Бесплатная электронная книга

УЧУСЬ

jasper-reports

Free unaffiliated eBook created from
Stack Overflow contributors.

#jasper-reports

Около

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: jasper-reports

It is an unofficial and free jasper-reports ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official jasper-reports.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

глава 1: Начало работы с отчетами о яшме

замечания

Существует несколько библиотек, используемых *JasperReports Java API* для создания отчетов с Java:

- **DynamicReports**
- **DynamicJasper**

Эти библиотеки / фреймворки могут создавать отчеты «на лету» с использованием или без использования шаблона отчета (файл *jrxml*)

Версии

Библиотека *JasperReports*

Версия	Дата выхода
6.3.0	2016-06-20
6.2.0	2015-11-11
5.6.0	2014-05-27
5.5.0	2013-10-24
5.0.4	2013-03-26
5.0.0	2012-11-12
4.8.0	2012-11-05
4.7.0	2012-07-02
4.6.0	2012-05-21
4.5.0	2011-12-06
4.1.1	2011-04-18
4.0.0	2010-12-31
3.7.6	2010-10-27
3.7.5	2010-09-22

Версия	Дата выхода
3.7.0	2009-12-08
3.6.0	2009-08-31
3.5.3	2009-07-29
3.5.0	2009-03-25
3.1.4	2009-02-10
3.1.2	2008-11-04
3.1.0	2008-09-17
3.0.1	2008-08-07
3.0.0	2008-05-19
2.0.5	2008-03-12
2.0.3	2007-12-12
2.0.0	2007-08-14
1.3.4	2007-06-11
1.3.0	2006-12-22
1.2.8	2006-11-14
1.2.0	2006-02-06
1.1.0	2005-10-21
1.0.3	2005-10-10
1.0.0	2005-07-20
0.6.8	2005-05-31
0.2.3	2002-02-06

IDE для разработки отчетов

Текущая версия дизайнера основана на *Eclipse* : *Jaspersoft Studio* .

Предыдущая версия конструктора была основана на *NetBeans* : *iReport Desigher* .

Первая версия *iReport Designer* была независимой приложением - *iReport Classic*

Examples

Установка или настройка

Библиотека JasperReports

JasperReports - это инструмент для создания отчетов на основе Java с открытым исходным кодом. Библиотека *JasperReports* может быть загружена из сообщества Jaspersoft для последней версии .

> В последних выпусках сторонние банки в папке lib **не** распространяются, их нужно загружать из общедоступных репозиториев, см. `pom.xml` распределенного `pom.xml` для зависимостей. Maven может использоваться для извлечения всех зависимостей, включая переходные, в папку целевых / зависимых.

```
mvn dependency:copy-dependencies
```

Jaspersoft Studio (IDE)

Jaspersoft Studio является официальным клиентом дизайна для JasperReports - построена на платформе Eclipse - для замены iReport Designer.

iReport Designer (IDE)

iReport Designer - это предыдущий разработчик отчетов для JasperReports. Версия 5.6.0 (выпущенная в мае 2014 года) была последней официальной версией; поддержка поставщиков закончилась в конце 2015 года.

Ресурсы JasperReport Commuity

Часто задаваемые вопросы о библиотеке JasperReports

- Часто задаваемые вопросы

Исходный код

- Исходный код библиотеки JasperReports

Учебники

- Точки обучения
- Руководство пользователя JasperReports

образцы

- Образец ссылки

Рекомендации

- Официальная документация
- Сообщество

Официальный трекер

- Баг трекер

Рабочий поток

Рабочий поток в отчетах яшмы:

1. Создайте отчет, создайте файл jrxml, определяющий макет отчета. Jrxml можно создать с помощью простого текстового редактора, но обычно IDE (JasperSoft Studio или iReport) используется как для ускорения разработки отчетов, так и для визуального представления макета.

2. Скомпилируйте отчет (jrxml), чтобы получить файл .jasper или объект JasperReport . Этот процесс можно сравнить с файлом `.java` , скомпилированным в `.class` .

3. Заполните отчет , передайте параметры и источник данных в отчет для создания объекта печати JasperPrint, который также можно сохранить в файле `.jprint`

4. Просмотрите, распечатайте и / или экспортируйте JasperPrint. Наиболее распространенный формат экспорта поддерживается как pdf, excel, word, html, cvs и т. Д.

Понимание различных диапазонов отчетов

заглавие

Эта группа показана один раз в начале отчета. Его можно использовать как первую страницу, установив атрибут `isTitleNewPage="true"`

Заголовок страницы

Это появляется в начале каждой страницы, исключая первую страницу, если используется полоса заголовка, и последняя страница, если используется `isSummaryWithPageHeaderAndFooter="false"` с настройкой `isSummaryWithPageHeaderAndFooter="false"`

Заголовок столбца

Это отображается перед диапазоном деталей на каждой странице.

подробность

Этот раздел повторяется **для каждой записи** в поставляемом источнике данных. Разрешено иметь несколько полос деталей (деталь 1, деталь 2 .. деталь n), они повторяются следующим образом

```
Row 1
    detail 1
    detail 2
    detail n
Row 2
    detail 1
    detail 2
    detail n
```

Нижний колонтитул

Это появляется ниже диапазона деталей на каждой странице, где присутствует подробная полоса. Значение по умолчанию - конец страницы (до нижнего колонтитула страницы), но это может быть переключено на последний диапазон данных (последняя запись), установив атрибут `isFloatColumnFooter="true"`

Нижний колонтитул страницы

Это отображается внизу каждой страницы, за исключением полосы заголовка, сводной полосы (без нижнего колонтитула страницы) и последней не суммируемой полосы, если используется нижний колонтитул последней страницы.

