

Platypus 3.0

Руководство администратора

Руководство по установке и администрированию платформы Platypus

Platypus 3.0 Руководство администратора

Руководство по установке и администрированию платформы

Platypus

Редакция 0

Данное руководство содержит информацию по установке и администрированию платформы Platypus

Введение	v
1. Соглашения документа	v
1.1. Типографические соглашения	v
1.2. Соглашения по выделению текста	vi
1.3. Примечания и предупреждения	vii
2. Мы ждем ваших отзывов	vii
1. Установка сервера и клиента	1
1.1. Системные требования платформы Platypus	1
1.2. Установка платформы Platypus	2
1.3. Удаление платформы Platypus	3
2. Конфигурирование и запуск клиента Platypus	5
2.1. Режимы работы клиента Platypus	5
2.2. Запуск клиента	7
2.3. Настройка скриптов и параметры командной строки запуска клиента	9
3. Конфигурирование и запуск сервера приложений Platypus	11
3.1. Запуск сервера приложений Platypus в автономном режиме	11
3.2. Настройка скрипта и параметры командной строки запуска сервера приложений в автономном режиме	11
4. Конфигурирование серверных компонентов Platypus для работы в сервлет-контейнере или J2EE сервере	15
4.1. Настройка J2EE сервера для работы с Platypus	15
4.2. Конфигурирование сервера приложений Platypus и развертывание в сервлет-контейнер или J2EE сервер	16
5. Безопасность в Platypus	19
5.1. Настройка прав доступа пользователей при использовании встроенного хранилища информации о пользователях	19
5.2. Использование внешнего хранилища информации о пользователях	20
6. Развертывание приложений и миграции БД	21
6.1. Развертывание и импорт приложения	22
6.2. Миграции БД	23
A. Режимы протоколирования	25
B. Настройка J2EE сервера Glassfish 3.1.2	27
C. Настройка Apache Tomcat 7	31
D. History	33
Предметный указатель	35

Введение

1. Соглашения документа

В этом руководстве используются различные стили для выделения текста.

В PDF и бумажной версиях руководства используются шрифты семейства [Liberation](https://fedorahosted.org/liberation-fonts/)¹. Эти же шрифты будут использоваться для отображения HTML-версии, если они установлены в вашей системе. В противном случае будут использоваться аналогичные шрифты. Red Hat Enterprise Linux 5 и более поздние версии включают в свой состав комплект Liberation по умолчанию.

1.1. Типографические соглашения

Для выделения текста используются четыре стиля, которые будут перечислены далее.

Моноширинный жирный шрифт

Используется для выделения вводимого текста, включая команды оболочки, а также имен файлов, путей и комбинаций клавиш. Пример:

Чтобы просмотреть содержимое файла **my_next_bestselling_novel** в текущем каталоге, в строке приглашения оболочки введите **cat my_next_bestselling_novel** и нажмите **Enter** для выполнения этой команды.

Приведенный текст содержит имя файла, команду оболочки и имя клавиши, которые выделены моноширинным жирным шрифтом.

Для разделения клавиш в составе комбинаций используется дефис. Пример:

Нажмите **Enter** для исполнения команды.

Нажмите **Ctrl+Alt+F2** для перехода в первый виртуальный терминал. Нажмите **Ctrl+Alt+F1**, чтобы вернуться в сессию X-Windows.

В первом примере жирным шрифтом выделено название отдельной клавиши, во втором — комбинаций клавиш.

Этим же шрифтом выделяются имена классов, методов, функций, переменных и возвращаемые ими значения. Пример:

Классы файлов включают **filesystem** для файловых систем, **file** для файлов, **dir** для каталогов. Каждому классу соответствует набор разрешений.

Пропорциональный жирный

Выделяет системные слова и фразы, что включает имена приложений, текст диалогов, названия меню, текст кнопок, флажков и других элементов графического интерфейса. Пример:

В главном меню выберите **Система** → **Параметры** → **Мышь** для запуска утилиты **Настройки мыши**. На вкладке **Кнопки** установите флажок **Настроить мышь под левую руку** и нажмите кнопку **Заккрыть**, чтобы настроить мышь для левши.

