
from stimulsoft_reports.viewer import StiViewer
from stimulsoft_reports.events import StiReportEventArgs

def printReport(args: StiReportEventArgs):
    printAction = args.printAction
    report = args.report

viewer = StiViewer()
viewer.onPrintReport += printReport
viewer.onPrintReport += 'printReport'
```

viewer.html

```
<script>
    function printReport(args) {
        let printAction = args.printAction;
        let report = args.report;
    }
</script>
```

Подробное описание доступных значений аргументов находится в разделе События вьювера.

Печать отчета из кода

Предусмотрена возможность распечатать отчет из кода без использования функций вьювера. Подробное описание данного функционала находится в разделе генератора отчетов Печать отчета из кода.

10.2.6 Экспорт отчета

Вьювер позволяет экспортировать отображаемый отчет либо дашборд в различные форматы. Для работы экспорта не требуется дополнительных настроек вьювера. В таблице ниже перечислены все доступные форматы экспорта для отчетов и дашбордов:

© 2003-2026 Stimulsoft

Формат экспорта	Отчеты	Дашборды
Document (Snapshot)	+	+
Adobe PDF	+	+
XPS (XML Paper Specification)	+	-
Microsoft PowerPoint	+	-
HTML	+	+
HTML5	+	-
Text	+	-
Microsoft Word	+	-
Microsoft Excel	+	+
OpenDocument Writer	+	-
OpenDocument Calc	+	-
RTF (Rich Text Format)	+	-

Формат данных	Отчеты	Дашборды
CSV (Comma Separated Value)	+	+
JSON (JavaScript Object Notation)	+	+
XML (Extensible Markup Language)	+	+
DBF (dBase/FoxPro)	+	+
DIF	+	+
SYLK (Symbolic Link)	+	+

Формат изображения	Отчеты	Дашборды
PNG (Portable Network Graphics)	+	+
JPEG (Joint Photographic	+	+

© 2003-2026 Stimulsoft

Experts Group)		
GIF (Graphics Interchange)	+	+
TIFF (Tagged Image File Format)	+	+
SVG (Scalable Vector Graphics)	+	+
SVGZ (Compressed SVG)	+	+
PCX (Picture Exchange)	+	+
BMP (Windows Bitmap)	+	+

Событие начала экспорта

При необходимости выполнения каких-либо действий перед экспортированием отчета, предназначено событие `onBeginExportReport`. Событие будет вызвано после отображения диалогового окна экспорта с настройками. В аргументах события будут переданы тип экспорта отчета, сам отчет, а также все настройки выбранного экспорта. Допускается изменение отчета либо его параметров, настроек экспорта и название файла.

app.py

```
from stimulsoft_reports.report.enums import StiExportFormat
from stimulsoft_reports.viewer import StiViewer
from stimulsoft_reports.events import StiExportEventArgs
from stimulsoft_reports.export import StiPdfExportSettings

def beginExportReport(args: StiExportEventArgs):
    if args.format == StiExportFormat.PDF:
        settings: StiPdfExportSettings = args.settings
        settings.imageQuality = 0.90
        settings.imageResolution = 200

viewer = StiViewer()
viewer.onBeginExportReport += beginExportReport
viewer.onBeginExportReport += 'beginExportReport'
```

© 2003-2026 Stimulsoft

viewer.html

```
<script>
    function beginExportReport(args) {
        if (args.format == Stimulsoft.Report.StiExportFormat.Pdf) {
            args.settings.imageQuality = 0.90;
            args.settings.imageResolution = 200;
        }
    }
</script>
```

Подробное описание доступных значений аргументов находится в разделе События вьювера.

Событие завершения экспорта

При необходимости выполнения каких-либо действий после экспортирования отчета перед его сохранением, предназначено событие `onEndExportReport`. В аргументах события будут переданы тип экспорта отчета, а также имя и байтовые данные экспортированного файла. Допускается изменение имени и байтовых данных экспортированного файла.

app.py

```
from stimulsoft_reports.report.enums import StiExportFormat
from stimulsoft_reports.viewer import StiViewer
from stimulsoft_reports.events import StiExportEventArgs

def endExportReport(args: StiExportEventArgs):
    if args.format == StiExportFormat.HTML:
        htmlText = args.data

viewer = StiViewer()
viewer.onEndExportReport += endExportReport
viewer.onEndExportReport += 'endExportReport'
```

© 2003-2026 Stimulsoft

viewer.html

```
<script>
    function endExportReport(args) {
        if (args.format == Stimulsoft.Report.StiExportFormat.Html) {
            htmlText = args.data
        }
    }
</script>
```

Подробное описание доступных значений аргументов находится в разделе События вьювера.

Настройки экспорта

Иногда требуется отключить неиспользуемые форматы экспорта отчета, и оставить только нужные. Это позволяет разгрузить интерфейс и упростить использование вьювера. Для отключения неиспользуемых форматов экспортов достаточно установить значение `False` для соответствующих свойств вьювера, например:

app.py

```
from stimulsoft_reports.viewer import StiViewer

viewer = StiViewer()
viewer.options.exports.showExportToDocument = False
viewer.options.exports.showExportToWord = False
viewer.options.exports.showExportToCsv = False
```

Также, при необходимости, можно полностью убрать отображение диалоговых окон экспорта, экспортирование всегда будет осуществляться с настройками по умолчанию. В этом случае, настройками можно управлять в событии экспорта. Для отключения диалоговых окон достаточно установить значение `False` для свойства `showExportDialog`:

© 2003-2026 Stimulsoft

app.py

```
from stimulsoft_reports.viewer import StiViewer

viewer = StiViewer()
viewer.options.exports.showExportDialog = False
```

Полный список доступных опций находится в разделе Настройка вьювера.

Экспортирование отчета из кода

Также доступен экспорт отчета при помощи кода. Для этого предназначен специальный метод `exportDocument()` у объекта отчета. Подробное описание находится в разделе Экспортирование отчета из кода.

10.2.7 Режимы отображения

Вьювер позволяет настраивать различные режимы отображения интерфейса и страниц отчета, а также управлять отображением на мобильных устройствах.

Полосы прокрутки

Во вьювере предусмотрено два режима отображения отчета: с полосами прокрутки и без них. По умолчанию установлен режим просмотра без полос прокрутки. Для включения режима просмотра с полосами прокрутки достаточно установить значение `True` для свойства `scrollbarsMode`.

app.py

```
from stimulsoft_reports.viewer import StiViewer

viewer = StiViewer()
viewer.options.appearance.scrollbarsMode = True
```

В первом режиме (без полос прокрутки), вьювер отображает страницу или отчет целиком, автоматически растягивая область просмотра. Если заданы размеры по ширине и высоте, то вьювер будет осуществлять обрезку вышедшей за границы страницы. Во втором режиме, в отличие от первого, при выходе страницы за границы размеров вьювера, обрезка осуществляться не будет. Вместо этого появятся полосы прокрутки, при помощи которых можно

© 2003-2026 Stimulsoft

просмотреть страницу или отчет целиком.

Информация

В режиме просмотра отчета с полосами прокрутки необходимо задать высоту вьювера, иначе будет задана высота по умолчанию, равная 650 пикселей.

Полноэкранный режим

Во вьювере предусмотрен режим полноэкранного отображения отчета и дашборда. По умолчанию включен стандартный режим просмотра, вьювер имеет заданные в настройках размеры. Для включения полноэкранного режима просмотра достаточно установить значение `True` для свойства `fullScreenMode`.

app.py

```
from stimulsoft_reports.viewer import StiViewer

viewer = StiViewer()
viewer.options.appearance.fullScreenMode = True
```

Также, для включения либо отключения полноэкранного режима можно воспользоваться соответствующей кнопкой на панели управления вьювера.

Отображение страниц отчета

Во вьювере доступно три режима отображения отчета:

- постраничное отображение;
- отчет целиком в виде лент;
- табличное отображение страниц отчета.

Для управления режимами предназначено свойство `viewMode`, которое соответственно принимает одно из указанных значений:

Наименование	Описание

© 2003-2026 Stimulsoft

StiWebViewMode.SINGLE_PAGE	Отображается одна страница, выбранная на панели инструментов.
StiWebViewMode.CONTINUOUS	Отображаются все страницы отчета в виде ленты.
StiWebViewMode.MULTIPLE_PAGES	Отображаются все страницы отчета в виде таблицы.

Например, требуется установить режим отображения всех страниц в виде ленты:

app.py

```
from stimulsoft_reports.viewer import StiViewer
from stimulsoft_reports.viewer.enums import StiWebViewMode

viewer = StiViewer()
viewer.options.toolbar.viewMode = StiWebViewMode.CONTINUOUS
```

Мобильный режим

Во вьювере реализована поддержка работы как с обычным компьютером, так и с сенсорными экранами, мобильными устройствами. Для управления режимами интерфейса предназначено свойство `interfaceType`, которое принимает одно из следующих значений:

Наименование	Описание
StiInterfaceType.AUTO	Тип интерфейса вьювера будет выбран автоматически в зависимости от используемого устройства (значение по умолчанию).
StiInterfaceType.MOUSE	Принудительное использование стандартного интерфейса для управления вьювером при помощи мыши.

© 2003-2026 Stimulsoft

`StiInterfaceType.TOUCH`	Принудительное использование `Touch` интерфейса для управления вьювером при помощи сенсорного экрана монитора, в этом режиме элементы интерфейса вьювера имеют увеличенные размеры для удобства управления.
`StiInterfaceType.MOBILE`	Принудительное использование `Mobile` интерфейса для управления вьювером при помощи экрана смартфона, в этом режиме интерфейс вьювера имеет упрощенный вид и адаптирован для управления при помощи мобильного устройства.

Например, требуется полностью отключить мобильный режим отображения:

app.py

```
from stimulsoft_reports.viewer import StiViewer
from stimulsoft_reports.viewer.enums import StiInterfaceType

viewer = StiViewer()
viewer.options.appearance.interfaceType = StiInterfaceType.MOUSE
```

10.2.8 Работа с переменными отчета

Для работы с переменными отчета во вьювере реализована поддержка специальной панели параметров. Для добавления параметра на панель необходимо в отчете определить переменную, запрашиваемую у пользователя. При просмотре отчета во вьювере такая переменная будет автоматически добавлена на панель параметров. Поддерживаются все типы переменных отчета (обычные переменные, дата и время, диапазон, списки и др.).

Значения переменных на панели параметров

Для выполнения каких-либо действий перед применением параметров предназначено специальное событие `onInteraction`, которое будет вызвано при любых интерактивных действиях вьювера. В аргументах события будут

© 2003-2026 Stimulsoft

переданы тип действия, объект отчета, а также коллекция переменных и их значений, расположенных на панели параметров. Тип действия в данном случае будет иметь значение `"Variables"`. Подробное описание доступных значений аргументов события находится в разделе События вьювера.

app.py

```
from stimulsoft_reports.viewer import StiViewer

viewer = StiViewer()
viewer.onInteraction += 'interaction'
```

viewer.html

```
<script>
   function interaction(args) {
      if (args.action == "Variables") {
         let variables = args.variables;
      }
   }
</script>
```

Коллекция переменных представляет собой объект, содержащий все переменные панели параметров. Допустимо изменять значения переменных, при этом тип нового значения должен соответствовать типу изменяемой переменной.

viewer.html

```
<script>
   let variables = {
      VariableString: "Text value",
      VariableInt: 20
   }
</script>
```

Прямой доступ к значениям переменных на панели параметров на стороне Python сервера не предусмотрен, событие работает только на стороне JavaScript клиента.

© 2003-2026 Stimulsoft

Настройка панели параметров

Если работа с переменными во вьювере не требуется, то можно полностью отключить данную возможность. Для этого предназначено свойство `showParametersButton`, которому необходимо установить значение `False`.

app.py

```
from stimulsoft_reports.viewer import StiViewer

viewer = StiViewer()
viewer.options.toolbar.showParametersButton = False
```

Информация

При такой конфигурации вьювера панель параметров не будет показана даже в том случае, если параметры присутствуют в отображаемом отчете.

Значения переменных при построении отчета

При необходимости контролирования всех переменных отчета, предназначено специальное событие `onPrepareVariables`, которое будет вызвано перед построением отчета.

app.py

```
from stimulsoft_reports.viewer import StiViewer
from stimulsoft_reports.events import StiReportEventArgs

def prepareVariables(args: StiVariablesEventArgs):
    variables = args.variables

viewer = StiViewer()
viewer.onPrepareVariables += onPrepareVariables
viewer.onPrepareVariables += 'onPrepareVariables'
```

© 2003-2026 Stimulsoft

viewer.html

```
<script>
   function prepareVariables(args) {
      let variables = args.variables;
   }
</script>
```

Подробное описание данного события находится в разделе Работа с переменными отчета в описании генератора отчетов.

10.2.9 Работа с закладками

Во вьювере реализована поддержка закладок отчета. При отображении такого отчета слева от страницы будет отображена панель с закладками. При выборе закладки отчета вьювер осуществит автоматический переход на нужную страницу, а элемент отчета с закладкой будет подсвечен.

Настройка закладок

По умолчанию ширина панели закладок равна 180 пикселей, вьювер позволяет изменить это значение. Для этого предназначено свойство `bookmarksTreeWidth`, значение которого указывается в пикселях.

app.py

```
from stimulsoft_reports.viewer import StiViewer

viewer = StiViewer()
viewer.options.appearance.bookmarksTreeWidth = 300
```

Если работа с закладками отчета не требуется, то можно полностью отключить данную возможность. Для этого предназначено свойство `showBookmarksButton`, которое необходимо установить в значение `False`.

© 2003-2026 Stimulsoft

app.py

```
from stimulsoft_reports.viewer import StiViewer

viewer = StiViewer()
viewer.options.toolbar.showBookmarksButton = False
```

Информация

В этом случае закладки отчета не будут показаны, даже если они присутствуют в отображаемом отчете. Данное свойство не оказывает влияние на печать и экспортирование отчета с закладками.

При печати отчета с закладками, дерево закладок будет скрыто. Если помимо самого отчета требуется распечатать и закладки, то необходимо установить свойство `bookmarksPrint` в значение `True`.

app.py

```
from stimulsoft_reports.viewer import StiViewer

viewer = StiViewer()
viewer.options.appearance.bookmarksPrint = True
```

10.2.10 Динамическое сворачивание, сортировка и детализация

Помимо переменных, значения которых можно задавать на панели параметров, вьювер поддерживает другие виды интерактивности, добавляющей удобства и функциональности при использовании генератора отчетов. Это сортировка, сворачивание и детализация.

Сортировка

Динамическая сортировка предоставляет возможность изменять направление сортировки в построенном отчете. Для этого следует щелкнуть по компоненту, у которого была установлена динамическая сортировка. Динамическая сортировка осуществляется в следующих направлениях: **По возрастанию (Ascending)** и **По убыванию (Descending)**. Каждый раз при щелчке по компоненту направление меняется на противоположное.

© 2003-2026 Stimulsoft

Допускается многоуровневая сортировка в отчете. Для этого необходимо удерживать клавишу **Ctrl** и последовательно нажимать на сортируемые компоненты отчета. Для сброса сортировки можно нажать на любой сортируемый компонент без удержания клавиши **Ctrl**.

Сворачивание

Отчет с динамическим сворачиванием представляет собой интерактивный отчет, в котором определенные блоки могут сворачивать/разворачивать свое содержимое при нажатии на заголовок блока. Элементы отчета, которые можно свернуть/развернуть, обозначены специальной иконкой со знаком **[-]** или **[+]**.

Детализация

При детализации данных под главной панелью вьювера будет отображена панель детализации с закладками детализированных отчетов. Отображенный в данный момент отчет будет подсвечен. Предусмотрена возможность закрывать не требуемые в данный момент детализированные страницы.

Сортировка дашборда

При просмотре дашборда, во многих элемента можно изменять сортировку по полям данных элемента **По возрастанию (Ascending)** и **По убыванию (Descending)**.

Фильтрация дашборда

Фильтрация на дашборде возможна как при помощи специальных элементов фильтрации, так и при помощи других элементов. Фильтрация будет применяться для всех взаимосвязанных элементов на дашборде.

Детализация дашборда

В дашбордах, как и в отчетах, возможно открытие детализирующего дашборда, а также отчета. Кроме этого, для некоторых элементов дашборда возможна детализация данных на уровне элемента.

Событие интерактивности вьювера

Для работы любых интерактивных действий в отчете или дашборде не требуется дополнительных настроек вьювера. Предусмотрена возможность выполнения каких-либо операций перед интерактивным действием вьювера. Для этого достаточно определить событие `onInteraction`, которое будет

© 2003-2026 Stimulsoft

вызвано в момент совершения интерактивного действия перед его применением к отчету. В аргументах события будут переданы тип действия, объект отчета, а также все параметры, используемые для текущего интерактивного действия. Подробное описание доступных значений аргументов находится в разделе События вьювера.

Для каждого вида интерактивности вьювера предусмотрен определенный тип действия, указанного в данной таблице:

Наименование	Описание
`InitVars`	Действие происходит при инициализации переменных отчета, запрашиваемых у пользователя.
`Variables`	Действие происходит при использовании переменных на панели параметров. Подробное описание находится в разделе Работа с переменными отчета.
`Sorting`	Действие происходит при использовании сортировки колонок.
`DrillDown`	Действие происходит при использовании детализации колонок.
`Collapsing`	Действие происходит при использовании сворачивания блоков отчета.
`DashboardFiltering`	Действие происходит при использовании фильтров внутри элемента дашборда.
`DashboardSorting`	Действие происходит при использовании сортировки внутри элемента дашборда.
`DashboardResetAllFilters`	Действие происходит при сбросе сортировки и фильтров внутри

© 2003-2026 Stimulsoft

	элемента дашборда на значения, заданные в шаблоне.
`DashboardElementDrillDown`	Действие происходит при использовании детализации элемента дашборда.
`DashboardElementDrillUp`	Действие происходит при использовании детализации элемента дашборда.

Тип действия передается в аргументах события:

app.py

```
from stimulsoft_reports.viewer import StiViewer

viewer = StiViewer()
viewer.onInteraction += 'interaction'
```

viewer.html

```
<script>
  function interaction(args) {
    switch (args.action) {
      case "Sorting":
        break;

      case "DrillDown":
        break;

      case "Collapsing":
        break;
    }
  }
</script>
```

В аргументах передаются соответствующие коллекции параметров `sortingParameters`, `collapsingParameters`, `drillDownParameters` и `filteringParameters`, содержащие данные в специальном формате, необходимые для текущего интерактивного действия. При необходимости, значения коллекции параметров допускается корректировать, сохраняя

© 2003-2026 Stimulsoft

структуру и порядок передаваемых значений. Пример передаваемых значений параметров для некоторых интерактивных действий:

viewer.html

```
<script>
   let sortingParameters = {
      ComponentName: "Text10;false",
      DataBand: "DataBand1;DESC;CompanyName"
   };

   let collapsingParameters = {
      CollapsingStates: {
         GroupHeaderBand1: {
            keys: [1],
            values: [false]
         },
         ComponentName: "GroupHeaderBand1"
      };

   let drillDownParameters = [{
      ComponentIndex: "1"
      DrillDownMode: null
      ElementIndex: "6"
      PageGuid: "b916d048d3f446dc97c356d4ff47f48f"
      PageIndex: "0"
      ReportFile: null
   }];
</script>
```

Прямой доступ к значениям параметров интерактивных действий вьювера на стороне Python сервера не предусмотрен, событие работает только на стороне JavaScript клиента.

10.2.11 Редактирование построенного отчета

Вьювер имеет возможность редактировать элементы построенного отчета, такие как текстовые поля и чекбоксы. Для того, чтобы редактирование стало возможным, в самом шаблоне отчета необходимо пометить нужные компоненты как редактируемые. Никаких дополнительных настроек вьювера не требуется. После отображения отчета во вьювере, для начала редактирования необходимо нажать на соответствующую кнопку на панели вьювера. После завершения редактирования, необходимо нажать указанную кнопку еще раз, и все внесенные изменения будут применены к отчету.

© 2003-2026 Stimulsoft

Информация

Отредактированные значения будут применены при печати либо экспортировании отчета, при этом исходный отчет останется нетронутым. После перезагрузки вьювера все значения будут возвращены к исходным.

10.2.12 Отправка отчета по Email

Во вьювере предусмотрена возможность отправки отчета по Email. Для активации данной возможности достаточно установить свойство вьювера `showSendEmailButton` в значение `True` и добавить обработчик события `onEmailReport` на стороне Python сервера, в котором необходимо задать все параметры отправки Email. При необходимости, можно добавить вызов JavaScript события, в аргументах которого можно узнать необходимые данные для отправки письма, получить тип экспорта отчета, получить сам отчет, а также получить настройки экспорта отчета и при необходимости изменить их. Подробное описание доступных значений аргументов находится в разделе События вьювера.

app.py

```
from stimulsoft_reports.viewer import StiViewer
from stimulsoft_reports.events import StiEmailEventArgs

def emailReport(args: StiEmailEventArgs):
    settings = args.settions

viewer = StiViewer()
viewer.onEmailReport += emailReport
viewer.onEmailReport += 'emailReport'
```

viewer.html

```
<script>
  function emailReport(args) {
    let settings = args.settings;
  }
</script>
```

При отправке отчета по Email отображается меню выбора формата вложения,

© 2003-2026 Stimulsoft

которое соответствует меню выбора формата экспортирования отчета. После выбора формата отобразиться диалог ввода параметров отправки, таких как Email получателя, тема и текст письма. После подтверждения отправки будет вызвано событие `onEmailReport`, в котором можно проверить и скорректировать данные, введенные в этой форме, например:

viewer.html

```
<script>
   function emailReport(args) {
      args.settings.subject = "Invoice: " + args.settings.subject;
   }
</script>
```

Чистый JavaScript не имеет функций работы с Email, для этой возможности реализованы функции для отправки на стороне Python сервера. Для отправки Email, на стороне Python сервера необходимо установить данные, такие как логин и пароль аккаунта, с которого будет производится отправка, а также настройки сервера - его адрес, порт и другие параметры. Для этого предназначено свойство settings, которое находится в аргументах события и представляет собой объект класса `StiEmailSettings`, содержащий все необходимые свойства. Дополнительно, в аргументах можно получить и изменить данные для отправки письма (тема письма, текст и имя файла отчета). Подробное описание доступных настроек находится в разделе События вьювера.

app.py

```
from stimulsoft_reports.viewer import StiViewer
from stimulsoft_reports.events import StiEmailEventArgs

def emailReport(args: StiEmailEventArgs):
    args.settings.fromAddr = 'mail.sender@stimulsoft.com'
    args.settings.host = 'smtp.stimulsoft.com'
    args.settings.port = 456
    args.settings.login = '********'
    args.settings.password = '********'
    return 'The Email has been sent successfully.'

viewer = StiViewer()
viewer.onEmailReport += emailReport
```

© 2003-2026 Stimulsoft

Пример проверки и изменения темы письма перед отправкой:

app.py

```
from stimulsoft_reports.viewer import StiViewer
from stimulsoft_reports.events import StiEmailEventArgs

def emailReport(args: StiEmailEventArgs):
    if len(args.settings.subject) == 0:
        args.settings.subject = args.formatName + ' report ' +
args.settings.attachmentName

viewer = StiViewer()
viewer.onEmailReport += emailReport
```

Пример добавления адресов дополнительных получателей письма с отчетом:

app.py

```
from stimulsoft_reports.viewer import StiViewer
from stimulsoft_reports.events import StiEmailEventArgs

def emailReport(args: StiEmailEventArgs):
    args.settings.cc.append('extra_recipient_one@stimulsoft.com')
    args.settings.bcc.append('hidden_recipient_one@stimulsoft.com')
    args.settings.bcc.append('hidden_recipient_two@stimulsoft.com John
Smith')

viewer = StiViewer()
viewer.onEmailReport += emailReport
```

Настройки отправки Email

Вьювер позволяет установить значения по умолчанию для формы отправки Email. Для этого предназначены свойства `defaultEmailAddress`, `defaultEmailSubject` и `defaultEmailMessage`. По умолчанию данные свойства являются пустыми.

© 2003-2026 Stimulsoft

app.py

```
from stimulsoft_reports.viewer import StiViewer

viewer = StiViewer()
viewer.options.email.defaultEmailAddress =
'recipient_address@stimulsoft.com'
viewer.options.email.defaultEmailSubject = 'New Invoice'
viewer.options.email.defaultEmailMessage = 'Please check the new invoice
in the attachment'
```

10.2.13 Вызов дизайнера

Вьювер имеет возможность вызвать дизайнер отчетов. Для этого предназначена специальная кнопка **Дизайн (Design)** на панели инструментов вьювера. По умолчанию данная кнопка отключена. Для использования данной возможности необходимо установить свойство `showDesignButton` в значение `True`, а также определить событие `onDesignReport`.

app.py

```
from stimulsoft_reports.viewer import StiViewer

viewer = StiViewer()
viewer.onDesignReport += 'designReport'
```

viewer.html

```
<script>
   function designReport(args) {
      window.location.href = "/designer?report=" + args.fileName;
   }
</script>
```

Подробное описание доступных значений аргументов находится в разделе [События вьювера].

© 2003-2026 Stimulsoft

Информация

Сам вьювер не запускает дизайнер, он лишь вызывает указанное событие и передает в качестве аргументов имя файла и просматриваемый отчет. В событии можно выполнить перенаправление на Web-страницу, на которой расположен дизайнер отчетов, и передать необходимые параметры.

10.2.14 Визуальное оформление

Вьювер имеет возможность смены тем оформления визуальных элементов управления. Для этого достаточно установить свойство `theme` в опциях компонента:

app.py

```
from stimulsoft_reports.viewer import StiViewer
from stimulsoft_reports.viewer.enums import StiViewerTheme

viewer = StiViewer()
viewer.options.appearance.theme = StiViewerTheme.OFFICE_2022_DARKGRAY_BLUE
```

На данный момент доступно **8 тем** оформления с различными цветовыми акцентами. В результате, доступно более **60** вариантов оформления. Это позволяет настроить внешний вид вьювера практически под любое оформление Web-проекта.

Дополнительные настройки

По умолчанию, вьювер имеет только верхнюю панель инструментов, на которой находятся все элементы управления отчетом. При необходимости, панель инструментов можно разделить на верхнюю и нижнюю. На верхней панели будут находиться меню печати и экспорта отчета, а также кнопки работы с параметрами и закладками. Нижняя панель инструментов будет содержать элементы переключения между страницами отчета и меню управления масштабом. Для включения указанного режима предназначено свойство `displayMode`, которое может иметь следующие значения:

Наименование	Описание

© 2003-2026 Stimulsoft

StiToolbarDisplayMode.SIMPLE	Простой режим отображения, все элементы управления расположены на одной панели управления (значение по умолчанию).
StiToolbarDisplayMode.SEPARATED	Раздельный режим отображения, панель управления разделена на верхнюю - взаимодействие с отчетом, и нижнюю - взаимодействие со страницами.

app.py

```
from stimulsoft_reports.viewer import StiViewer

viewer = StiViewer()
viewer.options.toolbar.displayMode = StiToolbarDisplayMode.SIMPLE
viewer.options.appearance.scrollbarsMode = True
```

Дополнительно предусмотрена возможность установки параметров оформления основных элементов вьювера. Например, можно изменить шрифт и цвет надписей панели управления вьювера, установить фон вьювера, задать цвет границ страницы и др. Ниже представлен список доступных свойств, изменяющих оформление вьювера, и их значения по умолчанию:

app.py

```
from stimulsoft_reports.viewer import StiViewer

viewer = StiViewer()

viewer.options.appearance.backgroundColor = 'white'
viewer.options.appearance.pageBorderColor = 'gray'
viewer.options.appearance.showPageShadow = False

viewer.options.toolbar.backgroundColor = 'transparent';
viewer.options.toolbar.borderColor = 'transparent';
viewer.options.toolbar.fontColor = 'transparent';
viewer.options.toolbar.fontFamily = 'Arial';
```

© 2003-2026 Stimulsoft

В качестве значения для цвета указывается одна из стандартных констант для HTML цветов, либо код цвета в RGB формате, например '`#ff2020`'.

10.2.15 События вьювера

Вьювер отчетов поддерживает события, которые предоставляют возможность выполнить необходимые операции перед определенными действиями - как на стороне JavaScript клиента, так и на стороне Python сервера. Подробное описание работы событий находится в разделе События генератора отчетов.

Некоторые аргументы событий принимают значения из перечислений, которые находятся в определенных пространствах имен. Все используемые в событиях вьювера перечисления указаны в блоке кода ниже:

app.py

```
from stimulsoft_reports.enums import StiEventType
from stimulsoft_reports.report.enums import StiExportFormat
from stimulsoft_reports.viewer.enums import StiPrintAction,
StiExportAction
```

viewer.html

```
<script>
   StiExportAction = Stimulsoft.Viewer.StiExportAction;
   StiExportFormat = Stimulsoft.Report.StiExportFormat;
</script>
```

Вьювер поддерживает следующие события:

- onPrepareVariables
- onBeginProcessData
- onEndProcessData
- onOpenReport
- onOpenedReport
- onPrintReport
- onBeginExportReport
- onEndExportReport

© 2003-2026 Stimulsoft

- onInteraction
- onEmailReport
- onDesignReport

onPrepareVariables
[v] JavaScript [v] Python

Событие вызывается перед построением отчета после подготовки переменных отчета. Список аргументов события находится в разделе События генератора отчетов. Подробное описание и примеры использования находятся в разделе Работа с переменными отчета.

onBeginProcessData
[v] JavaScript [v] Python

Событие вызывается перед запросом данных, необходимых для построения отчета. Список аргументов события находится в разделе События генератора отчетов. Подробное описание и примеры использования находятся в разделах Подключение файловых данных и Подключение SQL адаптеров данных.

onEndProcessData
[v] JavaScript [v] Python

Событие вызывается после загрузки данных перед построением отчета. Список аргументов события находится в разделе События генератора отчетов. Подробное описание и примеры использования находятся в разделах Подключение файловых данных и Подключение SQL адаптеров данных.

onOpenReport
[v] JavaScript [x] Python

Событие вызывается перед открытием отчета после нажатия кнопки на панели инструментов.

© 2003-2026 Stimulsoft

Список свойств, передаваемых в аргументах события на стороне JavaScript клиента:

Наименование	Описание
`event`	Идентификатор текущего события, имеет значение "`OpenReport`".
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • "`Viewer`" • "`Designer`"
`report`	Текущий объект отчета, в аргументах данного события будет передано значение `null`.
`preventDefault`	Данный флаг предоставляет возможность остановить дальнейшую обработку события вьювером. По умолчанию установлено значение `false`.

onOpenedReport
[v] JavaScript [v] Python

Событие вызывается после открытия файла отчета до его передачи вьюверу.

Список свойств, передаваемых в аргументах события на стороне JavaScript клиента:

Наименование	Описание
`event`	Идентификатор текущего события, имеет значение "`OpenedReport`".
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать

© 2003-2026 Stimulsoft

	следующие значения: • `"Viewer"` • `"Designer"`
`report`	Текущий объект отчета.
`preventDefault`	Данный флаг предоставляет возможность остановить дальнейшую обработку события вьювером. По умолчанию установлено значение `false`.

Список свойств, передаваемых в аргументах события на стороне Python сервера. Аргументы имеют тип `StiReportEventArgs`:

Наименование	Описание
`event`	Идентификатор текущего события, для данного события имеет значение `StiEventType.OPENED_REPORT`
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • `StiViewer` • `StiDesigner`
`report`	Текущий объект отчета в виде JSON строки либо объекта.

onPrintReport
[v] JavaScript [v] Python

Событие вызывается перед печатью отчета. В таблице ниже представлен список аргументов обработчика событий на стороне JavaScript клиента:

Наименование	Описание

© 2003-2026 Stimulsoft

event	Идентификатор текущего события, имеет значение "PrintReport".
sender	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • "Viewer" • "Designer"
report	Текущий объект отчета.
printAction	Тип печати отчета. Может принимать следующие значения: • PrintPdf - печать в PDF формат; • PrintWithoutPreview - печать в HTML формат непосредственно на принтер, будет отображен системный диалог печати; • PrintWithPreview - печать в HTML формат с предварительным просмотром во всплывающем окне.
preventDefault	Данный флаг предоставляет возможность остановить дальнейшую обработку события вьювером. По умолчанию установлено значение false.

Список свойств, передаваемых в аргументах события на стороне Python сервера. Аргументы имеют тип StiReportEventArgs:

Наименование	Описание
event	Идентификатор текущего события, для данного события имеет значение

© 2003-2026 Stimulsoft

	`StiEventType.PRINT_REPORT`.
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • `StiViewer` • `StiDesigner`
`report`	Текущий объект отчета.
`printAction`	Тип печати отчета. Может принимать следующие значения: • `StiPrintAction.PRINT_PDF` - печать в PDF формат; • `StiPrintAction.PRINT_WITHOUT_PREVIEW` - печать в HTML формат непосредственно на принтер, будет отображен системный диалог печати; • `StiPrintAction.PRINT_WITH_PREVIEW` - печать в HTML формат с предварительным просмотром во всплывающем окне.

Подробное описание и примеры использования находятся в разделе Печать отчета.

onBeginExportReport
[v] JavaScript [v] Python

Событие вызывается перед экспортированием отчета после диалога настроек экспорта. В таблице ниже представлен список аргументов обработчика событий на стороне JavaScript клиента:

Наименование	Описание

© 2003-2026 Stimulsoft

event	Идентификатор текущего события, имеет значение "BeginExportReport".
sender	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • "Viewer" • "Designer"
report	Текущий объект отчета.
action	Действие, которое вызвало событие экспорта, может принимать следующие значения: • StiExportAction.ExportReport • StiExportAction.SendEmail
format	Выбранный формат экспорта отчета. Может принимать следующие значения: • StiExportFormat.Document • StiExportFormat.Pdf • StiExportFormat.Xps • StiExportFormat.POWERPOINT • StiExportFormat.Html • StiExportFormat.Html5 • StiExportFormat.Text • StiExportFormat.Word • StiExportFormat.Excel2007 • StiExportFormat.Odt • StiExportFormat.Ods

© 2003-2026 Stimulsoft

	• StiExportFormat.Csv • StiExportFormat.ImageSvg
formatName	Название выбранного формата экспорта отчета, соответствует названию констант в перечислении форматов.
settings	Настройки выбранного формата экспорта. Список доступных свойств будет зависеть от выбранного типа экспорта.
fileName	Имя файла отчета для сохранения после завершения экспорта.
openAfterExport	Флаг указывает на то, что экспортирование отчета будет происходить в новой вкладке браузера (значение true), либо по завершению экспорта будет вызвано сохранение файла (значение false).
preventDefault	Данный флаг предоставляет возможность остановить дальнейшую обработку события вьювером. По умолчанию установлено значение false.

Список свойств, передаваемых в аргументах события на стороне Python сервера. Аргументы имеют тип `StiExportEventArgs`:

Наименование	Описание
event	Идентификатор текущего события, для данного события имеет значение StiEventType.BEGIN_EXPORT_REPORT.

© 2003-2026 Stimulsoft

sender	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • StiViewer • StiDesigner
report	Текущий объект отчета.
action	Действие, которое вызвало событие экспорта, может принимать следующие значения: • StiExportAction.EXPORT_REPORT • StiExportAction.SEND_EMAIL
format	Выбранный формат экспорта отчета. Может принимать следующие значения: • StiExportFormat.DOCUMENT • StiExportFormat.PDF • StiExportFormat.XPS • StiExportFormat.POWERPOINT • StiExportFormat.HTML • StiExportFormat.HTML5 • StiExportFormat.TEXT • StiExportFormat.Word • StiExportFormat.EXCEL2007 • StiExportFormat.ODT • StiExportFormat.ODS • StiExportFormat.CSV • StiExportFormat.IMAGE_S

© 2003-2026 Stimulsoft

	VG
formatName	Название выбранного формата экспорта отчета, соответствует названию констант в перечислении форматов.
settings	Настройки выбранного формата экспорта. Список доступных свойств будет зависеть от выбранного типа экспорта.
fileName	Имя файла отчета для сохранения после завершения экспорта.
openAfterExport	Флаг указывает на то, что экспортирование отчета будет происходить в новой вкладке браузера (значение True), либо по завершению экспорта будет вызвано сохранение файла (значение False).

onEndExportReport
[v] JavaScript [v] Python

Событие вызывается после экспортирования отчета до его сохранения в виде файла. В таблице ниже представлен список аргументов обработчика событий на стороне JavaScript клиента:

Наименование	Описание
event	Идентификатор текущего события, имеет значение "EndExportReport".
sender	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • "Viewer"

© 2003-2026 Stimulsoft

	• "Designer"
report	Текущий объект отчета.
format	Выбранный формат экспорта отчета. Может принимать следующие значения: • StiExportFormat.Document • StiExportFormat.Pdf • StiExportFormat.Xps • StiExportFormat.POWERPOINT • StiExportFormat.Html • StiExportFormat.Html5 • StiExportFormat.Text • StiExportFormat.Word • StiExportFormat.Excel2007 • StiExportFormat.Odt • StiExportFormat.Ods • StiExportFormat.Csv • StiExportFormat.ImageSvg
formatName	Название выбранного формата экспорта отчета, соответствует названию констант в перечислении форматов.
data	Байтовые данные экспортированного отчета, подготовленные для сохранения в файл.
fileName	Имя файла отчета для сохранения после завершения экспорта.
openAfterExport	Флаг указывает на то, что

© 2003-2026 Stimulsoft

	экспортирование отчета будет происходить в новой вкладке браузера (значение `true`), либо по завершению экспорта будет вызвано сохранение файла (значение `false`).
`preventDefault`	Данный флаг предоставляет возможность остановить дальнейшую обработку события вьювером. По умолчанию установлено значение `false`.

Список свойств, передаваемых в аргументах события на стороне Python сервера. Аргументы имеют тип `StiExportEventArgs`:

Наименование	Описание
`event`	Идентификатор текущего события, для данного события имеет значение `StiEventType.END_EXPORT_REPORT`.
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • `StiViewer` • `StiDesigner`
`format`	Выбранный формат экспорта отчета. Может принимать следующие значения: • `StiExportFormat.DOCUMENT` • `StiExportFormat.PDF` • `StiExportFormat.XPS`

© 2003-2026 Stimulsoft

	• StiExportFormat.POWERPOINT • StiExportFormat.HTML • StiExportFormat.HTML5 • StiExportFormat.TEXT • StiExportFormat.Word • StiExportFormat.EXCEL2007 • StiExportFormat.ODT • StiExportFormat.ODS • StiExportFormat.CSV • StiExportFormat.IMAGE_SVG
formatName	Название выбранного формата экспорта отчета, соответствует названию констант в перечислении форматов.
data	Байтовые данные экспортированного отчета, подготовленные для сохранения в файл.
fileName	Имя файла отчета для сохранения после завершения экспорта.
fileExtension	Расширение файла отчета для сохранения после завершения экспорта, соответствует типу выбранного формата.
mimeType	MIME тип для выбранного формата экспортирования.
openAfterExport	Флаг указывает на то, что экспортирование отчета будет происходить в новой вкладке браузера (значение True), либо по завершению экспорта будет

© 2003-2026 Stimulsoft

	вызвано сохранение файла (значение `False`).

onInteraction
[v] JavaScript [x] Python

Событие вызывается в момент интерактивного действия вьювера (динамическая сортировка, сворачивание, детализация, применение параметров) до обработки значений генератором отчетов. В таблице ниже представлен список аргументов обработчика событий на стороне JavaScript клиента:

Наименование	Описание
`event`	Идентификатор текущего события, имеет значение "`Interaction`".
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • "`Viewer`" • "`Designer`"
`report`	Текущий объект отчета.
`action`	Идентификатор текущего интерактивного действия, может принимать следующие значения: • "`InitVars`" - действие происходит при инициализации переменных отчета, запрашиваемых у пользователя; • "`Variables`" - действие происходит при применении значений переменных, запрашиваемых у пользователя; • "`Sorting`" - действие

© 2003-2026 Stimulsoft

	происходит при использовании сортировки колонок; • "DrillDown" - действие происходит при детализации отчета; • "Collapsing" - действие происходит при использовании сворачивания блоков отчета; • "DashboardFiltering" - действие происходит при использовании фильтров внутри элемента дашборда; • "DashboardSorting" - действие происходит при использовании сортировки внутри элемента дашборда; • "DashboardResetAllFilters" - действие происходит при сбросе сортировки и фильтров внутри элемента дашборда на значения, заданные в шаблоне; • "DashboardElementDrillDown" - действие происходит при использовании детализации элемента дашборда; • "DashboardElementDrillUp" - действие происходит при использовании детализации элемента дашборда.
variables	Коллекция переменных отчета и их значений, заданных на панели параметров.
sortingParameters	Коллекция параметров, необходимых для динамической сортировки отчета.

© 2003-2026 Stimulsoft

collapsingParameters	Коллекция параметров, необходимых для динамического сворачивания элементов отчета.
drillDownParameters	Коллекция параметров, необходимых для детализации отчета.
filteringParameters	Коллекция параметров, необходимых для сортировки, фильтрации и детализации элементов дашборда.
preventDefault	Данный флаг предоставляет возможность остановить дальнейшую обработку события. По умолчанию установлено значение false.

onEmailReport
[v] JavaScript [v] Python

Событие вызывается после экспортирования отчета перед его отправкой по Email. В таблице ниже представлен список аргументов обработчика событий на стороне JavaScript клиента:

Наименование	Описание
event	Идентификатор текущего события, имеет значение "EmailReport".
sender	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • "Viewer" • "Designer"
report	Текущий объект отчета.
format	Выбранный формат экспорта отчета. Может принимать

© 2003-2026 Stimulsoft

	следующие значения: • StiExportFormat.Document • StiExportFormat.Pdf • StiExportFormat.Xps • StiExportFormat.POWERPOINT • StiExportFormat.Html • StiExportFormat.Html5 • StiExportFormat.Text • StiExportFormat.Word • StiExportFormat.Excel2007 • StiExportFormat.Odt • StiExportFormat.Ods • StiExportFormat.Csv • StiExportFormat.ImageSvg
formatName	Название выбранного формата экспорта отчета, соответствует названию констант в перечислении форматов.
data	Байтовые данные экспортированного отчета, подготовленные для отправки по Email.
fileName	Имя файла отчета для отправки по Email.
settings	Объект, содержащий параметры, заполненные в диалоге вьювера отправки отчета по Email. Описание всех параметров находится в отдельной таблице ниже.

© 2003-2026 Stimulsoft

В таблице ниже представлен список параметров отправки Email на стороне JavaScript клиента:

Наименование	Описание
`settings.email`	Email адрес, на который будет отправлен экспортированный отчет.
`settings.subject`	Тема письма.
`settings.message`	Текст письма.

Список свойств, передаваемых в аргументах события на стороне Python сервера. Аргументы имеют тип `StiExportEventArgs`:

Наименование	Описание
`event`	Идентификатор текущего события, для данного события имеет значение `StiEventType.EMAIL_REPORT`.
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • `StiViewer` • `StiDesigner`
`report`	Текущий объект отчета в виде JSON строки.
`format`	Выбранный формат экспорта отчета. Может принимать следующие значения: • `StiExportFormat.DOCUMENT`

© 2003-2026 Stimulsoft

	• StiExportFormat.PDF • StiExportFormat.XPS • StiExportFormat.POWERPOINT • StiExportFormat.HTML • StiExportFormat.HTML5 • StiExportFormat.TEXT • StiExportFormat.Word • StiExportFormat.EXCEL2007 • StiExportFormat.ODT • StiExportFormat.ODS • StiExportFormat.CSV • StiExportFormat.IMAGE_SVG
formatName	Название выбранного формата экспорта отчета, соответствует названию констант в перечислении форматов.
data	Байтовые данные экспортированного отчета, подготовленные для отправки по Email.
fileName	Имя файла отчета для отправки по Email.
settings	Объект, содержащий параметры отправки Email на стороне сервера. Описание всех параметров находится в отдельной таблице ниже.

Список параметров отправки Email на стороне Python сервера. Все настройки находятся в классе `StiEmailSettings`:

© 2003-2026 Stimulsoft

Наименование	Описание
fromAddr	Email адрес отправителя письма.
name	Имя и фамилия отправителя письма.
toAddr	Email адрес, на который будет отправлен экспортированный отчет, передается из диалогового окна вьювера.
subject	Тема письма, передается из диалогового окна вьювера.
message	Текст письма, передается из диалогового окна вьювера.
attachmentName	Имя отчета во вложении, по умолчанию используется имя файла отчета.
charset	Кодировка, используемая для текста письма, по умолчанию используется "UTF-8".
host	Адрес SMTP сервера. Является обязательным для заполнения.
port	Порт SMTP сервера, по умолчанию используется 465.
login	Логин для подключения к почтовому серверу. Является обязательным для заполнения.
password	Пароль для подключения к почтовому серверу. Является обязательным для заполнения.
secure	Тип шифрования соединения с почтовым сервером, может использоваться "ssl" (по умолчанию) либо "tls" шифрование.

© 2003-2026 Stimulsoft

`cc`	Массив CC (Carbon Copy) адресов вторичных получателей письма.
`bcc`	Массив BCC (Blind Carbon Copy) адресов скрытых получателей письма.

onDesignReport
[v] JavaScript [x] Python

Событие вызывается при нажатии на кнопку **Дизайн (Design)** на панели вьювера. В таблице ниже представлен список аргументов обработчика событий на стороне JavaScript клиента:

Наименование	Описание
`event`	Идентификатор текущего события, имеет значение "`DesignReport`".
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • "`Viewer`"
`report`	Текущий объект отчета.
`fileName`	Имя файла отчета для передачи и загрузки в дизайнер.

10.2.16 Настройки вьювера

Настройка вьювера выполняется при помощи изменения значений свойств, находящихся в главном контейнере свойств `options` у компонента. Все свойства разделены на группы для удобства использования. Все перечисления, используемые в настройках вьювера, находятся в пространстве имен `stimulsoft_reports.viewer.enums`.

Пример изменения некоторых настроек вьювера:

© 2003-2026 Stimulsoft

app.py