Последняя нижняя колонтитула страницы

Это отображается на последней странице (если не сводная таблица без нижнего колонтитула страницы) вместо обычного нижнего колонтитула страницы

Резюме

Это появляется в конце отчета на новой странице, если `isSummaryNewPage="true"` задано и с заголовком и нижним колонтитулом страницы, если `isSummaryWithPageHeaderAndFooter="true"`

Заголовок группы

Этот раздел появляется, если группа определяется каждый раз, когда выражение группы изменяется, перед полосой детали.

Нижний колонтитул группы

Этот раздел появляется, если группа определяется каждый раз *до* изменения группового выражения после группы деталей.

Фон

Эта полоса отображается на каждой странице в качестве фона для всех других диапазонов.

Нет данных

Это появляется только в том случае, если источник данных не передан или источник данных пуст (0 записей), а `whenNoDataType="NoDataSection"` .

Форматы файлов отчета Jasper

- `.jrxml` - это файл дизайна отчета, его формат находится в человеко-читаемом XML, его можно выполнить в объект `JasperReport` и сохранить как `.jasper`
- `.jasper` - это скомпилированная версия `.jrxml` и может быть загружена непосредственно в объект `JasperReport` готовый к заполнению данными
- `.jrprint` - это сериализованный объект `JasperPrint` , отчет, который уже заполнен данными и может быть загружен для печати, просмотра и / или экспорта в желаемый формат.

- `.jrpxml` - это XML-представление о объекте `JasperPrint` которое может быть изменено, а затем немаршалировано для извлечения объекта `JasperPrint`

Прочитайте Начало работы с отчетами о яшме онлайн: https://riptutorial.com/ru/jasper-reports/topic/3594/начало-работы-с-отчетами-о-яшме

глава 2: Font-расширения

Examples

Создание и использование расширений шрифтов

Создайте расширение шрифта, используя IDE. Подробнее см. Документацию iReport или Jaspersoft Studio . Расширение шрифта также можно создать вручную .

Что такое расширения шрифтов?

С помощью `textElement` вы можете указать шрифт (если не указан стандартный шрифт `SansSerif`)

```
<textElement>
    <font fontName="DejaVu Sans"/>
</textElement>
```

Чтобы рассчитать шрифт-метрику (для разрывов строк, выравнивания и т. Д.) И правильно отредактировать шрифт, **шрифт** нужно **сопоставить в JVM** (виртуальный macchine Java). Вы можете установить файл шрифта непосредственно в JVM, но это не поощряет

Из руководства по управлению JasperReport Ultimate:

> Мы настоятельно рекомендуем людям использовать только шрифты, полученные из расширений шрифтов, поскольку это единственный способ убедиться, что шрифты будут доступны для приложения, когда отчеты будут выполнены во время выполнения. Использование системных шрифтов всегда приводит к тому, что отчеты не будут работать должным образом при развертывании на новом компьютере, на котором не могут быть установлены эти шрифты

Расширение шрифта по умолчанию

JasperReports предоставляет стандартное расширение шрифта (см. Дистрибутив maven jasperreports-fonts.jar). Добавляя это в classpath, вы можете использовать следующее имя шрифта без создания собственного расширения шрифта

> DejaVu Sans
> DejaVu Serif
> DejaVu Sans Mono

Общие вопросы

Вопросы, которые следует учитывать при использовании шрифта в pdf (itext):

- При экспорте в PDF, если текст не отображается правильно (отсутствующие части, символы не отображаются, а не обертываются или имеют правильный размер), **шрифтовые расширения** , скорее всего, отсутствуют.
- **Поддерживается** ли фактическое `.tff` (OpenType) и может ли шрифт **визуализировать** персонажа? Не все шрифты отображают все символы в `UTF-8` .
- **Правильно ли** передается **кодировка** iText? В сомнениях (или вообще) используется **кодировка** `Identity-H` это рекомендуется для новых стандартов PDF и дает вам возможность смешивать различные кодировки.
- Является ли **встроенный** шрифт таким образом, чтобы общий доступ к файлам PDF на компьютерах мог отображать контент, даже если шрифт не установлен? Если шрифт не входит в один из 14 стандартных шрифтов Type 1, всегда вставляйте его.

Обратите внимание, что версия iText, используемая в отчете jasper, не будет отображать все шрифты (проблема с лигатуратором), вы можете проверить шрифт `ttf` и кодировку напрямую. Как проверить, правильно ли мой шрифт отображается в pdf?

Прочитайте Font-расширения онлайн: https://riptutorial.com/ru/jasper-reports/topic/5773/font-расширения