¹ <https://fedorahosted.org/liberation-fonts/>

Чтобы вставить специальный символ в файл **gedit**, выберите **Приложения** → **Стандартные** → **Таблица символов**. Затем в меню выберите **Поиск** → **Поиск...**, введите имя символа и нажмите кнопку **Найти следующее**. Найденный символ будет выделен в **таблице символов**. Дважды щелкните на этом символе, чтобы вставить его в поле **Текст для копирования** и нажмите кнопку **Копировать**. Теперь вернитесь к вашему документу и в меню выберите **Правка** → **Вставить**.

Приведенный выше текст содержит имя приложения, названия меню, кнопок и текста элементов графического интерфейса.

Моноширинный жирный курсив или **пропорциональный жирный курсив**

Оба типа выделения обозначают изменяемый или заменяемый текст. Курсив сообщает о том, что не следует вводить приведенный текст напрямую, а изменить в соответствии с вашими настройками. Пример:

Для подключения к удаленной машине с помощью SSH в строке приглашения выполните **ssh имя_пользователя@имя_домена**. Скажем, имя удаленной машины – **example.com**, а ваше имя пользователя – **john**, тогда команда будет выглядеть так: **ssh john@example.com**.

Команда **mount -o remount файловая_система** повторно подключит заданную файловую систему. Например, для **/home** команда будет выглядеть так: **mount -o remount /home**.

Чтобы просмотреть версию установленного пакета, выполните команду **rpm -q пакет**. Результат команды будет представлен в формате **пакет-версия-выпуск**.

В приведенных примерах жирным курсивом выделяются имя пользователя, имя домена, файловой системы, пакет, его версия и выпуск.

Также курсивом выделяются термины, которые встречаются в тексте документа впервые. Пример:

Publiсan — система публикации *DocBook*.

1.2. Соглашения по выделению текста

Вывод экрана и листинг исходного кода будут отделены от окружающего текста.

Для отображения текста, который вы увидите на экране, используется **моноширинный шрифт**:

```
books      Desktop  documentation  drafts  mss    photos  stuff  svn
books_tests Desktop1  downloads      images  notes  scripts  svgs
```

Для отображения содержимого исходного кода используется **моноширинный шрифт**:

```
package org.jboss.book.jca.ex1;

import javax.naming.InitialContext;

public class ExClient
{
    public static void main(String args[])
        throws Exception
```

```
{
    InitialContext iniCtx = new InitialContext();
    Object          ref    = iniCtx.lookup("EchoBean");
    EchoHome        home   = (EchoHome) ref;
    Echo            echo    = home.create();

    System.out.println("Created Echo");

    System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
}
```

1.3. Примечания и предупреждения

Наконец, чтобы привлечь внимание читателя к важной информации, используются три стиля.



Примечание

Примечания обычно содержат дополнительную информацию. Если вы их проигнорируете, это не критично, но вы можете пропустить совет, который, возможно, поможет сэкономить время при выполнении задания.



Важно

На информацию, отмеченную как важную, следует обратить особое внимание. Она может включать изменения настроек текущего сеанса или, например, перечень служб, которые нужно запустить, прежде чем обновления вступят в силу. Ознакомление с важной информацией значительно облегчит вашу работу.



Предупреждение

Не стоит игнорировать предупреждения, так как они содержат важную информацию, которая позволит избежать потери данных.

2. Мы ждем ваших отзывов

Переопределите эти настройки, создав собственный локальный файл Feedback.xml.

Установка сервера и клиента

Платформа Platypus поставляется в следующих вариантах:

- На компакт или DVD диске;
- Через Интернет - в виде архивного файла в формате zip.

В случае поставки в виде архивного файла распакуйте его содержимое в удобное для вас место на жестком диске.



Примечание

Инструкции по установке инструмента разработчика **Platypus Application Designer** смотри в документе "Руководство разработчика".

