

## ultima-reports-api:1.4.2

В данном разделе приведено описание API подсистемы подготовки отчётов Платформы НЕЙРОСС.

Основные классы API в составе артефакта `ultima-reports-api_<версия>.jar`:

Класс	Описание
<code>extensions.reports. ReportDefinition</code>	<p>Базовый абстрактный класс для всех классов реализации шаблонов отчётов.</p> <p>Основные методы класса:</p> <p><code>def init(conf: Config, loader: ReportDefinitionLoader): Unit</code> В реализации данного метода следует выполнить инициализацию шаблона. Метод вызывается логикой Платформы НЕЙРОСС в тот момент, когда пользователь хочет сформировать отчёт по выбранному шаблону. В метод передаются объекты классов <code>Config</code> (соответствует содержимому файла <code>report.conf</code>) и <code>ReportDefinitionLoader</code> — последний позволяет получить доступ к файлам в папке шаблона отчёта (например, для загрузки <code>jasper</code>-файла).</p> <p><code>def title: String</code> Должен возвращать название шаблона, которое будет выведено в веб-интерфейсе Платформы НЕЙРОСС.</p> <p><code>def description: String</code> Должен возвращать описание шаблона, которое будет выведено в веб-интерфейсе Платформы НЕЙРОСС.</p> <p><code>def renderForm(...): Result</code> <code>def renderFormAction(...): Action</code> Генерация и выдача формы ввода параметров для формирования отчёта. Два метода позволяют как переопределить работу с <code>Action</code> (при использовании <code>views</code> в <code>Play Framework</code>, например), так и просто обработать «голый» <code>HTTP</code>-запрос и вернуть «голый» <code>HTTP</code>-ответ.</p>

	<p><b>def processForm(...): AnyRef</b>  Обработка введенных пользователем параметров. Метод должен вернуть объект (любого типа), содержащий разобранные параметры. Этот объект впоследствии будет передан в метод <b>generateReport</b>. Если параметры заданы неверно, то метод должен выбросить исключение.</p> <p><b>def generateReport(format: String, form: AnyRef, dataSource: javax.sql.DataSource, to: File)</b>  Данный метод вызывается логикой Платформы НЕЙРОСС для непосредственной генерации отчёта. В параметрах передаются:</p> <ol style="list-style-type: none"> <li>1. требуемый формат отчёта (один из заданных в <code>report.conf</code>)  — например, "pdf".</li> <li>2. объект параметров, который ранее был получен вызовом метода <code>processForm</code>.</li> <li>3. объект доступа к источнику данных;</li> <li>4. хэндлер файла, в который следует записать сгенерированный отчёт.</li> </ol> <p><b>def customAction(path: String): Action</b>  Данный метод позволяет реализовать кастомное HTTP-API для шаблона отчёта. Например, выдавать в форму отчёта список объектов из базы данных по запросу и пр.</p>
<code>extensions.reports. JasperReportDefinition</code>	<p>Класс реализации шаблона отчёта с поддержкой JasperReports Library, который формирует отчёт на основе jasper-шаблонов. Является потомком <code>extensions.reports.ReportDefinition</code>.</p> <p>Можно использовать этот класс как есть, если не требуется поддержки формы ввода параметров.</p>

<p><code>extensions.reports.GenericFormField</code></p>	<p>Интерфейс для класса поля ввода. Определяет обязательные методы любого класса поля ввода:</p> <p><code>def key(): String</code> Должен возвращать имя параметра, для которого определено данное поле ввода.</p> <p><code>def title(): String</code> Должен возвращать удобочитаемое название поля ввода. Данный текст выводится в форме в интерфейс пользователя.</p> <p><code>def render(): Html</code> Должен возвращать объект-представление HTML-фрагмента поля ввода.</p> <p><code>def parse(value: String): AnyRef</code> Преобразует введенное строковое значение параметра к Java-объекту требуемого типа.</p>
<p><code>extensions.reports.GenericFormSupport</code></p>	<p>Утилитарный трэйт поддержки базовой формы ввода параметров. Хранит карту объектов полей ввода (потомков <code>extensions.reports.GenericFormField</code>), генерирует базовую форму ввода параметров со списком полей ввода. Основные методы:</p> <p><code>def registerField(field: GenericFormField): Unit</code> Регистрирует новое поле ввода. Данный метод должен вызываться в классах реализации шаблонов, наследуемых от <code>extensions.reports.GenericFormSupport</code>, для составления списка полей ввода.</p> <p><code>def renderFormAction: Action[AnyContent]</code> Генерирует базовую форму ввода параметров на основе заданного через <code>registerField</code> списка полей ввода.</p> <p><code>def processForm(request: Request[MultipartFormData[TemporaryFile]]): AnyRef</code> Выполняет разбор значений формы ввода параметров, вызывая метод <code>extensions.reports.GenericFormField.parse(...)</code> и формируя карту вида «имя параметра — значение».</p>