глава 3: Заполнить отчет

параметры

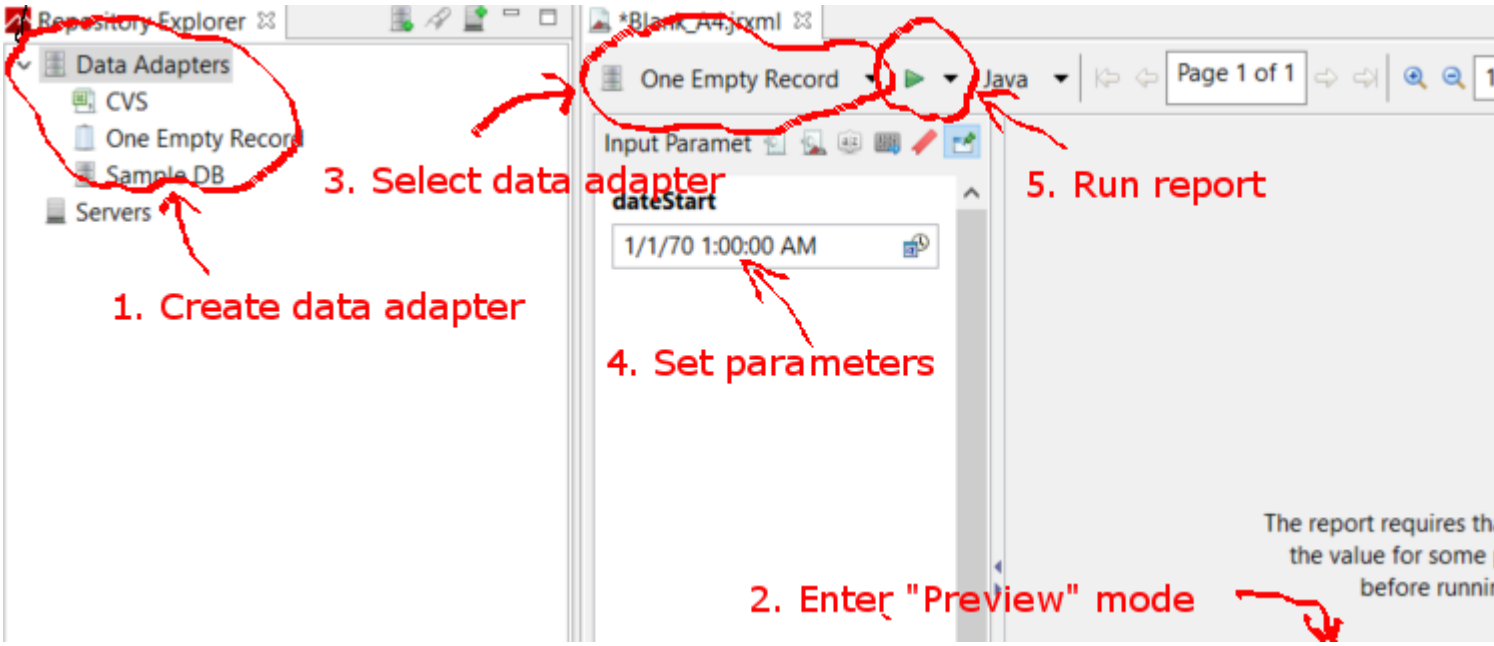
параметры	колонка
jasperPrint	Вывод процесса заполнения, который может быть экспортирован в желаемый формат
reportTemplate	Скомпилированный файл проекта `.jasper`
параметры	Карта параметров, которая, если определена, может быть ссылкой внутри отчета с помощью `$P{key}`
источник данных	Файл net.sf.jasperreports.engine.JRDataSource
соединение	Соединение с базой данных java.sql.Connection

Examples

С интегрированной средой разработки (IDE)

Студия JasperSoft

1. Если для заполнения отчета требуется источник данных или соединение с базой данных, создайте свой адаптер данных в проводнике репозитория, щелкнув правой кнопкой мыши «Адаптеры данных», выбрав «Создать адаптер данных»,
2. Войдите в режим предварительного просмотра, выбрав вкладку « **Предварительный просмотр** » (никаких ошибок в подключении не должно присутствовать)
3. Выберите желаемый источник данных (если нет источника данных, выберите «Одна пустая запись»,
4. Задайте параметр по желанию
5. Заполните отчет, нажав зеленую стрелку «Запустить отчет»



Заполнить шаблон JasperReport с помощью Java

Общие требования

Все отчеты, независимо от того, как представлены данные, проходят путь к шаблону отчета и карте параметров. Переменные используются во всех приведенных ниже примерах:

```
// Parameters passed into the report.
Map<String, Object> parameters = new HashMap<>();

// Arbitrary parameter passed into the report.
parameters.put("KEY", "Value");

// The compiled report design.
String path = "path/to/template.jasper";
```

Использование файла `.jrxml` дополнительного этапа компиляции, который не требуется в большинстве ситуаций. Если вы не написали специальное программное обеспечение для изменения `.jrxml` до `.jrxml` отчета (например, добавление или удаление столбцов динамически), используйте файл `.jasper` как показано в последующих примерах.

Использование подключения к базе данных

```
// Establish a database connection.
Connection connection = DriverManager.getConnection(url, username, password);
```

```
// Fill the report, get the JasperPrint that can be exported to desired format.
JasperPrint jasperPrint = JasperFillManager.fillReport(
  path, parameters, connection);
```

Использование настраиваемого источника данных

```
// Populate this list of beans as per your requirements.
List<Bean> beans = new ArrayList<>();

// Wrap the beans in a beans in a JRBeanCollectionDataSource.
JRBeanCollectionDataSource datasource = new JRBeanCollectionDataSource(beans);

// Fill the report, get the JasperPrint that can be exported to desired format.
JasperPrint jasperPrint = JasperFillManager.fillReport(
  path, parameters, datasource);
```

Без источника данных, неиспользуемый блок данных

```
// Fill the report, get the JasperPrint that can be exported to desired format.
JasperPrint jasperPrint = JasperFillManager.fillReport(path, parameters);
```

> Без данных, атрибут `whenNoDataType="AllSectionsNoDetail"` в элементе `JasperReport` должен быть установлен, иначе будет создан пустой (пустой) отчет.