1.1. Системные требования платформы Platypus

Ниже указаны необходимые условия для запуска компонентов платформы на компьютере:

- 32-разрядный (x86) или 64-разрядный (x64) процессор с тактовой частотой 1 гигагерц (ГГц) или выше;
- 1 гигабайт (ГБ) (для 32-разрядной системы) или 2 ГБ (для 64-разрядной системы) оперативной памяти (ОЗУ);
- 5 гигабайт (ГБ) свободного пространства на жестком диске;
- операционная система Windows, Linux, Mac OS X или другая Unix подобная;
- Oracle JRE 7 или OpenJDK 7;
- графический интерфейс пользователя для работы клиента;
- доступ к Интернет для получения обновлений и работы с внешним картографическим сервисом. В корпоративной сети такой доступ может быть организован через прокси-сервер;
- Web-браузер Google Chrome 19, Mozilla Firefox 10, Internet Explorer 9 или их более новые версии для работы HTML5 клиента Platypus.

Для работы приложений Platypus обычно необходим сервер баз данных (БД), который может быть установлен как на том же компьютере что и клиент или сервер приложений, так и на другом компьютере, доступном по сети.

Платформа Platypus поддерживает работу со следующими базами данных:

- Oracle Database 10g и выше;
- IBM DB2 9 и выше;
- Microsoft SQL Server 2008 и выше;
- MySQL (InnoDB) 5.5 и выше;
- PostgreSQL 9 и выше;

- H2 Database 1.3 и выше.

Для получения информации об установке и настройке сервера баз данных обратитесь к соответствующим руководствам по установке и администрированию, предоставляемым производителями серверов БД.



Примечание

В комплект поставки Platypus Platform включена база данных H2. Использование данной БД доступно "из коробки" после установки платформы и не требует дополнительных действий по установке и настройке.

Для работы серверной части приложений внутри сервлет-контейнера или J2EE сервера установите или используйте уже установленный сервлет-контейнер или J2EE сервер. Работа Platypus тестировалась в следующем окружении:

- Apache Tomcat 7;
- Glassfish 3.1.2;
- JBoss AS7;

Для получения информации об установке и настройке сервлет-контейнера или J2EE сервера обратитесь к соответствующим руководствам по установке и администрированию, предоставляемым их производителями.

1.2. Установка платформы Platypus

Для того, чтобы установить компоненты Платформы выполните следующие действия:

1. перейдите в подкаталог **install** дистрибутива Platypus;
2. запустите файлы **install.bat** для операционной системы Windows или выполните команду **sh install.sh** для Linux;
3. выберите язык интерфейса пользователя программы по установке;
4. выберите каталог установки;
5. выберите компоненты которые будут установлены;
 - Platypus Platform - ядро платформы Platypus;
 - Java SE клиент;
 - модули работы с картами;
 - сервер приложений - включает в себя модули необходимые для работы автономного сервера приложений и серверных компонентов в сервлет/J2EE-контейнерах;
 - Platypus Communications - модули, необходимые для работы с COM/DCOM и OPC серверами;
6. выберите параметры настроек ярлыков для запуска клиента Platypus в вашей системе.



Примечание

Компоненты ядра платформы являются обязательными при установке любой конфигурации.

После установки, каталог будет иметь следующую структуру:

- **platypus/**
 - **run/**
 - **web/**
 - **examples/**
 - **docs/**
 - **bin/**
 - **lib/**

1.3. Удаление платформы Platypus

Для того, чтобы удалить платформу Platypus с компьютера выполните следующие действия:

1. запустите программу **UninstallPlatypus** из меню установленных программ или с помощью ярлыка. Для ОС Windows используйте штатный механизм удаления программ;
2. подтвердите удаление компонентов Platypus, при необходимости активируйте опцию удаления рабочего каталога.

Конфигурирование и запуск клиента Platypus

Клиент Platypus является Java SE программой и предназначен для организации интерфейса пользователя и/или исполнения логики приложения на компьютере конечного пользователя. Клиент Platypus загружает требуемые элементы приложения с сервера приложений, из базы данных или читает непосредственно с диска, из папки, указанной при конфигурировании.

Клиент Platypus обеспечивает автоматическую проверку обновлений и предлагает из установку.