```
from flask import Flask, request
from stimulsoft_reports.viewer import StiViewer
from stimulsoft_reports.viewer.enums import StiHtmlExportMode,
StiToolbarDisplayMode, StiViewerTheme

@app.route('/viewer', methods = ['GET', 'POST'])
def viewer():
    viewer = StiViewer()
    viewer.options.localization = 'de.xml'
    viewer.options.appearance.theme =
StiViewerTheme.OFFICE_2022_DARKGRAY_BLUE
    viewer.options.appearance.fullScreenMode = True
    viewer.options.appearance.scrollbarsMode = True
    viewer.options.appearance.bookmarksTreeWidth = 200
    viewer.options.toolbar.displayMode = StiToolbarDisplayMode.SEPARATED
    viewer.options.exports.showExportToWord = False
    viewer.options.exports.showExportToCsv = False

    if viewer.processRequest(request):
        return viewer.getFrameworkResponse()

    # Here is the code for working with the report

    return viewer.getFrameworkResponse()
```

Главная (без группы)

Наименование	Описание
width	Задает ширину компонента в "px" или "%". По умолчанию установлено значение "100%".
height	Задает высоту компонента в "px" или "%". По умолчанию установлено значение "100%" для стандартного режима, и "650px" для режима отображения с полосами прокрутки.
localization	Устанавливает выбранную локализацию компонента. По умолчанию установлена английская локализация, которая

© 2003-2026 Stimulsoft

	встроена в сам компонент.

Оформление (Appearance)

Наименование	Описание
`theme`	Задает тему оформления вьювера. Список доступных тем оформления находится в перечислении `StiViewerTheme`. По умолчанию установлено значение `StiViewerTheme.OFFICE_2022_WHITE_BLUE`.
`iconSet`	Предоставляет возможность установить набор иконок: • `StiWebUIIconSet.AUTO` (значение по умолчанию) - устанавливает набор иконок автоматически. Для тем `Office2022` - используется набор иконок в стиле `Monoline`, для тем `Office2013` - используется набор иконок в стиле `Regular`. • `StiWebUIIconSet.MONOLINE` - устанавливает набор иконок в стиле `Monoline`; • `StiWebUIIconSet.REGULAR` - устанавливает набор иконок в стиле `Regular`.
`backgroundColor`	Устанавливает цвет фона вьювера. По умолчанию, установлено значение `'white'`.
`pageBorderColor`	Устанавливает цвет границы страниц отчета. По умолчанию, установлено значение `'gray'`.

© 2003-2026 Stimulsoft

`rightToLeft`	Устанавливает режим отображения **Right to Left** для элементов управления вьювера. По умолчанию, установлено значение `False`
`fullScreenMode`	Устанавливает полноэкранный режим отображения вьювера. Если свойство установлено в значение `True`, то значения свойств **width** и **height** игнорируются. По умолчанию, установлено значение `False`.
`scrollbarsMode`	Устанавливает режим отображения отчета с полосами прокрутки. По умолчанию, установлено значение `False`.
`openLinksWindow`	Устанавливает целевое окно или фрэйм для открытия гиперссылок из отчета. По умолчанию, установлено значение `'_blank'` (новая вкладка браузера). Может принимать одно из стандартных значений `'_blank'`, `'_self'`, `'_top'`, а так же имя окна либо фрэйма.
`showTooltips`	Включает или выключает отображение подсказок при наведении курсора на инструмент вьювера. По умолчанию, установлено значение `True`.
`showTooltipsHelp`	Предоставляет возможность отображать или не отображать ссылку на документацию в подсказках при наведении курсора на инструмент вьювера. По умолчанию, установлено значение `True`.

© 2003-2026 Stimulsoft

showDialogsHelp	Предоставляет возможность отображать или не отображать кнопку вызова справки в различных меню. По умолчанию, установлено значение True.
pageAlignment	Устанавливает выравнивание страниц отчета во вьювере. • StiContentAlignment.DEFAULT – выравнивание страниц определяется из настроек шаблона; • StiContentAlignment.LEFT– выравнивание страниц по левому краю; • StiContentAlignment.CENTER – выравнивание страниц по центру (значение по умолчанию); • StiContentAlignment.RIGHT – выравнивание страниц по правому краю.
showPageShadow	Включает или выключает отображение тени страниц отчета. По умолчанию, установлено значение False.
bookmarksPrint	Включает печать закладок в отчете. По умолчанию, установлено значение False.
bookmarksTreeWidth	Устанавливает ширину панели закладок в пикселях. По умолчанию, установлено значение 180.
parametersPanelPosition	Устанавливает расположение панели параметров во вьювере: • StiParametersPanelPosition.FROM_REPORT – расположение панели

© 2003-2026 Stimulsoft

	определяется из настроек шаблона (значение по умолчанию); • `StiParametersPanelPosition.TOP` - панель располагается вверху над страницей отчета; • `StiParametersPanelPosition.LEFT` - панель располагается слева от страницы отчета.
`parametersPanelMaxHeight`	Устанавливает максимальную высоту панели параметров в пикселях. По умолчанию, установлено значение 300.
`parametersPanelColumnsCount`	Устанавливает количество колонок на панели параметров. По умолчанию, установлено значение 2.
`minParametersCountForMultiColumns`	Устанавливает минимальное количество переменных на панели параметров для режима отображения в несколько колонок. По умолчанию, установлено значение 5.
`parametersPanelDateFormat`	Устанавливает формат даты и времени для переменных, которые отображаются на панели параметров. По умолчанию, значение не задано.
`parametersPanelSortDataItems`	Устанавливает или выключает режим сортировки значений переменной. По умолчанию, опция установлена в значение `False`, т.е. значения переменной отображаются в исходном порядке..

© 2003-2026 Stimulsoft

interfaceType	Устанавливает тип интерфейса вьювера. Могут использоваться следующие значения: • StiInterfaceType.AUTO – тип интерфейса вьювера будет выбран автоматически в зависимости от используемого устройства (значение по умолчанию); • StiInterfaceType.MOUSE – принудительное использование стандартного интерфейса для управления вьювером при помощи мыши; • StiInterfaceType.TOUCH – принудительное использование Touch интерфейса для управления вьювером при помощи сенсорного экрана монитора, в этом режиме элементы интерфейса вьювера имеют увеличенные размеры для удобства управления; • StiInterfaceType.MOBILE – принудительное использование Mobile интерфейса для управления вьювером при помощи экрана смартфона, в этом режиме интерфейс вьювера имеет упрощенный вид и адаптирован для управления при помощи мобильного устройства.
allowMobileMode	Включает или выключает возможность отображения отчета или дашборда в мобильном режиме. Если опция установлена в

© 2003-2026 Stimulsoft

	значение `False`, то мобильный режим просмотра не будет использоваться ни при каких обстоятельства. Если опция установлена в значение `True`, то мобильный режим просмотра будет использоваться запуске вьювера на мобильных устройствах. По умолчанию, опция установлена в значение `True`.
`chartRenderType`	Устанавливает тип отрисовки диаграмм в отчете: • `StiChartRenderType.ANIMATED_VECTOR` – диаграммы будут отрисованы в векторном режиме с анимацией (значение по умолчанию); • `StiChartRenderType.VECTOR` – диаграммы будут отрисованы как векторное изображение без анимации.
`reportDisplayMode`	Устанавливает режим экспорта для отображения страниц отчета. Может принимать одно из следующих значений: • `StiHtmlExportMode.FROM_REPORT` - режим экспорта элементов отчета определяется из настроек шаблона - Div или Table (значение по умолчанию); • `StiHtmlExportMode.TABLE` – элементы отчета экспортируются с использованием таблиц HTML; • `StiHtmlExportMode.DIV` – элементы отчета

© 2003-2026 Stimulsoft

	экспортируются с использованием разметки DIV;
datePickerFirstDayOfWeek	Предоставляет возможность установить первый день недели для инструмента **Выбор даты (Date picker)**. • StiFirstDayOfWeek.AUTO - Будет установлен понедельник или воскресенье как первый день недели, в зависимости от культуры браузера. • StiFirstDayOfWeek.MONDAY- Понедельник будет установлен как первый день недели. • StiFirstDayOfWeek.SUNDAY - Воскресенье будет установлен как первый день недели.
datePickerIncludeCurrentDayForRanges	Предоставляет возможность включать или не включать текущий день в диапазон значений элемента **Выбор даты (Date Picker)**. По умолчанию, опция установлена в значение False, т.е. текущий день не включается в диапазон значений элемента.
appearance.allowScrollZoom	Предоставляется возможность изменять масштаб вьювера при помощи колеса прокрутки мыши. По умолчанию, опция установлена в значение true.
allowTouchZoom	Предоставляется возможность изменять масштаб вьювера касанием. По умолчанию, опция установлена в значение True.

© 2003-2026 Stimulsoft

`combineReportPages`	Предоставляет возможность объединить обработанные страницы шаблона отчета в один шаблон или представить каждую страницу шаблона отдельной вкладкой во вьювере. По умолчанию, опция установлена в значение `False`, т.е. каждая страница шаблона отчета будет представлена отдельной вкладкой во вьювере.
`allowPropagationEvents`	Предоставляет возможность разрешить распространение событий нажатия клавиш, когда вьювер отчетов не в фокусе. По умолчанию свойство имеет значение `true`.
`dashboardFilterElementItemHeight`	Предоставляет возможность изменить высоту флажка в элементе дашборда **Список (List Box)**.

Панель инструментов (Toolbar)

Наименование	Описание
`visible`	Предоставляет возможность отображать или не отображать панель инструментов вьювера. По умолчанию, установлено значение `True`.
`displayMode`	Задает режим отображения панели инструментов вьювера. Может принимать одно из указанных ниже значений перечисления `displayMode`: • `StiToolbarDisplayMode.SI`

© 2003-2026 Stimulsoft

	MPLE – простой режим отображения, все элементы управления расположены на одной панели управления (значение по умолчанию); • StiToolbarDisplayMode.SEPARATED – раздельный режим отображения, панель управления разделена на верхнюю и нижнюю.
backgroundColor	Предоставляет возможность изменить цвет панели инструментов. По умолчанию, установлено значение 'transparent'.
borderColor	Предоставляет возможность изменить цвет границы панели инструментов. По умолчанию, установлено значение 'transparent'.
fontColor	Предоставляет возможность изменить цвет шрифта для всех элементов на панели инструментов и во всех меню этой панели. По умолчанию, установлено значение 'transparent'.
fontFamily	Предоставляет возможность изменить шрифт для всех элементов на панели инструментов и во всех меню этой панели. По умолчанию, установлено значение 'Arial'.
alignment	Устанавливает выравнивание элементов на панели управления. • StiContentAlignment.DEFAULT – выравнивание

© 2003-2026 Stimulsoft

	элементов зависит от опции **RightToLeft** (значение по умолчанию); • `StiContentAlignment.LEFT` – все элементы будут выровнены по левому краю панели инструментов; • `StiContentAlignment.CENTER` – все элементы будут выровнены по центру панели инструментов; • `StiContentAlignment.RIGHT` – все элементы будут выровнены по правой стороне панели инструментов.
`showButtonCaptions`	Включает или выключает отображение текста кнопок панели инструментов вьювера. По умолчанию, установлено значение `True`.
`showPrintButton`	Предоставляет возможность отображать или не отображать кнопку **Печать (Print)** на панели инструментов. По умолчанию, установлено значение `True`.
`showOpenButton`	Включает отображение кнопки **Открыть (Open)** на панели инструментов вьювера при просмотре отчетов или панели индикаторов (дашбордов). По умолчанию свойство имеет значение `True`.
`showSaveButton`	Включает отображение кнопки **Сохранить (Save)** на панели инструментов вьювера при просмотре отчетов или панели индикаторов (дашбордов). По умолчанию свойство имеет

© 2003-2026 Stimulsoft

	значение `True`.
`showSendEmailButton`	Предоставляет возможность отображать или не отображать кнопку **Отправить по Email (Send Email)** на панели инструментов. По умолчанию, установлено значение `False`. Также, необходимо добавить обработчик события onEmailReport.
`showFindButton`	Предоставляет возможность отображать или не отображать кнопку **Поиск (Find)** на панели инструментов. По умолчанию, установлено значение `True`.
`showSignatureButton`	Предоставляет возможность отображать или не отображать кнопку **Подпись (Signature)** на панели инструментов. По умолчанию, установлено значение `True`.
`showBookmarksButton`	Предоставляет возможность отображать или не отображать кнопку **Закладки (Bookmarks)** на панели инструментов. Если эта кнопка не отображается, то и панель закладок в отчете отображаться не будет. По умолчанию, установлено значение `True`.
`showParametersButton`	Предоставляет возможность отображать или не отображать кнопку **Параметры (Parameters)** на панели инструментов. Если эта кнопка не отображается, то и панель параметров в отчете отображаться не будет. По умолчанию, установлено

© 2003-2026 Stimulsoft

	значение `True`.
`showResourcesButton`	Предоставляет возможность отображать или не отображать кнопку **Ресурсы (Resources)** на панели инструментов. Если эта кнопка не отображается, то и панель ресурсов в отчете отображаться не будет. По умолчанию, установлено значение `True`.
`showEditorButton`	Предоставляет возможность отображать или не отображать кнопку **Редактировать (Editor)** на панели инструментов. Если эта кнопка не отображается, то невозможно будет изменить редактируемые элементы. По умолчанию, установлено значение `True`.
`showFullScreenButton`	Включает отображение кнопки **Полноэкранный просмотр (Full Screen)** на панели инструментов вьювера при просмотре отчетов или панели индикаторов (дашбордов). По умолчанию свойство имеет значение `True`.
`showRefreshButton`	Предоставляет возможность отображать или не отображать кнопку **Обновить (Refresh)** на панели инструментов вьювера при просмотре панели индикаторов (дашбордов). По умолчанию свойство имеет значение `True`.
`showFirstPageButton`	Предоставляет возможность отображать или не отображать кнопку **Первая страница (First Page)** на панели инструментов. По

© 2003-2026 Stimulsoft

	умолчанию, установлено значение `True`.
`showPreviousPageButton`	Предоставляет возможность отображать или не отображать кнопку **Последняя страница (Previous Page)** на панели инструментов. По умолчанию, установлено значение `True`.
`showCurrentPageControl`	Предоставляет возможность отображать или не отображать индикатор текущей страницы на панели инструментов. По умолчанию, установлено значение `True`.
`showNextPageButton`	Предоставляет возможность отображать или не отображать кнопку **Следующая страница (Next Page)** на панели инструментов. По умолчанию, установлено значение `True`.
`showLastPageButton`	Предоставляет возможность отображать или не отображать кнопку **Предыдущая страница (Last Page)** на панели инструментов. По умолчанию, установлено значение `True`.
`showZoomButton`	Предоставляет возможность отображать или не отображать кнопку выбора масштаба на панели инструментов. По умолчанию, установлено значение `True`.
`showViewModeButton`	Предоставляет возможность отображать или не отображать кнопку выбора режимов отображения страниц отчета. По умолчанию, установлено

© 2003-2026 Stimulsoft

	значение `True`.
`showDesignButton`	Включает отображение кнопки **Дизайн (Design)** на панели инструментов вьювера при просмотре отчетов или панели индикаторов (дашбордов). По умолчанию свойство имеет значение `False`.
`showAboutButton`	Предоставляет возможность отображать или не отображать кнопку **О программе (About)** на панели инструментов. По умолчанию, установлено значение `True`.
`showPinToolbarButton`	Предоставляет возможность отображать или не отображать кнопку **Закрепить (Pin)** в мобильном режиме просмотра отчета. По умолчанию, установлено значение `True`.
`printDestination`	Устанавливает режим печати отчета. Может принимать одно из указанных ниже значений перечисления: • `StiPrintDestination.DEFAULT` – будет отображено меню с предоставлением выбора режима печати (значение по умолчанию); • `StiPrintDestination.PDF` – печать будет осуществлена в формате PDF; • `StiPrintDestination.DIRECT` – печать будет осуществлена в формате HTML непосредственно на принтер, будет отображен системный диалог печати;

© 2003-2026 Stimulsoft

	• StiPrintDestination.WITH_PREVIEW – печать будет осуществлена в формате HTML через всплывающее окно предварительного просмотра отчета.
viewMode	Устанавливает режим отображения страниц отчета: • StiWebViewMode.SINGLE_PAGE – отображается одна страница, выбранная на панели инструментов (значение по умолчанию); • StiWebViewMode.CONTINUOUS – отображаются все страницы отчета в виде ленты; • StiWebViewMode.MULTIPLE_PAGES – отображаются все страницы отчета в виде таблицы.
zoom	Предоставляет возможность установить масштаб страниц отчета при загрузке вьювера. По умолчанию, установлено значение 100 процентов. Максимальное значение 500 процентов. Дополнительно, можно установить одно из следующих значений для масштаба: • StiZoomMode.PAGE_WIDTH – масштаб страниц отчета по ширине страницы. • StiZoomMode.PAGE_HEIGHT – масштаб страниц отчета по высоте страницы.
menuAnimation	Предоставляет возможность включить или выключить анимацию отображения и

© 2003-2026 Stimulsoft

	закрытия различных меню во вьювере. По умолчанию, установлено значение `True`.
`showMenuMode`	Устанавливает режим раскрытия различных меню во вьювере. Может принимать одно из указанных ниже значений: • `StiShowMenuMode.CLICK` – режим раскрытия меню по щелчку (значение по умолчанию); • `StiShowMenuMode.HOVER` – режим раскрытия меню при наведении курсора.
`autoHide`	Устанавливает режим автоматического сворачивания панели инструментов при просмотре отчета в мобильном режиме. По умолчанию, установлено значение `False`.

Экспорт (Exports)

Наименование	Описание
`storeExportSettings`	Предоставляет возможность сохранять настройки экспорта в cookies. По умолчанию, установлено значение `True`.
`showExportDialog`	Предоставляет возможность отображать или не отображать меню настроек экспорта. Если меню спрятано, экспорт будет выполнен со значениями по умолчанию. По умолчанию, установлено значение `True`.

© 2003-2026 Stimulsoft

`showExportToDocument`	Предоставляет возможность отображать или не отображать элемент **Document File** в меню **Сохранить (Save)**. По умолчанию, установлено значение `True`.
`showExportToPdf`	Включает отображение пункта меню экспорта **Adobe PDF**. По умолчанию свойство имеет значение `True`.
`showExportToXps`	Включает отображение пункта меню экспорта **XPS**. По умолчанию свойство имеет значение `False`.
`showExportToPowerPoint`	Включает отображение пункта меню экспорта **Microsoft PowerPoint**. По умолчанию свойство имеет значение `True`.
`showExportToHtml`	Включает отображение пункта меню экспорта типа **HTML** в настройках пункта меню экспорта **HTML**. По умолчанию, установлено значение `True`.
`showExportToHtml5`	Включает отображение пункта меню экспорта типа **HTML5** в настройках пункта меню экспорта **HTML**. По умолчанию, установлено значение `True`.
`showExportToText`	Включает отображение пункта меню экспорта **Text**. По умолчанию свойство имеет значение `True`.
`showExportToWord`	Включает отображение пункта меню экспорта **Microsoft Word**. По умолчанию, установлено значение `True`.
`showExportToOpenDocumentWriter`	Включает отображение пункта меню экспорта **OpenDocument**

© 2003-2026 Stimulsoft

	Writer. По умолчанию свойство имеет значение `True`.
`showExportToExcel`	Включает отображение пункта меню экспорта **Microsoft Excel**. По умолчанию свойство имеет значение `True`.
`showExportToOpenDocumentCalc`	Включает отображение пункта меню экспорта **OpenDocument Calc**. По умолчанию свойство имеет значение `True`.
`showExportToRtf`	Включает отображение пункта меню экспорта RTF. По умолчанию свойство имеет значение `True`.
`showExportToCsv`	Включает отображение типа **CSV** в настройках пункта меню экспорта **Данные (Data)**. По умолчанию свойство имеет значение `True`.
`showExportToJson`	Включает отображение типа **JSON** в настройках пункта меню экспорта **Данные (Data)**. По умолчанию свойство имеет значение `True`.
`showExportToXml`	Включает отображение типа **XML** в настройках пункта меню экспорта **Данные (Data)**. По умолчанию свойство имеет значение `True`.
`showExportToDbf`	Включает отображение типа **DBF** в настройках пункта меню экспорта **Данные (Data)**. По умолчанию свойство имеет значение `True`.
`showExportToDif`	Включает отображение типа **DIF** в настройках пункта меню экспорта **Данные (Data)**. По умолчанию свойство имеет значение `True`.
`showExportToSylk`	Включает отображение типа **SYLK** в настройках пункта меню

© 2003-2026 Stimulsoft

	экспорта **Данные (Data)**. По умолчанию свойство имеет значение `True`.
`showExportToImagePng`	Включает отображение типа **PNG** в настройках пункта меню экспорта **Изображение (Image)**. По умолчанию свойство имеет значение `True`.
`showExportToImageJpeg`	Включает отображение типа **JPEG** в настройках пункта меню экспорта **Изображение (Image)**. По умолчанию свойство имеет значение `True`.
`showExportToImageSvg`	Включает отображение типа **SVG** в настройках пункта меню экспорта **Изображение (Image)**. По умолчанию свойство имеет значение `True`.
`showExportToImageSvgz`	Включает отображение типа **SVGZ** в настройках пункта меню экспорта **Изображение (Image)**. По умолчанию свойство имеет значение `True`.
`showExportToImagePcx`	Включает отображение типа **PCX** в настройках пункта меню экспорта **Изображение (Image)**. По умолчанию свойство имеет значение `True`.
`showExportToImageBmp`	Включает отображение типа **BMP** в настройках пункта меню экспорта **Изображение (Image)**. По умолчанию свойство имеет значение `True`.
`showExportToImageGif`	Включает отображение типа **GIF** в настройках пункта меню экспорта **Изображение (Image)**. По

© 2003-2026 Stimulsoft

	умолчанию свойство имеет значение `True`.
`showExportToImageTiff`	Включает отображение типа **TIFF** в настройках пункта меню экспорта **Изображение (Image)**. По умолчанию свойство имеет значение `True`.
`showExportDataOnly`	Включает отображение опции **Экспортировать только данные (Export Data Only)** в настройках пункта меню экспорта **Данные (Data)**. По умолчанию свойство имеет значение `True`.

Отправить по Email (Email)

Наименование	Описание
`showEmailDialog`	Включает отображение диалогового окна параметров отправки отчета по Email. Если диалоговое окно отключено, то отправка по Email будет производиться с настройками, заданными по умолчанию `onEmailReport`. По умолчанию, установлено значение `True`.
`showExportDialog`	Включает отображение диалогового окна параметров экспорта при отправке Email. Если свойство имеет значение `False`, то экспортирование будет производиться с заданными по умолчанию настройками. По умолчанию, установлено значение `True`.

© 2003-2026 Stimulsoft

`defaultEmailAddress`	Устанавливает Email получателя по умолчанию, т.е. адрес на который будет отправлено письмо с вложенным отчетом. По умолчанию, значение не задано.
`defaultEmailSubject`	Устанавливает тему (заголовок) электронного письма по умолчанию. По умолчанию, значение не задано.
`defaultEmailMessage`	Устанавливает сообщение (текст) электронного письма по умолчанию. По умолчанию, значение не задано.

10.3 HTML5 дизайнер

Дизайнер отчетов представляет собой Python компонент `StiDesigner`, предназначенный для разработки отчетов и дашбордов в окне браузера на любом компьютере с любой установленной операционной системой. Дизайнер имеет современный интерфейс и различные темы оформления, доступно для использования множество профессионально подготовленных шаблонов отчетов и дашбордов, мастера создания отчетов, большое количество компонентов для построения отчета и дашборда практически любой сложности.

Дизайнер имеет удобный режим предварительного просмотра отчета во встроенном вьювере, что значительно ускоряет разработку отчета. Работа встроенного вьювера не отличается от работы отдельного компонента, более подробно рассмотрена в разделе Вьювер отчетов.

Интерфейс дизайнера построен с использованием HTML5, что позволяет использовать его практически на любой современной платформе. Компонент использует технологию AJAX для выполнения всех действий (загрузка и построение отчета, подключение к данным, работа с компонентами и их настройками, предварительный просмотр отчета, печать, экспортирование и др.), что позволяет избавиться от перезагрузки всей страницы и повысить скорость работы, а также использовать компонент в One-Page приложениях. Технология JavaScript, применяемая для построения отчетов, позволяет использовать практически любую серверную сторону невысокой

© 2003-2026 Stimulsoft

производительности.

Информация

Вследствие того, что для панелей индикаторов (дашбордов) и отчетов используется одинаковый унифицированный формат шаблона MRT, методы для загрузки шаблона и работы с данными, то в тексте документации будет использоваться слово “отчет”.

- Развертывание
- Активация лицензии
- Создание и редактирование отчета
- Сохранение отчета
- Локализация
- Просмотр
- Визуальное оформление
- События дизайнера
- Настройки

10.3.1 Использование

Примеры

Полный код примера можно найти на GitHub по данной ссылке.

Для использования дизайнера достаточно установить пакет `stimulsoft-reports` либо `stimulsoft-dashboards` при помощи установщика пакетов, выполнив следующую команду:

console

```
python -m pip install stimulsoft-reports
```

console

© 2003-2026 Stimulsoft

```
python -m pip install stimulsoft-dashboards
```

По умолчанию устанавливается минимальный набор драйверов данных. Необходимые драйверы можно дополнительно установить вручную, либо воспользоваться командой для установки всех доступных драйверов:

console

```
python -m pip install stimulsoft-reports[ext]
```

console

```
python -m pip install stimulsoft-dashboards[ext]
```

Будет установлена последняя доступная версия продукта в текущее рабочее пространство, после этого можно использовать классы и функции по работе с отчетами и дашбордами.

Информация

Для примеров кода в документации использован **Flask** фреймворк, как один из самых популярных и простых для понимания кода. Можно использовать любой Web-фреймворк, все классы и функции для работы с компонентами являются универсальными.

Для добавления дизайнера в Web-проект, предназначен класс `StiDesigner`. При помощи данного класса можно создать объект дизайнера, установить необходимые настройки, обработать запрос и вернуть результат его выполнения, получить подготовленный JavaScript и HTML код компонента. Пример отображения дизайнера на HTML странице:

© 2003-2026 Stimulsoft

app.py

```
from flask import Flask, render_template, url_for, request
from stimulsoft_reports.designer import StiDesigner

app = Flask(__name__)

@app.route('/designer', methods = ['GET', 'POST'])
def designer():
    designer = StiDesigner()
    designer.options.appearance.fullScreenMode = True

    if designer.processRequest(request):
        return designer.getFrameworkResponse()

    js = designer.javascript.getHtml()
    html = designer.getHtml()
    return render_template('designer.html', designerJavaScript = js,
designerHtml = html)
```

designer.html

```
<!DOCTYPE html>
<html>

<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>Editing a Report Template in the Designer</title>

  {{ designerJavaScript|safe }}
</head>

<body>
  {{ designerHtml|safe }}
</body>

</html>
```

В этом примере создается экземпляр объекта `StiDesigner` и задаются необходимые настройки.

Специальная функция компонента `processRequest(request)` выполняет обработку текущего запроса. Если функция возвращает `True`, значит запрос успешно обработан, и необходимо вернуть результат его выполнения. Более подробно про это рассказано в разделе Обработчик событий.

© 2003-2026 Stimulsoft

Далее, необходимо сгенерировать подготовленный JavaScript и HTML код, необходимый для работы дизайнера. Функция `designer.javascript.getHtml()` генерирует HTML код для подключения необходимых скриптов и ресурсов, функция `designer.getHtml()` генерирует HTML код самого компонента. Сгенерированный код передается в качестве параметров в HTML шаблон designer.html, и выводится в заданных местах.

Информация

Наши продукты Stimulsoft Reports.PYTHON и Stimulsoft Dashboards.PYTHON не имеют нативного ядра генератора отчетов на Python, построение и экспорт отчета выполняется на стороне клиента при помощи JavaScript кода. При использовании Python кода для работы с компонентами, необходимо вызвать функцию `getHtml()`, которая вернет весь необходимый JavaScript и HTML код для работы генератора отчетов и компонентов.

Предусмотрено упрощенное развертывание дизайнера без использования HTML шаблона страницы. Например, этот же пример можно реализовать, использовав только Python код:

app.py

```
from flask import Flask, url_for, request
from stimulsoft_reports.designer import StiDesigner

app = Flask(__name__)

@app.route('/designer', methods = ['GET', 'POST'])
def designer():
    designer = StiDesigner()
    designer.options.appearance.fullScreenMode = True

    if designer.processRequest(request):
        return designer.getFrameworkResponse()

    # Here is the code for working with the report

    return designer.getFrameworkResponse()
```

В данном случае, после вызова обработчика событий и действий с отчетом,

© 2003-2026 Stimulsoft

сразу возвращается финальный результат запроса. Т.е. дизайнер вернет полностью готовую HTML страницу со всеми необходимыми скриптами и кодом.

Информация

В большинстве случаев для работы с продуктом достаточно использовать только Python код, который обеспечивает взаимодействие со всеми основными возможностями компонентов. Для детальной настройки продукта и использования всех возможностей JS генератора, необходимо использовать JavaScript код. Вариант развертывания продукта при помощи только JavaScript кода описан в разделе Reports and Dashboards for JS, в этом случае использование Python кода требуется только для подключения адаптеров данных.

Для формирования ответа используется функция `getFrameworkResponse()`, которая возвращает готовый объект, подходящий для используемого в данный момент Web-фреймворка. Текущий фреймворк определяется в момент обработки запроса. Наши компоненты поддерживают работу с такими популярными фреймворками, как **Django**, **Flask**, **Tornado**. Для обработки запроса и формирования ответа в Web-проектах, не использующих указанные фреймворки, необходимо передать текущие GET и POST данные в функцию обработки запроса, а для формирования ответа воспользоваться функцией `getResponse()`, которая вернет все необходимые данные. Например, обработку событий дизайнера можно реализовать указанным ниже способом:

app.py