Прочитайте Заполнить отчет онлайн: https://riptutorial.com/ru/jasper-reports/topic/3958/заполнить-отчет

глава 4: Использование подписок

параметры

параметр	подробности
parametersMapExpression	Карта с параметрами. *Не требуется*
subreportParameter	Пара имени и значения (устанавливается с помощью ***subreportParameterExpression***). *Не требуется* . Несколько параметров могут быть переданы в отчет
connectionExpression	Соединение для получения данных. *Не требуется*
dataSourceExpression	Выражение для передачи Datasource. *Не требуется*
subreportExpression	Путь / URI подрепортажа или даже объект JasperReport. *Не требуется*
ReturnValue	Пара имени и значения. *Не требуется* . Несколько значений могут быть возвращены из подрепортажа в основной отчет

замечания

- Subreports может использоваться для построения сложных отчетов. Повторное использование существующих отчетов - еще одна цель использования подзаголовков.
- Подрепорт будет отображаться как часть основного отчета в случае использования элемента `<subreport>` .
- Значение параметра ***subreportExpression*** отличается для использования на *сервере JasperReports* или только с помощью инфраструктуры *JasperReports* (некоторые *API* используют или используют в среде IDE).

  Для *JasperReports Server* это выглядит так:

  ```
  <subreportExpression><![CDATA["repo:subreport.jrxml"]]></subreportExpression>
  ```

  Для использования только *JasperReports* :

  ```
  <subreportExpression><![CDATA["/somePath/subreport.jasper"]]></subreportExpression>
  ```

Большое объяснение от *@AndreasDietrich* можно найти на *JasperServer: невозможно найти сообщение об исключении подрепортажа*

- По некоторым причинам субрепорт может использоваться как общий отчет - без вызова основного отчета (с помощью элемента `<subreport>`). Подрепорт всегда является отчетом.

Examples

Передача подключения к отчету; вернуть значения в основной отчет

Это фрагмент основного отчета. Два параметра и соединение (например, *jdbc*) передаются в подрегистр. Одно значение возвращается из подчиненного отчета обратно в главный отчет, это значение (*переменная*) может использоваться в основном отчете

```
<subreport>
    <reportElement x="0" y="80" width="200" height="100"/>
    <subreportParameter name="someSubreportParameter">

<subreportParameterExpression><![CDATA[$P{someMasterReportParamter}]]></subreportParameterExpression>

    </subreportParameter>
    <subreportParameter name="anotherSubreportParameter">
        <subreportParameterExpression><![CDATA["Some text - constant
value"]]></subreportParameterExpression>
    </subreportParameter>
    <connectionExpression><![CDATA[$P{REPORT_CONNECTION}]]></connectionExpression>
    <returnValue subreportVariable="someVariableInSubreport"
toVariable="someVariableInMasterReport"/>
    <subreportExpression><![CDATA["$P{SUBREPORT_DIR} +
"subreport.jasper"]]></subreportExpression>
</subreport>
```

Передача данных для отчета

Это фрагмент основного отчета. Источник данных передается в подчиненный отчет с помощью ***net.sf.jasperreports.engine.data.JRBeanCollectionDataSource*** конструктора

```
<field name="someFieldWithList" class="java.util.List"/>
<!-- ...... -->
<subreport>
    <reportElement x="0" y="0" width="200" height="70"/>
    <parametersMapExpression><![CDATA[$P{REPORT_PARAMETERS_MAP}]]></parametersMapExpression>

<dataSourceExpression><![CDATA[net.sf.jasperreports.engine.data.JRBeanCollectionDataSource($F{someFiel

    <subreportExpression><![CDATA[$P{SUBREPORT_DIR} +
"subreport.jasper"]]></subreportExpression>
</subreport>
```

Прочитайте Использование подписок онлайн: https://riptutorial.com/ru/jasper-reports/topic/5452/использование-подписок