2.1. Режимы работы клиента Platypus

Клиент Platypus может работать в следующих режимах:

- 2-х звенный режим - Java SE клиент подключается непосредственно к серверу БД. Данная схема рекомендуется для относительно несложных приложений, не требующих исполнения бизнес-логики на сервере.

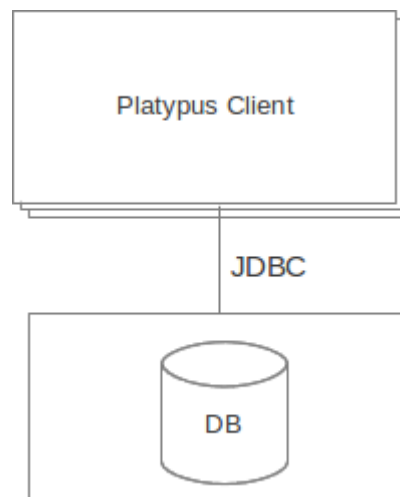


Рисунок 2.1. 2-х звенная схема в Platypus

- автономный 3-х звенный режим - клиент подключается к серверу **Platypus Application Server**. Для передачи данных используется двоичный протокол Platypus Protocol. Сервер приложений в свою очередь взаимодействует с сервером БД. Этот режим рекомендуется для приложений средней сложности, предполагающих исполнение бизнес-логики на сервере приложений.

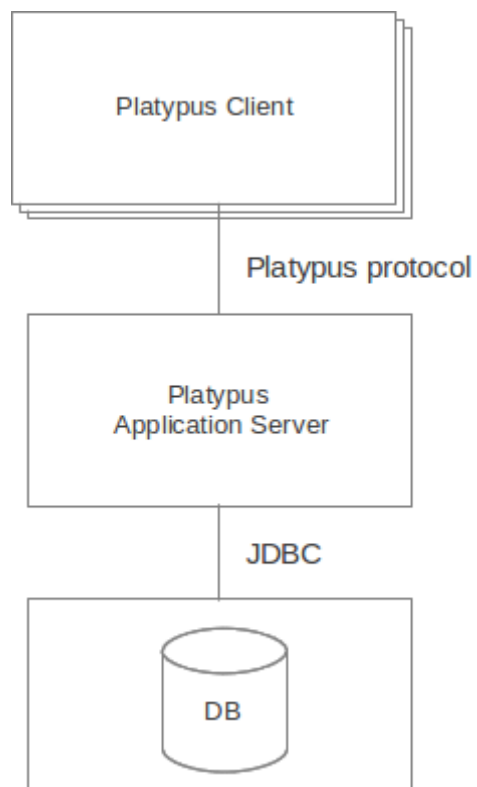


Рисунок 2.2. Автономный 3-х звенный режим

- режим сервлета - клиент подключается к сервлет-контейнеру или J2EE серверу , где развернуты серверные компоненты Platypus. Клиентом в данном случае может являться как Java SE клиент, так и HTML5 браузер (Chrome, Opera, Firefox и т.д.). Для передачи данных используется протокол HTTP/HTTPS. В этом случае, для подключения к базам данных используется ресурс источника данных, предоставляемого контейнером. Для данного варианта подключения требуется развертывание серверных компонентов Platypus в сервлет-контейнер или J2EE сервер.