```
from flask import Flask, make_response, render_template, url_for, request
from stimulsoft_reports.designer import StiDesigner

app = Flask(__name__)

@app.route('/designer', methods = ['GET', 'POST'])
def designer():
    designer = StiDesigner()
    designer.options.appearance.fullScreenMode = True

    query = request.args.to_dict()
    body = request.get_data(False)
```

© 2003-2026 Stimulsoft

```
    if designer.processRequest(None, query, body):
        designerResponse = designer.getResponse()
        response = make_response(designerResponse.data)
        response.mimetype = designerResponse.mimetype
        response.headers.add('Access-Control-Allow-Origin',
request.origin)
        return response

    js = designer.javascript.getHtml()
    html = designer.getHtml()
    return render_template('designer.html', designerJavaScript = js,
designerHtml = html)
```

10.3.2 Активация лицензии

После приобретения продукта вам необходимо активировать лицензию для используемых компонентов. Предусмотрено несколько способов подключения лицензионного ключа.

Все варианты активации компонентов описаны в разделе Активация лицензии генератора отчетов, имеют одинаковые функции и параметры вызова.

10.3.3 Создание и редактирование отчета

Для запуска дизайнера без отчета не требуется никаких действий. После загрузки компонента, отобразится главное меню дизайнера. При необходимости запуска дизайнера с новым (пустым) отчетом, можно создать новый объект отчета `StiReport` и присвоить его дизайнеру.

Для редактирования отчета в дизайнере, достаточно создать объект `StiReport`, загрузить в него шаблон отчета, и присвоить полученный объект дизайнеру. Все остальные действия будут выполнены автоматически, дизайнер отобразит первую страницу шаблона:

© 2003-2026 Stimulsoft

app.py

```
from flask import Flask, url_for, request
from stimulsoft_reports.report import StiReport
from stimulsoft_reports.designer import StiDesigner

app = Flask(__name__)

@app.route('/designer', methods = ['GET', 'POST'])
def designer():
    designer = StiDesinger()
    designer.options.appearance.fullScreenMode = True

    if designer.processRequest(request):
        return designer.getFrameworkResponse()

    report = StiReport()
    report.loadFile(url_for('static', filename='reports/SimpleList.mrt'))
    designer.report = report

    return designer.getFrameworkResponse()
```

Дизайнер умеет работать со стандартными, упакованными и зашифрованными шаблонами. Подробное описание работы с различными форматами отчетов находится в разделе Загрузка и сохранение отчета.

Событие создания отчета

Новый отчет можно создать, используя главное меню дизайнера. Для предварительной загрузки данных для нового отчета, либо выполнения какие-либо других необходимых действий с новым отчетом, предназначено событие `onCreateReport`. Данное событие будет вызвано при создании нового пустого отчета из главного меню, либо при создании отчета с использованием мастера.

В событии на стороне Python сервера допускается изменение отчета либо его параметров.

© 2003-2026 Stimulsoft

app.py

```
from stimulsoft_reports.designer import StiDesigner
from stimulsoft_reports.events import StiReportEventArgs

def createReport(args: StiReportEventArgs):
    args.report['ReportDescription'] = 'This is a report description from
the Python server-side.'

designer = StiDesigner()
designer.onCreateReport += createReport
```

В событии на стороне JavaScript клиента доступны все функции генератора отчетов. Например, можно подключить данные для нового отчета и синхронизировать их со словарем.

app.py

```
from stimulsoft_reports.designer import StiDesigner

designer = StiDesigner()
designer.onCreateReport += 'createReport'
```

designer.html

```
<script>
   function createReport(args) {
      let dataSet = new Stimulsoft.System.Data.DataSet("Demo");
      dataSet.readJsonFile("/static/data/Demo.json");

      let report = args.report;
      report.regData(dataSet.dataSetName, "", dataSet);
      report.dictionary.synchronize();
   }
</script>
```

Подробное описание доступных значений аргументов находится в разделе События дизайнера.

© 2003-2026 Stimulsoft

10.3.4 Сохранение отчета

В дизайнере предусмотрено два варианта сохранения отчета, которые доступны в главном меню и на главной панели дизайнера: **Сохранить (Save)** и **Сохранить как (Save As)**. В свою очередь, каждый из этих вариантов сохранения имеет свои режимы и настройки.

Сохранение отчета на стороне JavaScript клиента

При нажатии кнопки **Сохранить (Save)**, сохранение файла шаблона отчета осуществляется средствами браузера, для этого не требуется каких-либо настроек. При необходимости сохранения отчета своими методами, предназначено событие `onSaveReport`. В аргументах события будут переданы имя файла отчета и сам отчет. Отчет можно сохранить, например, в JSON строку и передать на сторону сервера своими методами.

app.py

```
from stimulsoft_reports.designer import StiDesigner

designer = StiDesigner()
designer.onSaveReport += 'saveReport'
```

designer.html

```
<script>
   function saveReport(args) {
      let fileName = args.fileName;
      let report = args.report;

      let jsonReport = report.saveToJsonString();
   }
</script>
```

Если событие определено, то после завершения события дизайнер продолжает работу без каких-либо дополнительных действий и сообщений. При необходимости, после сохранения отчета можно отобразить диалоговое окно с ошибкой или текстовым сообщением. Для этого предназначена специальная функция `StiError.showError()`. Вы сами определяете необходимость отображения сообщения об ошибке.

© 2003-2026 Stimulsoft

designer.html

```
<script>
   function saveReport(args) {
      let fileName = args.fileName;
      let report = args.report;

      // Error message
      Stimulsoft.System.StiError.showError("An error occurred while saving
      the report.");

      // Info message
      Stimulsoft.System.StiError.showError("The report was saved
      successfully.", true, true);
   }
</script>
```

При нажатии кнопки **Сохранить как (Save As)**, будет отображено диалоговое окно с запросом имени файла отчета. После этого сохранение файла шаблона отчета будет осуществляться средствами браузера. При необходимости сохранения отчета своими методами, предназначено событие `onSaveAsReport`. В аргументах события будут переданы имя файла отчета и сам отчет.

Работа данного события не отличается от работы события `onSaveReport`, за исключением того, что после завершения события произойдет автоматическое сохранение шаблона отчета на компьютере средствами браузера. Для предотвращения этого действия, достаточно установить свойство `preventDefault` в значение `true` в аргументах события, в этом случае автоматическое сохранение не будет выполнено.

app.py

```
from stimulsoft_reports.designer import StiDesigner

designer = StiDesigner()
designer.onSaveAsReport += 'saveAsReport'
```

© 2003-2026 Stimulsoft

designer.html

```
<script>
   function saveAsReport(args) {
      args.preventDefault = true;
   }
</script>
```

При необходимости, вы можете получить доступ к оригинальному имени отчета или имени отчета из диалога сохранения следующим образом:

designer.html

```
<script>
   function saveAsReport(args) {
      // Report name from the designer save dialog
      var reportName1 = args.fileName;

      // Original report name from properties
      var reportName2 = args.report.reportName;
   }
</script>
```

Подробное описание доступных значений аргументов находится в разделе События дизайнера.

Сохранение отчета на стороне Python сервера

Для сохранения отчета на стороне Python сервера, достаточно определить событие `onSaveReport`, в аргументах события будут переданы имя файла отчета и сам отчет в виде объекта. Для сохранения отчета можно воспользоваться стандартными функциями, например сохрани редактируемый отчет в виде файла в заданной директории:

© 2003-2026 Stimulsoft

app.py

```
import json
import os
from stimulsoft_reports import StiResult
from stimulsoft_reports.designer import StiDesigner
from stimulsoft_reports.events import StiReportEventArgs
from stimulsoft_reports.report import StiReport

def saveReport(args: StiReportEventArgs):
    filePath = os.path.normpath(os.getcwd() + '\\static\\reports\\' +
args.fileName)
    try:
        with open(filePath, mode='w', encoding='utf-8') as file:
            jsonReport = json.dumps(args.report, indent = 4)
            file.write(jsonReport)
            file.close()
    except Exception as e:
        return StiResult.getError(str(e))

    return f'The report was successfully saved to a {args.fileName} file.'

designer = StiDesigner()
designer.onSaveReport += saveReport
```

Аналогичным образом реализована работа события `onSaveAsReport` на стороне Python сервера, все аргументы события имеют те же имена и значения.

Подробное описание доступных значений аргументов находится в разделе События дизайнера.

10.3.5 Локализация

Дизайнер поддерживает полную локализацию своего интерфейса. Для локализации интерфейса на требуемый язык, достаточно установить требуемое название файла для опции `localization` у дизайнера:

app.py

```
from stimulsoft_reports.designer import StiDesigner

designer = StiDesigner()
designer.options.localization = 'de.xml'
```

© 2003-2026 Stimulsoft

Все доступные локализационные XML файлы находятся в ресурсах установленного пакета продукта. При необходимости, локализационный файл можно загрузить из любого другого места, для этого необходимо для опции `localization` указать полный путь к нужному XML файлу:

app.py

```
from stimulsoft_reports.designer import StiDesigner

designer = StiDesigner()
designer.options.localization = '/resources/loc/de.xml'
```

Если файл доступен для чтения из Python приложения, локализация будет загружена в дизайнер. Иначе будет использована встроенная английская локализация интерфейса.

Дизайнер имеет возможность выбрать необходимую локализацию интерфейса при помощи специального меню на панели инструментов. По умолчанию, в данное меню добавлена английская (встроенная) локализация, а также заданная при помощи свойства `localization`. Для добавления дополнительных локализаций в меню, предназначена специальная опция дизайнера `localizations`, которая представляет собой коллекцию локализаций. В качестве значений указывается локализационный файл либо полный путь к этому файлу:

app.py

```
from stimulsoft_reports.designer import StiDesigner

designer = StiDesigner()
designer.options.localization = 'de.xml'
designer.options.localizations.append('fr.xml')
designer.options.localizations.append('pl.xml')
designer.options.localizations.append('/resources/loc/it.xml')
```

10.3.6 Просмотр

В дизайнере предусмотрен режим предварительного просмотра редактируемого отчета. Для этого достаточно перейти на соответствующую вкладку в окне дизайнера. Шаблон отчета будет построен и отображен во

© 2003-2026 Stimulsoft

встроенном вьювере.

Событие предварительного просмотра

Перед просмотром отчета предусмотрена возможность выполнить какие-либо необходимые действия. Для этого предназначено специальное событие `onPreviewReport`, которое будет вызвано перед просмотром отчета. В аргументах события будет передан отчет, предназначенный для просмотра.

В событии на стороне Python сервера допускается изменение отчета либо его параметров

app.py

```
from stimulsoft_reports.designer import StiDesigner
from stimulsoft_reports.events import StiReportEventArgs

def previewReport(args: StiReportEventArgs):
    args.report['ReportDescription'] = 'This is a report description from
the Python server-side.'

designer = StiDesigner()
designer.onPreviewReport += previewReport
```

В событии на стороне JavaScript клиента доступны все функции генератора отчетов. Например, можно подключить данные для отчета.

app.py

```
from stimulsoft_reports.designer import StiDesigner

designer = StiDesigner()
designer.onPreviewReport += 'previewReport'
```

© 2003-2026 Stimulsoft

designer.html

```
<script>
   function previewReport(args) {
      let dataSet = new Stimulsoft.System.Data.DataSet("Demo");
      dataSet.readJsonFile("/static/data/Demo.json");

      let report = args.report;
      report.regData(dataSet.dataSetName, "", dataSet);
   }
</script>
```

Список доступных событий встроенного вьювера и описание вариантов использования находится в разделе События вьювера отчетов.

Дополнительные возможности

Окно предварительного просмотра отчета в дизайнере представляет собой полноценный интерактивный вьювер, который умеет осуществлять печать и экспортирование отчета, поддерживает работу с параметрами отчета. Поддерживаются все доступные интерактивные действия, такие как динамическая сортировка, детализация, сворачивание. Для использования указанных возможностей не требуется каких-либо дополнительных настроек дизайнера отчетов.

10.3.7 Визуальное оформление

Дизайнер имеет возможность смены тем оформления визуальных элементов управления. Для этого достаточно установить свойство `theme` в опциях компонента:

app.py

```
from stimulsoft_reports.designer import StiDesigner
from stimulsoft_reports.designer.enums import StiDesignerTheme

designer = StiDesigner()
designer.options.appearance.theme =
StiDesignerTheme.OFFICE_2022_DARKGRAY_BLUE
```

На данный момент доступно **2 различные темы** со своими цветовыми

© 2003-2026 Stimulsoft

акцентами. В результате, доступно **более 50 вариантов** оформления. Это позволяет настроить внешний вид дизайнера практически под любое оформление Web-проекта.

10.3.8 События дизайнера

Дизайнер отчетов поддерживает события, которые предоставляют возможность выполнить необходимые операции перед определенными действиями - как на стороне JavaScript клиента, так и на стороне Python сервера. Подробное описание работы событий находится в разделе События генератора отчетов.

Некоторые аргументы событий принимают значения из перечислений, которые находятся в определенных пространствах имен. Все используемые в событиях дизайнера перечисления указаны в блоке кода ниже:

app.py

```
from stimulsoft_reports.enums import StiEventType
```

Дизайнер поддерживает следующие события:

- onPrepareVariables
- onBeginProcessData
- onEndProcessData
- onCreateReport
- onOpenReport
- onOpenedReport
- onSaveReport
- onSaveAsReport
- onPreviewReport
- onCloseReport
- onExit

onPrepareVariables
[v] JavaScript [v] Python

© 2003-2026 Stimulsoft

Событие вызывается перед построением отчета после подготовки переменных отчета. Список аргументов события находится в разделе События генератора отчетов. Подробное описание и примеры использования находятся в разделе Работа с переменными отчета.

onBeginProcessData
[v] JavaScript [v] Python

Событие вызывается перед запросом данных, необходимых для построения отчета. Список аргументов события находится в разделе События генератора отчетов. Подробное описание и примеры использования находятся в разделах Подключение файловых данных и Подключение SQL адаптеров данных.

onEndProcessData
[v] JavaScript [v] Python

Событие вызывается после загрузки данных перед построением отчета. Список аргументов события находится в разделе События генератора отчетов. Подробное описание и примеры использования находятся в разделах Подключение файловых данных и Подключение SQL адаптеров данных.

onCreateReport
[v] JavaScript [v] Python

Событие вызывается после создания нового отчета в дизайнере. В таблице ниже представлен список аргументов обработчика событий на стороне JavaScript клиента:

Наименование	Описание
`event`	Идентификатор текущего события, имеет значение "`CreateReport`".
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать

© 2003-2026 Stimulsoft

	следующие значения: • "Designer"
report	Текущий объект отчета.
isWizardUsed	Данный флаг указывает на то, что создание нового отчета осуществляется при помощи мастера (значение true), либо создается пустой отчет (значение false).
preventDefault	Данный флаг предоставляет возможность остановить дальнейшую обработку события. По умолчанию установлено значение false.

Список свойств, передаваемых в аргументах события на стороне Python сервера. Аргументы имеют тип StiReportEventArgs:

Наименование	Описание
event	Идентификатор текущего события, для данного события имеет значение StiEventType.CREATE_REPORT
sender	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • StiDesigner
report	Текущий объект отчета.
isWizardUsed	Флаг указывает на то, что создание нового отчета осуществляется при помощи мастера (значение True), либо создается пустой отчет (значение False).

© 2003-2026 Stimulsoft

onOpenReport
[v] JavaScript [x] Python

Список свойств, передаваемых в аргументах события на стороне JavaScript клиента:

Наименование	Описание
`event`	Идентификатор текущего события, имеет значение "`OpenReport`".
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • "`Designer`"
`preventDefault`	Данный флаг предоставляет возможность остановить дальнейшую обработку события дизайнером. По умолчанию установлено значение `true`.

onOpenedReport
[v] JavaScript [v] Python

Событие вызывается после открытия отчета из меню дизайнера до его передачи в сам дизайнер.

Список свойств, передаваемых в аргументах события на стороне JavaScript клиента:

Наименование	Описание
`event`	Идентификатор текущего события, имеет значение "`OpenReport`".
`sender`	Идентификатор компонента,

© 2003-2026 Stimulsoft

	который инициировал данное событие, может принимать следующие значения: • "Designer"
report	Текущий объект отчета.
preventDefault	Данный флаг предоставляет возможность остановить дальнейшую обработку события дизайнером. По умолчанию установлено значение true.

Список свойств, передаваемых в аргументах события на стороне Python сервера. Аргументы имеют тип StiReportEventArgs:

Наименование	Описание
event	Идентификатор текущего события, для данного события имеет значение StiEventType.OPENED_REPORT
sender	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • StiDesigner
report	Текущий объект отчета.

onSaveReport
[v] JavaScript [v] Python

Событие вызывается при сохранении отчета в дизайнере. В таблице ниже представлен список аргументов обработчика событий на стороне JavaScript клиента:

© 2003-2026 Stimulsoft

Наименование	Описание
`event`	Идентификатор текущего события, имеет значение "`SaveReport`".
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • "`Designer`"
`report`	Текущий объект отчета.
`fileName`	Имя файла отчета для сохранения.
`autoSave`	Данный флаг на то, что сохранение отчета осуществляется автоматически (значение `true`), либо при нажатии на кнопку сохранения (значение `false`).
`preventDefault`	Данный флаг предоставляет возможность остановить дальнейшую обработку события дизайнером. По умолчанию установлено значение `false`.

Список свойств, передаваемых в аргументах события на стороне Python сервера. Аргументы имеют тип `StiReportEventArgs`:

Наименование	Описание
`event`	Идентификатор текущего события, для данного события имеет значение `StiEventType.SAVE_REPORT`
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • `StiDesigner`

© 2003-2026 Stimulsoft

report	Текущий отчет, представленный в виде объекта.
fileName	Имя файла отчета для сохранения.
autoSave	Данный флаг на то, что сохранение отчета осуществляется автоматически (значение True), либо при нажатии на кнопку сохранения (значение False).

onSaveAsReport
[v] JavaScript [v] Python

Событие вызывается при сохранении отчета в дизайнере с предварительным вводом имени файла. Список свойств, передаваемых в аргументах события на стороне JavaScript клиента:

Наименование	Описание
event	Идентификатор текущего события, имеет значение "SaveAsReport".
sender	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • "Designer"
report	Текущий объект отчета.
fileName	Имя файла отчета для сохранения.
preventDefault	Данный флаг предоставляет возможность остановить дальнейшую обработку события дизайнером. По умолчанию установлено значение false.

Список свойств, передаваемых в аргументах события на стороне Python

© 2003-2026 Stimulsoft

сервера. Аргументы имеют тип `StiReportEventArgs`:

Наименование	Описание
`event`	Идентификатор текущего события, для данного события имеет значение `StiEventType.SAVE_AS_REPORT`
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • `StiDesigner`
`report`	Текущий объект отчета.
`fileName`	Имя файла отчета для сохранения.

onPreviewReport
[v] JavaScript [v] Python

Событие вызывается при переходе на вкладку предварительного просмотра отчета.

Список свойств, передаваемых в аргументах события на стороне JavaScript клиента:

Наименование	Описание
`event`	Идентификатор текущего события, имеет значение "`PreviewReport`".
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • "`Designer`"
`report`	Текущий объект отчета.

© 2003-2026 Stimulsoft

viewer	Текущий объект встроенного в дизайнер компонента вьювера.
preventDefault	Данный флаг предоставляет возможность остановить дальнейшую обработку события дизайнером. По умолчанию установлено значение true.

Список свойств, передаваемых в аргументах события на стороне Python сервера. Аргументы имеют тип StiReportEventArgs:

Наименование	Описание
event	Идентификатор текущего события, для данного события имеет значение StiEventType.PREVIEW_REPORT
sender	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • StiDesigner
report	Текущий объект отчета.

onCloseReport
[v] JavaScript [v] Python

Событие вызывается после закрытия отчета из меню дизайнера до его отмены в дизайнере.

Список свойств, передаваемых в аргументах события на стороне JavaScript клиента:

Наименование	Описание
event	Идентификатор текущего события,

© 2003-2026 Stimulsoft

	имеет значение "CloseReport".
sender	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • "Designer"
report	Текущий объект отчета.
preventDefault	Данный флаг предоставляет возможность остановить дальнейшую обработку события дизайнером. По умолчанию установлено значение true.

Список свойств, передаваемых в аргументах события на стороне Python сервера. Аргументы имеют тип StiReportEventArgs:

Наименование	Описание
event	Идентификатор текущего события, для данного события имеет значение StiEventType.CLOSE_REPORT
sender	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • StiDesigner
report	Текущий объект отчета.

onExit
[v] JavaScript [x] Python

Событие вызывается при нажатии на кнопку **Выход (Exit)** в главном меню дизайнера.

© 2003-2026 Stimulsoft

Список свойств, передаваемых в аргументах события на стороне JavaScript клиента:

Наименование	Описание
`event`	Идентификатор текущего события, имеет значение "`Exit`".
`sender`	Идентификатор компонента, который инициировал данное событие, может принимать следующие значения: • "`Designer`"

10.3.9 Настройки

Настройка дизайнера выполняется при помощи изменения значений свойств, находящихся в главном контейнере свойств `options` у компонента. Все свойства разделены на группы для удобства использования. Все перечисления, используемые в настройках вьювера, находятся в пространстве имен `stimulsoft_reports.designer.enums`.

Пример изменения некоторых настроек дизайнера:

© 2003-2026 Stimulsoft

app.py

```
from flask import Flask, request
from stimulsoft_reports.designer import StiDesigner
from stimulsoft_reports.designer.enums import StiReportUnitType,
StiDesignerTheme

@app.route('/designer', methods = ['GET', 'POST'])
def designer():
    designer = StiDesigner()
    designer.options.appearance.theme =
StiDesignerTheme.OFFICE_2022_DARKGRAY_BLUE
    designer.options.appearance.defaultUnit =
StiReportUnitType.CENTIMETERS
    designer.options.appearance.showReportTree = False
    designer.options.appearance.showTooltips = False
    designer.options.bands.showChildBand = False
    designer.options.components.showPanel = False
    designer.options.toolbar.showFileMenuExit = False
    designer.options.toolbar.showFileMenuOptions = False

    if designer.processRequest(request):
        return designer.getFrameworkResponse()

    # Here is the code for working with the report

    return designer.getFrameworkResponse()
```

Главная (без группы)

Наименование	Описание
width	Задает ширину компонента в "px" или "%". По умолчанию установлено значение "100%".
height	Задает высоту компонента в "px" или "%". По умолчанию установлено значение "800px".
localization	Устанавливает выбранную локализацию компонента. По умолчанию установлена английская локализация, которая встроена в сам компонент.

© 2003-2026 Stimulsoft

Оформление (Appearance)

Наименование	Описание
`theme`	Задает тему оформления дизайнера. Список доступных тем оформления находится в перечислении `StiDesignerTheme`. По умолчанию установлено значение `StiDesignerTheme.OFFICE_2022_WHITE_BLUE`.
`iconSet`	Предоставляет возможность установить набор иконок: • `StiWebUIIconSet.AUTO` (значение по умолчанию) - устанавливает набор иконок автоматически. Для тем `Office2022` - используется набор иконок в стиле `Monoline`, для тем `Office2013` - используется набор иконок в стиле `Regular`. • `StiWebUIIconSet.MONOLINE` - устанавливает набор иконок в стиле `Monoline`; • `StiWebUIIconSet.REGULAR` - устанавливает набор иконок в стиле `Regular`.
`defaultUnit`	Устанавливает единицы измерения размеров для отчета и всех его компонентов: • `StiReportUnitType.CENTIMETERS` (значение по уомлчанию); • `StiReportUnitType.HUNDREDTHS_OF_INCH`;

© 2003-2026 Stimulsoft

	• StiReportUnitType.INCHES; • StiReportUnitType.MILLIMETERS.
zoom	Задает масштаб отображения страниц отчета. По умолчанию установлен масштаб 100 процентов. Допускаются значения от 10 до 200 процентов. Может принимать одно из указанных ниже значений перечисления StiZoomMode: • StiZoomMode.PAGE_WIDTH – масштаб страниц отчета по ширине страницы; • StiZoomMode.PAGE_HEIGHT – масштаб страниц отчета по высоте страницы.
interfaceType	Устанавливает тип интерфейса дизайнера. Могут использоваться следующие значения: • StiInterfaceType.AUTO – тип интерфейса дизайнера будет выбран автоматически в зависимости от используемого устройства (значение по умолчанию); • StiInterfaceType.MOUSE – принудительное использование стандартного интерфейса для управления дизайнером при помощи мыши; • StiInterfaceType.TOUCH – принудительное использование Touch интерфейса для управления

© 2003-2026 Stimulsoft

	дизайнером при помощи сенсорного экрана монитора, в этом режиме элементы интерфейса дизайнера имеют увеличенные размеры для удобства управления.
showAnimation	Предоставляет возможность включить или выключить анимацию отображения и закрытия различных меню в дизайнере. По умолчанию, установлено значение True.
showSaveDialog	Включает отображение диалога ввода имени отчета при его сохранении. Имя отчета будет передано в параметрах дизайнера. По умолчанию свойство имеет значение True.
showTooltips	Включает отображение подсказок для элементов управления дизайнера при наведении мыши. По умолчанию свойство имеет значение True.
showTooltipsHelp	Включает отображение ссылки на онлайн документацию в подсказках для элементов управления дизайнера. По умолчанию свойство имеет значение True.
showDialogsHelp	Предоставляет возможность отображать или не отображать кнопку вызова справки в различных меню. По умолчанию, установлено значение True.
fullScreenMode	Устанавливает полноэкранный режим отображения дизайнера. Если свойство установлено в

© 2003-2026 Stimulsoft

	значение True, то значения свойств width и height игнорируются. По умолчанию, установлено значение `False`.
`maximizeAfterCreating`	Предоставляет возможность установить максимальный размер дизайнера отчетов. По умолчанию свойство имеет значение `False`.
`showLocalization`	Предоставляет возможность отображать или не отображать элемент управления локализацией в дизайнере отчете. По умолчанию свойство имеет значение `True`.
`allowChangeWindowTitle`	Разрешает использование заголовка окна браузера для отображения имени файла редактируемого отчета. По умолчанию свойство имеет значение `True`.
`showPropertiesGrid`	Включает отображение панели свойств в дизайнере отчетов. По умолчанию свойство имеет значение `True`.
`showReportTree`	Включает отображение дерева компонентов отчета. По умолчанию свойство имеет значение `True`.
`propertiesGridPosition`	Предоставляет возможность определить позицию панели свойств. Могут использоваться следующие значения: • `StiPropertiesGridPosition.LEFT` – панель свойств будет отображаться слева (значение по умолчанию); • `StiPropertiesGridPosition.RIGHT` – панель свойств

© 2003-2026 Stimulsoft

	будет отображаться справа.
showSystemFonts	Предоставляет возможность отображать или не отображать системные шрифты в списке шрифтов. По умолчанию свойство имеет значение True, т.е. системные шрифты отображаются в списке шрифтов.
datePickerFirstDayOfWeek	Предоставляет возможность установить первый день недели для инструмента **Выбор даты (Date picker)**. • StiFirstDayOfWeek.AUTO - Будет установлен понедельник или воскресенье как первый день недели, в зависимости от культуры браузера (значение по умолчанию); • StiFirstDayOfWeek.MONDAY - Понедельник будет установлен как первый день недели; • StiFirstDayOfWeek.SUNDAY - Воскресенье будет установлен как первый день недели.
formatForDateControls	Предоставляет возможность переопределить формат для элементов управления дат. По умолчанию, значение для текущей опции не задан и формат для дат определяется из локали браузера.
undoMaxLevel	Устанавливает максимальную глубину отмены изменений отчета при его редактировании. Влияет на потребление памяти. По умолчанию установлено 6

© 2003-2026 Stimulsoft

	изменений.
wizardTypeRunningAfterLoad	Предоставляет возможность вызывать мастер создания отчетов после запуска дизайнера отчетов. Может принимать одно из указанных ниже значений перечисления: • StiWizardType.NONE - дизайнер отчетов будет запущен без вызова мастера создания отчета (значение по умолчанию); • StiWizardType.STANDARD_REPORT - дизайнер отчетов будет запущен с вызовом стандартного мастера создания отчета; • StiWizardType.MASTER_DETAIL_REPORT - дизайнер отчетов будет запущен с вызовом master-detail мастера создания отчета; • StiWizardType.LABEL_REPORT - дизайнер отчетов будет запущен с вызовом мастера создания отчета с этикетками; • StiWizardType.INVOICES_REPORT - дизайнер отчетов будет запущен с вызовом мастера создания инвойса; • StiWizardType.ORDERS_REPORT - дизайнер отчетов будет запущен с вызовом мастера создания ордера; • StiWizardType.QUOTATION_REPORT - дизайнер отчетов будет запущен с вызовом мастера создания квоты.

© 2003-2026 Stimulsoft

allowWordWrapTextEditors	Включает или выключает перенос строк в редакторах текста в дизайнере. По умолчанию свойство имеет значение True.
allowLoadingCustomFontsToClientSide	Включает или выключает передачу пользовательских шрифтов на сторону клиента. По умолчанию свойство имеет значение False.
enableShortCutKeys	Включает или выключает обработку сочетаний клавиш. По умолчанию свойство имеет значение True.
defaultRibbonType	Предоставляет возможность установить используемый по умолчанию стиль панели инструментов дизайнера. Может принимать одно из указанных ниже значений перечисления: • StiDesignerRibbonType.CLASSIC - Будет установлен классический стиль (значение по умолчанию); • StiDesignerRibbonType.SINGLE_LINE – Будет установлен компактный однострочный стиль.
allowPropagationEvents	Предоставляет возможность разрешить распространение событий нажатия клавиш, когда дизайнер отчетов не в фокусе. По умолчанию свойство имеет значение true.
propertiesPanelViewMode	Предоставляет возможность закрепить или открепить панель свойств, словарь отчета, дерево отчета. Может принимать одно из указанных ниже значений

© 2003-2026 Stimulsoft

	перечисления: › `Pinned` - панели закреплены (значение по умолчангию); › `Unpinned` - панели откреплены.

Панель инструментов (Toolbar)

Наименование	Описание
`visible`	Включает отображение панели инструментов в дизайнере отчетов. По умолчанию свойство имеет значение `True`.
`showPreviewButton`	Включает или выключает отображение кнопки **Просмотр (Preview)** на панели инструментов дизайнера. По умолчанию свойство имеет значение `True`.
`showSaveButton`	Включает отображение кнопки **Сохранить (Save)** на панели инструментов дизайнера. По умолчанию свойство имеет значение `False`.
`showAboutButton`	Включает отображение кнопки **О программе (About)** на панели инструментов дизайнера. По умолчанию свойство имеет значение `False`.
`showFileMenu`	Включает отображение главного меню дизайнера отчетов. По умолчанию свойство имеет значение `True`.
`showFileMenuNew`	Включает отображение пункта главного меню **Создать (New)**. По умолчанию свойство имеет значение `True`.

© 2003-2026 Stimulsoft

showFileMenuOpen	Включает отображение пункта главного меню **Открыть (Open)**. По умолчанию свойство имеет значение True.
showFileMenuSave	Включает отображение пункта главного меню **Сохранить (Save)**. По умолчанию свойство имеет значение True.
showFileMenuSaveAs	Включает отображение пункта главного меню **Сохранить как (Save As)**. По умолчанию свойство имеет значение True.
showFileMenuClose	Включает отображение пункта главного меню **Закрыть (Close)**. По умолчанию свойство имеет значение True.
showFileMenuExit	Включает отображение пункта главного меню **Выход (Exit)**. По умолчанию свойство имеет значение False.
showFileMenuReportSetup	Включает отображение пункта главного меню **Свойства отчета (Report Setup)**. По умолчанию свойство имеет значение True.
showFileMenuOptions	Включает отображение пункта главного меню **Параметры (Options)**. По умолчанию свойство имеет значение True.
showFileMenuInfo	Включает отображение пункта главного меню **Сведения (Info)**. По умолчанию свойство имеет значение True.
showFileMenuAbout	Включает отображение пункта главного меню **О программе (About)**. По умолчанию свойство имеет значение True.

© 2003-2026 Stimulsoft

showFileMenuNewReport	Включает или выключает отображение пункта главного меню **Новая страница (New Page)**. По умолчанию свойство имеет значение True.
showFileMenuNewDashboard	Включает или выключает отображение пункта главного меню **Новая панель индикаторов (New Dashboard)**. По умолчанию свойство имеет значение True.
showSetupToolboxButton	Включает или выключает отображение кнопки вызова окна настроек боковой панели компонентов отчета. По умолчанию свойство имеет значение True.
showNewPageButton	Включает или выключает отображение кнопки **Новая страница (New Page)** на панели инструментов. По умолчанию свойство имеет значение True.
showNewDashboardButton	Включает или выключает отображение кнопки **Новая панель индикаторов (New Dashboard)** на панели инструментов. По умолчанию свойство имеет значение True.

Секции (Bands)

Наименование	Описание
showReportTitleBand	Включает отображение секции **Заголовок отчета (Report Title)** в списке меню **Секции**. По умолчанию свойство имеет значение True.

© 2003-2026 Stimulsoft

showReportSummaryBand	Включает отображение секции **Итог отчета (Report Summary)** в списке меню **Секции**. По умолчанию свойство имеет значение True.
showPageHeaderBand	Включает отображение секции **Заголовок страницы (Page Header)** в списке меню **Секции**. По умолчанию свойство имеет значение True.
showPageFooterBand	Включает отображение секции **Итог страницы (Page Footer)** в списке меню **Секции**. По умолчанию свойство имеет значение True.
showGroupHeaderBand	Включает отображение секции **Заголовок группы (Group Header)** в списке меню **Секции**. По умолчанию свойство имеет значение True.
showGroupFooterBand	Включает отображение секции **Итог группы (Group Footer)** в списке меню **Секции**. По умолчанию свойство имеет значение True.
showHeaderBand	Включает отображение секции **Заголовок данных (Header)** в списке меню **Секции**. По умолчанию свойство имеет значение True.
showFooterBand	Включает отображение секции **Итог данных (Footer)** в списке меню **Секции**. По умолчанию свойство имеет значение True.
showColumnHeaderBand	Включает отображение секции **Заголовок колонки (Column**

© 2003-2026 Stimulsoft

	Header) в списке меню **Секции**. По умолчанию свойство имеет значение `True`.
`showColumnFooterBand`	Включает отображение секции **Итог колонки (Column Footer)** в списке меню **Секции**. По умолчанию свойство имеет значение `True`.
`showDataBand`	Включает отображение секции **Данные (Data)** в списке меню **Секции**. По умолчанию свойство имеет значение `True`.
`showHierarchicalBand`	Включает отображение секции **Дерево (Hierarchical)** в списке меню **Секции**. По умолчанию свойство имеет значение `True`.
`showChildBand`	Включает отображение секции **Подчиненный (Child)** в списке меню **Секции**. По умолчанию свойство имеет значение `True`.
`showEmptyBand`	Включает отображение секции **Пустые данные (Empty)** в списке меню **Секции**. По умолчанию свойство имеет значение `True`.
`showOverlayBand`	Включает отображение секции **Водяной знак (Overlay)** в списке меню **Секции**. По умолчанию свойство имеет значение `True`.
`showTable`	Включает отображение компонента **Таблица (Table)** в списке меню **Секции**. По умолчанию свойство имеет значение `True`.
`showTableOfContents`	Включает или выключает отображение компонента **Оглавление (Table of Contents)** в

© 2003-2026 Stimulsoft

	списке меню **Секции**. По умолчанию свойство имеет значение `True`.

Кросс секции (Cross-Bands)

Наименование	Описание
`showCrossTab`	Включает отображение компонента **Кросс-таблица (Cross-Tab)** в списке меню **Кросс**. По умолчанию свойство имеет значение `True`.
`showCrossGroupHeaderBand`	Включает отображение секции **Кросс-заголовок группы (Cross Group Header)** в списке меню **Кросс**. По умолчанию свойство имеет значение `True`.
`showCrossGroupFooterBand`	Включает отображение секции **Кросс-итог группы (Cross Group Footer)** в списке меню **Кросс**. По умолчанию свойство имеет значение `True`.
`showCrossHeaderBand`	Включает отображение секции **Кросс-заголовок данных (Cross Header)** в списке меню **Кросс**. По умолчанию свойство имеет значение `True`.
`showCrossFooterBand`	Включает отображение секции **Кросс-итог данных (Cross Footer)** в списке меню **Кросс**. По умолчанию свойство имеет значение `True`.
`showCrossDataBand`	Включает отображение секции **Кросс- данные (Cross Data)**в списке меню **Кросс**. По

© 2003-2026 Stimulsoft

	умолчанию свойство имеет значение `True`.

Элементы дашборда (dashboardElements)

Наименование	Описание
`showTableElement`	Включает отображение элемента панели индикаторов **Таблица (Table)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `True`.
`showCardsElement`	Включает отображение элемента панели индикаторов **Карточки (Cards)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `True`.
`showChartElement`	Включает отображение элемента панели индикаторов **Диаграмма (Chart)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `True`.
`showGaugeElement`	Включает отображение элемента панели индикаторов **Измерительный прибор (Gauge)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `True`.
`showPivotTableElement`	Включает отображение элемента панели индикаторов **Сводная таблица (Pivot)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По

© 2003-2026 Stimulsoft

	умолчанию свойство имеет значение `True`.
`showIndicatorElement`	Включает отображение элемента панели индикаторов **Индикатор (Indicator)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `True`.
`showProgressElement`	Включает отображение элемента панели индикаторов **Прогресс (Progress)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `True`.
`showRegionMapElement`	Включает отображение элемента панели индикаторов **Региональная карта (Region Map)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `True`.
`showOnlineMapElement`	Включает отображение элемента панели индикаторов **Онлайн карта (Online Map)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `True`.
`showImageElement`	Включает отображение элемента панели индикаторов **Изображение (Image)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `True`.
`showWebContentElement`	Включает отображение элемента панели индикаторов **Веб-контент (Web Content)** на инструментарии

© 2003-2026 Stimulsoft

	или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `True`.
`showTextElement`	Включает отображение элемента панели индикаторов **Текст (Text)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `True`.
`showPanelElement`	Включает отображение элемента панели индикаторов **Панель (Panel)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `True`.
`showShapeElement`	Включает отображение элемента панели индикаторов **Геометрия (Shape)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `True`.
`showButtonElement`	Включает отображение элемента панели индикаторов **Кнопка (Button)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `True`.
`showListBoxElement`	Включает отображение элемента панели индикаторов **Список (ListBox)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `True`.
`showComboBoxElement`	Включает отображение элемента панели индикаторов **Выпадающий список (ComboBox)** на инструментарии

© 2003-2026 Stimulsoft

	или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `True`.
`showTreeViewElement`	Включает отображение элемента панели индикаторов **Иерархический список (TreeView)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `True`.
`showTreeViewBoxElement`	Включает отображение элемента панели индикаторов **Выпадающий иерархический список (TreeViewBox)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `True`.
`showDatePickerElement`	Включает отображение элемента панели индикаторов **Выбор даты (Date Picker)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `True`.
`showNumberBoxElement`	Включает отображение элемента панели индикаторов **Поле с номером (Number Box)** на инструментарии или вкладке **Вставить (Insert)** в дизайнере. По умолчанию свойство имеет значение `True`.

Компоненты (Components)

Наименование	Описание

© 2003-2026 Stimulsoft

showText	Включает отображение компонента **Текст (Text)** в списке меню **Компоненты**. По умолчанию свойство имеет значение True.
showTextInCells	Включает отображение компонента **Текст в ячейках (Text in Cells)** в списке меню **Компоненты**. По умолчанию свойство имеет значение True.
showRichText	Включает отображение компонента **Rich текст (Rich Text)** в списке меню **Компоненты**. По умолчанию свойство имеет значение False.
showImage	Включает отображение компонента **Картинка (Image)** в списке меню **Компоненты**. По умолчанию свойство имеет значение True.
showBarCode	Включает отображение компонента **Штрих-код (Bar Code)** на инструментарии или вкладке **Вставить** в дизайнере. По умолчанию свойство имеет значение True.
showShape	Включает отображение компонента **Геометрия (Shape)** на инструментарии или вкладке **Вставить** в дизайнере. По умолчанию свойство имеет значение True.
showPanel	Включает отображение компонента **Панель (Panel)** в списке меню **Компоненты**. По умолчанию свойство имеет значение True.

© 2003-2026 Stimulsoft

showClone	Включает отображение компонента **Клон (Clone)** в списке меню **Компоненты**. По умолчанию свойство имеет значение `False`.
showCheckBox	Включает отображение компонента **Флажок (Check Box)** в списке меню **Компоненты**. По умолчанию свойство имеет значение `True`.
showSubReport	Включает отображение компонента **Вложенный отчет (Sub Report)** в списке меню **Компоненты**. По умолчанию свойство имеет значение `True`.
showZipCode	Включает отображение компонента **Почтовый индекс (Zip Code)** в списке меню **Компоненты**. По умолчанию свойство имеет значение `False`.
showChart	Включает отображение компонента **Диаграмма (Chart)** на инструментарии или вкладке **Вставить** в дизайнере. Действует сразу на все типы диаграмм. По умолчанию свойство имеет значение `True`.
showGauge	Включает или выключает отображение компонента **Измерительный прибор (Gauge)** на инструментарии или вкладке **Вставить** в дизайнере. По умолчанию свойство имеет значение `True`.
showSparkline	Включает отображение компонента **Спарклайн (Sparkline)** в списке меню **Компоненты**. По

© 2003-2026 Stimulsoft

	умолчанию свойство имеет значение `True`.
`showMathFormula`	Включает или выключает отображение компонента **Математическая формула (Math Formula)** в списке меню **Компоненты**. По умолчанию свойство имеет значение `False`.
`showMap`	Включает или выключает отображение компонента **Карта (Map)** на инструментарии или вкладке **Вставить** в дизайнере. По умолчанию свойство имеет значение `True`.
`showElectronicSignature`	Включает или выключает отображение компонента **Электронная подпись (Electronic Signature)** на инструментарии или вкладке **Вставить** в дизайнере. По умолчанию свойство имеет значение `True`.
`showPdfDigitalSignature`	Включает или выключает отображение компонента **Цифровая подпись PDF (PDF Digital Signature)** на инструментарии или вкладке **Вставить** в дизайнере. По умолчанию свойство имеет значение `True`.
`showHorizontalLinePrimitive`	Включает или выключает отображение компонента **Горизонтальная линия (Horizontal Line)** на инструментарии или вкладке **Вставить** в дизайнере. По умолчанию свойство имеет значение `True`.

© 2003-2026 Stimulsoft

showVerticalLinePrimitive	Включает или выключает отображение компонента **Вертикальная линия (Vertical Line)** на инструментарии или вкладке **Вставить** в дизайнере. По умолчанию свойство имеет значение True.
showRectanglePrimitive	Включает или выключает отображение компонента **Прямоугольник (Rectangle)** на инструментарии или вкладке **Вставить** в дизайнере. По умолчанию свойство имеет значение True.
showRoundedRectanglePrimitive	Включает или выключает отображение компонента **Закругленный прямоугольник (Rounded Rectangle)** на инструментарии или вкладке **Вставить** в дизайнере. По умолчанию свойство имеет значение True.

Словарь данных (Dictionary)

Наименование	Описание
showAdaptersInNewConnectionForm	Включает отображение категории **Объекты (Object)** в окне создания нового соединения. По умолчанию свойство имеет значение True.
showDictionary	Включает отображение словаря данных отчета. По умолчанию свойство имеет значение True.
newReportDictionary	Предоставляет возможность создавать новый словарь данных или объединить с существующим,

© 2003-2026 Stimulsoft

	при создании нового отчета в дизайнере. Может принимать одно из указанных ниже значений: › `StiNewReportDictionary.AUTO` - определяет режим создания или объединения словаря данных из сохраненного значения в файлах `cookies` (значение по умолчанию); › `StiNewReportDictionary.DICTIONARY_NEW` - устанавливает режим создания нового словаря данных, при создании нового отчета; › `StiNewReportDictionary.DICTIONARY_MERGE` - устанавливает режим объединения существующего словаря данных с новым, при создании нового отчета в дизайнере.
`useAliases`	Предоставляет возможность использовать псевдонимы в словаре данных. Может принимать одно из указанных ниже значений: › `StiUseAliases.AUTO` - определяет режим использования псевдонимов из сохраненного значения в файлах cookies (значение по умолчанию); › `StiUseAliases.True` - устанавливает режим использования псевдонимов в словаре данных; › `StiUseAliases.False` - выключает режим использования псевдонимов в словаре данных.

© 2003-2026 Stimulsoft

showDictionaryContextMenuProperties	Устанавливает видимость пункта **Свойства (Properties)** в контекстном меню словаря данных. По умолчанию свойство имеет значение True.
showDictionaryActions	Устанавливает отображение меню **Действия (Actions)** на панели инструментов словаря данных. По умолчанию свойство имеет значение **True**.
dataSourcesPermissions	Устанавливает доступные действия над источниками данных отчета. Может принимать одно или несколько значений из перечисления StiDesignerPermissions.
dataConnectionsPermissions	Устанавливает доступные действия над подключениями к данным отчета. Может принимать одно или несколько значений из перечисления StiDesignerPermissions.
dataColumnsPermissions	Устанавливает доступные действия над колонками данных отчета. Может принимать одно или несколько значений из перечисления StiDesignerPermissions.
dataRelationsPermissions	Устанавливает доступные действия над связями данных отчета. Может принимать одно или несколько значений из перечисления StiDesignerPermissions.
businessObjectsPermissions	Устанавливает доступные действия над бизнес-объектами отчета. Может принимать одно или несколько значений из

© 2003-2026 Stimulsoft

	перечисления StiDesignerPermissions.
variablesPermissions	Устанавливает доступные действия над переменными отчета. Может принимать одно или несколько значений из перечисления StiDesignerPermissions.
resourcesPermissions	Устанавливает доступные действия над ресурсами в словаре данных отчета. Может принимать одно или несколько значений из перечисления StiDesignerPermissions.
dataTransformations	Устанавливает доступные действия над преобразованием данных. Может принимать одно или несколько значений из перечисления StiDesignerPermissions.
userFunctions	Устанавливает доступные действия над пользовательскими функциями. Может принимать одно или несколько значений из перечисления StiDesignerPermissions.

В таблице ниже приведены все доступные значения для перечисления StiDesignerPermissions, которые можно установить для элементов словаря отчета.

Значение	Описание
StiDesignerPermissions.NONE	Запрещает любое действие над элементом словаря данных.
StiDesignerPermissions.CREATE	Разрешает создавать определенный элемент словаря

© 2003-2026 Stimulsoft

	данных.
StiDesignerPermissions.DELETE	Разрешает удалять определенный элемент словаря данных.
StiDesignerPermissions.MODIFY	Разрешает редактировать определенный элемент словаря данных.
StiDesignerPermissions.VIEW	Разрешает просматривать определенный элемент словаря данных.
StiDesignerPermissions.MODIFY_VIEW	Разрешает редактировать и просматривать определенный элемент словаря данных.
StiDesignerPermissions.ALL	Разрешает любые действия над элементом словаря данных (установлено по умолчанию для всех свойств, использующих данное перечисление).

Предусмотрена настройка встроенного компонента `StiViewer`, используемого для предварительного просмотра отчета. Для доступа ко всем его настройкам предназначено свойство `viewerOptions`, которое представляет собой объект опций вьювера. Все его свойства описаны в разделе Настройки вьювера. Например, требуется изменить режим отображения отчета и отключить ненужные форматы экспорта:

© 2003-2026 Stimulsoft

app.py