глава 5: Компилировать JasperReports .jrxml в .jasper

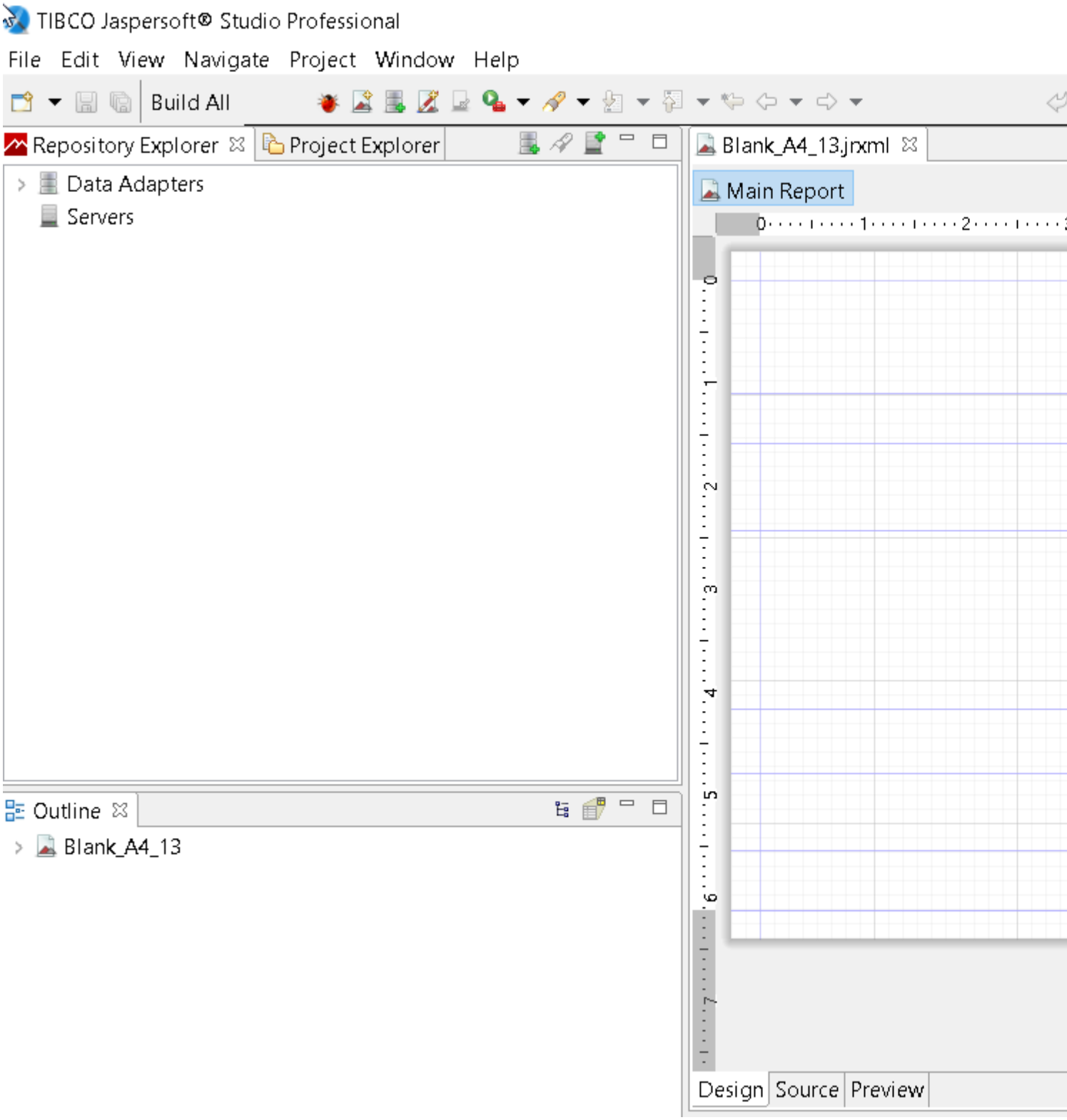
Examples

С интегрированной средой разработки (IDE)

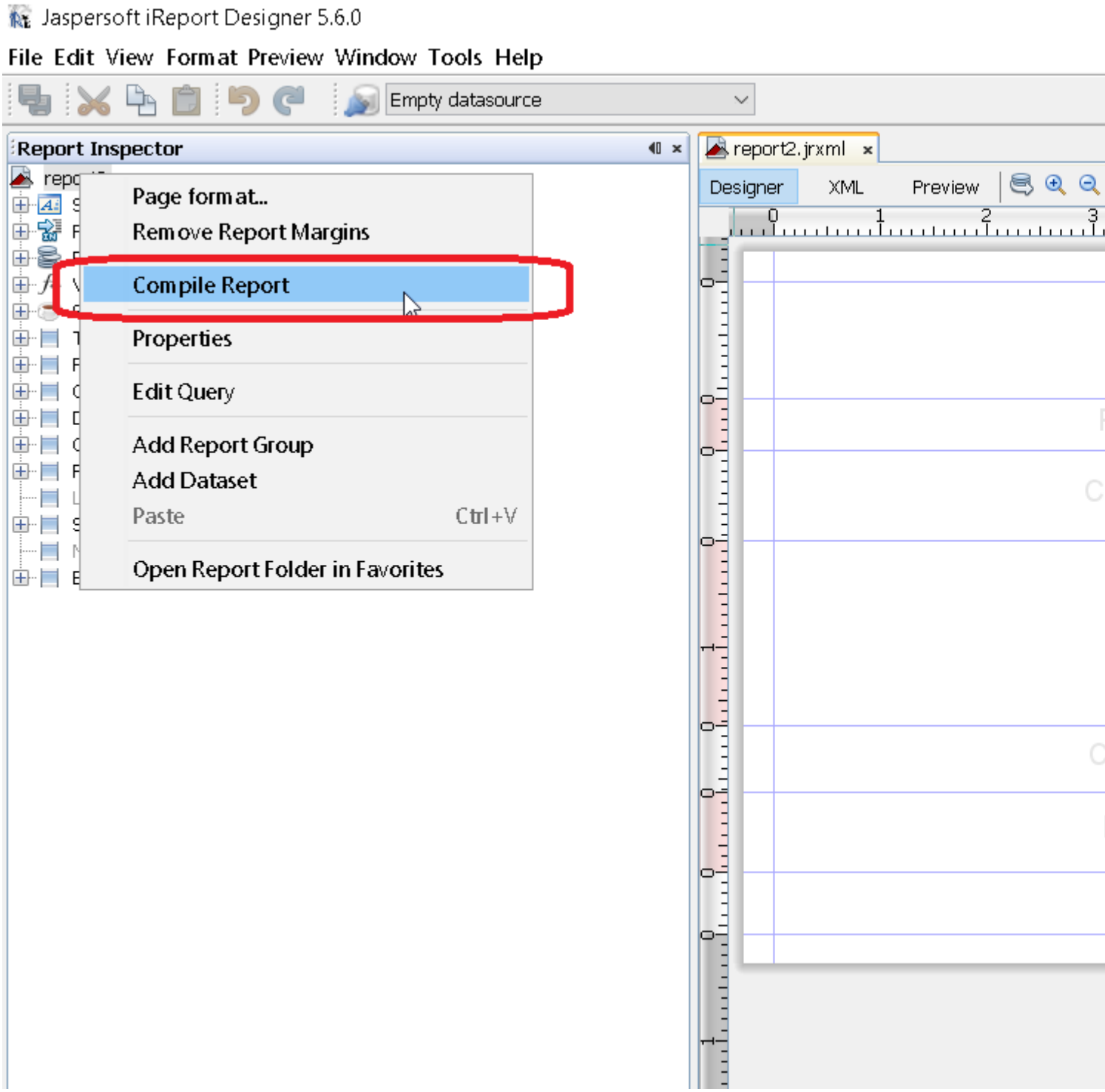
В IDE Jaspersoft Studio (*JSS*) или в более ранней версии iReport Designer достаточно нажать **Preview** .