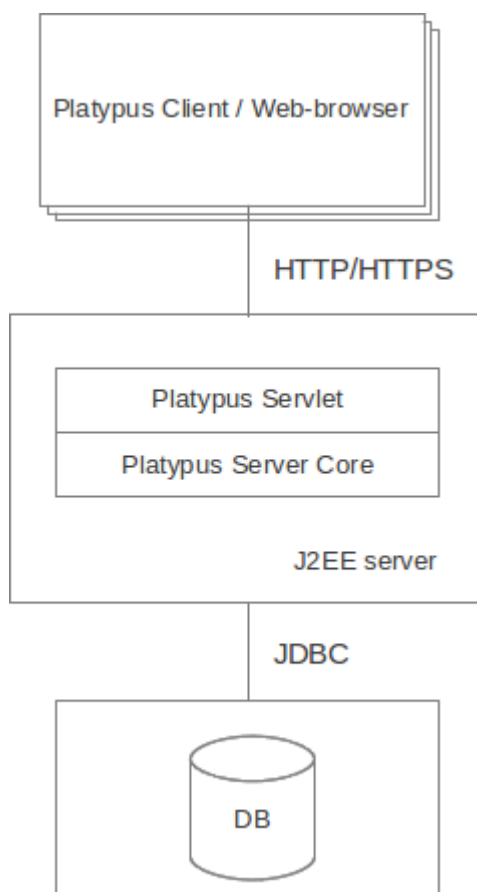


Рисунок 2.3. Режим сервлета Platypus

2.2. Запуск клиента

Для запуска клиента необходимо использовать ярлык **Platypus Client** на рабочем столе, меню **Пуск** или запустить стартовый скрипт.

Скрипты запуска и загрузчик клиента Platypus находятся в папке **/run** :

- **platypus.exe** - для запуска в ОС Windows без создания окна командной строки с предварительным запуском обновления клиента; данная команда обеспечивает корректную работу с UAC (User Account Control) в ОС Windows;
- **platypus.js** - скрипт для запуска в ОС Windows без проверки обновлений;
- **platypus.sh** - для запуска клиента в ОС Linux;
- **startupupdate.sh** - для запуска клиента в ОС Linux с предварительным запуском обновления клиента;

При успешном запуске, клиент выводит диалоговое окно для ввода имени пользователя и пароля и выбора предварительно настроенного подключения к серверу.

Вы можете создать и использовать несколько предварительно настроенных подключений к БД или серверу Platypus. В ваш дистрибутив может быть включено предварительно настроенное подключение по умолчанию.

В случае работы по 2-х звенной схеме введите имя пользователя и пароль соединения с базой данных, а также имя пользователя и пароль в приложениях Platypus.

В случае работы по одной из 3-х звенной схеме введите только имя пользователя и пароль приложений Platypus.

Используйте флажки **Запомнить пароль БД** и **Запомнить пароль**, если необходимо запомнить пароль БД и/или пароль приложений Platypus.

Используйте кнопку **>>>** и **<<<** для того чтобы отобразить и скрыть панель предварительно настроенных подключений.

Выберите предварительно настроенное подключение из списка для того чтобы подключиться к серверу и запустить клиента.

Создание и редактирование предварительно настроенных подключений.

Для того, чтобы отобразить диалог создания нового подключения нажмите кнопку **Новое** на панели предварительно настроенных подключений.

Для того, чтобы изменить предварительно настроенное подключение выберите его из списка и нажмите кнопку **Изменить** на панели предварительно настроенных подключений.

Введите или измените следующие поля диалога "Настройки соединения":

- **Наименование** - наименование подключения, например, **myserver**. В данном поле может быть введено любое название;
- **URL подключения** - строка соединения с сервером, для работы по 2х звенной схеме должна быть в формате JDBC URL, например: **jdbc:oracle:thin:@dbhost:1521:adb**,

для работы по 3-х звенной схеме по протоколу Platypus Protocol используйте URL формата **platypus://host:port**, например, **platypus://serverhost:8500**, порт 8500 является портом по умолчанию для протокола Platypus, и его можно в данном случае не указывать,

для работы по 3-х звенной схеме по протоколу HTTP используйте URL формата **http://host:port/platypus_servlet/application**, например, **http://localhost/myapp/application**.
- **Схема** - схема БД по умолчанию, например, **myschema**; В случае 3-х звенного подключения не используется.
- **Пользователь БД** - имя пользователя БД по умолчанию, например, **user1**; В случае 3-х звенного подключения не используется.
- Для сохранения новых настроек подключения нажмите кнопку **ОК** диалога. Для отмены кнопку **Отмена**.

Для того чтобы удалить предварительно настроенное соединение выберите соединение из списка и нажмите кнопку **Удалить** на панели предварительно настроенных соединений. Подключение будет удалено после того как пользователь нажмет кнопку **ОК** в диалоге подтверждения удаления.

Для запуска клиента с выбранным подключением нажмите кнопку **ОК**. Для того, чтобы закрыть диалог выбора подключений нажмите кнопку **Отмена**.