```
from flask import Flask, request
from stimulsoft_reports.designer import StiDesigner
from stimulsoft_reports.designer.enums import StiReportUnitType,
StiDesignerTheme

@app.route('/designer', methods = ['GET', 'POST'])
def designer():
    designer = StiDesigner()
    designer.options.viewerOptions.exports.showExportToWord = False
    designer.options.viewerOptions.exports.showExportToCsv = False

    if designer.processRequest(request):
        return designer.getFrameworkResponse()

    # Here is the code for working with the report

    return designer.getFrameworkResponse()
```

11 Reports and Dashboards for Net

YouTube

Посмотрите видеоуроки по работе с компонентами Stimulsoft Reports.NET. Подписывайтесь на канал Stimulsoft и узнайте первыми о новых видеоуроках. Вопросы и предложения оставляйте в комментариях к видео.

Samples

Ознакомьтесь с примерами по работе с компонентом Reports.Net на нашем сайте. Все примеры представляют собой отдельные проекты, сгруппированные в один solution для Visual Studio. Так же, вы можете просмотреть и скачать указанные примеры на GitHub.

- WinForms Viewer
- Активация

© 2003-2026 Stimulsoft

11.1 Активация

YouTube

Посмотрите видеоуроки по активации WinForms компонентов. Подписывайтесь на канал Stimulsoft и узнайте первыми о новых видеоуроках. Вопросы и предложения оставляйте в комментариях к видео.

После приобретения продукта Stimulsoft вам необходимо активировать лицензию для используемых компонентов. Это можно выполнить различными способами. Ниже представлен пример активации WinForms компонентов.

Form1.cs

```
...
public partial class Form1 : Form
{
   public Form1()
   {
      //Activation with using license code
      Stimulsoft.Base.StiLicense.Key = "Your activation code...";

      //Activation with using license file
      Stimulsoft.Base.StiLicense.LoadFromFile("license.key");

      //Activation from byte array
      Stimulsoft.Base.StiLicense.LoadFromBytes(bytes);

      //Activation from stream
      Stimulsoft.Base.StiLicense.LoadFromStream(stream);

      //Activation from assembly
      Stimulsoft.Base.StiLicense.LoadFromEntryAssembly(assembly,
      "stimulsoft-license.key");

      InitializeComponent();
   }
}
...
```

Получить лицензионный ключ или скачать файл с лицензионным ключом, вы можете в личном кабинете пользователя. Для авторизации в кабинете используйте имя пользователя и пароль, указанные при покупке подписки продукта..

© 2003-2026 Stimulsoft

11.2 WinForms вьювер

Для отображения отчетов в **WinForms** используется компонент **StiViewerControl**. Компонент умеет отображать отчет, масштабировать, сохранять построенный отчет в различные форматы, выводить его на печать, отправлять построенные отчеты по почте.

- Как отобразить отчет,
- Dot-Matrix режим.

11.2.1 Как отобразить отчет

Примечание

При присвоении отчета вьювер компоненту, происходит автоматическое построение отчета. Вызов метода `Report.Render()` необходим только в том случае, если требуется провести какие-либо действия с построенным отчетом перед его отображением во вьювере. Аналогично, при использовании режима компиляции, вызов метода `Report.Compile()` требуется только в том случае, если требуется провести какие-либо действия с откомпилированным отчетом перед его построением и отображением во вьювере.

Отображение вьювера возможно несколько способами. Отчет можно загрузить в размещенной на форме **StiViewerControl** вашего приложения, или отобразить вьювер с отчетом в отдельном диалоговом окне с использованием **Show()** или **ShowWithRibbonGUI()** метода объекта **StiReport**.

В следующих разделах документации **StiViewerControl**, мы будем рассматривать примеры кода с размещение вьювера на форме вашего приложения. Однако, вы можете использовать методы **Show()** или **ShowWithRibbonGUI()** для отображения отчета без каких либо ограничений.

Отображение отчета в размещенном **StiViewerControl** на форме вашего приложения.

© 2003-2026 Stimulsoft

Form1.cs

```
...
public Form1()
{
   InitializeComponent();

   var report = new StiReport();
   report.Load(@"Reports\SimpleList.mrt");

   report.Render();
   stiViewerControl1.Report = report;
}
...
```

Информация

Обратите внимание, что при отображении отчета в размещенном **StiViewerControl** на форме вашего приложения, необходимо использовать метод Render(). Этот метод осуществляет построение отчета.

Отображение отчета с использованием метода **Show()** объекта **StiReport**.

C#

```
...
private void buttonDialog_Click(object sender, EventArgs e)
{
   StiReport report = new StiReport();
   report.Load("report.mrt");
   report.Show();
   //report.ShowWithRibbonGUI();
}
...
```

© 2003-2026 Stimulsoft

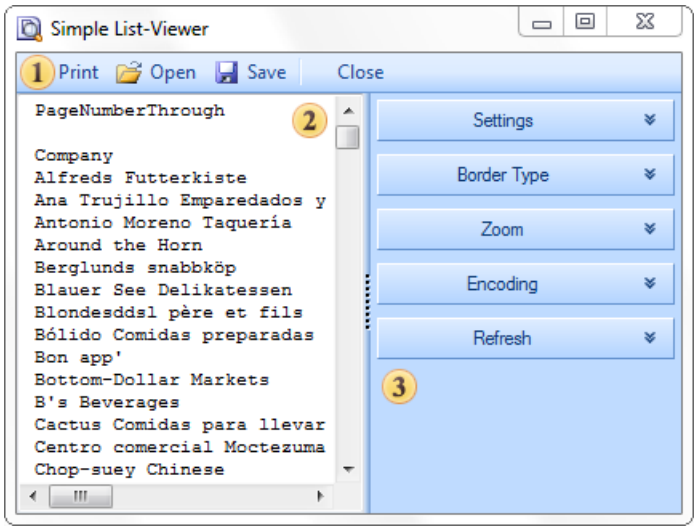
Информация

Методы **Show()** и **ShowWithRibbonGUI()** предоставляют возможность отобразить вьювер с отчетом в отдельном диалоговом окне. Разница, лишь в том, что команды управления вьювера будут представлены на Standard или Ribbon панели управления соответственно.

Также, стоит отметить, что при использовании отображения отчета с помощью этих методов, нет необходимости использовать метод **Render()** объекта **StiReport**.

11.2.2 Dot-Matrix режим

Dot-matrix вьювер предназначен для предварительного просмотра отчета перед печатью на матричном принтере. Матричный принтер предназначен для печати только текста и символов псевдографики. Соответственно вьювер отображает только текст и границы объектов в виде символов псевдографики. На рисунке снизу представлено окно **Dot-matrix** вьювера:

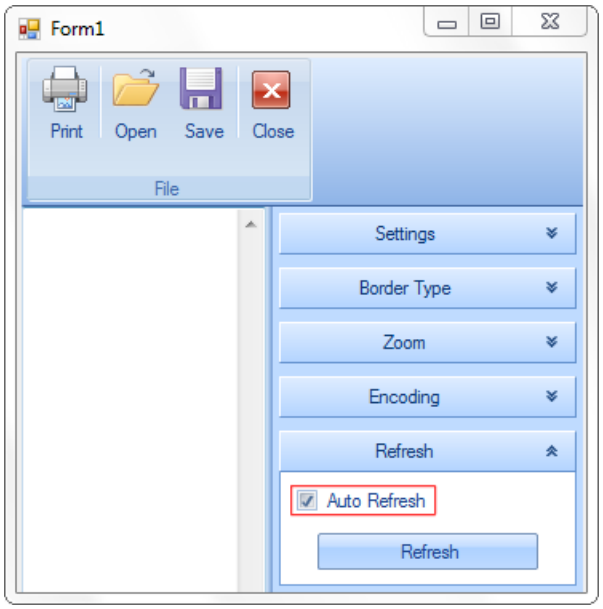


1. Панель инструментов (Toolbars) Dot-matrix вьювера.
2. Панель отображения текста отчета.
3. Панель параметров отображаемого отчета.

Настройка Dot-Matrix вьювера отчета для WinForms

© 2003-2026 Stimulsoft

Dot-Matrix вьювер можно настроить из кода используя статические свойства. В зависимости от значения статических свойств в **Dot-matrix** вьювере будут определены те или иные параметры. Рассмотрим на примере свойства **Автообновление (Auto Refresh)**. На рисунке снизу представлено окно Dot-matrix вьювера:



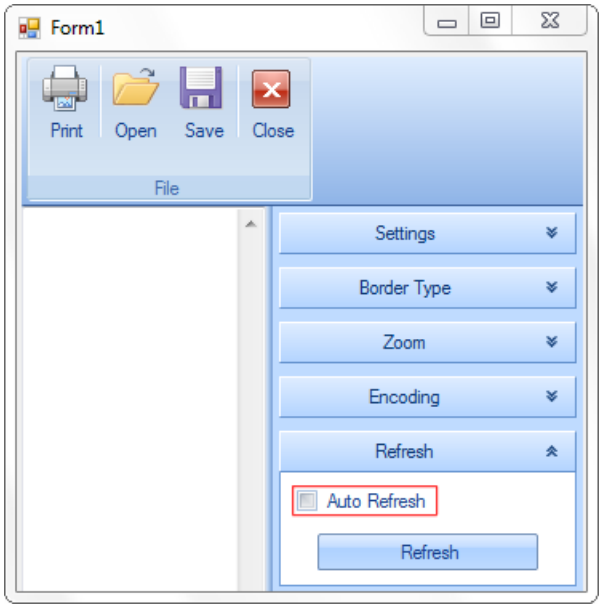
Как видно из рисунка, свойство **Автообновление (Auto Refresh)** включено, значит статическое свойство **Dot-matrix** вьювера **AutoRefresh** установлено в значение **true**. Если статическое свойство **AutoRefresh** установить в значение **false**, то свойство **Автообновление (Auto Refresh)** в **Dot-matrix** вьювере будет выключено. Добавим ниже приведенный код в код проекта:

C#

```
...
StiOptions.Viewer.DotMatrix.AutoRefresh = false;
...
```

Таким образом, свойство **Автообновление (Auto Refresh)** будет выключено. На рисунке снизу представлено окно **Dot-matrix** вьювера с выключенным автообновлением:

© 2003-2026 Stimulsoft



Большинство значений параметров можно установить с помощью статических свойств.

DotMatrix вьювер и Escape коды

Полноценного управления матричным принтером в **Stimulsoft Reports.Net** нет, но начиная с версии **2009.2** в **Dot-Matrix** вьювере можно использовать **Escape**-последовательности в тексте для изменения параметров: размер встроенных шрифтом матричного принтера, их начертание и др. Ниже приведен пример **Escape**-последовательности в тексте:

```
{(char)15 + "Your condensed text " + (char)18 + "and normal text"}
```

В версии 2010.2 **Stimulsoft Reports.Net** появилась более расширенная поддержка **Escape**-кодов. Для ввода Escape-последовательности в текст надо использовать команды вида **<#command>**, например:

```
Normal text <#b> Bold text <#/b><#i> Italic text <#/i> Again normal text
```

Также автоматически вставляются команды выбора жирного, наклонного или подчеркнутого начертания в зависимости от установленного стиля шрифта текстового компонента. При печати на матричный принтер и при экспорте в текстовый формат эти команды заменяются на соответствующие Escape-

© 2003-2026 Stimulsoft

последовательности. Для задания соответствий используется класс **StiEscapeCodesCollection**. Он унаследован от класса **Hashtable** и представляет собой коллекцию пар ключ-значение, в которой ключом является команда, а значением - соответствующая Escape-последовательность. Для разных типов принтеров можно задавать разные коллекции с разным набором команд. Хранятся коллекции в статической переменной **StiOptions.Export.Txt.EscapeCodesCollectionList**. По умолчанию созданы следующие коллекции: "**None**", "**EpsonFX**", "**Oki ML92/93**". Коллекция "**None**" пустая и служит для вывода текста без escape-кодов.

Команда \ Коллекция	EpsonFX	Oki ML92/93
b	ESC E	ESC T
/b	ESC F	ESC I
i	ESC 4	
/i	ESC 5	
u	ESC -1	ESC H
/u	ESC -0	ESC D
sup	ESC S0	ESC J
/sup	ESC T	ESC K
sub	ESC S1	ESC L
/sub	ESC T	ESC M
condensed	0x0F	0x1d
/condensed	0x12	0x1e
elite	ESC M	0x1c
pica	ESC P	0x1e
doublewidth	ESC W1	0x1f
/doublewidth	ESC W0	0x1e

Можно добавлять новые коллекции или изменять существующие. Выбор нужной коллекции осуществляется по имени. Если коллекция с нужным именем не найдена, используется коллекция "None". Имя коллекции можно выбрать в настройках Dot-Matrix Viewer или передать в качестве параметра в методы

© 2003-2026 Stimulsoft

экспорта и печати.

11.3 Добавление пользовательских функций

Информация

При работе в дизайнере отчетов при работе с командами undo/redo, сохранение стека операций выполняется в JSON. Если же необходимо стек операций сохранять в XML следует установить опции `StiOptions.Designer.UndoRedoMode = StiUndoRedoMode.Xml`.

При интеграции дизайнера отчетов в пользовательское приложение, присутствует возможность добавления custom функции в словарь данных дизайнера отчетов. После добавления функции, ее можно использовать при разработке отчетов и панелей индикаторов (дашбордов). Ниже представлен пример добавления функции расчета итога суммирования.

Designer.cs

```
...
StiFunctions.AddFunction("MyCategory", "MySum",
"description", typeof(MyClass),
typeof(decimal), "Calculates a sum of the specified set of values.",
new[] { typeof(object) },
new[] { "values" },
new[] { "A set of values" }).UseFullPath = false;
...
public static decimal MySum(object value)
{
   if (!ListExt.IsList(value))
   return Stimulsoft.Base.Helpers.StiValueHelper.TryToDecimal(value);
   return
   Stimulsoft.Data.Functions.Funcs.SkipNulls(ListExt.ToList(value)).TryCast
   ToDecimal().Sum();
}
...
```

© 2003-2026 Stimulsoft

12 Reports.JAVA

YouTube

Посмотрите видеоуроки по работе с компонентами продукта Stimulsoft Reports.JAVA. Подписывайтесь на канал Stimulsoft и узнайте первыми о новых видеоуроках. Вопросы и предложения оставляйте в комментариях к видео.

Samples

Ознакомьтесь с примерами по работе с компонентом продукта Reports.JAVA на нашем сайте. Все примеры представляют собой отдельные проекты, сгруппированные в один solution для Visual Studio. Так же, вы можете просмотреть и скачать указанные примеры на GitHub.

Технология Java позволяет использовать созданные программы на различных платформах, в различных операционных системах и на различном аппаратном обеспечении. Компания Stimulsoft предоставляет инструменты создания и отображения отчетов на любых устройствах с использованием этой технологии.

12.1 Активация

После приобретения продукта Stimulsoft вам необходимо активировать лицензию для используемых компонентов. Это можно выполнить указав лицензионный ключ или загрузив файл с лицензионным ключом. Ниже представлен пример активации компонента **StiWebViewer**.

© 2003-2026 Stimulsoft

index.jsp

```
...
<body>
   <%
      //Activation with using license code
      com.stimulsoft.base.licenses.StiLicense.setKey(
      "Your activation code...";

      //Activation with using license file
      com.stimulsoft.base.licenses.StiLicense.loadFromFile(request.getSessi
      on().getServletContext().getRealPath("/license/license.key"));
   %>
   <stiwebviewer:webviewer report="${report}" options="${options}" />
</body>
...
```

Получить лицензионный ключ или скачать файл с лицензионным ключом, вы можете в личном кабинете пользователя. Для авторизации в кабинете используйте имя пользователя и пароль, указанные при покупке подписки продукта.

12.2 HTML5 дизайнер

Работа с компонентом Java HTML5 дизайнер описано в этой главе:

- Установка и описание дизайнера
- Шаблоны соединений

12.2.1 Установка и описание дизайнера

Создание макета страницы при помощи дизайнера отчетов web

Рассмотрим пример, создания простой страницы в web-дизайнере. Для этого необходимо, в директорию **WebContent\WEB-INF\Lib** поместить следующие библиотеки: stimulsoft.lib.jar, stimulsoft.reports-base.jar, stimulsoft.reports-report.jar, stimulsoft.reports-flex.jar, stimulsoft.reports-web.jar, stimulsoft.reports-webdesigner.jar. Далее, следует отредактировать **web.xml**, пример представлен ниже:

© 2003-2026 Stimulsoft

web.xml

```
...
<?xml version="1.0" encoding="UTF-8" ?>
  <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://java.sun.com/xml/ns/javaee" xmlns:web="http://
    java.sun.com/xml/ns/javaee/webapp_2_5.xsd"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee"
    id="WebApp_ID" version="2.5">
  <display-name>sti_webdesigner</display-name>
  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
    <!-- configuration, this parameter indicates the main application
    directory -->
  <servlet>
    <servlet-name>StimulsoftResource</servlet-name>
    <servlet-class>com.stimulsoft.web.servlet.StiWebResourceServlet</
    servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>StimulsoftResource</servlet-name>
    <url-pattern>/stimulsoft_web_resource/*</url-pattern>
  </servlet-mapping>
  <servlet>
    <servlet-name>StimulsoftDesignerAction</servlet-name>
    <servlet-
    class
    >com.stimulsoft.webdesigner.servlet.StiWebDesignerActionServlet</
    servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>StimulsoftDesignerAction</servlet-name>
    <url-pattern>/stimulsoft_webdesigner_action</url-pattern>
  </servlet-mapping>
</web-app>
...
```

Оставить без изменений остальные блоки web.xml, которые определяют сервлеты, необходимые для работы. Затем, следует отредактировать **index.jsp**:

© 2003-2026 Stimulsoft

index.jsp

```
...
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://
www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<%@page import="java.io.FileOutputStream"%>
<%@page  import="java.io.FileInputStream"%>
<%@page  import="com.stimulsoft.report.utils.data.StiDataColumnsUtil"%>
<%@page
import="com.stimulsoft.report.dictionary.StiDataColumnsCollection"%>
<%@page  import="com.stimulsoft.report.dictionary.StiDataColumn"%>
<%@page  import="com.stimulsoft.report.utils.data.StiSqlField"%>
<%@page
import="com.stimulsoft.report.dictionary.dataSources.StiDataTableSource"%>
<%@page  import="com.stimulsoft.report.utils.data.StiXmlTable"%>
<%@page
import="com.stimulsoft.report.utils.data.StiXmlTableFildsRequest"%>
<%@page  import="com.stimulsoft.webdesigner.StiWebDesigerHandler"%>
<%@page  import="com.stimulsoft.webdesigner.StiWebDesignerOptions"%>
<%@page
  import="com.stimulsoft.report.dictionary.databases.StiXmlDatabase"%>
<%@page  import="java.io.File"%>
<%@page  import="com.stimulsoft.report.StiSerializeManager"%>
<%@page  import="com.stimulsoft.report.StiReport"%>
<%@page  language="java" contentType="text/html; charset=utf-8"
  pageEncoding="UTF-8"%>
<%@taglib uri="http://stimulsoft.com/webdesigner" prefix="stiwebdesigner"%
>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Stimulsoft Webdesigner for Java</title>
  <style type="text/css">
  </style>
</head>
<body>
  <%
    final String reportPath =
    request.getSession().getServletContext().getRealPath("/reports/
    Master-Detail.mrt");
    final String xmlPath =
    request.getSession().getServletContext().getRealPath("/data/
    Demo.xml");
    final String xsdPath =
    request.getSession().getServletContext().getRealPath("/data/
    Demo.xsd");
    final String savePath =
    request.getSession().getServletContext().getRealPath("/save/");

    StiWebDesignerOptions options = new StiWebDesignerOptions();

    StiWebDesigerHandler handler = new StiWebDesigerHandler(){
      public StiReport getEditedReport(HttpServletRequest request){
        try{
          StiReport report = StiSerializeManager.deserializeReport(new
          File(reportPath));
          report.getDictionary().getDatabases().add(new
          StiXmlDatabase("Demo", xsdPath, xmlPath));
          return report;
        } catch (Exception e){
            e.printStackTrace();
        }
        return null;
      }
```

© 2003-202

В **JSP** необходимо добавить **taglib** директивы. Они будут работать с пользовательскими тегами на странице.

Тег Custom Stimulsoft

index.jsp

```
...
<%@ taglib uri="http://stimulsoft.com/webdesigner"
prefix="stiwebdesigner"%>
...
```

Добавьте тег `<stiwebdesigner:resources />`, тег используется для загрузки необходимых ресурсов (CSS и JS) для веб-дизайнера. У него отсутствуют атрибуты, и он должен быть помещен внутрь тега HTML <head>.

Описание тэга webdesigner

Тэг содержит следующие атрибуты:

- **handler** [required] – com.stimulsoft.webdesigner.StiWebDesigerHandler объект для управления web-дизайнером;
- **options** [optional] – Объект StiWebdesignerOptions для настройки webdesigner. Если нет - опция используется по умолчанию;
- **designerID** [optional] – Строковое значение идентификатора элемента HTML веб-дизайнера. Если более чем один веб-дизайнер размещен на HTML странице, каждый веб-дизайнер должен иметь различные идентификаторы.

web.xml

```
...
<stiwebdesigner:webdesigner handler="${handler}" report="${report}" />
...
```

Описание StiWebDesigerHandler

Для управления событиями дизайнера, класс который применяет

© 2003-2026 Stimulsoft

StiWebDesigerHandler должен быть создан и настроен в теге StiWebDesigner-а. Происходит на открытии {@link StiReport}.

- **public StiReport getEditedReport(HttpServletRequest request)**;
  Возникает при загрузке веб-дизайнера. Здесь необходимо осуществление загрузки отчета и его заполнение (если необходимо).

- **public void onOpenReportTemplate(StiReport report, HttpServletRequest request)**;
  Возникает при открытии отчета (объекта StiReport). Метод предназначен для заполнения отчета данными из источника (если необходимо).

- **public void onNewReportTemplate(StiReport report, HttpServletRequest request);**
  Возникает при создании отчета (объекта StiReport). Метод предназначен для заполнения отчета данными из источника (если необходимо). Если необходимо создать новый отчет при помощи мастера, необходимо передать в отчет данные из источника.

- **public void onSaveReportTemplate(StiReport report, String reportName, HttpServletRequest request);**
  Возникает при сохранении отчета (объекта StiReport).

12.2.2 Шаблоны соединений

Общий шаблон соединения имеет вид:

***.mrt**

```
...
jdbc.driver={myDriver};
jdbc.url={myConnectionUrl};
jdbc.username={myUserName };
jdbc.password={ myUserPassword };
...
```

Ниже представлены примеры соединения для

SQLServer

© 2003-2026 Stimulsoft

***.mrt**

```
...
jdbc.driver=com.microsoft.sqlserver.jdbc.SQLServerDriver;
jdbc.url= jdbc:sqlserver://[serverName[\instanceName][:portNumber]]
[;property=value[;property=value]];
jdbc.username={myUserName };
jdbc.password={ myUserPassword };
...
```

http://msdn.microsoft.com/en-us/library/ms378428

Oracle

***.mrt**

```
...
jdbc.driver=oracle.jdbc.driver.OracleDriver;
jdbc.url=jdbc:oracle:thin:@[HOST][:PORT]:SID;
jdbc.username={myUserName };
jdbc.password={ myUserPassword };
...
```

http://www.orafaq.com/wiki/JDBC

POSTGreSQL

***.mrt**

```
...
jdbc.driver= org.postgresql.Driver
jdbc.url= jdbc:postgresql://[host]:[port]/[database]
jdbc.username={myUserName };
jdbc.password={ myUserPassword };
...
```

http://jdbc.postgresql.org/documentation/80/connect.html

© 2003-2026 Stimulsoft

13 Reports.WPF

YouTube

Посмотрите видеоуроки по работе с компонентами Stimulsoft Reports.WPF. Подписывайтесь на канал Stimulsoft и узнайте первыми о новых видеоуроках. Вопросы и предложения оставляйте в комментариях к видео.

Samples

Ознакомьтесь с примерами по работе с компонентом Reports.WPF на нашем сайте. Все примеры представляют собой отдельные проекты, сгруппированные в один solution для Visual Studio. Так же, вы можете просмотреть и скачать указанные примеры на GitHub.

- WPF Viewer
- Активация

13.1 Активация

YouTube

Посмотрите видеоуроки по активации WPF компонентов. Подписывайтесь на канал Stimulsoft и узнайте первыми о новых видеоуроках. Вопросы и предложения оставляйте в комментариях к видео.

После приобретения продукта Stimulsoft вам необходимо активировать лицензию для используемых компонентов. Это можно выполнить различными способами. Ниже представлен пример активации WPF компонентов.

© 2003-2026 Stimulsoft

MainWindows.xaml..cs

```
...
public partial class MainWindow : Window
{
   public MainWindow()
   {
      //Activation with using license code
      Stimulsoft.Base.StiLicense.Key = "Your activation code...";

      //Activation with using license file
      Stimulsoft.Base.StiLicense.LoadFromFile("license.key");

      //Activation from byte array
      Stimulsoft.Base.StiLicense.LoadFromBytes(bytes);

      //Activation from stream
      Stimulsoft.Base.StiLicense.LoadFromStream(stream);

      //Activation from assembly
      Stimulsoft.Base.StiLicense.LoadFromEntryAssembly(assembly,
      "stimulsoft-license.key");

      InitializeComponent();
   }
}
...
```

Получить лицензионный ключ или скачать файл с лицензионным ключом, вы можете в личном кабинете пользователя. Для авторизации в кабинете используйте имя пользователя и пароль, указанные при покупке подписки продукта.

13.2 WPF вьювер

Для отображения отчетов в **WPF** используется компонент **StiWpfViewerControl**. Компонент умеет отображать отчет, масштабировать, сохранять построенный отчет в различные форматы, выводить его на печать, отправлять построенные отчеты по почте.

- Как отобразить отчет,
- Dot-Matrix режим.

© 2003-2026 Stimulsoft

13.2.1 Как отобразить отчет

Примечание

При присвоении отчета вьювер компоненту, происходит автоматическое построение отчета. Вызов метода `Report.Render()` необходим только в том случае, если требуется провести какие-либо действия с построенным отчетом перед его отображением во вьювере. Аналогично, при использовании режима компиляции, вызов метода `Report.Compile()` требуется только в том случае, если требуется провести какие-либо действия с откомпилированным отчетом перед его построением и отображением во вьювере.

Для того, чтобы отобразить отчет потребуется вызов всего одного метода:

C#