Файл проекта `.jrxml` проекта JasperReports будет автоматически скомпилирован в `.jasper` в той же папке, что и `.jrxml` если **ошибок** нет.

Другой способ - нажать кнопку *«Скомпилировать отчет»* в *JSS*



или используйте контекстное меню *«Отчет компиляции»*, вызванный из *Report Inspector* в *iReport*



C Apache Ant

```
<target name="compile" description="Compiles report designs specified using the 'srcdir' in
the &lt;jrc&gt; tag." depends="prepare-compile-classpath">
  <mkdir dir="./build/reports"/>
    <taskdef name="jrc" classname="net.sf.jasperreports.ant.JRAntCompileTask">
        <classpath refid="project-classpath"/>
    </taskdef>
    <jrc
            srcdir="./reports"
            destdir="./build/reports"
            tempdir="./build/reports"
            keepjava="true"
```

```
            xmlvalidation="true">
        <classpath refid="sample-classpath"/>
        <include name="**/*.jrxml"/>
    </jrc>
</target>
```

Инструмент сборки Apache Ant должен быть правильно установлен в вашей системе

С Java

Хотя можно компилировать файлы `.jrxml` файлы `.jasper` с помощью Java-кода, это приводит к хиту производительности, который лучше всего избегать путем предварительной компиляции файлов `.jrxml` с использованием среды IDE. Имея это в виду, компиляция файлов `.jrxml` может быть выполнена с помощью JasperCompileManager следующим образом:

```
JasperCompileManager.compileReportToFile(
            "designFile.jrxml", //Relative or absoulte path to the .jrxml file to compile
            "compiled.jasper"); //Relative or absolute path to the compiled file .jasper
```

С Apache Maven

Плагин JasperReports от Alex Nederlof является хорошей альтернативой заброшенному org.codehaus.mojo: плагину jasperreports-maven- plugin.

Добавление плагина - типичная простая процедура:

```
<build>
    <plugins>
        <plugin>
            <groupId>com.alexnederlof</groupId>
            <artifactId>jasperreports-plugin</artifactId>
            <version>2.3</version>
            <executions>
                <execution>
                    <phase>process-sources</phase>
                    <goals>
                        <goal>jasper</goal>
                    </goals>
                </execution>
            </executions>
            <configuration>
                <sourceDirectory>src/main/resources/jrxml</sourceDirectory>
                <outputDirectory>${project.build.directory}/jasper</outputDirectory>
            </configuration>
        </plugin>
    </plugins>
</build>
```

Команда для компиляции с *Maven* :

mvn jasperreports: jasper

Файлы *jasper* будут созданы в папке *$ {project.build.directory} / jasper* (например, в */ target / jasper*)

Прочитайте Компилировать JasperReports .jrxml в .jasper онлайн:
https://riptutorial.com/ru/jasper-reports/topic/4943/компилировать-jasperreports--jrxml-в--jasper

глава 6: Экспорт в pdf

замечания

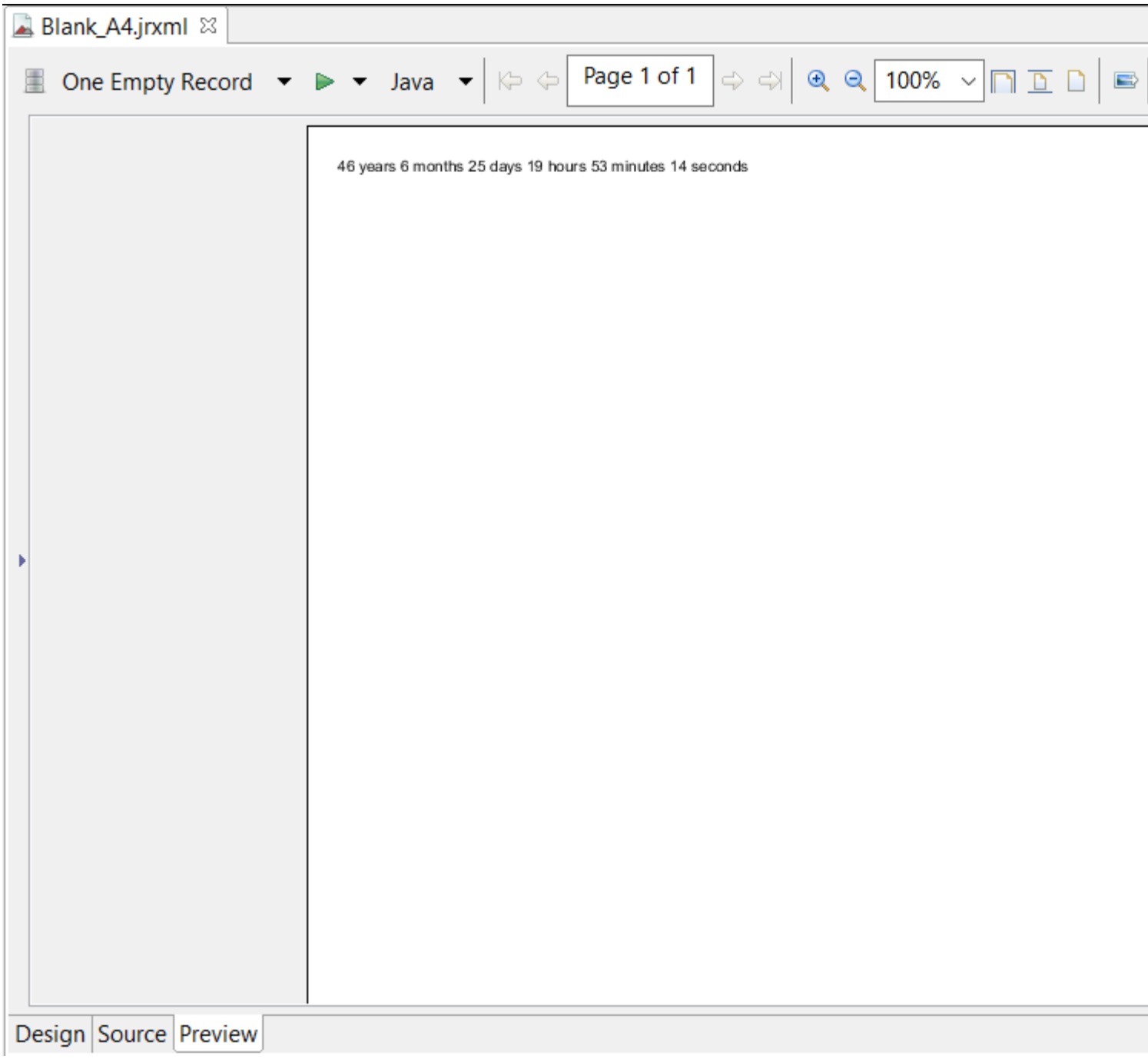
Для правильного отображения **шрифтов** в pdf шрифтовые расширения всегда должны использоваться (в classpath)