После запуска клиента будет запущен элемент приложения по умолчанию (форма, отчет или модуль) для данного пользователя или элемент приложения явно указанный в параметре - *appElement* командной строки запуска.

2.3. Настройка скриптов и параметры командной строки запуска клиента

Данный раздел содержит подробную справочную информацию по параметрам командной строки запуска клиента Platypus.

Для настройки запуска клиента Platypus отредактируйте содержимое файлов скриптов запуска. Могут быть настроены следующие параметры:

- элемент приложения по умолчанию - идентификатор формы, отчета или модуля который будет открыт при запуске клиента, например **appStartForm**, в случае, если этот параметр не будет указан, клиент будет использовать элемент по умолчанию для текущего пользователя;
- тема оформления (LaF - Look and Feel) внешнего вида клиента. Клиент Platypus может использовать LaF графической оболочки Swing платформы Java SE. Допустимо использовать как стандартные LaF для данной операционной системы так и библиотечные от сторонних производителей;
- уровень настройки логов и файл записи логов.

Команда для запуска клиента Platypus имеет следующий вид:

```
java -cp
Application.jar;EXT_CLASSPATHcom.eas.client.application.PlatypusClientApplication OPTIONS
```

где *EXT_CLASSPATH* - пути, которые должны быть добавлены в путь поиска загрузчика классов Java, *OPTIONS* - дополнительные параметры запуска.

В начале настройки рекомендуется запускать скрипты через командную строку, чтобы увидеть возможные сообщения об ошибках. В случае удачного запуска клиента появится диалог авторизации.

Так же клиента можно запустить с использованием параметра *-jar*:

```
java -jar Application.jar OPTIONS
```

Такой вариант запуска удобен при работе с командной строкой, однако такой подход не позволяет добавлять внешние библиотеки в путь по которому загрузчик Java ищет классы, например, для того чтобы подключать произвольные темы оформления Look and Feel.

Таблица 2.1. Параметры запуска клиента из командной строки

Параметр	Описание
<i>-appElement APP_ELEMENT</i>	Идентификатор элемента приложения для запуска.
<i>-ap APP_PATH</i>	Путь к папке, содержащей элементы приложения Platypus. Если этот параметр задан, то клиент Platypus будет искать элементы приложения Platypus в этой папке, в противном случае клиент Platypus будет искать элементы приложения в базе данных в соответствующих таблицах.
<i>-url JDBC_URL</i>	URL соединения с БД в формате JDBC.
<i>-dbuser DB_USER</i>	Имя пользователя для авторизации в БД.

Глава 2. Конфигурирование и запуск клиента Platypus

<code>-dbpassword DB_PASSWORD</code>	Пароль пользователя для авторизации в БД.
<code>-dbschema DB_SCHEMA</code>	Схема БД.
<code>-user USER_NAME</code>	Имя пользователя приложения Platypus.
<code>-password PASSWORD</code>	Пароль пользователя приложения Platypus.
<code>-loglevel LOG_LEVEL</code>	Уровень ведения журнала. Данный параметр указывает минимальный уровень который будет выводиться в журнал. Описание уровней журнала смотрите в Приложении А.
<code>-log LOG_FILE_PATH</code>	Имя и путь к файлу журнала.

Если параметры настройки соединения с БД или имя пользователя и пароль Platypus не заданы будет отображен диалог "Настройки соединения" для ввода этих параметров.

Пример команды запуска клиента Platypus:

```
java -cp Application.jar;CoolLaf.jar -laf CoolLaf  
com.eas.client.application.PlatypusClientApplication -appElement MainForm
```

Пример запуска клиента с использованием параметра `-jar`:

```
java -jar Application.jar -appElement MainForm
```

Конфигурирование и запуск сервера приложений Platypus

Сервер приложений **Platypus Application Server** предназначен для централизованного выполнения бизнес-логики и является промежуточным слоем при работе Platypus по 3-х звенной схеме.

Сервер приложений может быть развернут автономно или внутри сервелет-контейнера/J2EE сервера.