```
...
StiReport report = new StiReport();
report.Load("report.mrt");
report.ShowWithWpf();
...
```

VB.NET

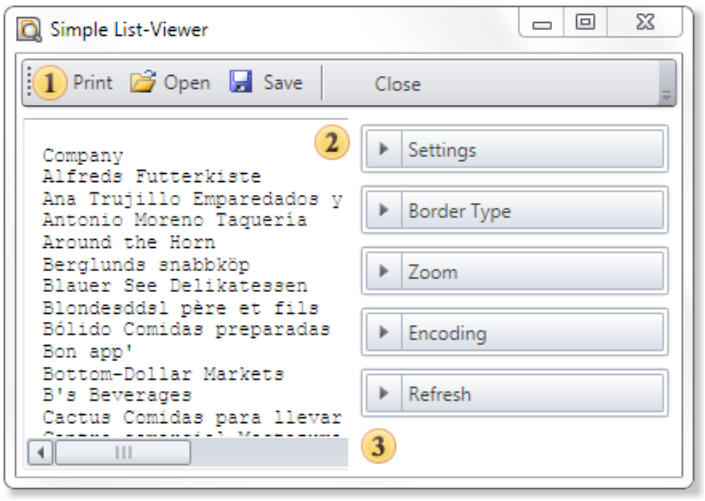
```
...
Dim Report As StiReport = New StiReport()
Report.Load("report.mrt")
Report.ShowWithWpf()
...
```

Если отчет не был построен до отображения метод **ShowWithWpf** также выполните построение отчета при помощи метода **RenderWithWpf**.

13.2.2 Dot-Matrix режим

Dot-matrix вьювер предназначен для предварительного просмотра отчета перед печатью на матричном принтере. Матричный принтер предназначен для печати только текста и символов псевдографики. Соответственно вьювер отображает только текст и границы объектов в виде символов псевдографики. На рисунке снизу представлено окно **Dot-matrix** вьювера:

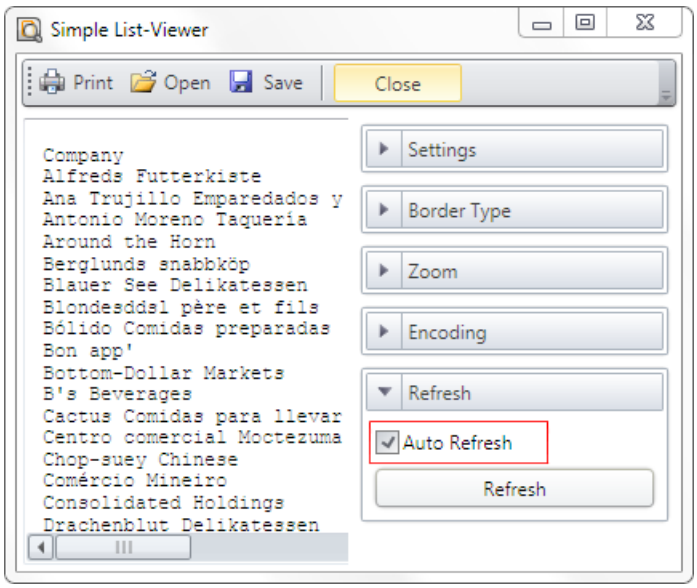
© 2003-2026 Stimulsoft



1. Панель инструментов (Toolbars) Dot-matrix вьювера.
2. Панель отображения текста отчета.
3. Панель параметров отображаемого отчета.

Dot-Matrix Viewer Settings for WPF

Dot-Matrix вьювер можно настроить из кода используя статические свойства. В зависимости от значения статических свойств в **Dot-matrix** вьювере будут определены те или иные параметры. Рассмотрим на примере свойства **Автообновление (Auto Refresh)**. На рисунке снизу представлено окно Dot-matrix вьювера:



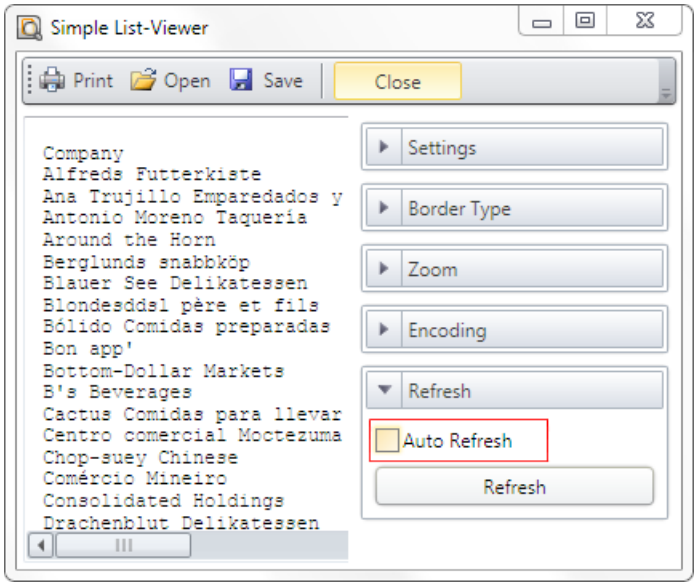
© 2003-2026 Stimulsoft

Как видно из рисунка, свойство **Автообновление (Auto Refresh)** включено, значит статическое свойство **Dot-matrix** вьювера **AutoRefresh** установлено в значение **true**. Если статическое свойство **AutoRefresh** установить в значение **false**, то свойство **Автообновление (Auto Refresh)** в **Dot-matrix** вьювере будет выключено. Добавим ниже приведенный код в код проекта:

C#

```
...
StiOptions.Viewer.DotMatrix.AutoRefresh = false;
...
```

Таким образом, свойство **Автообновление (Auto Refresh)** будет выключено. На рисунке снизу представлено окно **Dot-matrix** вьювера с выключенным автообновлением:



Большинство значений параметров можно установить с помощью статических свойств.

DotMatrix and Escape Codes

© 2003-2026 Stimulsoft

Полноценного управления матричным принтером в **Stimulsoft Reports.Wpf** нет, но начиная с версии **2009.xx** в **Dot-Matrix** вьювере можно использовать **Escape**-последовательности в тексте для изменения параметров: размер встроенных шрифтом матричного принтера, их начертание и др. Ниже приведен пример **Escape**-последовательности в тексте:

```
Normal text <#b> Bold text <#/b><#i> Italic text <#/i> Again normal text
```

В версии 2010.xx **Stimulsoft Reports.Wpf** появилась более расширенная поддержка **Escape**-кодов. Для ввода Escape-последовательности в текст надо использовать команды вида **<#command>**.

Также автоматически вставляются команды выбора жирного, наклонного или подчеркнутого начертания в зависимости от установленного стиля шрифта текстового компонента. При печати на матричный принтер и при экспорте в текстовый формат эти команды заменяются на соответствующие Escape-последовательности. Для задания соответствий используется класс **StiEscapeCodesCollection**. Он унаследован от класса **Hashtable** и представляет собой коллекцию пар ключ-значение, в которой ключом является команда, а значением - соответствующая Escape-последовательность. Для разных типов принтеров можно задавать разные коллекции с разным набором команд. Хранятся коллекции в статической переменной **StiOptions.Export.Txt.EscapeCodesCollectionList**. По умолчанию созданы следующие коллекции: "**None**", "**EpsonFX**", "**Oki ML92/93**". Коллекция "**None**" пустая и служит для вывода текста без escape-кодов.

Command/Collection	EpsonFX	Oki ML92/93
b	ESC E	ESC T
/b	ESC F	ESC I
i	ESC 4	
/i	ESC 5	
u	ESC -1	ESC H
/u	ESC -0	ESC D
sup	ESC S0	ESC J
/sup	ESC T	ESC K
sub	ESC S1	ESC L

© 2003-2026 Stimulsoft

Command/Collection	EpsonFX	Oki ML92/93
/sub	ESC T	ESC M
condensed	0x0F	0x1d
/condensed	0x12	0x1e
elite	ESC M	0x1c
pica	ESC P	0x1e
doublewidth	ESC W1	0x1f
/doublewidth	ESC W0	0x1e

Можно добавлять новые коллекции или изменять существующие. Выбор нужной коллекции осуществляется по имени. Если коллекция с нужным именем не найдена, используется коллекция "None". Имя коллекции можно выбрать в настройках Dot-Matrix Viewer или передать в качестве параметра в методы экспорта и печати.

14 Генератор отчетов

Информация

В отчете могут использоваться вложенные отчеты. По умолчанию, если у главного (вызывающего) отчета режим вычисления установлен как **Интерпретация (Interpretation)**, то все вложенные отчеты также будут обработаны в этом режиме. Если же у главного отчета режим вычисления установлен как **Компиляция (Compilation)**, то вложенные отчеты будут обработаны в соответствии с их собственным режимом вычисления.

Однако можно разрешить выполнение вложенных отчетов в режиме Компиляция, независимо от режима вычисления главного отчета. Для этого следует установить опцию `AllowOpenSubReportWithCompilation` в значение `true`. Например, `StiOptions.Engine.AllowOpenSubReportWithCompilation = true;`

В этой главе рассматриваются вопросы связанные с возможностями ядра

© 2003-2026 Stimulsoft

генератора отчетов, которые могут относится к различным продуктам и компонентам компании Stimulsoft.

- Данные
- Экспорт отчетов
- Наследование отчетов
- Скрипты
- Режим "Справа налево"

14.1 Данные

YouTube

Посмотрите видеоуроки по работе с данными. Подписывайтесь на канал Stimulsoft и узнайте первыми о новых видеоуроках. Вопросы и предложения оставляйте в комментариях к видео.

В этом разделе рассматриваются вопросы работы с различными данными для построения отчетов

- Бизнес-объекты в NET, Web
- Получение данных при помощи протокола OData
- Использование значения выражения в переменной

14.1.1 Бизнес-объекты в NET, Web

Бизнес-объекты - это тип данных, который является набором объектов связанных между собой, с помощью которых можно представлять данные в различных структурах: таблицы, списки, массивы и т.д. Эти данные могут быть переданы в генератор отчетов и на основе их можно построить отчет. Бизнес-объекты создаются, регистрируются и передаются в генератор отчетов из кода.

Заполнение бизнес-объектов вручную в Net

Рассмотрим пример создания отчета с Бизнес-объектом. Для начала

© 2003-2026 Stimulsoft

необходимо создать структуру бизнес-объекта. Ниже представлен пример кода создания класса бизнес-объекта:

C#

```
...
public class MyObject
{
   public class Category
   {
      public int number;
      public int Number
      {
         get
         {
            return number;
         }
      }

      public string name;
      public string Name
      {
         get
         {
            return name;
         }
      }

      public string description;
      public string Description
      {
         get
         {
            return description;
         }
      }
   }

   public Category[] list = null;
   public Category[] List
   {
      get
      {
         return list;
      }
   }
}
...
```

Теперь, следует заполнить данными бизнес-объект. Ниже представлен пример кода, с помощью которого бизнес-объект заполняется данными:

© 2003-2026 Stimulsoft

C#

```
...
MyObject obj = new MyObject();
obj.list = new MyObject.Category[2];

MyObject.Category c1 = new MyObject.Category();
c1.number = 1;
c1.name = "Cat1";
c1.description = "desc for n1";

MyObject.Category c2 = new MyObject.Category();
c2.number = 2;
c2.name = "Cat2";
c2.description = "desc for n2";

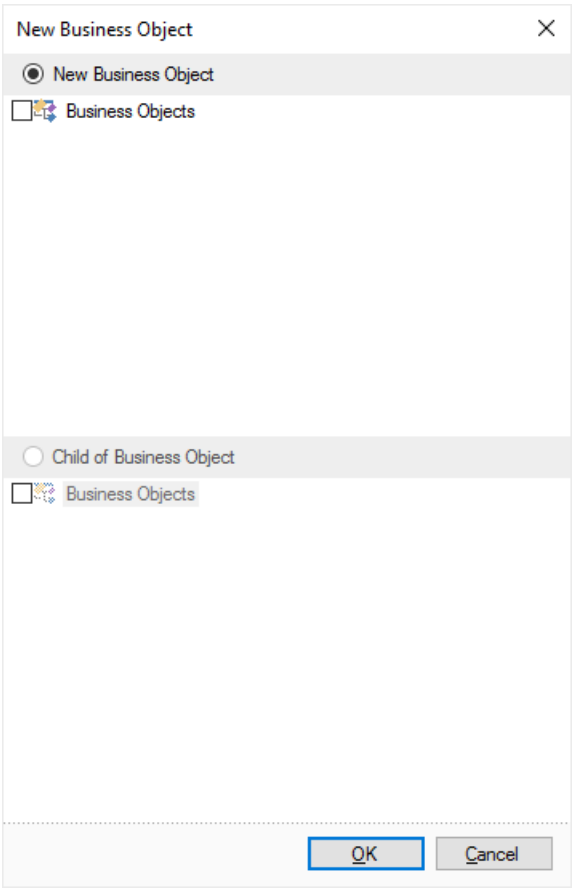
obj.list[0] = c1;
obj.list[1] = c2;

StiReport mainreport = new StiReport();
mainreport.RegBusinessObject("MyObject", obj);
mainreport.Design();
...
```

Использование бизнес-объектов в отчете

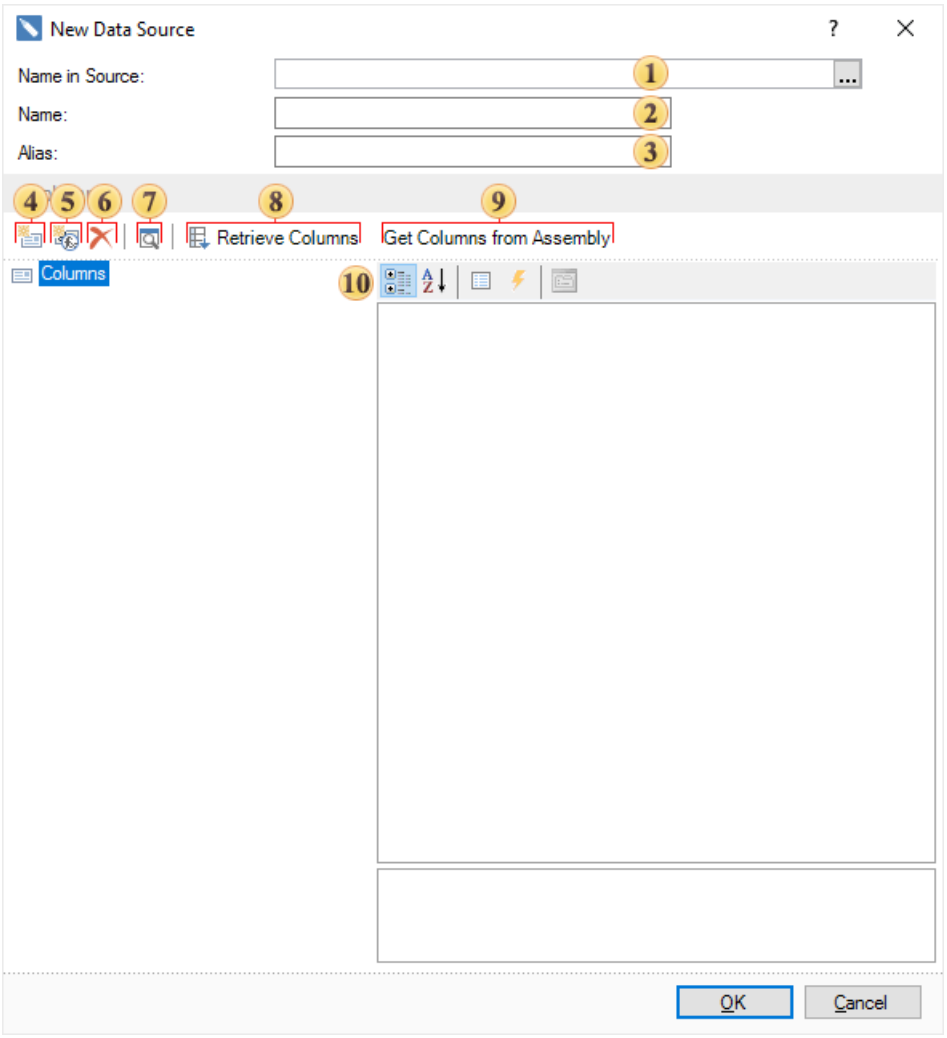
После этого бизнес-объект создан, заполнен данными, зарегистрирован и передан в генератор отчетов. Для того, чтобы построить отчет в дизайнере с использованием бизнес-объектов необходимо создать описание данных в словаре отчета. Для этого, следует в словаре отчета выделить созданный бизнес-объект **MyObject**, и выбрать пункт **Новый бизнес-объект... (New Business Object...)** в контекстном меню или меню **Новый элемент (New Item)**. После выбора данной команды, будет открыто окно **Новый бизнес-объект (New Business Object)**, в котором следует указать пункт **Подчиненный бизнес-объект (Child Business Object)** и выбрать списки данных. На рисунке снизу представлено окно **Новый бизнес-объект (New Business Object)**:

© 2003-2026 Stimulsoft



После нажатия кнопки **Ок**, пользователю будет отображено 2-ая диалоговая форма окна **Новый бизнес объект (New Business Object)**, в котором можно изменить параметры подчиненного бизнес объекта. На рисунке снизу представлена 2-ая диалоговая форма окна **Новый бизнес объект (New Business Object)**:

© 2003-2026 Stimulsoft



1 В поле **Категория (Category)** отображается имя категории. При создании бизнес объектов данное поле не доступно для редактирования и носит сугубо информативный характер. Также оно может быть пустым, как в данном случае.

2 Поле **Наименование (Name)** предназначено для указания имени бизнес-объекта. Это поле доступно для редактирования всегда и, в данном случае, используется имя **List**.

3 В поле **Псевдоним (Alias)** указывается псевдоним бизнес-объекта. Это поле доступно для редактирования всегда и, в данном случае, используется имя **List**.

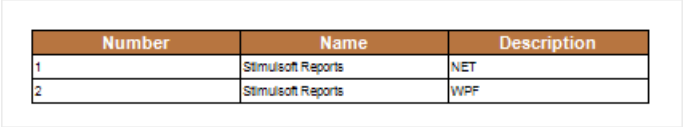
4 Кнопка **Новая колонка (New Column)**. При нажатии на которую в бизнес объекте будет создана новая колонка данных. Следует отметить, что созданная таким образом колонка данных, является виртуальной колонкой данных и реальных данных, она не содержит.

5 При нажатии кнопки **Новая рассчитываемая колонка (New Calculated Column)** в бизнес-объект будет вставлена новая рассчитываемая колонка.

© 2003-2026 Stimulsoft

6 При нажатии на кнопку **Удалить (Delete)** будет удалена выделенная колонка данных. Если выделена закладка **Колонки (Columns)**, то будут удалены все колонки, которые располагаются в данной вкладке.
7 Кнопка просмотра запроса.
8 С помощью кнопки **Получить колонки (Retrieve Columns)** можно получить колонки данных из бизнес объекта.
9 При нажатии кнопки **Взять колонки из сборки (Get Columns from Assembly)** будет открыто окно **Открыть сборку (Open Assembly)**, в котором следует выбрать файл сборки. После выбора файла сборки следует нажать кнопку **Открыть (Open)** и из этого файла будут извлечены колонки данных, если они там присутствуют.
10 Панель **Колонки (Columns)** представлена тремя полями. В этих полях отображается список колонок, их свойства и описание этих свойств.

После того как поля заполнены и параметры указаны следует нажать кнопку **OK**. После этого, в словаре данных отчета будет создано описание нового бизнес объекта, который можно использовать для построения отчетов. На рисунке снизу представлен отчет построенный с помощью бизнес-объекта:



Number	Name	Description
1	Stimulsoft Reports	NET
2	Stimulsoft Reports	WPF

Получение данных для бизнес-объектов из источника данных в Net

Созданные бизнес-объекты, которые зарегистрированы и переданы в генератор отчетов, но не содержащие реальные данные называются описанием бизнес-объектов. С помощью описания бизнес-объектов можно создать шаблон отчета (обозначить структуру и оформление отчета), а затем перед построением подключить источник с реальными данными и построить отчет. Это удобно, если необходимо создавать отчеты с одинаковой структурой и оформлением, но с различными данными. Для начала создадим структурное описание бизнес-объекта. Ниже представлен пример кода создания класса бизнес-объекта:

© 2003-2026 Stimulsoft

C#

```
...
public class MyObject
{
   public class Category
   {
      public int categoryID;
      public int CategoryID
      {
         get
         {
            return categoryID;
         }
      }

      public string categoryName;
      public string CategoryName
      {
         get
         {
            return categoryName;
         }
      }

      public string description;
      public string Description
      {
         get
         {
            return description;
         }
      }

   }

   public Category[] list = null;
   public Category[] Categories
   {
      get
      {
         return list;
      }
   }
}
...
```

Затем необходимо создать новый объект класса бизнес-объект, зарегистрировать и передать его в генератор отчетов. Ниже представлен пример кода создания и регистрации нового бизнес-объекта:

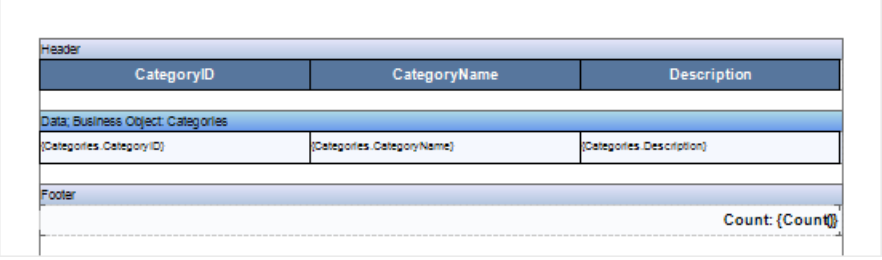
© 2003-2026 Stimulsoft

C#

```
...
MyObject.Category obj = new MyObject.Category();
int busobjLevel = 1;

StiReport report = new StiReport();
report.RegBusinessObject("Categories", obj);
report.Dictionary.SynchronizeBusinessObjects(busobjLevel);
report.Design();
...
```

Теперь при помощи созданного описания бизнес-объекта, создадим шаблон отчета в дизайнере. На рисунке снизу представлен созданный шаблон отчета с описанием бизнес-объекта:



После того, как шаблон отчета создан, следует сохранить его, к примеру, по следующему пути **D:\\Report.mrt**. Поскольку, описание бизнес-объекта не содержит реальных данных, то для того, чтобы построить отчет, необходимо будет получить реальные данные для бизнес-объектов. В этом примере, данные будут получены из базы данных **Northwind**. Для начала, необходимо в Visual Studio создать соединение к этой базе данных. После этого, следует указать код получения данных для бизнес-объекта. Получение реальных данных для бизнес-объекта происходит непосредственно перед построением отчета. Ниже представлен код получения данных для бизнес-объекта:

© 2003-2026 Stimulsoft

C#

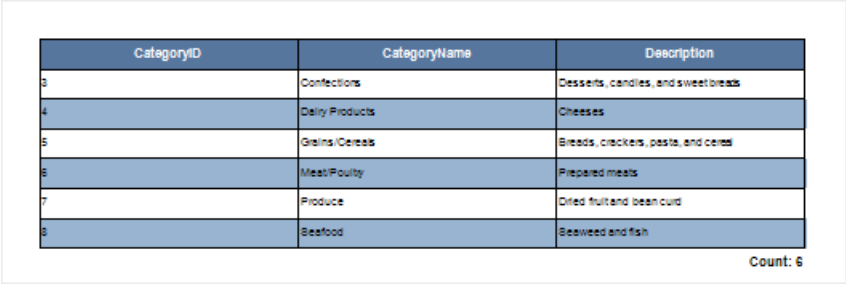
```
...
int busobjLevel = 1;

StiReport report = new StiReport();
report.Load("D:\\Report.mrt");

using (NorthwindDataContext context = new NorthwindDataContext())
{
   var categories =
      from c in context.Categories
      select new { c.CategoryID, c.CategoryName, c.Description };

   report.RegBusinessObject("Categories", categories);
   report.Show();
}
...
```

После этого, генератор отчетов получит данные для бизнес-объекта из указанного источника, в данном случае из базы данных **Northwind**. Далее, отчет будет построен по созданному шаблону. На рисунке снизу представлен построенный отчет:



CategoryID	CategoryName	Description
3	Confections	Desserts, candies, and sweet breads
4	Dairy Products	Cheeses
5	Grains/Cereals	Breads, crackers, pasta, and cereal
6	Meat/Poultry	Prepared meats
7	Produce	Dried fruit and bean curd
8	Seafood	Seaweed and fish

Count: 6

Бизнес-объекты в Web

Создание, заполнение, регистрация и передача бизнес-объектов в **Web** осуществляется практически также как и в **Net**. Для начала создадим класс бизнес-объекта, идентичный как в **Net**. Далее, следует создать объект класса бизнес-объекта, зарегистрировать его, заполнить вручную данными и передать их. Здесь отличия будут в том, что вместо метода **mainreport.Design()** следует использовать метод **StiWebDesigner1.Design(mainreport)**. А также обязательно произвести синхронизацию с помощью метода **mainreport.Dictionary.SynchronizeBusinessObjects()**, потому как в дизайнере **Web** отсутствует возможность создавать описание бизнес-объектов из словаря данных (описание может создаваться только из кода). Ниже представлен

© 2003-2026 Stimulsoft

пример кода создания, заполнения, регистрации и передачи бизнес-объекта:

C#

```
...
MyObject obj = new MyObject();
obj.list = new MyObject.Category[2];

MyObject.Category c1 = new MyObject.Category();
c1.number = 1;
c1.name = "Cat1";
c1.description = "desc for n1";

MyObject.Category c2 = new MyObject.Category();
c2.number = 2;
c2.name = "Cat2";
c2.description = "desc for n2";

obj.list[0] = c1;
obj.list[1] = c2;

int busobjLevel = 1;

StiReport mainreport = new StiReport();
mainreport.RegBusinessObject("MyObject", obj);
mainreport.Dictionary.SynchronizeBusinessObjects(busobjLevel);
StiWebDesigner1.Design(mainreport);
...
```

Также как и в **NET**, в **Web** можно сначала создать описание бизнес-объектов, затем шаблон отчета, а потом перед построением подключить источник с реальными данными и построить отчет. Для начала, создадим описание бизнес-объекта. Предварительно создав класс бизнес-объекта, идентичный классу бизнес-объекта в **NET**. Ниже представлен пример создания описания бизнес-объекта:

C#

```
...
MyObject.Category obj = new MyObject.Category();
int busobjLevel = 1;

StiReport report = new StiReport();
report.RegBusinessObject("Categories", obj);
report.Dictionary.SynchronizeBusinessObjects(busobjLevel);
StiWebDesigner1.Design(report);
...
```

© 2003-2026 Stimulsoft

Теперь при помощи созданного описания, создадим шаблон отчета, идентичный шаблону в **NET**. После того, как шаблон отчета создан, следует сохранить его, к примеру, по следующему пути **D:\\Report.mrt**. Поскольку, описание бизнес-объекта не содержит реальных данных, то для того, чтобы построить отчет, необходимо будет получить реальные данные для бизнес-объектов. В этом примере, данные будут получены из базы данных **Northwind**. Для начала, необходимо в **Visual Studio** создать соединение к этой базе данных. После этого, следует указать код получения данных для бизнес-объекта. Получение реальных данных для бизнес-объекта происходит непосредственно перед построением отчета. Ниже представлен код получения данных для бизнес-объекта:

C#

```
...
int busobjLevel = 1;

StiReport report = new StiReport();
report.Load("D:\\Report.mrt");

using (NorthwindDataContext context = new NorthwindDataContext())
{
   var categories =
      from c in context.Categories
      select new { c.CategoryID, c.CategoryName, c.Description };

   report.RegBusinessObject("Categories", categories);
   StiWebViewer1.Report = report;
}
...
```

14.1.2 Получение данных при помощи протокола OData

Протокол **Open Data Protocol (OData)** предназначен для получения доступа к данным из различных источников, включая, реляционные базы данных, файловые системы, системы управления контентом, а также традиционных веб-сайтов. Протокол **OData** реализует концепцию **CRUD** (Create - создать, Read - получить, Update - Обновить, Delete - удалить) в отношении данных. В **Visual Studio 2010** и **.NET Framework 4.0** поддержка протокола **OData** значительно упрощена за счет работы с технологией доступа **Entity Framework**. На основе полученных (при помощи протокола OData) данных можно создавать отчеты. Передача данных в отчет осуществляется через бизнес-объекты. Рассмотрим пример получения данных с web-ресурса и их передачу в отчет:

- Подключить сборки **Stimulsoft** в проекте;

© 2003-2026 Stimulsoft

› Добавить **Service Reference** с указанием адреса точки входа на **OData**-сервис. В данном случае, адрес http://services.odata.org/V3/OData/OData.svc;
› Использовать нижеприведенный код:

C#

```
...
//Подключение к Хранилищу Данных
Uri uri = new Uri("http://services.odata.org/V3/OData/OData.svc");
var container = new ServiceReference1.DemoService(uri);

//Создание запроса выборки с параметрами
var product = container.Products.Where(p => p.ID < 50).ToList();

//Передача данных в отчет при помощи Бизнес-Объекта
var report = var StiReport();
report.RegBusinessObject("Products", product);
report.Dictionary.SynchronizeBusinessObjects(2);
report.Design();
...
```

14.1.3 Использование значения выражения в переменной

Текстовые компоненты, кроме текста, могут содержать выражения. В момент построения компонента, выражение обрабатывается генератором отчетов, т.е. производиться его вычисление. За исключением, итоговых сумм по бэнду или странице. В этом случае, результат высчитывается после полного построения бэнда или отчета. Порядок обработки компонентов ядром генератора отчетов определяется иерархией компонентов отчета. Иначе говоря, чем выше компонент в этой иерархии, тем выше его приоритет обработки при построении отчета. Полученное значение (результат вычисления выражения) передается в обработанный (построенный) компонент.

Информация

Если текстовый компонент с выражением располагается на бэнде, то в построенном отчете, будет столько же экземпляров компонента, сколько строк в источнике данных бэнда.

Иногда возникает необходимость использовать результат выражения (рассчитанное значение) в переменной. Рассмотрим пример. Допустим есть выражение **{x+y}**, которое расположено в текстовом компонент **Text10**. Для того чтобы использовать результат этого выражения в переменной

© 2003-2026 Stimulsoft

недостаточно указать ссылку **{Text10.Text}**. Потому как обращение, в данном случае, будет не к результату выражения, а к текстовому выражению в шаблоне отчета. Для использования результата выражения в переменной, следует воспользоваться одним из следующих вариантов:

- Если необходимо получить значение компонента до его отрисовки, то следует: использовать такое же выражение в переменной, как в компоненте, или использовать два прохода, т.е на первом рассчитать значение, на втором использовать в переменной.
- Если значение будет использоваться после отрисовки компонента, можно в событии **GetValue** компонента получить необходимое значение и сохранить (передать) его в переменную.
- Также можно после построения всего отчета, в событии **EndRender**, пробежаться по всем компонентам построенного отчета и сделать необходимые вычисления. Ниже представлен пример скрипта, который подсчитывает сумму по странице, в случае, если у бэнда установлено свойство **Can Break** и заранее неизвестно, на какой странице окажется текстовый компонент.

C#

```
...
foreach (StiPage page in RenderedPages)
            {
                StiText sumComp = null;
                foreach (StiComponent component in page.Components)
                {
                    if (component.Name == "sum_comp")
                    {
                        sumComp = component as StiText;
                        break;
                    }
                }

                double sum = 0;
                foreach (StiComponent comp in page.Components)
                {
                    if (comp.Name == "aaa")
                    {
                        sum += (double)(comp as StiText).TagValue;
                    }
                }
                sumComp.Text = sum.ToString();
            }
...
```

© 2003-2026 Stimulsoft

14.2 Наследование отчетов

Есть следующие пути наследования отчета:

- Создание базового класса отчета;
- Создание **Master** отчета.

При любом пути наследования отчета, необходимо создать в дизайнере базовый отчет, который включает все необходимые элементы. Добавить к базовому отчету можно следующие компоненты:

- Страницы;
- Источники данных;
- Переменные;
- Подключения;
- Компоненты.

Узнать больше об основных подходах

14.2.1 Основные подходы

После того, как базовый отчет был создан, можно либо сохранить отчет, как специальный базовый класс для наследования (для этого следует использовать команду Сохранить как), или сохранить базовый отчет, как обычный отчет, и использовать его как Master отчет в последующем. В первом случае будет получен **C#** или **VB.NET** класс, и можно будет создавать новые отчеты. Например:

C#

```
...
Reports.Report master = new Reports.Report();
master.RegData(dataSet);
master.Design();
...
```

Если необходимо использовать базовый отчет для создания новых отчетов в дизайнере, то следует добавить нижеприведенный код:

C#

```
...
StiReport.ReportType = typeof(Reports.Report);
...
```

© 2003-2026 Stimulsoft

Тогда все новые отчеты будут автоматически наследованы от базового класса. При втором способе наследования необходимо использовать следующий код:

C#

```
...
StiReport masterReport = new StiReport();
masterReport.Load("d:\\master-detail.mrt");

StiReport report = new StiReport();
report.RegData(dataSet);

report.MasterReport = masterReport.SaveToString();
report.Design();
...
```

14.3 Режим "справа налево"

По умолчанию, отображение компонентов в отчете происходит в режиме "слева направо". Свойство **Справа налево (Right to Left)** предоставляет возможность изменить режим отображения элементов отчета. Рассмотрим применение данного свойства к отдельным элементам отчета более подробно.

- WinForms вьювер отчетов
- WPF вьювер отчетов
- Значки

14.3.1 Вьювер отчетов для WinForms

Во вьювере отчетов для **WinForms** есть возможность изменить режим отображения элементов вьювера и порядок отображения страниц отчета. По умолчанию отображение всех элементов вьювера и порядок отображения страниц отчета происходит в режиме "слева направо", но при необходимости можно отобразить в режиме "справа налево". Режим отображения элементов вьювера и порядок отображения страниц отчета зависит от значения статического свойства **RightToLeft** вьювера. Если свойство **RightToLeft** установлено в значение **No**, то отображение элементов вьювера и порядок отображения страниц отчета происходит в режиме "слева направо". Ниже представлен код для отображения в режима "слева направо":

C#

```
...
StiOptions.Viewer.RightToLeft = StiRightToLeftType.No;
...
```

© 2003-2026 Stimulsoft

Если свойство **RightToLeft** установлено в значение **Yes**, то отображение элементов вьювера и порядок отображения страниц отчета происходит в режиме "справа налево". Ниже представлен код для режима "справа налево":

C#

```
...
StiOptions.Viewer.RightToLeft = StiRightToLeftType.No;
...
```

14.3.2 Вьювер и дизайнер отчетов WPF

В дизайнере и вьювере отчетов **WPF** версии есть возможность установить режим отображения элементов управления "справа налево". По умолчанию отображение всех элементов управления и порядок отображения страниц отчета происходит в режиме "слева направо", но при необходимости можно изменить его. Режим отображения элементов управления и порядок отображения страниц в построенном отчете зависит от значения свойства **FlowDirection**. Если свойство **FlowDirection** установлено в значение **LeftToRight**, то отображение происходит в режиме "слева направо". Ниже представлен код для отображения в режиме "слева направо":

XAML

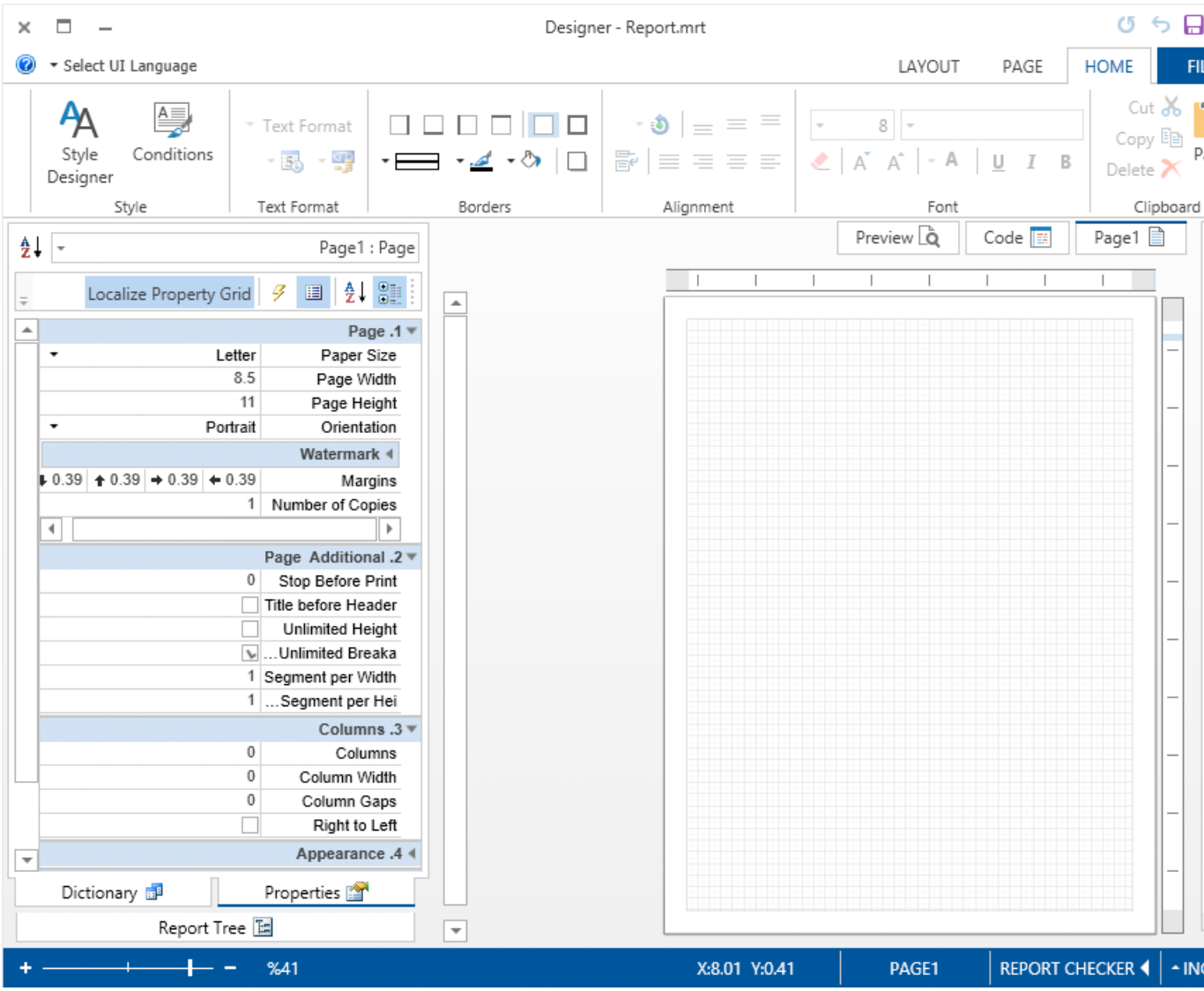
```
...
FlowDirection="LeftToRight"
...
```

Если свойство **FlowDirection** установлено в значение **RightToLeft**, то отображение элементов управления и порядок отображения страниц в построенном отчете происходит в режиме "справа налево". Ниже представлен код для режима "справа налево" и приведен пример окна дизайнера с отображением элементов в данном режиме:

XAML