<code>extensions.reports. JasperFormField</code>	<p>Интерфейс поля ввода для Jasper-параметра. Является потомком <a href="#">extensions.reports.GenericFormField</a>.</p> <p>Определяет новый метод <code>def init(jrParameter: JRParameter): Unit</code>, который вызывается в <code>extensions.reports.GenericJasperReportDefinition</code> при инициализации списка полей ввода по Jasper-параметрам.</p>
<code>extensions.reports. GenericJasperReportDefinition</code>	<p>Класс базовой реализации шаблона отчёта, который формирует отчёт на основе Jasper-шаблонов и умеет генерировать по Jasper-файлу форму для ввода параметров. Является потомком <code>extensions.reports.JasperReportDefinition</code> и <code>extensions.reports.GenericFormSupport</code>.</p>
<code>extensions.reports.generic. AbstractJasperFormField</code>	<p>Абстрактный класс базовой реализации поля ввода для Jasper-параметра. Реализует интерфейс <a href="#">extensions.reports.JasperFormField</a>:</p> <p><code>def key(): String</code> Возвращает имя (name) Jasper-параметра в роли ключа.</p> <p><code>def title(): String</code> Возвращает описание (description) Jasper-параметра в роли названия поля.</p>
<code>extensions.reports.generic. DateTimeFormField</code>	<p>Реализация поля ввода даты / времени. Возможности по использованию данного поля описаны в <a href="#">Поля ввода параметров для отчётов JasperReports</a>.</p>
<code>extensions.reports.generic. InputFormField</code>	<p>Реализация полей ввода текста / числа. Возможности по использованию данного поля описаны в <a href="#">Поля ввода параметров для отчётов JasperReports</a>.</p>
<code>extensions.reports.generic. PassFormField</code>	<p>Реализация поля для поиска и выбора пропуска (пропусков) из Платформы НЕЙРОСС.</p> <p>Возможности по использованию данного поля описаны в <a href="#">Поля ввода параметров для отчётов JasperReports</a>.</p>

<pre>extensions.reports.generic. SQLChoiceFormField</pre>	<p>Реализация поля выбора опций из списка. Опции загружаются из базы данных в соответствии с заданным SQL-запросом.</p> <p>Возможности по использованию данного поля описаны в <a href="#">Поля ввода параметров для отчётов JasperReports</a>.</p>
<pre>extensions.reports.generic. OriginFormField</pre>	<p>Реализация скрытого поля для передачи в отчёт базового URL сервера Платформы НЕЙРОСС. Для встраивания гиперссылок в отчёт.</p> <p>Возможности по использованию данного поля описаны в <a href="#">Поля ввода параметров для отчётов JasperReports</a>.</p>
<pre>extensions.reports.generic. TimeFormField</pre>	<p>Реализация поля ввода времени суток. Возможности по использованию данного поля описаны в <a href="#">Поля ввода параметров для отчётов JasperReports</a>.</p>
<pre>extensions.reports.generic. TimeZoneFormField</pre>	<p>Скрытое поле для автоматического получения временной зоны АРМ пользователя: чтобы дата / время в отчётах выводились во временной зоне пользователя.</p> <p>Системный параметр для задания временной зоны всегда есть в Jasper-шаблоне, а потому данное скрытое поле всегда будет присутствовать в стандартной форме ввода параметров.</p>