3х-звенная схема имеет ряд преимуществ по сравнению с 2-х звенной:

- лучшая безопасность в следствии того, что у клиента нет прямого доступа к базе данных и по сети не передается текст запросов к БД;
- лучшая производительность из-за того что бизнес-логика выполняется на сервере;
- лучшая сопровождаемость системы;
- масштабируемость конечного приложения;
- возможность использовать HTML5 браузерного клиента (работа с сервелет-контейнером/J2EE сервером);

3.1. Запуск сервера приложений Platypus в автономном режиме

В данном режиме сервер приложений Platypus запускается в собственном процессе и собственной виртуальной машине Java (JVM) .

Для запуска сервера приложений Platypus необходимо запустить стартовый скрипт.

Скрипт запуска сервера Platypus находятся в папке **/run** :

- **server.bat** - для запуска сервера в ОС Windows;
- **server.sh** - для запуска сервера в ОС Linux;

Для промышленной эксплуатации сервера приложений Platypus рекомендуется настроить его запуск в качестве службы ОС.

3.2. Настройка скрипта и параметры командной строки запуска сервера приложений в автономном режиме

Данный раздел содержит подробную справочную информацию по параметрам командной строки запуска сервера приложений Platypus.

Для настройки запуска сервера Platypus отредактируйте содержимое файлов скрипта запуска. Могут быть настроены следующие параметры:

- путь к приложению Platypus, в этом случае параметры соединения с БД будут получены из конфигурационного файла;
- параметры соединения с БД;

- сетевые интерфейсы и протоколы;
- модули обработки входящих данных и фоновых задач;
- уровень логов и файл записи логов.

Для настройки отредактируйте переменные и параметры строки запуска скрипта.

Команда для запуска клиента имеет следующий вид:

```
java -cp Server.jar;EXT_CLASSPATH com.eas.server.ServerMain OPTIONS
```

где *EXT_CLASSPATH* - пути, которые должны быть добавлены в путь поиска загрузчика классов Java, *OPTIONS* - дополнительные параметры запуска.

Так же сервер приложений можно запустить с использованием параметра *-jar*:

```
java -jar Server.jar OPTIONS
```

Такой вариант запуска удобен, при работе с командной строкой, однако такой подход не позволяет добавлять внешние библиотеки в путь по которому загрузчик Java ищет классы. Например, библиотеки отвечающие за соединение с нетиповыми базами данных или компоненты сторонних производителей, используемые конечным приложением в процессе обработки бизнес-логики.

Таблица 3.1. Параметры запуска сервера приложений из командной строки

Параметр	Описание
<i>applicationpath APP_PATH</i>	Путь к папке, содержащей приложение. Если задан этот параметр, то приложение будет запущено с диска из этой папки, в противном случае приложение будет запущено из базы данных. Этот параметр имеет короткий синоним <i>ap</i>
<i>-url JDBC_URL</i>	URL соединения с БД в формате JDBC.
<i>-dbuser DB_USER</i>	Имя пользователя для авторизации в БД.
<i>-dbpassword DB_PASSWORD</i>	Пароль пользователя для авторизации в БД.
<i>-dbschema DB_SCHEMA</i>	Схема БД.
<i>-iface ADDRESS:PORT,ADDRESS:PORT</i>	Сетевые интерфейсы и порты прослушивания для сервера.
<i>-protocols PORT:PROTOCOL,PORT:PROTOCOL</i>	Сетевые протоколы для портов прослушивания.
<i>-loglevel LOG_LEVEL</i>	Уровень ведения журнала. Данный параметр указывает минимальный уровень который будет выводиться в журнал. Описание уровней журнала смотрите в Приложении А.
<i>-log LOG_FILE_PATH</i>	Имя и путь к файлу журнала.
<i>-acceptorTask MODULE_NAME</i>	Серверный модуль для обработки входящих данных. У модуля должна быть определена функция <i>received(data)</i>
<i>-backgroundTask MODULE_NAME, MODULE_NAME</i>	Серверные модули, который будут запущены в фоновом потоке при старте сервера