```
...
FlowDirection="RightToLeft"
...
```

© 2003-2026 Stimulsoft

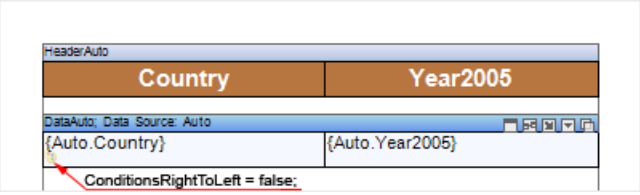


14.3.3 Значки

С помощью значков в отчетах отображается информация о компонентах, параметрах и инструментах, которые применяются к данному компоненту. К этим значкам относятся значки: фильтрации, условия, события, унаследованного отчета, динамического сворачивания, динамической сортировки, порядка, быстрой информации. По умолчанию, эти значки отображаются в режиме "слева направо", но в случае необходимости, их можно отобразить в режиме "справа налево". Изменить режим отображения значков можно с помощью свойств: **ConditionsRightToLeft**, **EventsRightToLeft**, **InheritedRightToLeft**, **InteractionCollapsingRightToLeft**, **InteractionSortRightToLeft**, **OrderAndQuickInfoRightToLeft**, **FiltersRightToLeft**, **QuickButtonsRightToLeft**, которые относятся к классу **StiOptions.Viewer.Pins**. Рассмотрим эти свойства более подробно:

© 2003-2026 Stimulsoft

› Режим отображения значка **Conditions** зависит от значения свойства **ConditionsRightToLeft**. К примеру, если на бэнде **Данные (DataBand)** разместить **Условие (Condition)**, то значок **Conditions** будет отображаться, по умолчанию, в нижнем левом углу бэнда **Данные (DataBand)**, т.к. свойство **ConditionsRightToLeft** установлено в значение **false**, а значит используется режим "слева направо". На рисунке снизу представлен пример шаблона отчета с отображением значка **Conditions** в режиме "слева направо":

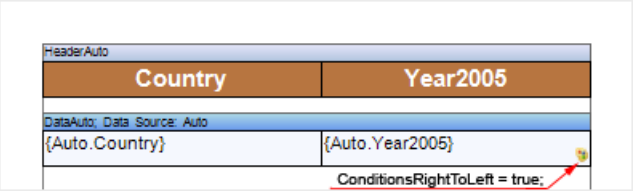


Если необходимо изменить режим отображения значка Conditions, следует установить свойство ConditionsRightToLeft в значение true:

C#

```
...
StiOptions.Viewer.Pins.ConditionsRightToLeft = true;
...
```

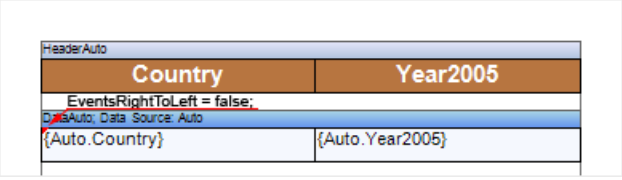
И тогда значок Conditions будет отображаться в режиме "справа налево", т.е. в нижнем правом углу бэнда Данные (DataBand). На рисунке снизу представлен пример шаблона отчета с отображением значок Conditions в режиме "справа налево":



› Для изменения режима отображения значка Events следует изменить значение свойства EventsRightToLeft. К примеру, если на текстовом компоненте разместить Событие (Event), то закладка Events будет отображаться, по умолчанию, в верхнем левом углу текстового компонента, т.к. свойство

© 2003-2026 Stimulsoft

EventsRightToLeft установлено в значение false, а значит используется режим "слева направо". На рисунке снизу представлен пример шаблона отчета с отображением значка Events в режиме "слева направо":

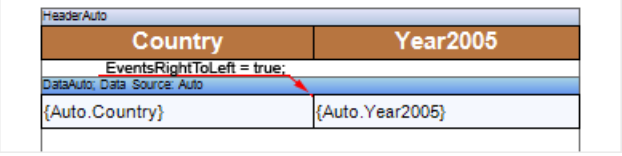


Если необходимо изменить режим отображения значка Events, следует установить свойство EventsRightToLeft в значение true:

C#

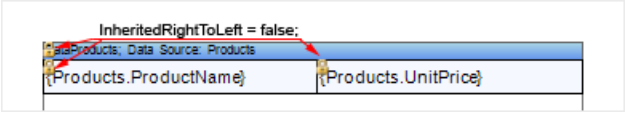
```
...
StiOptions.Viewer.Pins.EventsRightToLeft = true;
...
```

И тогда значок Events будет отображаться в режиме "справа налево", т.е. в верхнем правом углу текстового компонента. На рисунке снизу представлен пример шаблона отчета с отображением значка Events в режиме "справа налево":



› В унаследованных отчетах значок Inherited можно отобразить в режиме "справа налево". Для этого, следует изменить значение свойства InheritedRightToLeft. По умолчанию данное свойство установлено в значение false, т.е. значок Inherited отображается в режиме "слева направо". На рисунке снизу представлена страница шаблона отчета с отображением значка Inherited в режиме "слева направо":

© 2003-2026 Stimulsoft

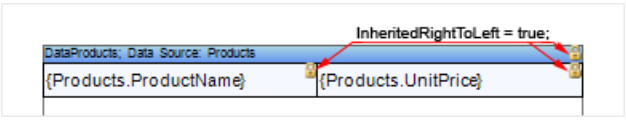


Если свойство InheritedRightToLeft установить в значение true:

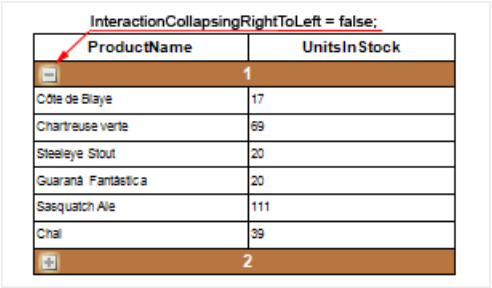
C#

```
...
StiOptions.Viewer.Pins.InheritedRightToLeft = true;
...
```

то значок Inherited будет отображаться в режиме "справа налево". На рисунке снизу представлена страница шаблона отчета с отображением значка Inherited в режиме "справа налево":



› При использовании динамического сворачивания в построенном отчете, можно отобразить значки в режиме "справа налево". Режим отображения значков зависит от значения свойства InteractionCollapsingRightToLeft. По умолчанию, данное свойство установлено в значение false и значки располагаются в режиме "слева направо". На рисунке снизу представлена страница отчета с динамическим сворачиванием, где значки отображаются в режиме "слева направо":



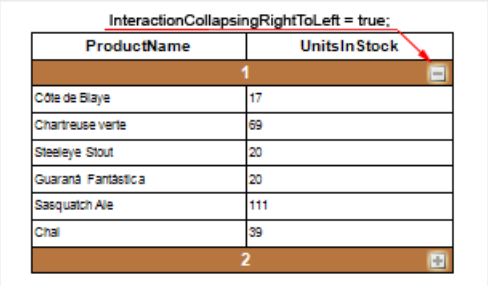
Для того, чтобы изменить режим отображения значков, следует установить свойство InteractionCollapsingRightToLeft в значение true:

© 2003-2026 Stimulsoft

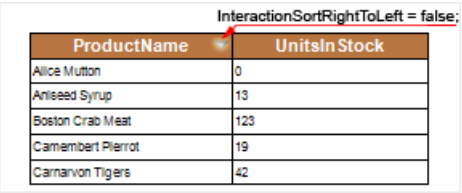
C#

```
...
StiOptions.Viewer.Pins.InteractionCollapsingRightToLeft = true;
...
```

и тогда значки при использовании динамического сворачивания будут располагаться в режиме "справа налево". На рисунке снизу представлен пример отчета с динамическим сворачиванием и значками в режиме "справа налево":



› В отчете с динамической сортировкой можно изменить режим отображения значка InteractionSort. Для этого, следует изменить значение свойства InteractionSortRightToLeft. По умолчанию, данное свойство установлено в значение false, т.е. используется режим "справа налево". К примеру, если динамическая сортировка установлена на текстовом компоненте, то в построенном отчете значок динамической сортировки отображается у правой границы этого компонента. На рисунке снизу представлен отчет, в котором значок InteractionSort отображается в режиме "слева направо":



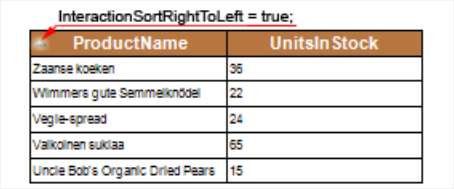
Для того, чтобы режим отображения значка, следует установить свойство InteractionSortRightToLeft в значение true:

© 2003-2026 Stimulsoft

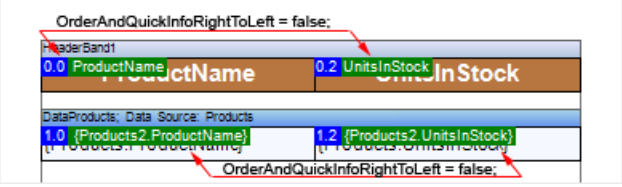
C#

```
...
StiOptions.Viewer.Pins.InteractionSortRightToLeft = true;
...
```

и тогда значок, при использовании динамической сортировки, будет располагаться в режиме "справа налево", у левой границы компонента. На рисунке снизу представлен пример отчета с динамической сортировкой и значком в режиме "справа налево":



› При создании шаблона отчета, можно включать/выключать значки с информацией о компонентах. Включение/выключение значков с информацией осуществляется в меню Быстрая информация (Quick Info) и с помощью кнопки Показывать порядок (Show Order). По умолчанию, значки с этой информацией отображается в режиме "слева направо". Для того, чтобы изменить режим отображения значков, необходимо изменить значение свойства OrderAndQuickInfoRightToLeft. На рисунке снизу представлена страница шаблона отчета с отображением некоторых значков с информацией в режиме "слева направо" (свойство OrderAndQuickInfoRightToLeft установлено в значение false):



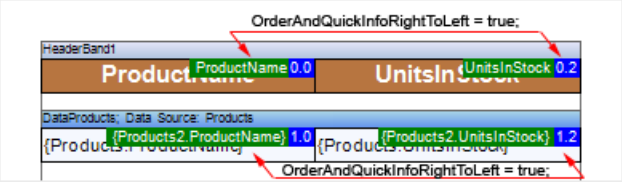
Если свойство OrderAndQuickInfoRightToLeft установить в значение true:

© 2003-2026 Stimulsoft

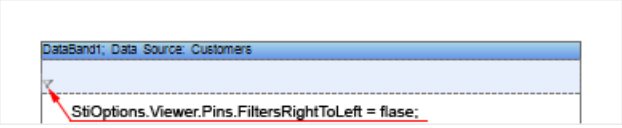
C#

```
...
StiOptions.Viewer.Pins.InteractionSortRightToLeft = true;
...
```

то значки с информацией о компонентах будут отображаться в режиме "справа налево". На рисунке снизу представлена страница шаблона отчета, в которой значки с информацией о компонентах располагается в режиме "справа налево":



› Режим отображения значка Filters зависит от значения свойства FiltersRightToLeft. К примеру, если на бэнде Данные (DataBand) установить фильтр, то значок Filters будет отображаться, по умолчанию, в нижнем левом углу бэнда Данные (DataBand), т.к. свойство FiltersRightToLeft установлено в значение false, а значит используется режим отображения "слева направо". На рисунке снизу представлен пример шаблона отчета с отображением значка Filters в режиме "слева направо":



Если необходимо изменить режим отображения значка Filters, следует установить свойство FiltersRightToLeft в значение true:

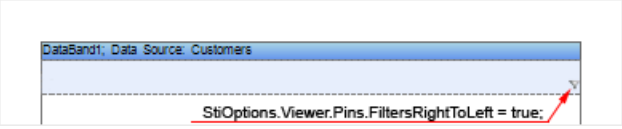
C#

```
...
StiOptions.Viewer.Pins.FiltersRightToLeft = true;
...
```

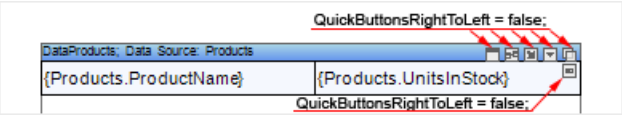
И тогда значок Filters будет отображаться в режиме "справа налево", т.е. в нижнем правом углу бэнда Данные (DataBand). На рисунке снизу представлен

© 2003-2026 Stimulsoft

пример шаблона отчета с отображением значок Filters в режиме "справа налево":



› Помимо значков, можно изменить режим отображения быстрых кнопок, которые представляют изменить те или иные параметры компонента в один шаг. Быстрые кнопки (QuickButtons) располагаются на компонентах в режиме "слева направо", как правило, в верхнем правом углу компонента. Для того, чтобы изменить режим расположения этих, следует изменить значение свойства QuickButtonsRightToLeft. По умолчанию, данное свойство установлено в значение false. На рисунке снизу представлена страница шаблона отчета с отображением быстрых кнопок в режиме "слева направо":

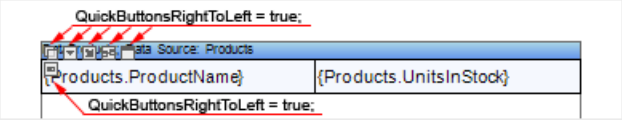


Если свойство **QuickButtonsRightToLeft** установить в значение **true**:

C#

```
...
StiOptions.Viewer.Pins.QuickButtonsRightToLeft = true;
...
```

то быстрые кнопки будут отображаться в режиме "справа налево". На рисунке снизу представлена страница шаблона отчета в которой быстрые кнопки располагаются в режиме "справа налево":



© 2003-2026 Stimulsoft

14.4 Экспорт отчетов

YouTube

Посмотрите видеоуроки по экспорту отчетов. Подписывайтесь на канал Stimulsoft и узнайте первыми о новых видеоуроках. Вопросы и предложения оставляйте в комментариях к видео.

В этом разделе описаны возможности **Stimulsoft Reports** по сохранению построенных отчетов в другие форматы. Особенности работы методов для экспорта отчетов. Рекомендации по оптимизации отчетов для последующего экспорта. Структуры данных, которые используются в методах экспорта. **Stimulsoft Reports** поддерживает большое количество форматов файлов для сохранения построенных отчетов (на момент написания этих строк около 30 форматов). Многие наши клиенты даже считают, слишком большое количество форматов. Но поверьте, в тот момент когда понадобится быстро получить файл определенного типа формата, написать при этом только одну строку кода, и при этом этим форматом будет не **PDF**, **HTML** или **RTF**, поможет только **Stimulsoft Reports**. Мы не считаем, что большое количество поддерживаемых форматов экспорта является минусом продукта и постоянно работаем над добавлением новых форматов. Как говорится много не мало.

- Экспорт отчетов из кода
- Форматы с фиксированной разметкой страницы
- Web документы
- Текстовые форматы
- Электронные таблицы
- Данные
- Изображения

Таблица форматов файлов для экспорта отчета

Перечисление **StiExportFormat** описывает немало форматов экспортов. Ниже приводится краткая информация по каждому формату.

© 2003-2026 Stimulsoft

Формат	Описание
Форматы, позволяющие точно передать вид отчета	
PDF	экспорт в формат Adobe PDF.
XPS	экспорт в формат Microsoft XPS.
Microsoft Power Point 2007/2010	экспорт в формат Microsoft Power Point 2007/2010
Форматы Web документов	
HTML	экспорт в формат HTML по умолчанию. Этот элемент дублирует HTMLTable режим.
HTMLTable	экспорт в формат HTML с использованием элемента HTML Table, для создания структуры отчета.
HTMLSpan	экспорт в формат HTML с использованием элемента HTML Span, для создания структуры отчета.
HTMLDiv	экспорт в формат HTML с использованием элемента HTML Div, для создания структуры отчета.
HTML5	экспорт в формат HTML5
MHT	экспорт в формат WebArchive. Этот формат поддерживается только Microsoft IE.
Текстовые форматы	
TXT	экспорт в текстовый формат.
RTF	экспорт в Rich Text Format по умолчанию. Этот элемент дублирует HtmlTable режим.
RTFTable	экспорт в формат Rich Text Format с использованием элемента Rtf Table, для создания структуры отчета.
RTFFrame	экспорт в формат Rich Text Format с использованием элемента Rtf Frame, для создания структуры отчета.
RTFWinWord	экспорт в формат Rich Text Format с

© 2003-2026 Stimulsoft

	использованием элемента Microsoft Word graphic, для создания структуры отчета.
RTFTabbedText	экспорт в формат Rich Text Format с использованием символов табуляции для форматирования структуры отчета.
Word	экспорт в формат Microsoft Word 2007. Этот формат поддерживается начиная с Microsoft Office 2007.
ODT	экспорт в формат OpenDocument Writer файл.
Электронные таблицы	
Excel	экспорт в формат Microsoft Excel. Файл создается с использованием формата BIFF (Binary Interchange File Format).
ExcelXml	экспорт в формат Microsoft Excel Xml. Файл создается с использованием формата Xml. Этот формат поддерживается начиная с Microsoft Office 2003.
Excel2007	экспорт в формат Microsoft Excel 2007. Этот формат поддерживается начиная с Microsoft Office 2007.
ODS	экспорт в формат OpenDocument Calc.
Экспорт в виде данных	
CSV	экспорт в формат CSV (Comma Separated Value).
DBF	экспорт в формат dBase/FoxPro.
XML	экспорт отчета в формате Xml в виде данных. Этот формат представляет из себя сохраненный DataSet.
DIF	экспорт в формат DIF (Data Interchange Format).
SYLK	экспорт в формат SYLK (Symbolic Link).
Экспорт в виде изображения	
GIF	экспорт в формат GIF.

© 2003-2026 Stimulsoft

BMP	экспорт в формат BMP.
PCX	экспорт в формат PCX.
PNG	экспорт в формат PNG.
TIFF	экспорт в формат TIFF.
JPEG	экспорт в формат JPEG.
EMF	экспорт в формат Windows Metafile.

14.4.1 Экспорт отчетов из кода

Stimulsoft Reports предоставляет большое количество всевозможных вариантов экспорта, построенного отчета в другие форматы. Кроме этого, каждый метод экспорта в другой формат, содержит большое количество разнообразных настроек. Для экспорта построенных отчетов **Stimulsoft Reports** использует систему сервисов. Это означает, что все объекты, ответственные за экспорт отчетов, представлены в коллекции сервисов и когда необходимо выполнить экспорт в какой либо формат построенного отчета, ядро генератора отчетов ищет соответствующий сервис в коллекции сервисов. Существует два варианта экспорта построенного отчета в другой формат из кода: с использованием метода **ExportDocument** класса **StiReport** и при помощи прямого создания или получения из коллекции сервисов нужного сервиса экспорта.

- Метод ExportDocument
- Сервисы экспорта отчетов

14.4.1.1 Использование метода ExportDocument

Метод **ExportDocument** предоставляет упрощенную обертку для экспортов отчетов. Вам не нужно получать нужный сервис экспорта. Все что нужно это указать тип экспорта, передать параметры экспорта и указать место куда положить результат. Например:

C#

```
...
StiPdfExportSettings pdfSettings = new StiPdfExportSettings();
report.ExportDocument(StiExportFormat.Pdf, "MyReport.Pdf", pdfSettings);
...
```

В данном коде мы экспортируем отчет в формат **PDF**. Созданный **PDF** файл

© 2003-2026 Stimulsoft

будет размещен в файле **MyReport.Pdf**. Также мы передаем параметры экспорта в объекте типа **StiPdfExportSettings**. Подробно класс **StiPdfExportSettings** описан в описании формата PDF. Если нам не нужно изменять какие либо параметры экспорта мы можем использовать более короткий код:

C#

```
...
report.ExportDocument(StiExportFormat.Pdf, "MyReport.Pdf");
...
```

В данном случае параметры экспорта не передаются и генератор отчетов будет использовать параметры, которые заданы по умолчанию для каждого экспорта. Кроме этого, Вы можете разместить результат экспорта в потоке. Например:

C#

```
...
MemoryStream stream = new MemoryStream();
report.ExportDocument(StiExportFormat.Pdf, stream);
...
```

Информация

Метод **ExportDocument** не вызывает автоматически метод **Render**. Перед вызовом метода **ExportDocument** нужно обязательно построить отчет или загрузить ранее построенный отчет.

Как видите, никаких сервисов в примерах не создавали, примеры содержат понятный и простой код. Всю работу по созданию сервисов и проверке параметров проводит метод **ExportDocument**. Приведенный выше код требует подключения следующих пространств имен из сборки **Stimulsoft.Reports.dll**:

C#

```
...
Stimulsoft.Report
...
```

© 2003-2026 Stimulsoft

14.4.1.2 Использование сервисов экспорта

Теперь рассмотрим вариант с созданием сервиса экспорта. Рассмотрим код:

C#

```
...
StiPdfExportService service = new StiPdfExportService();
StiPdfExportSettings settings = new StiPdfExportSettings();
MemoryStream stream = new MemoryStream();
service.ExportPdf(report, stream, settings);
...
```

Этот код сначала создает сервис экспорта в формат PDF. Затем методу **ExportPdf** передается поток, куда необходимо положить результат и параметры экспорта. Сервис экспорта **StiPdfExportService** имеет множество перегрузок метода **ExportPdf**. Этот метод собственно и производит экспорт отчета. Как и в случае с методом **ExportDocument**, отчет уже должен быть построенным! В целом использование метода **ExportPdf** аналогично использованию метода **ExportDocument**. Стоит отметить, что для каждого сервиса название этого метода разное, к примеру в сервисе **StiHtmlExportService** он называется **ExportHtml**.

Если Вы экспортировали отчет из **WinForms** вьювера, то Вы скорей всего заметили, что для каждого экспорта предоставляется специальная форма для настройки параметров экспорта. Эту форму можно также вызвать и из кода. Ниже приводится пример как это сделать для экспорта в формат **PDF**:

C#

```
...
service.Export(report, "MyReport.pdf");
...
```

Этот код вызовет диалоговую форму для настройки параметров экспорта отчета перед процессом построения файла в формате **PDF**. Если пользователь нажмет кнопку "**OK**", то файл будет создан. Если кнопку "**Cancel**", то создание файла в формате **PDF** будет прервано. Обратите внимание на то, что наименование метода для экспорта отчета с формой диалогов отличается от наименования метода экспорта без параметров. Сервис экспорта отчета также содержит еще одну возможность. Вы можете отослать отчет по почте. Например:

© 2003-2026 Stimulsoft

C#

```
...
bool sendEMail = true;
service.Export(report, "MyReport.pdf", sendEMail);
...
```

Этот код покажет диалоговую форму настройки параметров отчетов и, если пользователь нажмет кнопку "OK", то генератор отчетов вызовет **Email** клиент по умолчанию и создаст в нем новое письмо, в приложении которого будет находится экспортированный отчет. Приведенный выше код требует подключения следующих пространств имен из сборки **Stimulsoft.Report.dll**:

C#

```
...
Stimulsoft.Report
Stimulsoft.Report.Export
...
```

Полный список сервисов

Сервисы экспортов в форматы **Adobe PDF** и **Microsoft XPS**:

- StiPdfExportService
- StiXpsExportService

Сервисы экспортов в форматы **HTML** и **MHT**:

- StiHtmlExportService
- StiMhtExportService

Сервисы экспортов в текстовые форматы:

- StiTxtExportService
- StiRtfExportService
- StiWordExportService
- StiOdtExportService

Сервисы экспортов в форматы **Microsoft Excel** и **Open Document Calc**:

- StiExcelXmlExportService
- StiExcelExportService
- StiExcel2007ExportService
- StiOdsExportService

Сервисы экспортов в данные:

- StiCsvExportService

Сервисы экспортов в графические форматы:

- StiBmpExportService

© 2003-2026 Stimulsoft

- StiDbfExportService
- StiXmlExportService
- StiDifExportService
- StiSylkExportService
- StiGifExportService
- StiJpegExportService
- StiPcxExportService
- StiPngExportService
- StiTiffExportService
- StiEmfExportService

14.4.2 Форматы с фиксированной разметкой страницы

Stimulsoft Reports поддерживает два формата с фиксированной разметкой страницы. Что такое фиксированная разметка страницы? Это когда элементы страницы могут находится в любом месте страницы. При этом изменение расположения одного элемента не ведет к изменению расположения других компонентов.

- PDF (Portable Document Format)
- XPS (XML Paper Specification)
- Microsoft Power Point 2007/2010

14.4.2.1 PDF

Формат **PDF (Portable Document Format)** – переносимый формат документов, разработанный компанией Adobe Systems, используется как основа для создания электронных изданий в среде программного пакета Adobe Acrobat. Формат PDF – это файловый текстовой формат, используемый для представления публикаций или других документов на любой аппаратной платформе и для любой операционной среды. Документ PDF содержит одну или более страниц. Каждая страница может содержать любые компоненты электронного издания: текст, графику и иллюстрации, информацию, обеспечивающую навигацию в электронной публикации. Также документ содержит всю служебную информацию, необходимую для правильного отображения документа.

Экспорт в формат **PDF** основан на документе "**Adobe Portable Document Format, Version 1.3, second edition**", с использованием некоторых элементов более поздних спецификаций формата.

Для уменьшения размера файла **PDF** использует различные методы сжатия. Для сжатия текстового материала используется алгоритм LZW ("Flate"). Для сжатия графической информации используются алгоритмы **JPEG** или **LZW**. Алгоритм **JPEG** - это сжатие с потерями, рекомендуется для полноцветных иллюстраций и изображений. Алгоритм LZW - это сжатие без потерь, рекомендуется для

© 2003-2026 Stimulsoft

иллюстраций и изображений с небольшим количеством цветов, например графики, чарты, схемы.

Для обеспечения независимости от шрифтов **PDF** файл содержит описание для каждого шрифта, использованного в документе. Описание включает название, размер, стиль и другие параметры шрифта. В режиме просмотра документа, если шрифт, описанный в документе, доступен, то он используется. Если не доступен, то заменяется на подобный с тем же размером и другими характеристиками. Шрифты можно внедрять в документ. Это значительно увеличивает размер файла, но гарантирует правильное отображение документа на любом компьютере.

Информация

При экспорте в PDF поля, для которых свойство **Printable** установлено в значение **false**, экспортируются в отдельный слой. Этот слой не выводится на печать в Adobe Acrobat. При этом стоит учитывать, что в режиме совместимости PDF-A есть ограничение - слои не поддерживаются, поэтому свойство **Printable** игнорируется.

Если же происходит **Печать в PDF** из web-вьювера - то такие поля не экспортируются.

Цифровая подпись

Электронная цифровая подпись (ЭЦП)— реквизит электронного документа, предназначенный для защиты данного электронного документа от подделки, полученный в результате криптографического преобразования информации с использованием закрытого ключа электронной цифровой подписи и позволяющий идентифицировать владельца сертификата ключа подписи, а также установить отсутствие искажения информации в электронном документе.

Управление цифровой подписью из кода осуществляется через класс **StiPdfExportSettings**. За это отвечают следующие свойства этого класса:

© 2003-2026 Stimulsoft

C#

```
...
public bool UseDigitalSignature
public bool UseLocalMachineCertificates
public bool GetCertificateFromCryptoUI
public string SubjectNameString
...
```

По умолчанию устанавливаются следующие значения:

C#

```
...
UseDigitalSignature = false;
UseLocalMachineCertificates = true;
GetCertificateFromCryptoUI = true;
SubjectNameString = string.Empty;
...
```

Пример использования:

C#

```
...
StiReport report = new StiReport();
report.Load("c:\\test.mrt");
report.Render(false);

StiPdfExportSettings settings = new StiPdfExportSettings();
settings.UseDigitalSignature = true;
settings.GetCertificateFromCryptoUI = false;
settings.UseLocalMachineCertificates = true;
settings.SubjectNameString = "John Smith <johns@google.com>";

report.ExportDocument(StiExportFormat.Pdf, "c:\\test.pdf", settings);
...
```

В JavaScript свойства такие же как и в C#, которые располагаются в классе `Stimulsoft.Report.Export.StiPdfExportSettings`. Пример использования:

© 2003-2026 Stimulsoft

JS

```
...
const report = new StiReport();
report.Load("c:\\test.mrt");

report.renderAsync(function () {
   const settings = new Stimulsoft.Report.Export.StiPdfExportSettings();
   settings.useDigitalSignature = true;
   settings.getCertificateFromCryptoUI = false;
   settings.useLocalMachineCertificates = true;
   settings.subjectNameString = "John Smith <johns@google.com>";

   report.exportDocumentAsync(function (data) {
      Stimulsoft.System.StiObject.saveAs(data, "Report.pdf", "application/
      pdf");
   }, Stimulsoft.Report.StiExportFormat.Pdf, null, settings);
});
...
```

Шифрование

Документ PDF может быть зашифрован для защиты своего содержимого от несанкционированного доступа. Пользователь может задать следующие параметры шифрования:

- Пароль пользователя;
- Пароль владельца;
- Разрешения доступа;
- Длина ключа.

Управление шифрованием из кода осуществляется через класс StiPdfExportSettings. За шифрование отвечают следующие свойства этого класса:

C#

```
...
public string PasswordInputUser
public string PasswordInputOwner
public StiUserAccessPrivileges UserAccessPrivileges
public StiPdfEncryptionKeyLength KeyLength
...
```

Перечисление **StiUserAccessPrivileges** содержит следующие элементы (флаги):

- None,
- PrintDocument,

© 2003-2026 Stimulsoft

- ModifyContents,
- CopyTextAndGraphics,
- AddOrModifyTextAnnotations,
- All

Перечисление StiPdfEncryptionKeyLength содержит следующие элементы:
- Bit40 - 40-битное шифрование;
- Bit128 - 128-битное шифрование;
- Bit256_r5 - 256-битное шифрование с внутренней ревизией версии 5 (Acrobat 9);
- Bit256_r6 - 256-битное шифрование с внутренней ревизией версии 6 (Acrobat X).

По умолчанию устанавливаются следующие значения:

C#

```
...
PasswordInputUser = string.Empty;
PasswordInputOwner = string.Empty;
UserAccessPrivileges = StiUserAccessPrivileges.All;
KeyLength = StiPdfEncryptionKeyLength.Bit40;
...
```

Пример использования:

C#

```
...
StiReport report = new StiReport();
report.Load("c:\\test.mrt");
report.Render(false);

StiPdfExportSettings settings = new StiPdfExportSettings();
settings.PasswordInputUser = "user";
settings.PasswordInputOwner = "owner";
settings.UserAccessPrivileges = StiUserAccessPrivileges.PrintDocument;
settings.KeyLength = StiPdfEncryptionKeyLength.Bit128;

report.ExportDocument(StiExportFormat.Pdf, "c:\\test.pdf", settings);
...
```

В JavaScript свойства такие же как и в C#, которые располагаются в классе `Stimulsoft.Report.Export.StiPdfExportSettings`. Пример

© 2003-2026 Stimulsoft

использования:

JS

```
...
const report = new StiReport();
report.Load("c:\\test.mrt");

report.renderAsync(function () {
   const settings = new Stimulsoft.Report.Export.StiPdfExportSettings();
   settings.passwordInputUser = "user";
   settings.passwordInputOwner = "owner";
   settings.userAccessPrivileges = StiUserAccessPrivileges.PrintDocument;
   settings.keyLength = StiPdfEncryptionKeyLength.Bit256_r6;

   report.exportDocumentAsync(function (data) {
      Stimulsoft.System.StiObject.saveAs(data, "Report.pdf", "application/
      pdf");
   }, Stimulsoft.Report.StiExportFormat.Pdf, null, settings);
});
...
```

Встроенные шрифты

По умолчанию все встроенные шрифты оптимизированы. Символы, которые не используются в отчете, исключаются. Это позволяет уменьшить размер файла. Однако для корректной работы редактируемого поля шрифт должен быть полным. Поэтому для шрифтов, которые используются в редактируемых полях, оптимизация не выполняется. Это увеличивает размер выходного файла. Если используются азиатские языки, размер файла может составлять 15-20 МБ.

Если по каким-то причинам оптимизация шрифтов работает некорректно, ее можно принудительно отключить с помощью статического свойства:

C#

```
...
StiOptions.Export.Pdf.ReduceFontFileSize = false;
...
```

Редактируемые поля

Чтобы включить экспорт редактируемых полей, необходимо установить статическое свойство:

© 2003-2026 Stimulsoft

C#

```
...
StiOptions.Export.Pdf.AllowEditablePdf = true;
...
```

Редактируемые поля в PDF-файле имеют два состояния:

- Первое — состояние до редактирования, которое отображается при открытии файла. Это состояние соответствует типу текстового поля в превью.
- Второе — тип в режиме редактирования поля и после редактирования. В этом состоянии невозможно установить вертикальное выравнивание текста (всегда "сверху") и некоторые параметры шрифта. Поэтому после редактирования поля, даже если содержимое не изменено, тип этого поля может измениться.

Если необходимо иметь редактируемое поле с несколькими строками, то необходимо установить свойство **WordWrap** текстового поля в значение **true**.

Параметры экспорта

Параметры экспорта в формат **PDF** описываются в классе **StiPdfExportSettings**. Ниже приводится таблица с описанием всех свойств этого класса.

Название	Тип	Описание
AutoPrintMode	bool	Если необходимо, чтобы при открытии документа Pdf появлялось окно печати, необходимо данное свойство установить в значение **true**.
AllowEditable	bool	При создании отчета, у компоненты можно устанавливать включить (установив флажок) свойство Редактируемый.

© 2003-2026 Stimulsoft

		Это предоставляет возможность, в построенном отчете, редактировать эти компненты. Свойство **AllowEditable** предоставляет возможность разрешить или запретить изменение редактируемых компонентов в документе Pdf.
CertificateData	byte	При помощи данного свойства, можно передавать сертификат как массив-байт.
CertificatePassword	string	При помощи данного, если это необходимо (зависит от сертификата), свойства указывается пароль к сертификату.
DigitalSignatureReason	string	Свойство цифровой подписи, в котором указывается основание ее использования.
DigitalSignatureLocation	string	Свойство цифровой подписи, в котором указывается ее применение.
DigitalSignatureContactInfo	string	Свойство цифровой подписи, в котором указывается контактная информация того, кто подписал.
DigitalSignatureSignedBy	string	Свойство цифровой подписи. Как правило, данные о том, кем

© 2003-2026 Stimulsoft

		подписан документ, берутся из сертификата и встраиваются в документ Pdf. Иногда, возникает необходимость изменить лицо, которое подписало документ. Это возможность предоставляет данное свойство.
ImageQuality	float	качество изображений; может принимать значение от 0.0 (низкое качество) до 1.0 (высокое качество); по умолчанию 0.75
ImageResolution	float	разрешение изображений, точек на дюйм; может принимать любое значение, по умолчанию 100
ImageFormat		при помощи данного свойства можно указать формат для изображений: монохромный или черно-белые. По умолчанию, изображения экспортируются как цветные.
DitheringType	enum	при помощи данного свойства, определяется формат монохромного изображения (с дизерингом или без).
EmbeddedFonts	bool	внедрять файлы шрифтов

© 2003-2026 Stimulsoft

		внутрь файла PDF; если true, то все необходимые шрифты внедряются внутрь, и PDF показывается одинаково на любом компьютере (никакие дополнительные установленные шрифты не нужны); если false, то шрифты не внедряются; по умолчанию true
StandardPdfFonts	bool	использовать только стандартные шрифты, которые устанавливаются вместе с Adobe Acrobat Reader и их не требуется внедрять в файл; все шрифты заменяются на похожие стандартные (Courier, Helvetica, Times-Roman); по умолчанию false
Compressed	bool	сжимать файл PDF; уменьшает размер файла за счет сжатия текстовой информации (изображения сжимаются всегда); по умолчанию true
UseUnicode	bool	использовать для записи текста в кодировке Unicode; если false, то в отчете может быть только 190 разных символов, и много проблем с национальными

© 2003-2026 Stimulsoft

		символами; если true – можно использовать любые символы; по умолчанию true
ExportRtfTextAsImage	bool	экспортировать объекты RichText как изображения; если false, то экспорт пытается преобразовать RichText объекты в примитивы PDF; если true, то RichText записывается как изображение; по умолчанию false
PasswordInputUser	string	пароль пользователя (см. Encryption); по умолчанию пустая строка
PasswordInputOwner	string	пароль владельца (см. Encryption); по умолчанию пустая строка
PdfACompliance	bool	Использовать или не использовать Pdf/A соглашение. Если значение true, то соглашение будет использоваться.
UserAccessPrivileges	enum	разрешения доступа пользователя (см. Encryption); по умолчанию StiUserAccessPrivileges.All
KeyLength	enum	длина ключа шифрования (см. Encryption); по умолчанию StiPdfEncryptionKeyLength.Bit40

© 2003-2026 Stimulsoft

UseDigitalSignature	bool	использовать цифровую подпись документа; по умолчанию false
GetCertificateFromCryptoUI	bool	для выбора необходимого сертификата использовать интерфейс системной библиотеки криптографии; если false - поиск сертификата ведется по идентификатору сертификата без использования интерфейса; по умолчанию true
SubjectNameString	string	идентификатор сертификата; это имя владельца сертификата (полная строка) или часть имени (подстрока); по умолчанию пустая строка
UseLocalMachineCertificates	bool	производить поиск сертификатов в хранилище локального компьютера; если false, то поиск сертификатов производится в хранилище текущего пользователя; по умолчанию false
CreatorString	string	поле "Creator" в описании документа (имя приложения, которое создало оригинальный файл); если не задано (пустая строка), то используется статическое

© 2003-2026 Stimulsoft

		свойство StiOptions.Export.Pdf.CreatorString; по умолчанию пустая строка
KeywordsString	string	поле "Keywords" в описании документа (ключевые слова для поиска документа); если не задано (пустая строка), то используется статическое свойство StiOptions.Export.Pdf.KeywordsString; по умолчанию пустая строка
ImageCompressionMethod	enum	метод сжатия изображений - Jpeg (с потерями) или Flate (без потерь); по умолчанию StiPdfImageCompressionMethod.Jpeg

Если используется **UseUnicode**, то для **Acrobat Reader 5.0** нужно обязательно **Embedded fonts = true**.
Если используется **UseUnicode + шифрование**, то обязательно должно быть **Embedded fonts = true**.

Чтобы уменьшить размер выходного файла, можно сделать следующее:
- Включить **Compressed**;
- Не включать **Embedded fonts**;
- Если Embedded fonts необходим, то обязательно включить **ReduceFontFileSize**.

Статические параметры

Кроме класса **StiPdfExportSettings** параметры экспорта в **PDF** также задаются при помощи статических свойств. Каждое свойство имеет глобальное влияние на любой экспорт в формат **PDF** из кода или при помощи **Web** или **WinForms**

© 2003-2026 Stimulsoft

Viewer-а. Все свойства с описанием приведены ниже. Для доступа к свойствам нужно добавлять префикс **StiOptions.Export.Pdf**... например, **StiOptions.Export.Pdf.DivideSegmentPages**.

Название	Тип	Описание
DivideSegmentPages	bool	разбивать сегментированные страницы на отдельные страницы выбранного формата; если false - страницы передаются как есть, без разбивки на части; по умолчанию true
ConvertDigitsToArabic	bool	конвертировать ASCII цифры в арабские цифры; по умолчанию false
ArabicDigitsType	enum	выбор типа арабских цифр; по умолчанию Standard
ReduceFontFileSize	bool	оптимизировать внедряемые шрифты - исключать символы, которые не встречаются в отчете; если false - шрифты не изменяются; по умолчанию true
AllowEditablePdf	bool	экспортировать редактируемые поля как редактируемые объекты PDF (при этом шрифты, которые используются в редактируемых полях, не оптимизируются); если false - редактируемые поля экспортируются как простой текст; по

© 2003-2026 Stimulsoft

		умолчанию false
AllowImageComparer	bool	использовать оптимизатор изображений, т.е. удалять дубликаты изображений (см. Common export settings); если false - изображения передаются как есть; по умолчанию true
AllowImageTransparency	bool	использовать прозрачность при экспорте изображений; по умолчанию true
AllowInheritedPageResources	bool	хранить ресурсы страниц в родительском словаре и наследовать от него; если false - ресурсы страниц указываются в каждой странице; это свойство критично для некоторых программ обработки PDF-файлов; по умолчанию true
AllowExtGState	bool	использовать команды управления прозрачностью при создании документа; если false - команды не используются; это свойство критично для некоторых программ обработки PDF-файлов; по умолчанию true
CreatorString	string	поле "Creator" в описании документа (имя приложения, которое

© 2003-2026 Stimulsoft

		создало оригинальный файл); по умолчанию строка "Stimulsoft Reports.NET"
KeywordsString	string	поле "Keywords" в описании документа (ключевые слова для поиска документа); по умолчанию пустая строка

14.4.2.2 ZUGFeRD

В продуктах компании Stimulsoft добавлена поддержка формата электронных инвойсов ZUGFeRD 2.0.

Счета в формате **ZUGFeRD** передают как человекочитаемые счета-фактуры, так и его структурированное машиночитаемое XML-представление. Человекочитаемое представление кодируется в виде обычных страниц PDF-документа. XML-представление встроено в PDF-документ в качестве вложенного файла. PDF-файл должен соответствовать стандарту PDF/A-3, который разрешает использование вложенных файлов. Также XMP-метаданные pdf-файла должны содержать необходимую информацию согласно используемой версии стандарта **ZUGFeRD**.

На данный момент использовать формат **ZUGFeRD** можно только из кода. Для этого следует в настройках экспорта в PDF выбрать с помощью опции **ZUGFeRDComplianceMode** нужный вариант формата (V1 или V2), с помощью опции **ZUGFeRDConformanceLevel** выбрать нужный **Conformance Level**, а также загрузить заранее приготовленный XML-файл в свойство **ZUGFeRDInvoiceData**.
При этом файл будет автоматически добавлен в коллекцию **EmbeddedFiles** со стандартным **FileName** и **Description**. Если вам надо использовать другой **Description**, вы можете сами добавить файл в коллекцию **EmbeddedFiles** с нужным **FileName** и **Description**.

Обращаем внимание:
- Имя XML-файла в разных версиях стандарта отличается регистром.
- ConformanceLevel COMFORT в ZUGFeRD 2.0 заменен на EN 16931.

Ниже представлен пример кода для экспорта отчета с использованием

© 2003-2026 Stimulsoft

формата **ZUGFeRD**:

C#