Examples

С интегрированной средой разработки (IDE)

Студия JasperSoft

В окне «Предварительный просмотр» запустите отчет, нажав зеленую стрелку, если нет ошибок в меню экспорта, нажмите кнопку «Экспорт» (образ диска) и выберите «Экспорт как Pdf»,



С Java

Для экспорта вам необходимо заполнить отчет, чтобы получить объект JasperPrint .

Экспорт одного JasperPrint (одного jrxml) в файл

```
// 1. Create exporter instance
JRPdfExporter exporter = new JRPdfExporter();

// 2. Set exporter input document
```

```
exporter.setExporterInput(new SimpleExporterInput(jasperPrint));

// 3. Set file path for exporter output
exporter.setExporterOutput(new SimpleOutputStreamExporterOutput("/path/filename.pdf"));

// 4. Create configuration instance
SimplePdfExporterConfiguration configuration = new SimplePdfExporterConfiguration();

// 5. Associate configuration with exporter
exporter.setConfiguration(configuration);

// 6. Fill export and write to file path
exporter.exportReport();
```

Экспортировать несколько JasperPrint (несколько jrxml) в один файл

Только первые шаги отличаются от предыдущих:

```
List<JasperPrint> jasperPrintList = new ArrayList<>();
jasperPrintList.add(jasperPrint1);
jasperPrintList.add(jasperPrint2);

JRPdfExporter exporter = new JRPdfExporter();
exporter.setExporterInput(SimpleExporterInput.getInstance(jasperPrintList));
```

Остальные шаги одинаковы:

```
exporter.setExporterOutput(new SimpleOutputStreamExporterOutput("/path/filename.pdf"));
SimplePdfExporterConfiguration configuration = new SimplePdfExporterConfiguration();
exporter.setConfiguration(configuration);
exporter.exportReport();
```

См. API-интерфейс SimplePdfExporterConfiguration для получения сведений о конфигурации.

Прочитайте Экспорт в pdf онлайн: https://riptutorial.com/ru/jasper-reports/topic/4190/экспорт-в-pdf

глава 7: Экспорт в xls / xlsx

Examples

C Java

Экспорт в формат *xlsx*

```
try (InputStream inputStream = JRLoader.getResourceInputStream(path)) {  // read report as
input stream
    JasperReport  jasperReport =
JasperCompileManager.compileReport(JRXmlLoader.load(inputStream)); // compile report

    Map<String, Object> params = new HashMap<>(); // init map with report's parameters
    params.put(JRParameter.REPORT_LOCALE, Locale.US);
    params.put(JRParameter.IS_IGNORE_PAGINATION, true);
    JasperPrint jasperPrint = JasperFillManager.fillReport(jasperReport, params, connection);
// prepare report - passs parameters and jdbc connection

    JRXlsxExporter exporter = new JRXlsxExporter(); // initialize exporter
    exporter.setExporterInput(new SimpleExporterInput(jasperPrint)); // set compiled report as
input
    exporter.setExporterOutput(new SimpleOutputStreamExporterOutput(destFile));  // set output
file via path with filename
    SimpleXlsxReportConfiguration configuration = new SimpleXlsxReportConfiguration();
    configuration.setOnePagePerSheet(true); // setup configuration
    configuration.setDetectCellType(true);
    exporter.setConfiguration(configuration); // set configuration
    exporter.exportReport();
}
```

