

Базовый шаблон отчёта с JasperReports

Для построения отчётов в программе [АРМ НЕЙРОСС Отчёты](#) с помощью JasperReports Library достаточно указать в своём наборе шаблонов в файле `reports.conf` значение параметра `definition.class` в `extensions.reports`.

GenericJasperReportDefinition. Это имя программного компонента (Java-класса) базовой реализации шаблона отчёта, который позволяет формировать отчёт средствами JasperReports Library по шаблону в форме *.jasper-файла.

Базовая реализация шаблона:

1. Анализирует *.jasper-файл шаблона отчёта и получает из него список параметров, которые должны быть заданы при формировании отчёта:
 - a. в список параметров попадают те, для которых в Jasper-шаблоне задан флаг Is For Prompting;
 - b. описание параметра (поле Description в Jasper-шаблоне) используется для текста надписи в форме ввода параметров;
 - c. тип данных параметра определяется свойством Class в Jasper-шаблоне.
2. Для каждого параметра выбирает оптимальное поле ввода в соответствии с заданными для параметра свойствами (см. [Привязка полей ввода](#)).
3. Генерирует форму ввода параметров из соответствующих полей ввода.
4. Обрабатывает введенные пользователем значения параметров и передаёт их JasperReports Library для генерации отчёта по шаблону.

Требования

В данном разделе приведены требования к шаблонам, которые используют базовую реализацию или производные от неё.

report.conf

При выборе базовой или производной от базовой реализации шаблона в файле `report.conf` также должны быть заданы следующие параметры:

Ключ	Значение	Комментарий	Пример значения
<code>definition. generic.key</code>	Уникальный идентификатор шаблона отчёта	В данном параметре следует указать уникальную для вашего шаблона строку, которая будет однозначно отличать данный шаблон от всех остальных. Замечание для программистов: Базовая реализация возвращает это значение в методе <code>key</code> .	<code>"ultima-reports- itrium- timecontrol: timeControl"</code>
<code>definition. generic. title</code>	Удобочитаемое название шаблона отчёта	Данное значение выводится в списке доступных шаблонов в пользовательском интерфейсе. Замечание для программистов: Базовая реализация возвращает это значение в методе <code>title</code> .	Учёт рабочего времени
<code>definition. generic. description</code>	Подробное описание шаблона отчёта	Подробное описание шаблона отчёта. Выводится в интерфейсе пользователя. Замечание для программистов: Базовая реализация возвращает это значение в методе <code>description</code> .	""
<code>definition. jasper. design</code>	Перечень jasper-файлов шаблона отчёта	В данном параметре должно быть указано имя файла с расширением <code>*.jasper</code> , который будет использоваться для формирования отчёта. Можно указать различные файлы для предпросмотра (HTML) и основного отчёта. Для этого задайте различные значения дочерним ключам <code>main</code> (основной отчёт) и <code>html</code> (предпросмотр).	<pre>{ main: "report. jasper", html: "report. jasper" }</pre>

JasperReports

В директории шаблона в наборе должен быть размещён хотя бы один `*.jasper`-файл. Имя файла должно быть указано в файле `report.conf` в параметре `definition.jasper.design`.

Для всех Jasper-параметров в шаблоне, которые должен ввести пользователь, должен быть задан флаг `Is For Prompting` равный `true`. Для таких параметров также должно быть задано свойство с именем класса поля ввода (см. [Привязка полей ввода](#)).

В Jasper-шаблоне необходимо определить системный параметр `DEFINITION_DIR` типа `java.lang.String`, для которого установить флаг `Is For Prompting` равный `false`. В данный параметр базовая реализация передаст абсолютный путь к директории с файлами шаблона (директории, в которой находятся `*.jasper`-файлы и другие файлы шаблона). Это значение необходимо, например, при использовании подотчётов — чтобы указать корректный путь к `*.jasper`-файлу подотчёта.

Все Java-классы, кроме классов API и классов стандартной библиотеки, которые используются в Jasper-шаблоне (например, для форматирования вывода данных), должны быть помещены в форме *.jar-библиотек в директорию [libs](#) набора шаблонов.

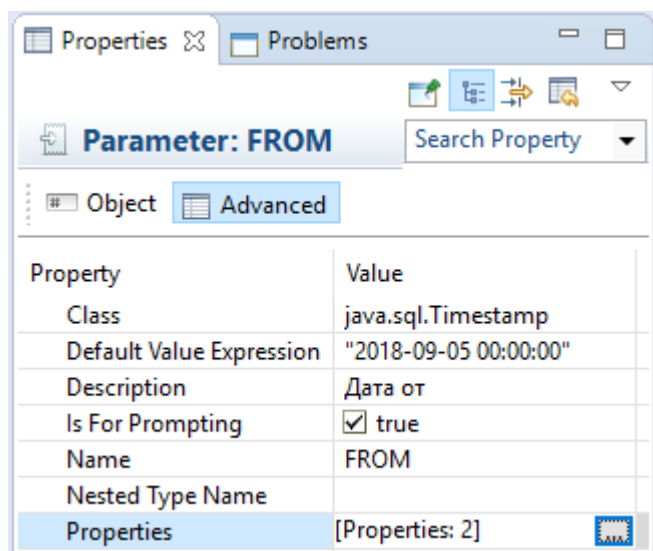
Возможности

В данном разделе описаны возможности, которые предоставляет базовая реализация для создания шаблонов.