```
...
FileStream fileStream = new FileStream(@"d:\test.pdf", FileMode.Create);
byte[] buf = File.ReadAllBytes(@"d:\ZUGFeRD-invoice.xml");

//for ZUGFeRD 1.0
var pdfExportSettings = new StiPdfExportSettings()
{
   ZUGFeRDComplianceMode = StiPdfZUGFeRDComplianceMode.V1,
   ZUGFeRDInvoiceData = buf,
   ZUGFeRDConformanceLevel = "COMFORT"     //BASIC, COMFORT, EXTENDED
};

//for ZUGFeRD 2.0
var pdfExportSettings = new StiPdfExportSettings()
{
   ZUGFeRDComplianceMode = StiPdfZUGFeRDComplianceMode.V2,
   ZUGFeRDInvoiceData = buf,
   ZUGFeRDConformanceLevel = "EN 16931"    //BASIC, EN 16931, EXTENDED
};

report.ExportDocument(StiExportFormat.Pdf, fileStream, pdfExportSettings);
fileStream.Close();
...
```

Ниже представлен пример кода для экспорта отчета с использованием формата **ZUGFeRD**, если необходимо использовать альтернативный **Description** для XML-файла:

© 2003-2026 Stimulsoft

C#

```
...
FileStream fileStream = new FileStream(@"d:\test.pdf", FileMode.Create);
byte[] buf = File.ReadAllBytes(@"d:\ZUGFeRD-invoice.xml");

//for ZUGFeRD 1.0, Custom settings
var pdfExportSettings = new StiPdfExportSettings();
pdfExportSettings.ZUGFeRDComplianceMode = StiPdfZUGFeRDComplianceMode.V1;
pdfExportSettings.EmbeddedFiles.Add(new StiPdfEmbeddedFileData("ZUGFeRD-
invoice.xml", "ZUGFeRD Invoice", buf));
pdfExportSettings.ZUGFeRDConformanceLevel = "COMFORT";

//for ZUGFeRD 2.0, Custom settings
var pdfExportSettings = new StiPdfExportSettings();
pdfExportSettings.ZUGFeRDComplianceMode = StiPdfZUGFeRDComplianceMode.V2;
pdfExportSettings.EmbeddedFiles.Add(new StiPdfEmbeddedFileData("zugferd-
invoice.xml", "ZUGFeRD Invoice", buf));
pdfExportSettings.ZUGFeRDConformanceLevel = "EN 16931";

report.ExportDocument(StiExportFormat.Pdf, fileStream, pdfExportSettings);
fileStream.Close();
...
```

14.4.2.3 Особенности экспорта PDF/A

PDF/A — это стандартизированная по ISO версия PDF, предназначенная для использования при архивировании и долговременном хранении электронных документов. Стандарт PDF/A не определяет стратегию архивирования или цели системы архивирования. Он определяет «профиль» для электронных документов, который гарантирует, что документы могут быть воспроизведены точно таким же образом с использованием различного программного обеспечения в ближайшие годы. Ключевым элементом этой воспроизводимости является требование к документам PDF/A быть на 100% автономными. Вся информация, необходимая для отображения документа таким же образом, встроена в файл. Это включает, помимо прочего, все содержимое (текст, растровые изображения и векторную графику), шрифты и информацию о цвете. Документ PDF/A не может зависеть от информации из внешних источников.

При включенном режиме "PDF/A Compliance" на экспортируемый отчет накладываются следующие ограничения:

- не используется прозрачность (PDF/A-1 only);
- не используются гиперлинки;
- не используются подсказки при наведении;
- шрифты всегда включены;
- не используется **шифрование** документа.

© 2003-2026 Stimulsoft

При экспорте отчета в PDF документ с использованием стандарта PDF/A могут возникать некоторые ошибки.

Ошибка "Font not embedded".
Стандарт PDF-A требует, чтобы шрифты были включены в pdf-файл. Поэтому, если установлен режим "PDF/A Compliance", то независимо от параметра "Embedded fonts" в настройках экспорта, шрифты будут принудительно включены в pdf-экспорт. Однако в приложениях в которых используются .NET Core и JS компоненты, ядро генератора отчетов имеет доступ только к тем шрифтам, которые загружены в `StiFontCollection`. Соответственно, если какой-то шрифт не загружен, он не будет включен в PDF файл. В следствие чего, проверка на PDF/A покажет ошибку "Font not embedded".

HomeController.cs

```
StiFontCollection.AddFontFile(StiNetCoreHelper.MapPath(this, "Reports/
Font.ttf"));
```

Ошибка "Width information for rendered glyphs is inconsistent"
("Glyph widths in font dictionary are not consistent with embedded font program widths")
TrueType, OpenType представляют собой векторные шрифты. В них контуры всех символов хранятся в виде точек, которые соединены прямыми или дугами. Ширина символов не хранится в шрифте, она рассчитывается на основании геометрии символа. Внутри PDF файла, кроме самого файла шрифта, записывается также блок информации об этом шрифте:

- Название;
- Кодировка;
- Таблица "Widths", в которой указана ширина всех используемых символов. Эту таблицу использует вьювер для ускорения работы с текстом (как кэш), чтобы не рассчитывать ширину символов каждый раз из данных шрифта.

Проверка на соответствие стандарту PDF/A состоит из многих этапов. Один из этапов - это проверка, чтобы данные из таблицы Widths соответствовали данным шрифта. При этом утилита проверки рассчитывает ширину символов и сравнивает ее с записанной в таблице Widths. По стандартам допускается отклонение на 0.1%. В этот момент и проявляется проблема. Различные

© 2003-2026 Stimulsoft

графические библиотеки рассчитывают ширину символов по-разному, в следствие чего, значения могут отличаться, иногда до 1-2%, что гораздо больше, чем допустимые 0.1%. Причем даже разные утилиты проверки могут по-разному рассчитывать ширину, и соответственно ругаться на разные файлы.

Примечание

Пример из нашей практики: PDF файл прошел проверку в Adobe Acrobat Pro, но не прошел в одной из онлайн-утилит. Наша графическая библиотека рассчитала ширину одного из символов как 554, и это значение подошло для Adobe Acrobat Pro. Когда мы вручную заменили число на 553, файл прошел проверку в онлайн-утилите, но перестал проходить в Adobe Acrobat Pro.

Универсального решения данной проблемы не существует. Однако, при возникновении такой ошибки стоит учитывать этот нюанс.

14.4.2.4 Microsoft Power Point

Обратите внимание

Для десктопных версий конкретных ограничений размеров нет, размер открываемого файла ограничен свободной памятью компьютера. Для web-версий есть ограничения: таймаут операций загрузки/сохранения установлен на 1 минуту, и обычно файлы более 1 гигабайта не успевают сохраниться.

Microsoft PowerPoint - программа для создания и проведения презентаций, являющаяся частью пакета **Microsoft Office**. PowerPoint презентации состоят из ряда отдельных страниц или "слайдов". Слайды могут содержать текст, графику, фильмы и другие объекты, которые могут быть размещены на слайде. Презентации могут быть отображены на ПК либо распечатаны. В Stimulsoft Reports каждой странице отчета соответствует один слайд.

Настройки экспорта

Параметры PPT экспорта описаны в классе **StiPowerPointExportSettings**.

© 2003-2026 Stimulsoft

Описание всех свойств класса приведены в таблице ниже:

Название	Тип	Описание
ImageQuality	float	качество изображений; может принимать значение от 0.0 (низкое качество) до 1.0 (высокое качество); по умолчанию 0.75
ImageResolution	float	разрешение изображений, точек на дюйм; может принимать любое значение, по умолчанию 100

Статические параметры

Кроме класса **StiPowerPointExportSettings** параметры экспорта в **PPT** также задаются при помощи статических свойств. Каждое свойство имеет глобальное влияние на любой экспорт в формат PPT из кода или при помощи **Web** или **WinForms Viewer**-а. Все свойства с описанием приведены ниже. Для доступа к свойствам нужно добавлять префикс **StiOptions.Export.PowerPoint.** например, **StiOptions.Export.PowerPoint.ReduceFontFileSize**.

Название	Тип	Описание
AllowImageComparer	bool	использовать оптимизатор изображений, т.е. удалять дубликаты изображений (см. Common export settings); если **false** - изображения передаются как есть; по умолчанию **true**

© 2003-2026 Stimulsoft

14.4.2.5 XPS

Формат **XPS (XML Paper Specification)** — открытый графический формат фиксированной разметки страниц на базе **XML** (точнее на базе **XAML**), предназначен для хранения макетов печатной продукции в электронном виде. Формат разработан компанией Microsoft как альтернатива широко известному формату PDF.

Документ **XPS** содержит структурированную разметку **XML**, которая определяет структуру документа, внешний вид каждой страницы, а также правила распространения, архивирования, рендеринга, обработки и печати документа. Язык разметки XPS является подмножеством XAML, что позволяет включать в документы элементы векторной графики, используя **XAML** для описания **WPF**-примитивов.

Файл **XPS** является **ZIP**-архивом, который содержит все файлы - части документа. Архив включает в себя разметку страниц (по одному файлу на каждую страницу документа), текст, внедренные шрифты, растровые изображения, векторную графику и другую информацию.

Параметры экспорта

Параметры экспорта в формат **XPS** описываются в классе **StiXpsExportSettings**. Ниже приводится таблица с описанием всех свойств этого класса.

Название	Тип	Описание
ImageQuality	float	качество изображений; может принимать значение от 0.0 (низкое качество) до1.0 (высокое качество); по умолчанию 0.75
ImageResolution	float	разрешение изображений, точек на дюйм; может принимать любое значение, по умолчанию 100

© 2003-2026 Stimulsoft

Статические параметры

Кроме класса **StiXpsExportSettings** параметры экспорта в **XPS** также задаются при помощи статических свойств. Каждое свойство имеет глобальное влияние на любой экспорт в формат XPS из кода или при помощи Web или WinForms Viewer-а. Все свойства с описанием приведены ниже. Для доступа к свойствам нужно добавлять префикс **StiOptions.Export.Xps...** например, **StiOptions.Export.Xps.ReduceFontFileSize**.

Название	Тип	Описание
ReduceFontFileSize	bool	оптимизировать внедряемые шрифты - исключать символы, которые не встречаются в отчете; если false - шрифты не изменяются; по умолчанию true
AllowImageComparer	bool	использовать оптимизатор изображений, т.е. удалять дубликаты изображений (см. Common export settings); если false - изображения передаются как есть; по умолчанию true
AllowImageTransparency	bool	использовать прозрачность при экспорте изображений; по умолчанию true

14.4.3 Web-документы

К этой категории форматов отнесены два формата - **HTML (HyperText Markup Language)** и **MHTML (MIME HTML)**. Первый формат используется для разметки **web**-страниц. Второй - для сохранения файлов первого формата в упакованном виде.

› HTML

© 2003-2026 Stimulsoft

- MHT

14.4.3.1 HTML

Формат **HTML (HyperText Markup Language)** - стандартный язык разметки документов во Всемирной паутине. Большинство web-страниц создаются при помощи языка **HTML**. Язык **HTML** интерпретируется браузером и отображается в виде документа, в удобной для человека форме. **HTML** — это теговый язык разметки документов. Любой документ на языке **HTML** представляет собой набор элементов, причем начало и конец каждого элемента обозначается специальными пометками — тегами. Элементы могут быть пустыми, то есть не содержащими никакого текста и других данных (например, тег перевода строки **
). В этом случае обычно не указывается закрывающий тег. Кроме того, элементы могут иметь атрибуты, определяющие какие-либо их свойства (например, размер шрифта для элемента **font). Атрибуты указываются в открывающем теге.

Параметры экспорта

Параметры экспорта в формат **HTML** описываются в классе **StiHtmlExportSettings**. Ниже приводится таблица с описанием всех свойств этого класса.

Название	Тип	Описание
Zoom	double	коэффициент масштабирования. По умолчанию установлено значение 1.0, что равно 100% в окне настройки экспорта
ImageFormat	ImageFormat	устанавливает, в каком формате будут экспортироваться изображения документа; по умолчанию ImageFormat.Png
ExportMode	StiHtmlExportMode	устанавливает режим экспорта документа - с использованием

© 2003-2026 Stimulsoft

		элементов div, span или table; по умолчанию StiHtmlExportMode.Table
ExportQuality	StiHtmlExportQuality	качество экспорта размеров компонентов; по умолчанию StiHtmlExportQuality.High
Encoding	Encoding	кодировка выходного файла; по умолчанию Encoding.UTF8
AddPageBreaks	bool	добавлять разделители страниц; по умолчанию false
BookmarksTreeWidth	int	ширина колонки закладок, в пикселях; по умолчанию 150
ExportBookmarksMode	StiHtmlExportBookmarksMode	режим экспорта документа с закладками; по умолчанию StiHtmlExportBookmarksMode.All
UseStylesTable	bool	использовать таблицу стилей; если false - таблица стилей будет пустая, а все свойства каждого компонента будут описываться прямо в стиле этого компонента; по умолчанию true

Статические параметры

Кроме класса **StiHtmlExportSettings** параметры экспорта в **HTML** также задаются при помощи статических свойств. Каждое свойство имеет глобальное влияние на любой экспорт в формат **HTML** из кода или при помощи **Web** или

© 2003-2026 Stimulsoft

WinForms Viewer-a. Все свойства с описанием приведены ниже. Для доступа к свойствам нужно добавлять префикс **StiOptions.Export.Html...** например, **StiOptions.Export.Html.ConvertDigitsToArabic**.

Название	Тип	Описание
ConvertDigitsToArabic	bool	конвертировать ASCII цифры в арабские цифры; по умолчанию false
ArabicDigitsType	enum	выбор типа арабских цифр; по умолчанию Standard
AllowImageComparer	bool	использовать оптимизатор изображений, т.е. удалять дубликаты изображений (см. Common export settings); если false - изображения передаются как есть; по умолчанию true
ForceWysiwygWordwrap	bool	принудительно разбивать текст на строки так же, как в режиме WYSIWYG; по умолчанию false
ReplaceSpecialCharacters	bool	заменять символы '<', '>', '&', ' " ' конструкциями < > & " по умолчанию true

14.4.3.2 MHT

Формат **MHTML** (**MIME HTML**) - это формат хранения web-страниц, который сохраняет **HTML**-файл и все ресурсы, на которые он ссылается (изображения, **flash**, **java-applets**, аудио), в одном файле. Полученный файл называется веб-архивом, и обычно имеет расширение «**.mht**». Содержимое файла записывается как обычное почтовое сообщение с использованием стандарта **MIME**: в начале

© 2003-2026 Stimulsoft

файла записан **HTML**-файл, затем все ресурсы в кодировке **base64** с соответствующими заголовками. С форматом **MHTML** могут работать **Internet Explorer**, **Opera**, **Microsoft Word**.

Параметры экспорта

Параметры экспорта в формат **MHT** описываются в классе **StiMhtExportSettings**. Ниже приводится таблица с описанием всех свойств этого класса.

Название	Тип	Описание
Zoom	double	коэффициент масштабирования. По умолчанию установлено значение 1.0, что равно 100% в окне настройки экспорта
ImageFormat	ImageFormat	устанавливает, в каком формате будут экспортироваться изображения документа; по умолчанию ImageFormat.Png
ExportMode	StiHtmlExportMode	устанавливает режим экспорта документа - с использованием элементов div, span или table; по умолчанию StiHtmlExportMode.Table
ExportQuality	StiHtmlExportQuality	качество экспорта размеров компонентов; по умолчанию StiHtmlExportQuality.High
Encoding	Encoding	кодировка выходного файла; по умолчанию Encoding.UTF8
AddPageBreaks	bool	добавлять разделители

© 2003-2026 Stimulsoft

		страниц; по умолчанию false
BookmarksTreeWidth	int	ширина колонки закладок, в пикселях; по умолчанию 150
ExportBookmarksMode	StiHtmlExportBookmarksMode	режим экспорта документа с закладками; по умолчанию StiHtmlExportBookmarksMode.All

14.4.4 Текстовые форматы

В этой группе экспортов собраны форматы текстовых файлов, т.е. файлов, которые в том или ином виде используются для создания текстовых документов.

- TXT
- RTF
- Word 2007/2010
- ODT

14.4.4.1 TXT

Text file (TXT) - текстовый файл, представляет из себя последовательность символов. Символы сгруппированы в строки. В современных системах строки разделяются переводом строки (0x0D, 0x0A), хотя в прошлом применялось хранение строк в виде записей постоянной или переменной длины. Также применяются символы табуляция (0x09) и перевод страницы (0x0C). Исторически для кодирования текстовых файлов применялись 7-битный набор символов **ASCII** и различные 8-битные расширения **ASCII**. В настоящее время рекомендуется использовать в 8-битных кодовых страницах только первую половину кодовой таблицы, а при необходимости использовать другие символы - применять Unicode (кодировки **UTF-8** или **UTF-16**).

Параметры экспорта

Параметры экспорта в формат **TXT** описываются в классе **StiTxtExportSettings**. Ниже приводится таблица с описанием всех свойств этого класса.

Название	Тип	Описание

© 2003-2026 Stimulsoft

Encoding	Encoding	кодировка текстового файла; по умолчанию Encoding.UTF8
DrawBorder	bool	рисовать линии бордюров; если false - бордюры не рисуются; по умолчанию true
BorderType	StiTxtBorderType	тип линии бордюра; по умолчанию StiTxtBorderType.UnicodeSingle
KillSpaceLines	bool	удалять из текста пустые строки; по умолчанию true
KillSpaceGraphLines	bool	удалять из текста строки, которые содержат только пробелы и символы вертикального бордюра; по умолчанию true
PutFeedPageCode	bool	помещать после каждой страницы управляющий код "перевод страницы"; по умолчанию true
CutLongLines	bool	обрезать длинные строки текста, которые не вмещаются в текстбоксы; по умолчанию true
ZoomX	float	горизонтальный коэффициент масштабирования по оси X. По умолчанию установлено значение 1.0, что равно 100% в окне настройки экспорта
ZoomY	float	вертикальный коэффициент

© 2003-2026 Stimulsoft

		масштабирования по оси Y. По умолчанию установлено значение 1.0, что равно 100% в окне настройки экспорта

Статические параметры

Статические свойства экспорта в формат **TXT** приведены в таблице ниже. Для доступа к свойствам нужно добавлять префикс **StiOptions.Export.Txt...** например, **StiOptions.Export.Txt.ColumnWidths**.

Название	Тип	Описание
ColumnWidths	string	принудительно установить ширину колонок текста (список через точку с запятой); если пустая строка - ширина колонок не изменяется; по умолчанию пустая строка
UseFullTextBoxWidth	bool	использовать для текста всю ширину текстбокса; в этом случае если текст накладывается на рамку, то рамка затирается; если false – при отрисовке текста справа всегда оставляется один пробел для корректной отрисовки рамок; по умолчанию false
UseOldMode	bool	использовать старый режим экспорта текста; свойство оставлено для совместимости со старыми версиями; по

© 2003-2026 Stimulsoft

		умолчанию false
UseFullVerticalBorder	bool	отрисовывать вертикальный бордюр снаружи ячейки, т.е. бордюр никогда не закрывается текстом; по умолчанию true
UseFullHorizontalBorder	bool	отрисовывать горизонтальный бордюр снаружи ячейки, т.е. бордюр никогда не закрывается текстом; по умолчанию true
CheckBoxTextForTrue	string	текст, которым отображается состояние чекбокса true; по умолчанию "+"
CheckBoxTextForFalse	string	текст, которым отображается состояние чекбокса false; по умолчанию "-"

14.4.4.2 RTF

Rich Text Format (RTF) — свободный межплатформенный формат хранения размеченных текстовых документов, предложенный **Microsoft**. Первая версия стандарта RTF появилась в 1987 году, с тех пор спецификация формата несколько раз изменялась и дополнялась. **RTF**-документы поддерживаются большинством современных текстовых редакторов. Под **Microsoft Windows** это, как правило, осуществляется с помощью стандартных библиотек, входящих в состав системы.

Параметры экспорта

Параметры экспорта в формат **RTF** описываются в классе **StiRtfExportSettings**. Ниже приводится таблица с описанием всех свойств этого класса.

Название	Тип	Описание

© 2003-2026 Stimulsoft

ImageQuality	float	качество изображений; может принимать значение от 0.0 (низкое качество) до1.0 (высокое качество); по умолчанию 0.75
ImageResolution	float	разрешение изображений, точек на дюйм; может принимать любое значение, по умолчанию 100
UsePageHeadersAndFooters	bool	обрабатывать хэдеры и футеры страницы (см. Table mode); по умолчанию false
ExportMode	enum	выбор режима экспорта (см. Common knowledge); по умолчанию StiRtfExportMode.Table
CodePage	int	свойство устарело и больше не используется, оставлено для совместимости с ранними версиями

Статические параметры

Кроме класса **StiRtfExportSettings** параметры экспорта в **RTF** также задаются при помощи статических свойств. Каждое свойство имеет глобальное влияние на любой экспорт в формат **RTF** из кода или при помощи **Web** или **WinForms Viewer**-а. Все свойства с описанием приведены ниже. Для доступа к свойствам нужно добавлять префикс **StiOptions.Export.Rtf...** например, **StiOptions.Export.Rtf.UsePageRefField**.

Название	Тип	Описание
UsePageRefField	bool	при экспорте оглавления

© 2003-2026 Stimulsoft

		с номерами страниц (например, отчет "Anchors") использовать для номеров страниц команду MS-Word "PAGEREF" - номера страниц в оглавлении будут динамически изменяться при изменении отчета; если false - номера страниц будут статические; по умолчанию true
ConvertDigitsToArabic	bool	конвертировать ASCII цифры в арабские цифры; по умолчанию false
ArabicDigitsType	enum	выбор типа арабских цифр; по умолчанию Standard
DivideSegmentPages	bool	разбивать сегментированные страницы на отдельные страницы выбранного формата; если false - страницы передаются как есть, без разбивки на части; по умолчанию true
LineHeightExactly	bool	экспортировать высоту строк таблицы "точно" ("exactly"); если false - высота строк экспортируется как "минимум" ("at least"); по умолчанию true
RemoveEmptySpaceAtBottom	bool	вырезать пустое пространство снизу

© 2003-2026 Stimulsoft

		страницы; по умолчанию true
LineSpacing	double	коэффициент корректировки высоты строки в многострочных текстовых полях; по умолчанию 0.965
RightMarginCorrection	int	величина коррекции правого поля ячейки; по умолчанию 0
SpaceBetweenCharacters	int	устанавливает разреженный или уплотненный интервал шрифта, в твипсах (twips); отрицательное значение соответствует уплотнению; по умолчанию -2
UseCanBreakProperty	bool	использовать свойство CanBreak при экспорте каждой строки таблицы; по умолчанию true
DivideBigCells	bool	разбивать очень высокие ячейки на более мелкие для удобства редактирования и скроллирования; по умолчанию true

14.4.4.3 Word

Обратите внимание

Ограничения со стороны Microsoft Word для максимального размера файла Word, который можно открыть:

- только текст - 32 MB;
- с картинакми - 512 MB.

© 2003-2026 Stimulsoft

Microsoft Word - это текстовый процессор, выпускаемый корпорацией **Microsoft** в составе пакета **Microsoft Office**. Первая версия была написана для **IBM PC**, использующих **DOS**, в 1983 году. Позднее выпускались версии для **Apple Macintosh (1984)**, **SCO UNIX и Microsoft Windows (1989)**. **Microsoft Word** является наиболее популярным из используемых в данный момент текстовых процессоров. Начиная с первых версий **MS Word** записывал файлы в двоичном формате с расширением «**.doc**». Спецификация форматов файлов Word была закрытой и опубликована **Microsoft** только в 2008 году. Последняя версия Word 2007 "использует по умолчанию" формат основанный на **XML** - Microsoft Office Open XML. Для нового формата используется и новое расширение файлов - «.docx». Формат представляет собой zip-архив, содержащий текст в виде XML, графику и другие данные. При экспорте в данный формат весь отчет преобразуется в одну большую сплошную таблицу. При преобразовании может происходить некоторое изменение объектов, но полученный документ очень удобен для редактирования.

Параметры экспорта

Параметры экспорта в формат **Word 2007** описываются в классе **StiWordExportSettings**. Ниже приводится таблица с описанием всех свойств этого класса.

Название	Тип	Описание
ImageQuality	float	качество изображений; может принимать значение от 0.0 (низкое качество) до 1.0 (высокое качество); по умолчанию 0.75
ImageResolution	float	разрешение изображений, точек на дюйм; может принимать любое значение, по умолчанию 100
UsePageHeadersAndFooters	bool	обрабатывать хедеры и футеры страницы; по

© 2003-2026 Stimulsoft

		умолчанию false

Статические параметры

Статические свойства экспорта в **Word 2007**. Свойства имеют глобальное влияние на все экспорты в формат **Word 2007**. Для доступа к свойствам нужно добавлять префикс **StiOptions.Export.Word...** например, **StiOptions.Export.Word.DivideSegmentPages**.

Название	Тип	Описание
DivideSegmentPages	bool	разбивать сегментированные страницы на отдельные страницы выбранного формата; если false - страницы передаются как есть, без разбивки на части; по умолчанию true
AllowImageComparer	bool	использовать оптимизатор изображений, т.е. удалять дубликаты изображений (см. Common export settings); если false - изображения передаются как есть; по умолчанию true
LineHeightExactly	bool	экспортировать высоту строк таблицы "точно" ("exactly"); если false - высота строк экспортируется как "минимум" ("at least"); по умолчанию true
RemoveEmptySpaceAtBottom	bool	вырезать пустое пространство снизу

© 2003-2026 Stimulsoft

		страницы; по умолчанию true
RightMarginCorrection	int	величина коррекции правого поля ячейки; по умолчанию 0
SpaceBetweenCharacters	int	устанавливает разреженный или уплотненный интервал шрифта, в твипсах (twips); отрицательное значение соответствует уплотнению; по умолчанию -2

14.4.4.4 ODT

Open Document Text (ODT) - открытый формат хранения документов текстового процессора **OpenOffice Writer**, который входит в состав пакета **OpenOffice.org**.

OpenOffice.org — это свободный пакет офисных приложений, разработанный с целью предоставить альтернативу **Microsoft Office** как на уровне форматов, так и на уровне интерфейса пользователя. Одним из первых стал поддерживать новый открытый формат **OpenDocument**. Работает на платформах **Microsoft Windows** и **UNIX**-подобных платформах: **GNU/Linux**, **Mac OS X**, **FreeBSD**, **Solaris**, **Irix**.

OpenDocument Format (ODF) — открытый формат файлов документов для хранения и обмена редактируемыми офисными документами, в том числе текстовыми документами (такими как заметки, отчеты и книги), электронными таблицами, рисунками, базами данных, презентациями. Формат основан на XML-формате. Стандарт был совместно и публично разработан различными организациями, доступен для всех и может быть использован без ограничений.

OpenOffice Writer — текстовый процессор и визуальный редактор **HTML**, входит в состав **OpenOffice** и является свободным программным обеспечением (выпускается под лицензией **LGPL**). **Writer** похож на **Microsoft Word** и функциональность этих редакторов примерно равна. **Writer** позволяет сохранять документы в различных форматах, включая **Microsoft Word**, **RTF**, **XHTML** и **OASIS Open Document Format**. Начиная с версии **OpenOffice 2.0**, для

© 2003-2026 Stimulsoft

хранения документов форматом по умолчанию используется **OpenDocument Format**, файлы сохраняются с расширением «**.odt**».

При экспорте в данный формат весь отчет преобразуется в одну большую сплошную таблицу. При преобразовании может происходить некоторое изменение объектов, но полученный документ удобен для редактирования.

Параметры экспорта

Параметры экспорта в формат **ODT** описываются в классе **StiOdtExportSettings**. Ниже приводится таблица с описанием всех свойств этого класса.

Название	Тип	Описание
ImageQuality	float	качество изображений; может принимать значение от 0.0 (низкое качество) до1.0 (высокое качество); по умолчанию 0.75
ImageResolution	float	разрешение изображений, точек на дюйм; может принимать любое значение, по умолчанию 100

Статические параметры

Статические свойства экспорта в **ODT**. Свойства имеют глобальное влияние на все экспорты в формат **ODT**. Для доступа к свойствам нужно добавлять префикс **StiOptions.Export.Odt...** например, **StiOptions.Export.Odt.DivideSegmentPages**.

Название	Тип	Описание
DivideSegmentPages	bool	разбивать сегментированные страницы на отдельные страницы выбранного

© 2003-2026 Stimulsoft

		формата; если false - страницы передаются как есть, без разбивки на части; по умолчанию true
AllowImageComparer	bool	использовать оптимизатор изображений, т.е. удалять дубликаты изображений (см. Common export settings); если false - изображения передаются как есть; по умолчанию true
RemoveEmptySpaceAtBottom	bool	вырезать пустое пространство снизу страницы; по умолчанию true

14.4.5 Электронные таблицы

Обратите внимание

Для десктопных версий конкретных ограничений размеров нет, размер открываемого файла ограничен свободной памятью компьютера. Но для web-версий есть ограничения в зависимости от версии продукта и используемого сервиса (10mb - 50mb - 250mb).

Эта группа экспортов, которые создают файлы электронных таблиц. Здесь представлены, как экспорты в различные форматы **Microsoft Excel**, так и в открытый формат хранения документов табличного процессора **OpenOffice Calc**.

- Excel
- Excel 2007/2010
- ODS

© 2003-2026 Stimulsoft

14.4.5.1 Excel

Microsoft Excel - программа для работы с электронными таблицами, созданная корпорацией **Microsoft для Microsoft Windows** и **Mac OS**. Она предоставляет возможности расчетов, графические инструменты и язык макро-программирования **VBA** (**Visual Basic** для приложений). Фактически, это самый популярный табличный процессор, доступный на этих платформах, и является таковым с версии **Microsoft Excel 5** от **1993** года, когда он вышел в составе пакета приложений **Microsoft Office**. На сегодняшний день **Excel** является одним из наиболее популярных приложений в мире. Для хранения документов в качестве основного формата **Microsoft Excel** вплоть до версии **Excel 2007** использовал свой собственный бинарный формат файлов (**Binary Interchange File Format**, сокращенно **BIFF**), и расширение файлов «**.xls**». Спецификация формата была закрытой и опубликована **Microsoft** только в 2008 году. Кроме того, большинство версий **Microsoft Excel** могут читать **CSV**, **DBF**, **SYLK**, **DIF** и другие форматы.

Параметры экспорта

Параметры экспорта в формат **XLS** описываются в классе **StiExcelExportSettings**. Ниже приводится таблица с описанием всех свойств этого класса.

Название	Тип	Описание
ImageQuality	float	качество изображений; может принимать значение от 0.0 (низкое качество) до1.0 (высокое качество); по умолчанию 0.75
ImageResolution	float	разрешение изображений, точек на дюйм; может принимать любое значение, по умолчанию 100
UseOnePageHeaderAndFooter	bool	убирать из отчета все заголовки страницы кроме первого и все подвалы страницы кроме

© 2003-2026 Stimulsoft

		последнего; по умолчанию false
ExportDataOnly	bool	экспортировать только данные, т.е. только компоненты, расположенные на бэндах данные; по умолчанию false
ExportPageBreaks	bool	экспортировать разделители страниц; по умолчанию false
ExportObjectFormatting	bool	экспортировать форматирование объектов; по умолчанию true
ExportEachPageToSheet	bool	экспортировать каждую страницу отчета в отдельный лист; по умолчанию false

Информация

Свойство **ExportObjectFormatting** работает только если **ExportDataOnly = true**.

Статические параметры

Статические свойства экспорта в **Excel**. Свойства имеют глобальное влияние на все экспорты в формат **Excel**. Для доступа к свойствам нужно добавлять префикс **StiOptions.Export.Excel...** например, **StiOptions.Export.Excel.AllowExportDateTime**.

Название	Тип	Описание
AllowExportDateTime	bool	экспортировать дату и

© 2003-2026 Stimulsoft

		время как соответствующие величины; если false - дата и время экспортируются как текстовые строки; по умолчанию false
ColumnsRightToLeft	bool	установить порядок расположения колонок справа налево; по умолчанию false
MaximumSheetHeight	int	максимальное количество строк на листе; лишние строки переносятся на следующий лист; по умолчанию 65534
RemoveEmptySpaceAtBottom	bool	вырезать пустое пространство снизу страницы; по умолчанию true
ShowGridLines	bool	показывать линии сетки; по умолчанию true

14.4.5.2 Excel 2007/2010

Для хранения документов в качестве основного формата **Microsoft Excel** вплоть до версии **Excel 2007** использовал свой собственный бинарный формат файлов (**BIFF**), и расширение файлов «**.xls**». **Excel 2007** в качестве своего основного формата использует новый формат **Microsoft Office Open XML**, и хранит документы в файлах с расширением «**.xlsx**». Несмотря на то, что **Excel 2007** поддерживает и направлен на использование новых **XML**-форматов в качестве основных, он по-прежнему совместим с традиционными бинарными форматами **CSV**, **DBF**, **SYLK**, **DIF** и другими.

Параметры экспорта

Параметры экспорта в формат **Excel 2007** описываются в классе **StiExcel2007ExportSettings**. Ниже приводится таблица с описанием всех

© 2003-2026 Stimulsoft

свойств этого класса.

Название	Тип	Описание
ImageQuality	float	качество изображений; может принимать значение от 0.0 (низкое качество) до1.0 (высокое качество); по умолчанию 0.75
ImageResolution	float	разрешение изображений, точек на дюйм; может принимать любое значение, по умолчанию 100
UseOnePageHeaderAndFooter	bool	убирать из отчета все заголовки страницы кроме первого и все подвалы страницы кроме последнего; по умолчанию false
ExportDataOnly	bool	экспортировать только данные, т.е. только компоненты, расположенные на бэндах данные; по умолчанию false
ExportPageBreaks	bool	экспортировать разделители страниц; по умолчанию false
ExportObjectFormatting	bool	экспортировать форматирование объектов; по умолчанию true
ExportEachPageToSheet	bool	экспортировать каждую страницу отчета в отдельный лист; по умолчанию false

© 2003-2026 Stimulsoft

Информация

Свойство **ExportObjectFormatting** работает только если **ExportDataOnly = true**.

Статические параметры

Статические свойства экспорта в **Excel 2007**. Свойства имеют глобальное влияние на все экспорты в формат **Excel 2007**.

Название	Тип	Описание
AllowImageComparer	bool	использовать оптимизатор изображений, т.е. удалять дубликаты изображений (см. Common export settings); если false - изображения передаются как есть; по умолчанию true
ColumnsRightToLeft	bool	установить порядок расположения колонок справа налево; по умолчанию false
MaximumSheetHeight	int	максимальное количество строк на листе; лишние строки переносятся на следующий лист; по умолчанию 1048574
RemoveEmptySpaceAtBottom	bool	вырезать пустое пространство снизу страницы; по умолчанию true

© 2003-2026 Stimulsoft

14.4.5.3 ODS

Open Document Spreadsheet (ODS) - открытый формат хранения документов табличного процессора **OpenOffice Calc**, который входит в состав пакета **OpenOffice.org**

OpenOffice.org — это свободный пакет офисных приложений, разработанный с целью предоставить альтернативу Microsoft Office как на уровне форматов, так и на уровне интерфейса пользователя. Одним из первых стал поддерживать новый открытый формат OpenDocument. Работает на платформах Microsoft Windows и UNIX-подобных платформах: GNU/Linux, Mac OS X, FreeBSD, Solaris, Irix.

OpenDocument Format (ODF) — открытый формат файлов документов для хранения и обмена редактируемыми офисными документами, в том числе текстовыми документами (такими как заметки, отчеты и книги), электронными таблицами, рисунками, базами данных, презентациями. Формат основан на XML-формате. Стандарт был совместно и публично разработан различными организациями, доступен для всех и может быть использован без ограничений.

OpenOffice Calc — табличный процессор, входит в состав OpenOffice и является свободным программным обеспечением (выпускается под лицензией LGPL). Calc похож на **Microsoft Excel** и функциональность этих табличных процессоров примерно равна. **Calc** позволяет сохранять документы в различных форматах, включая **Microsoft Excel**, **CSV**, **HTML**, **SXC**, **DBF**, **DIF**, **UOF**, **SLK**, **SDC**. Начиная с версии **OpenOffice 2.0**, для хранения документов форматом по умолчанию используется OpenDocument Format, файлы сохраняются с расширением «**.ods**».

Параметры экспорта

Параметры экспорта в формат **ODS** описываются в классе **StiOdsExportSettings**. Ниже приводится таблица с описанием всех свойств этого класса.

Название	Тип	Описание
ImageQuality	float	качество изображений; может принимать значение от 0.0 (низкое качество) до 1.0 (высокое качество); по умолчанию

© 2003-2026 Stimulsoft

		0.75
ImageResolution	float	разрешение изображений, точек на дюйм; может принимать любое значение, по умолчанию 100

Статические параметры

Статические свойства экспорта в **ODS**. Свойства имеют глобальное влияние на все экспорты в формат **ODS**. Для доступа к свойствам нужно добавлять префикс **StiOptions.Export.Ods...** например, **StiOptions.Export.Ods.AllowImageComparer**.

Название	Тип	Описание
AllowImageComparer	bool	использовать оптимизатор изображений, т.е. удалять дубликаты изображений (см. Common export settings); если false - изображения передаются как есть; по умолчанию true
DivideSegmentPages	bool	разбивать сегментированные страницы на отдельные страницы выбранного формата; если false - страницы передаются как есть, без разбивки на части; по умолчанию true
RemoveEmptySpaceAtBottom	bool	вырезать пустое пространство снизу страницы; по умолчанию true

© 2003-2026 Stimulsoft

14.4.6 Данные

Группа форматов файлов, которые являются форматом для хранения табличных данных.

- CSV
- DBF
- XML
- DIF
- SYLK

14.4.6.1 CSV

Формат **CSV** (**Comma Separated Values** — значения, разделенные запятыми) — это текстовый формат, предназначенный для представления табличных данных. Каждая строка файла — это одна строка таблицы. Значения отдельных колонок разделяются разделительным символом (delimiter), который зависит от региональных настроек. Значения, которые содержат зарезервированные символы (такие как запятая или новая строка), обрамляются символом двойные кавычки ("); если в значении встречаются кавычки — они представляются в файле в виде двух кавычек подряд.

Информация

В формат CSV экспортируются только данные, т.е. только те компоненты, которые расположены на бэндах данные. Если свойство **SkipColumnHeaders** в настройках экспорта установлено в **false**, то дополнительно первой строкой экспортируются заголовки/названия колонок.

Параметры экспорта

Параметры экспорта в формат **CSV** описываются в классе **StiCsvExportSettings**. Ниже приводится таблица с описанием всех свойств этого класса.

Название	Тип	Описание
Separator	string	задает символ - разделитель списка, используемый при

© 2003-2026 Stimulsoft

		экспорте; по умолчанию CurrentCulture.TextInfo.ListSeparator
Encoding	Encoding	кодировка текстового файла; по умолчанию Encoding.UTF8
SkipColumnHeaders	bool	пропускать заголовки колонок; по умолчанию false

Статические параметры

Статические свойства экспорта в **CSV**. Свойства имеют глобальное влияние на все экспорты в формат **CSV**. Для доступа к свойствам нужно добавлять префикс **StiOptions.Export.Csv...** например, **StiOptions.Export.Csv.ForcedSeparator**.

Название	Тип	Описание
ForcedSeparator	string	принудительно задает разделительный символ, используемый при экспорте; если задана пустая строка - используется символ из настроек экспорта; по умолчанию пустая строка

14.4.6.2 DBF

Формат **DBF (DataBase File)** - формат хранения данных, используемый в качестве одного из стандартных способов хранения и передачи информации системами управления базами данных, электронными таблицами и т. д. **DBF**-файл делится на заголовок, в котором хранится информация о структуре базы и количестве записей, и область данных, представляющую собой последовательно организованную таблицу из записей фиксированной длины. В свою очередь записи делятся на поля также фиксированной длины, которые хранятся в файле непосредственно друг за другом, образуя запись.

© 2003-2026 Stimulsoft

Информация

В формат **DBF** экспортируются только данные, т.е. только те компоненты, которые расположены на бэндах Данные.

Управление экспортом

В поле Tag каждого текстового компонента можно указать следующие элементы, которые управляют экспортом:

- ТипДанных [: ДлинаПоля [: ДлинаДробнойЧасти]]
- ТипЭкспорта : "НазваниеПоля"
- Column: "НазваниеПоля" "СтрокаДанных"

Несколько элементов должны разделяться точкой с запятой. Элемент “Тип данных” может быть только один и должен располагаться первым, остальные элементы – по необходимости. Возможные значения элемента "ТипДанных" приведены в таблице. Если тип данных не задан, то по умолчанию принимается тип данных string. Элемент "ДлинаПоля" задает фиксированную ширину поля данных. Если ширина поля не задана – принимается по умолчанию из таблицы. Для типа string по умолчанию принимается ширина самой длинной строки. Элемент "ДлинаДробнойЧасти" задает число знаков после запятой. Если не задано – принимается по умолчанию из таблицы.

Тип данных	Тип данных DBF (размер по умолчанию)	Описание
int	Numeric (15 : 0)	Целочисленный
long	Numeric (25 : 0)	Целочисленный
float	Numeric (15 : 5)	Дробный
double	Numeric (20 : 10)	Дробный
string	Character (auto)	Текстовый
date	Date (8)	Дата

Примеры использования элемента приведены в следующей таблице.

© 2003-2026 Stimulsoft

Тип данных	Описание
string : 25	задает ширину колонки 25 символов и обрезает все длинные строки
float	преобразует в дробное число длиной 15 знаков, 5 знаков после запятой
float :10	преобразует в дробное число длиной 10 знаков, 5 знаков после запятой
float :10 : 2	преобразует в дробное число длиной 10 знаков, 2 знаков после запятой
int :10 : 2	преобразует в целое число длиной 10 знаков; второй параметр игнорируется

Информация

Если целая часть числа не вмещается в указанное поле, она обрезается, т.е. данные теряются. Например, если записать число «-12345,678» в поле «float:8:3», то получится «2345,678».

Элемент "ТипЭкспорта" указывает, для какого экспорта задается название поля. Допустимые значения: “dbf”, “csv”, “xml”, “default”. Элемент "НазваниеПоля" определяет, как будет называться поле в файле (для DBF строка автоматически обрезается до 10 символов). Название можно указывать для каждого типа экспорта свое. Если название для какого-то экспорта не указано, то берется название для типа “default”; если его нет - берется название компонента. Например:

Пример