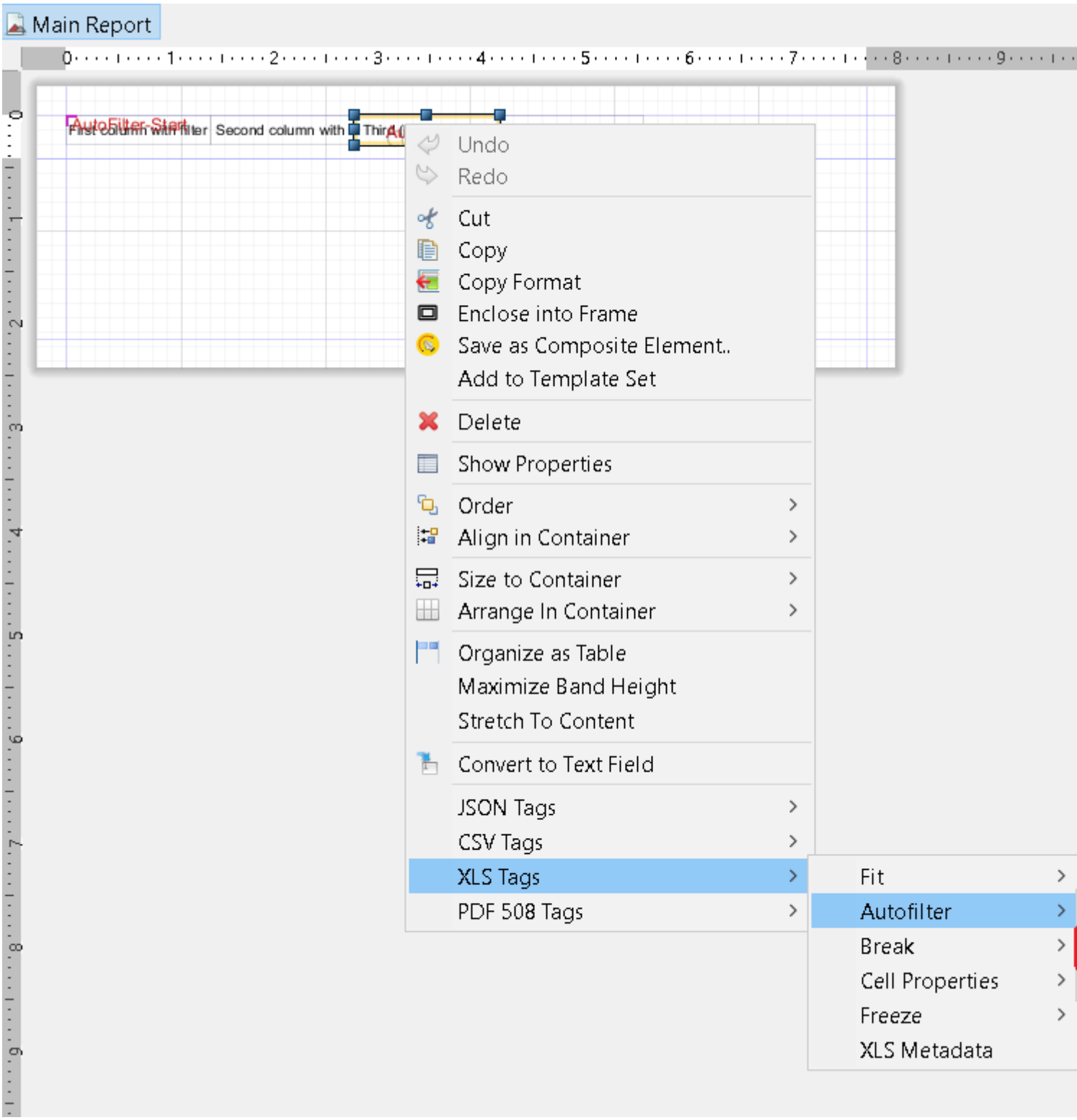
Добавление автофильтра для столбцов

Использование свойства ***net.sf.jasperreports.export.xls.auto.filter*** позволяет добавить автофильтр в сгенерированный файл xls.

```
<columnHeader>
    <band height="30" splitType="Stretch">
        <staticText>
            <reportElement x="0" y="0" width="100" height="20">
                <property name="net.sf.jasperreports.export.xls.auto.filter" value="Start"/>
            </reportElement>
            <text><![CDATA[First column with filter]]></text>
        </staticText>
        <staticText>
            <reportElement x="100" y="0" width="100" height="20"/>
            <text><![CDATA[Second column with filter]]></text>
        </staticText>
        <staticText>
            <reportElement x="200" y="0" width="100" height="20">
                <property name="net.sf.jasperreports.export.xls.auto.filter" value="End"/>
            </reportElement>
            <text><![CDATA[Third (Last) column with filter]]></text>
```

```
        </staticText>
        <staticText>
            <reportElement x="300" y="0" width="100" height="20"/>
            <text><![CDATA[Fourth column without filter]]></text>
        </staticText>
    </band>
</columnHeader>
```

Свойство можно установить в *Jaspersoft Studio* с помощью контекстного меню или вручную, отредактировав файл *jrxml* .



Прочитайте Экспорт в xls / xlsx онлайн: https://riptutorial.com/ru/jasper-reports/topic/5008/экспорт-в-xls---xlsx

кредиты

S. No	Главы	Contributors
1	Начало работы с отчетами о яшме	Alex K, Community, Dave Jarvis, Petter Friberg
2	Font-расширения	Dave Jarvis, Petter Friberg
3	Заполнить отчет	Alex K, Dave Jarvis, Petter Friberg
4	Использование подписок	Alex K
5	Компилировать JasperReports .jrxml в .jasper	Alex K, Dave Jarvis, Petter Friberg
6	Экспорт в pdf	Alex K, Dave Jarvis, Petter Friberg, RamenChef
7	Экспорт в xls / xlsx	Alex K