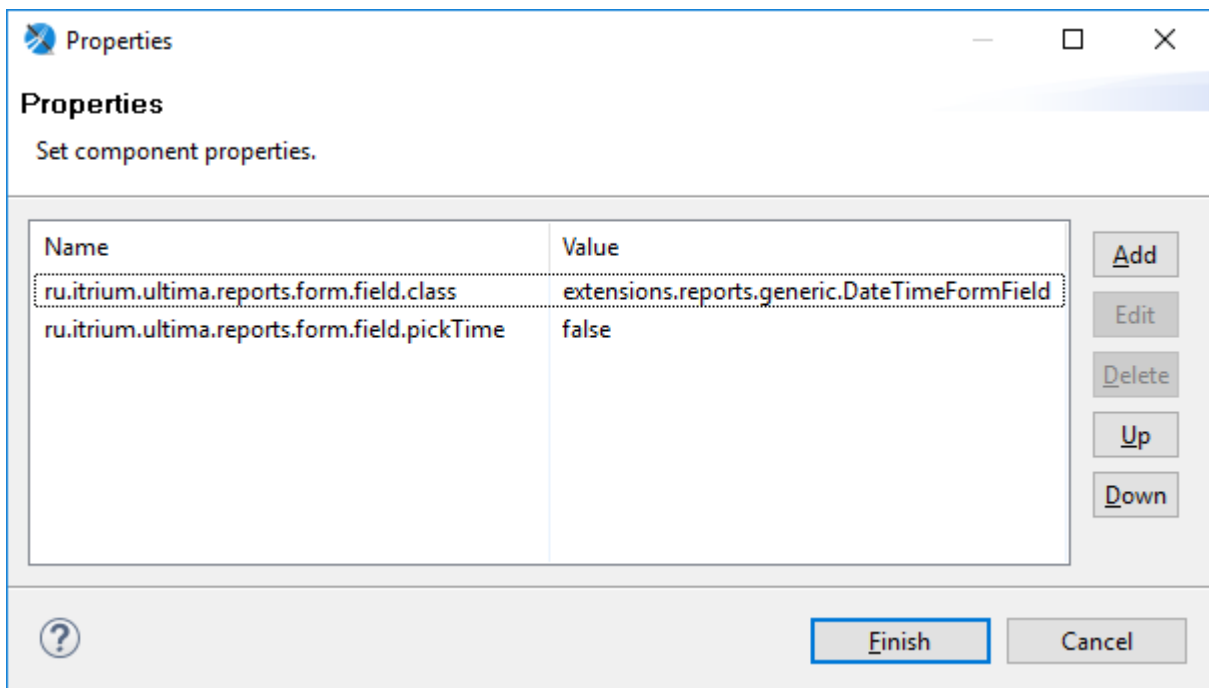
Привязка полей ввода

Базовая реализация позволяет определять, какие именно поля ввода будут использованы на форме ввода параметров, прямо в Jasper-шаблоне. Для этого для каждого параметра, который будет выведен на форме (для которого задан атрибут [Is For Prompting](#) равный [true](#)) должно быть задано свойство [ru.itrium.ultima.reports.form.field.class](#). С помощью свойств также можно определить дополнительные особенности поведения поля ввода — например, в каком формате должно быть введено значение в поле выбора даты / времени.

Управление расширенными свойствами Jasper-параметра в Jaspersoft Studio (версия 6.3.0) осуществляется на вкладке Advanced:



в диалоговом окне Properties:



В XML-документе шаблона такие свойства определяются следующим кодом в блоке параметра `<parameter />`:

```
<parameter name="FROM" class="java.sql.Timestamp">
  <property name="ru.itrium.ultima.reports.form.field.class" value="extensions.
reports.generic.DateTimeFormField"/>
  <property name="ru.itrium.ultima.reports.form.field.pickTime" value="false"/>
  <parameterDescription><![CDATA[Дата от]]></parameterDescription>
  <defaultValueExpression><![CDATA["2018-09-05 00:00:00"]]><
/defaultValueExpression>
</parameter>
```

✓ Полный список классов полей ввода, которые можно свободно использовать, а также перечень поддерживаемых ими свойств приведены [здесь](#).

Кроме полей ввода, перечисленных в списке, разработчик может также реализовать на языке Scala / Java свои собственные поля.