```
...
DBF : "Describe" ; XML : "Description" ; default: "Default name"
...
```

Элемент "Column" указывает, что в экспортируемые данные добавляется дополнительное поле. Элемент "НазваниеПоля" определяет название нового поля. Элемент "СтрокаДанных" определяет содержимое нового поля и может

© 2003-2026 Stimulsoft

быть выражением. Например:

Пример

```
...
Column: "SortField" "{Products.Categories.CategoryName}"
...
```

Параметры экспорта

Параметры экспорта в формат **DBF** описываются в классе **StiDbfExportSettings**. Ниже приводится таблица с описанием всех свойств этого класса.

Название	Тип	Описание
CodePage	StiDbfCodePages	кодовая страница файла; по умолчанию StiDbfCodePages.Default

14.4.6.3 XML

Формат **XML** (eXtensible Markup Language - расширяемый язык разметки) - текстовый формат, предназначенный для хранения структурированных данных (взамен существующих файлов баз данных), для обмена информацией между программами, а также для создания на его основе более специализированных языков разметки (например, **XHTML**), иногда называемых словарями. **XML** — это иерархическая структура, предназначенная для хранения любых данных, визуально структура может быть представлена как дерево. **XML** поддерживает **Unicode** и другие кодировки.

Информация

В формат **XML** экспортируются только данные, т.е. только те компоненты, которые расположены на датабэндах.

Управление экспортом

В поле **Tag** каждого текстового компонента можно указать следующие

© 2003-2026 Stimulsoft

элементы, которые управляют экспортом в формате **XML**:

- ТипДанных.
- ТипЭкспорта : "НазваниеПоля".
- Column: "НазваниеПоля" "СтрокаДанных".

Несколько элементов должны разделяться точкой с запятой. Элемент “Тип данных” может быть только один и должен располагаться первым, остальные элементы – по необходимости. Возможные значения элемента "ТипДанных" приведены в таблице. Если тип данных не задан, то по умолчанию принимается тип данных **string**.

Тип данных	Описание
int	Целочисленный
long	Целочисленный
float	Дробный
double	Дробный
string	Текстовый
date	Дата

Элемент "ТипЭкспорта" указывает, для какого экспорта задается название поля. Допустимые значения: “**dbf**”, “**csv**”, **“xml”**, **“default”**. Элемент "НазваниеПоля" определяет, как будет называться поле в файле. Название можно указывать для каждого типа экспорта свое. Если название для какого-то экспорта не указано, то берется название для типа “**default**”; если его нет - берется название компонента. Например:

Пример

```
...
DBF : "Describe" ; XML : "Description" ; default: "Default name"
...
```

Элемент "**Column**" указывает, что в экспортируемые данные добавляется дополнительное поле. Элемент "НазваниеПоля" определяет название нового поля. Элемент "СтрокаДанных" определяет содержимое нового поля и может быть выражением. Например:

© 2003-2026 Stimulsoft

Пример

```
...
Column: "SortField" "{Products.Categories.CategoryName}"
...
```

14.4.6.4 DIF

Формат **DIF (Data Interchange Format)** - это текстовый формат, предназначенный для обмена отдельными листами между электронными таблицами (**Microsoft Excel**, **OpenOffice.org Calc**, **Gnumeric**, **StarCalc**, **Lotus 1-2-3**, **FileMaker**, **dBase**, **Framework**, **Multiplan**, и т.д.). Ограничение данного формата в том, что формат **DIF** может содержать только один лист в одной книге.

Параметры экспорта

Параметры экспорта в формат **DIF** описываются в классе **StiDifExportSettings**. Ниже приводится таблица с описанием всех свойств этого класса.

Название	Тип	Описание
ExportDataOnly	bool	экспортировать только данные, т.е. только компоненты, расположенные на бэндах данные; по умолчанию false
Encoding	Encoding	кодировка файла; по умолчанию Encoding.ASCII
UseDefaultSystemEncoding	bool	использовать кодировку, установленную по умолчанию в системе; если false - использовать кодировку, задаваемую свойством Encoding; по умолчанию true

© 2003-2026 Stimulsoft

14.4.6.5 SYLK

Формат **SYLK (Symbolic Link)** - это текстовый формат, разработанный **Microsoft** для обмена данными между приложениями, преимущественно между табличными процессорами. Файлы формата **SYLK** имеют расширение «**.slk**». Microsoft не публиковала спецификацию этого формата, поэтому реализация работы с ним в разных приложениях может отличаться.

Информация

Файл **SYLK** может быть записан в кодировке **Unicode** и прочитан некоторыми приложениями в этой кодировке; но все равно большинство приложений (даже поддерживающие **Unicode**) записывают файл **SYLK** в текущей системной кодовой странице **ANSI**, а не в **Unicode**. Поэтому символы, которые не имеют представления в системной кодовой странице, буду записаны в виде знаков вопроса ('**?**').

Параметры экспорта

Параметры экспорта в формат **SYLK** описываются в классе **StiSylkExportSettings**. Ниже приводится таблица с описанием всех свойств этого класса.

Название	Тип	Описание
ExportDataOnly	bool	экспортировать только данные, т.е. только компоненты, расположенные на бэндах данные; по умолчанию false
Encoding	Encoding	кодировка файла; по умолчанию Encoding.ASCII
UseDefaultSystemEncoding	bool	использовать кодировку, установленную по умолчанию в системе;

© 2003-2026 Stimulsoft

		если false - использовать кодировку, задаваемую свойством Encoding; по умолчанию true

14.4.7 Изображения

Группа экспортов в графические форматы. Все графические форматы может разделить на два типа - растровые изображения, и векторные изображения. Также следует отметить, что экспорт монохромных изображений поддерживается только для форматов **BMP**, **GIF**, **PCX**, **PNG**, **TIFF**. Соответственно свойство **DitheringType** работает только для этих экспортов.

Параметры экспорта

Все экспорты изображений имеют одинаковые настройки экспортов. Они сведены в таблице ниже. Однако, класс **ExportSettings** каждый из форматов имеет свой. Для форматов **BMP**, **GIF**, **PNG**, **TIFF**, **JPEG**, **PCX** и **EMF**, классы **StiBmpExportSettings**,**StiGifExportSettings**, **StiPngExportSettings**, **StiTiffExportSettings**, **StiJpegExportSettings**, **StiPcxExportSettings** и **StiEmfExportSettings** соответственно.

Название	Тип	Описание
ImageZoom	double	коэффициент масштабирования. По умолчанию установлено значение 1.0, что равно 100% в окне настройки экспорта
CutEdges	bool	обрезать поля страниц; по умолчанию false
ImageFormat	StiImageFormat	формат изображения - цветное, оттенки серого или монохромное; по умолчанию StiImageFormat.Color
MultipleFiles	bool	записывать страницы отчета в отдельные

© 2003-2026 Stimulsoft

		файлы; актуально только для TIFF, так как только формат TIFF умеет сохранять несколько страниц в один файл, остальные форматы всегда сохраняют в отдельные файлы; по умолчанию false
DitheringType	StiMonochromeDitheringType	тип преобразования (дизеринга) исходного изображения для получения монохромного изображения; по умолчанию StiMonochromeDitheringType.FloydSteinberg

14.5 Скрипты

В этом разделе будут рассмотрены возможности связанные с использованием скриптов в генераторе отчетов.

- Язык программирования отчета
- Код отчета

14.5.1 Язык программирования отчета

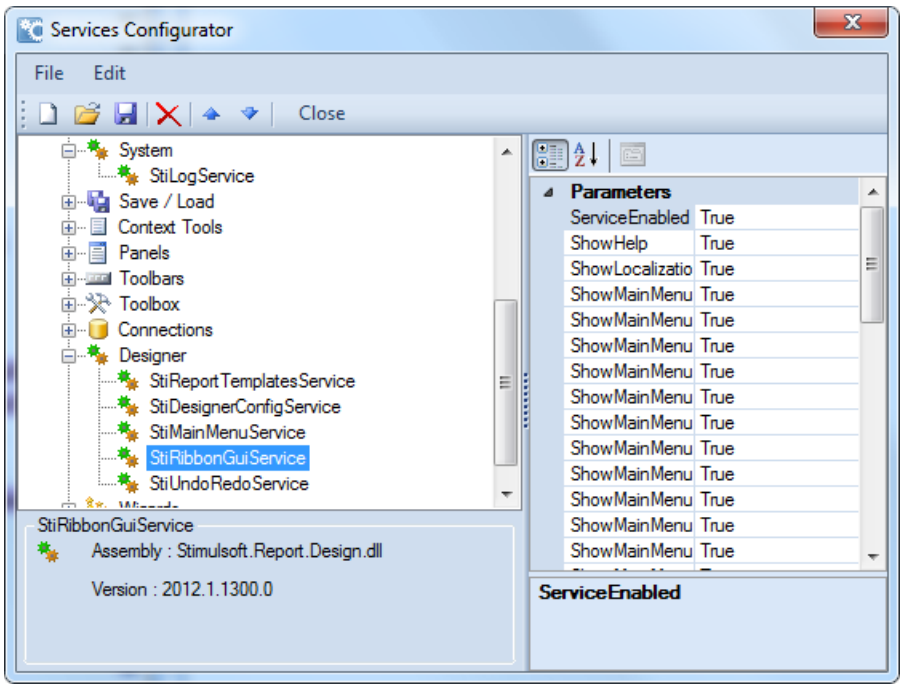
По умолчанию генератор отчетов использует, какой-то один язык программирования для генерации кода отчета. Язык используется для автоматической генерации кода отчета, для написания событий отчета. Если текущий язык отчета, по каким-то причинам не подходит, то Вам его можно изменить. Для этого необходимо выбрать меню **«File | Report Setup»**, выбрать новый язык программирования в группе **«Язык отчета»(«Language»)**. После нажатия на кнопку **«Ok»** текущий язык программирования будет изменен.

© 2003-2026 Stimulsoft

Информация

Будет сгенерирован новый код отчета для всего отчета и все изменения, которые были сделаны в коде отчета, будут потеряны.

Менять каждый раз после создания нового отчета язык программирования неудобно, поэтому можно изменить язык программирования по умолчанию. Для этого необходимо воспользоваться утилитой **Configurator**. Языки программирования доступные в **Stimulsoft Reports** находятся в папке **Languages**. Тот язык, который указан выше в дереве сервисов и является языком программирования по умолчанию. К примеру, на рисунке **C#**, является языком по умолчанию. Если необходимо сделать **VB.Net** языком программирования по умолчанию, то нужно передвинуть сервис **StiCSharpLanguage** на одну позицию вниз.



14.5.2 Код отчета

При создании нового отчета автоматически генерируется его исходный код. Этот код также часто называют скриптом отчета. Для генерации кода используется язык программирования **C#** или **VB.NET**. Одновременно можно использовать только один из указанных языков программирования. В коде

© 2003-2026 Stimulsoft

отчета полностью описывается класс отчета, унаследованный от класса **StiReport**, его структура и инициализация. При добавлении новых страниц, компонентов или изменении любых параметров отчета в класс вносятся изменения. Класс отчета содержит описание всех компонентов, данных, событий, свойств отчета, структуры источников данных. Также в код добавляются все события определенные пользователем. При написании событий или другого кода в отчете допускается использовать стандартный синтаксис языков программирования **.NET Framework**.

Информация

Код отчета генерируется на языке программирования **C#** или **VB.NET**. Соответственно и все события и любой другой код в этом отчете пишется на одном из этих языков.

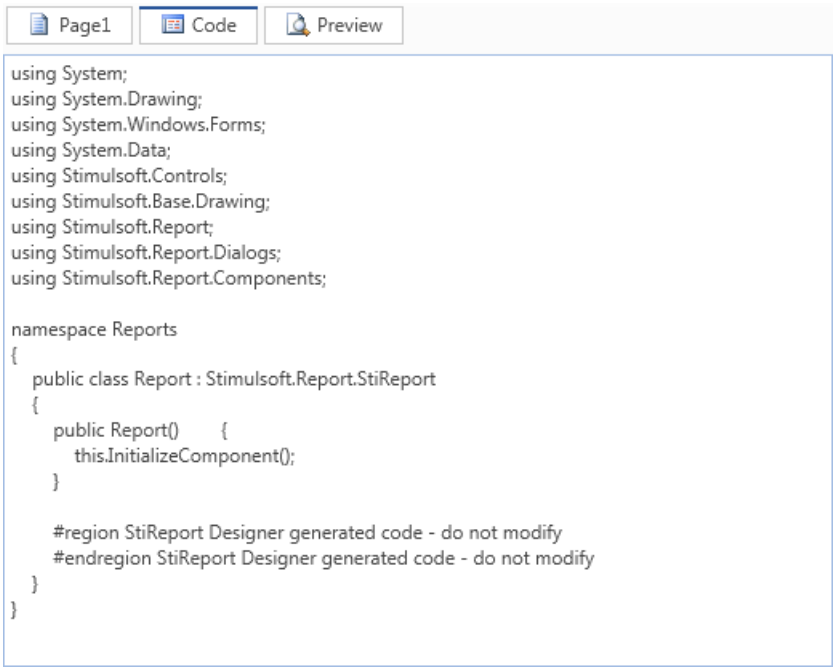
При построении отчета сначала происходит компиляция полученного класса. После чего скомпилированный отчет запускается на выполнение.

ОБРАТИТЕ ВНИМАНИЕ

Код отчета компилируется компилятором **.NET Framework**.

Для того чтобы увидеть код отчета необходимо перейти на закладку **«Code»** в дизайнере:

© 2003-2026 Stimulsoft



Stimulsoft Reports предоставляет возможность напрямую редактировать код отчета. При этом нужно помнить, что вносить изменения в автоматически обновляемый код отчета нельзя. Автоматически обновляемый код заключен в следующие препроцессорные директивы:

VB.Net

```
...
//начало текста
#region StiReport Designer generated code - do not modify

//Автоматически обновляемый код

//конец текста
#endregion StiReport Designer generated code - do not modify
...
```

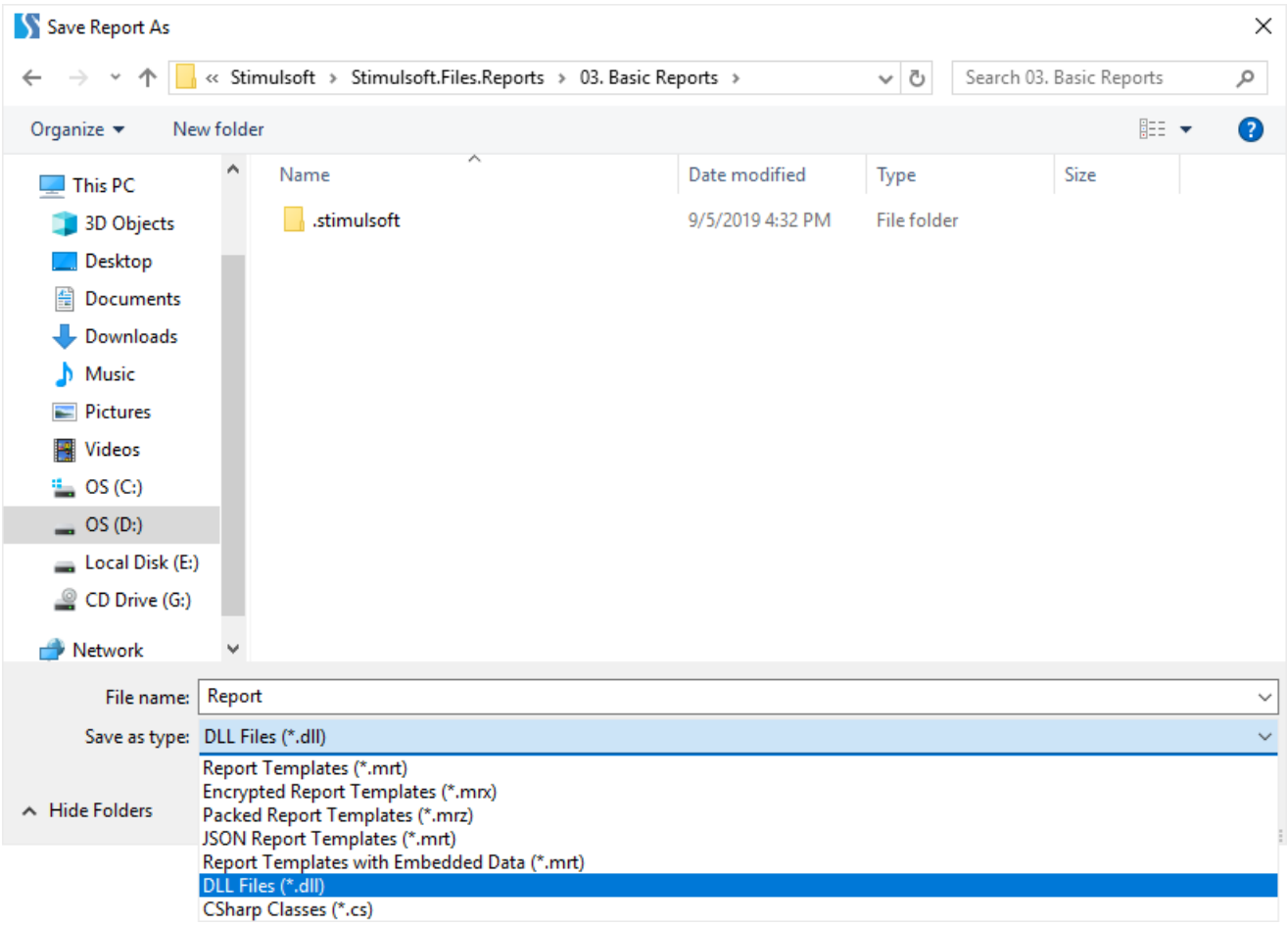
Информация

Не изменяйте препроцессорных директив или автоматически обновляемый код.

© 2003-2026 Stimulsoft

14.6 Отчеты как сборки

Генератор отчетов Stimulsoft Reports предоставляет уникальную возможность - компиляция отчета в сборку **.NET Framework**. При этом, все выражения преобразуются в методы, и их нельзя преобразовать обратно к исходному виду. Для того, чтобы сохранить отчет в сборку можно воспользоваться меню **File | Save Report As...**



Для загрузки отчета из сборки потребуется такой код:

C#

```
...
StiReport report = StiReport.GetReportFromAssembly("report.dll");
...
```

© 2003-2026 Stimulsoft

VB.NET

```
...
Dim Report As StiReport = StiReport.GetReportFromAssembly("report.dll")
...
```

Отчеты загруженные из сборки не требуют компиляции, кроме этого такие отчеты не возможно отредактировать в дизайнере отчетов. А также сохранить их как шаблон отчета.

14.7 Работа со сборками

Возможности создания метода, компилирующего в оперативную память и возвращающего сборку, не существует из-за ограничений **.Net Framework**. Если в приложении **.Net** загружается сборка, то эта сборка блокируется до момента закрытия приложения. Даже если сборка загружается из потока. В этом случае, сборка сохраняется в temp директорию windows, а потом грузится в память. Загрузка сборки отчета (как и любой другой сборки **.Net**) это заблокированный файл сборки на диске в том или ином месте, и занятая для нее память.

Есть некоторые варианты обхода данной проблемы:

› Использовать отчеты в приложении в виде **C#** классов (сохраняя их в виде классов из дизайнера отчетов). В этом случае, отчет компилируется вместе с Вашим приложением, и соответственно ему не требуется компиляция и загрузка сборки. При обновлении, неудобно администрировать отчеты.

› Использовать отчеты в виде скомпилированной сборки, т.е. сохранять отчеты в виде сборок из дизайнера отчетов. В этом случае, также неудобно администрировать отчеты при обновлении.

› Не использовать компиляцию отчетов. Вместо нее использовать интерпретацию выражений в отчете. Режим интерпретации можно включить при помощи свойства отчета **CalculationMode**. Однако, стоит учитывать, что интерпретатор выражений работает медленнее, чем скомпилированные выражения. И не все выражения языка **C#** и **VB.Net** интерпретатор может интерпретировать. Кроме того, нельзя использовать события у компонентов отчета.

› Компиляция и построение отчета в другом домене приложения. В этом

© 2003-2026 Stimulsoft

случае, после построения отчета можно выгрузить его сборку вместе со всем доменом приложения из памяти. Заблокированный файл сборки можно после выгрузки удалить. Для создания отчета можно использовать метод **CreateReportInNewAppDomain()** класса **StiReport**. Для его выгрузки - **UnloadReportAppDomain()**. Однако в этом случае, метод RegData и RegBusinessObject работают очень медленно (т.к. используется маршалинг данных в другом домене). Данные лучше всего запрашивать из самого отчета. Создание и удаление домена приложения занимает определенное время.

› Один из самых эффективных способов - это компиляция отчета при первом обращении к отчету. А при каждом последующем обращении - загрузка отчета из ранее скомпилированной сборки. В данном случае, отчет хранится в памяти только один раз, сборка на диске только один раз. Перед запуском приложения папку, где хранятся отчеты можно удалять (до использования отчетов), т.к. все отчеты будут созданы заново (при обращении к ним). Ниже приводится код, который использует данный способ. Новая сборка отчета создается если ее нет, или если отчет был изменен, а также в случае, если версия **.NET Framework**, для которой была собрана сборка, и версия под которой запущено приложение не совпадают. При этом способе только первый запуск медленный.

Sample

```
...
StiReport report = new StiReport();
report.Load(file);

string folder =
Environment.GetFolderPath(Environment.SpecialFolder.LocalApplicationData);
folder = Path.Combine(folder, "Stimulsoft\\CompiledReports");
folder = Path.Combine(folder,
System.Runtime.InteropServices.RuntimeEnvironment.GetSystemVersion().ToStr
ing());
string compiledReportFile = Path.Combine(folder,
report.GetReportAssemblyCacheName());

if (File.Exists(compiledReportFile))
{
  report = StiReport.GetReportFromAssembly(compiledReportFile, true);
}
else
{
  if (!Directory.Exists(folder)) Directory.CreateDirectory(folder);
  report.Compile(compiledReportFile);
}
report.Render(false);
...
```

© 2003-2026 Stimulsoft

› Прекомпиляция отчетов при их обновлении. Данный способ является вариацией предыдущего. После обновления отчетов в приложении, или если их нет вообще, или если это необходимо, запускается процесс компиляции всех отчетов, которые используются в приложении. Перед компиляцией старая версия сборок удаляется. Иногда процесс компиляции отчетов, при данном способе, может занять некоторое время.

14.8 Автономные отчеты

Создать автономные отчеты можно из дизайнера отчетов, с использованием мастера публикаций отчетов или из кода. Следующий код создаст **standalone** отчет:

C#

```
...
StiReport report = new StiReport();
report.Load("report.mrt");
report.CompileStandaloneReport("report.exe",
StiStandaloneReportType.Show);
...
```

VB.NET

```
...
Dim Report As New StiReport()
Report.Load("report.mrt")
Report.CompileStandaloneReport("report.exe", StiStandaloneReportType.Show)
...
```

Стоит учитывать, что для работы **standalone** отчета потребуются библиотеки **Stimulsoft Reports**.

15 Forms.Web

Stimulsoft Forms – инструмент для создания, редактирования, заполнения, публикации, распространения интерактивных форм и сбора результатов.

15.1 Интеграция с ASP.NET Core 3.1

1. Создайте **ASP.NET Core Empty** приложение;
2. Добавьте пакет **Stimulsoft.Forms.Web** в это приложение используя менеджер пакетов **NuGet**;

© 2003-2026 Stimulsoft

3. В файле проекта **Startup.cs**, настройте приложение для использования контроллера:

Startup.cs

```
public void ConfigureServices(IServiceCollection services)
{
   services.AddControllers();
}

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
   if (env.IsDevelopment())
   {
      app.UseDeveloperExceptionPage();
   }

   app.UseDefaultFiles();
   app.UseStaticFiles();

   app.UseRouting();

   app.UseCors(builder =>
   builder.AllowAnyOrigin().AllowAnyHeader().AllowAnyMethod());

   app.UseEndpoints(endpoints =>
   {
      endpoints.MapControllerRoute(
         name: "default",
         pattern: "{controller=Forms}/{action=Action}");
   });
}
```

4. Создайте папку **Controllers**;
5. Добавьте новый контроллер **FormsController.cs** в папку **Controllers**, в котором будут обрабатываться запросы от **Stimulsoft Forms** компонента:

© 2003-2026 Stimulsoft

FormsController.cs

```
public class FormsController : Controller
{
   [HttpPost]
   public IActionResult Action()
   {
      try
      {
         var data = JObject.Parse(this.HttpContext.Request.Form["data"]);
         var action = data["action"].ToString();
         switch (action)
         {
            case "Initialize":
               var initData = StiWebForm.Initialize(data, null);
               return Json(initData.Content);

            default:
               var result = StiWebForm.ProcessRequest(data);
               return result.ContentType switch
               {
                  "application/pdf" => new FileContentResult(result.Content as
                  byte[], result.ContentType),
                  _ => Json(result.Content),
               };
         }
      }
      catch (Exception e)
      {
         return new ContentResult()
         {
            Content = e.Message,
            ContentType = "text/plain"
         };
      }
   }
}
```

6. Внутри папки приложения, выполните команду в консоли для создания нового **Angular** приложения без `routing` и CSS стилей:

console

```
ng new sti-forms-designer --routing false --style css
```

7. Переименуйте папку **sti-forms-designer** в **ClientApp**;
8. В файле **package.json**, добавьте зависимости для **Stimulsoft Forms** и

© 2003-2026 Stimulsoft

необходимых компонентов:

package.json

```
"stimulsoft-forms": "^2023.2.4",
"ngx-color-picker": "^13.0.0",
"primeicons": "^6.0.1",
"primeng": "^14.1.2",
"resize-observer-polyfill": "^1.5.1",
"rxjs": "~6.5.0"
```

9. В файле **angular.json**, добавьте пути стилей и увеличьте максимальный допустимый размер `budgets`:

angular.json

```
"build": {
  "options": {
    "styles": [
      "src/styles.css",
      "node_modules/primeicons/primeicons.css",
      "node_modules/primeng/resources/primeng.min.css",
      "node_modules/stimulsoft-forms/theme.css"
    ]},

    "configurations": {
      "production": {
        "budgets": [
          {
            "type": "initial",
            "maximumWarning": "500kb",
            "maximumError": "5mb"
          }]
        }
      }
    }
  }
}
```

10. В файле **ClientApp\src\app\app.module.ts** добавьте необходимые импорты:

© 2003-2026 Stimulsoft

app.module.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { FormsModule } from "@angular/forms";
import { HttpClientModule } from "@angular/common/http";
import { BrowserAnimationsModule } from "@angular/platform-browser/
animations";
import { DropdownModule } from "primeng/dropdown";
import { StimulsoftFormsModule } from 'stimulsoft-forms';
import { AppComponent } from './app.component';


@NgModule({
   declarations: [
      AppComponent
   ],
   imports: [
      BrowserModule,
      HttpClientModule,
      FormsModule,
      BrowserAnimationsModule,
      DropdownModule,
      StimulsoftFormsModule
   ],
   providers: [],
   bootstrap: [AppComponent]
})
```

11. В файле **ClientApp\src\app\app.component.ts**, добавьте импорт `StimulsoftFormsService`, определите параметры компонента и добавьте конструктор в класс `AppComponent`:

© 2003-2026 Stimulsoft

app.component.ts

```
import { Component } from '@angular/core';
import { StimulsoftFormsService } from 'stimulsoft-forms';

@Component({
   selector: 'app-root',
   template: `
      <stimulsoft-forms
         [requestUrl]="'http://localhost:7536/Forms/Action'" /*URL to Forms
         controller, you can find this url in launchSettings.json file of
         the project*/
         [form]="form" /*StiForm object to use to create form object using
         StimulsoftFormsService*/
         [style.width]="'100%'"
         [style.height]="'100%'">
      </stimulsoft-forms>
`
})

export class AppComponent {
   public form!: any;

   constructor(public formService: StimulsoftFormsService) {
      this.form = this.formService.createElement("Form");
   }
}
```

12. Измените **ClientApp\src\app\index.html** файл, добавив настройки стилей для тега `<html>` и `<body>`:

index.html

```
<!doctype html>
<html lang="en" style="position: fixed;width:100%;height:100%">
<head>
   <meta charset="utf-8">
   <title>StiFormsDesigner</title>
   <base href="/">
   <meta name="viewport" content="width=device-width, initial-scale=1">
   <link rel="icon" type="image/x-icon" href="favicon.ico">
</head>
<body style="width:100%;height:100%">
   <app-root></app-root>
</body>
</html>
```

© 2003-2026 Stimulsoft

13. Перейдите в папку **ClientApp**;
14. Выполните команду для установки **Angular** компонента:

console

```
npm i --force
```

15. Выполните команду сборки **Angular** приложения:

console

```
ng build --output-hashing none
```

16. Добавьте папку **wwwroot** в корень проекта;
17. Скопируйте все файлы из папки **ClientApp\dist\sti-forms-designer** в папку **wwwroot**;
18. Запустите проект.

© 2003-2026 Stimulsoft