Пример команды запуска сервера приложений Platypus:

```
java -cp Server.jar com.eas.server.ServerMain -url jdbc:oracle:thin:@serverHost:1521:adb -  
dbuser user1 -dbpassword secret -dbschema testschema -loglevel INFO -log serverlog.log -iface  
0.0.0.0:8500 -protocols 8500:platypus
```

Пример команды запуска сервера приложений с использованием параметра *-jar*:

```
java -jar Server.jar -ap ~/apps/testApp -url jdbc:oracle:thin:@serverHost:1521:adb -dbuser  
user1 -dbpassword secret -dbschema testschema -loglevel INFO -log serverlog.log
```

Конфигурирование серверных компонентов Platypus для работы в сервлет-контейнере или J2EE сервере

Серверные компоненты **Platypus** могут работать в сервлет-контейнере или J2EE сервере. Данная конфигурация имеет следующие особенности:

- серверные компоненты Platypus развертываются в J2EE сервер как web-приложение - в виде WAR-архива или папки, взаимодействие с клиентом обеспечивает Platypus Servlet;
- возможна работа **Java SE клиента (Platypus Client)**, так и HTML5 браузерного клиента, а также Web-сайта с использованием Platypus Web API;
- обмен данными с клиентом происходит по протоколу HTTP/HTTPS;
- возможно использование внешнего сервиса аутентификации пользователей, например, службы **Microsoft Active Directory**, что позволяет интегрировать конечное приложение платформы Platypus в существующее на предприятии пространство пользователей;
- связь сервера приложений Platypus с БД осуществляется через настроенный в виде ресурса JNDI источник данных;

Использование Platypus Web API позволяет создавать Web-приложения, использующие в качестве источника данных сервер приложений Platypus. Web-приложение может строиться с использованием технологий ASP, JSP, GWT и прочих. Platypus Web API приложение может осуществлять удаленный вызов функций серверных модулей, при этом вызов функций является прозрачным для клиентского JavaScript кода.

4.1. Настройка J2EE сервера для работы с Platypus

Для запуска Platypus в сервлет-контейнере или J2EE сервере выполните следующие действия:

- настройте источник данных JDBC в виде ресурса JNDI, задайте его имя, например, **java:/jdbc/TestDB**, настройте JDBC-драйвер, пул ресурсов и ресурс JNDI, при необходимости скопируйте JAR соответствующего драйвера БД в каталог, доступный загрузчику классов сервера;
- настройте область безопасности (Realm) для работы со встроенным хранилищем или внешним сервисом аутентификации. Для работы со встроенным хранилищем настройте область безопасности JDBC для работы с таблицами **MTD_USERS** и **MTD_GROUPS** - см. раздел Безопасность.
- настройте параметры web-приложения в xml-дескрипторе **WEB-INF/web.xml** и, при необходимости, специфических для данного сервера файлах конфигурации;
- разверните web-приложение в виде WAR-архива или папки.

**Примечание**

Руководство по настройке J2EE сервера **Glassfish 3.1** смотрите в приложении В.

Руководство по настройке **Apache Tomcat 7** смотрите в приложении С.

4.2. Конфигурирование сервера приложений Platypus и развертывание в сервлет-контейнер или J2EE сервер

Для работы HTML5 браузерного клиента, использования Platypus Web API и Java SE клиента **Platypus** (по протоколам HTTP/HTTPS) серверные компоненты Platypus должны быть развернуты в сервлет-контейнере или J2EE сервере и соответствующим образом сконфигурированы.

В поставку Platypus входит папка `web/` в которой находится web-приложение Platypus, которое можно использовать для дальнейшей модификации. Сделайте копию данной папки, которая будет являться основой для вашего web-приложения.

Для конфигурирования web-приложения отредактируйте файл **WEB-INF/web.xml** и при необходимости добавьте или измените другие файлы - специфические для конкретного сервлет-контейнера или J2EE сервера.

Для корректной работы сервлета Platypus отредактируйте файл **WEB-INF/web.xml**:

1. задайте параметр инициализации с именем *url* и значением имени JNDI ресурса источника данных JDBC, например, **java:/jdbc/TestDB**.
2. задайте параметр инициализации с именем *dialect* и значением имени драйвера базы данных, например, **Oracle**.
3. задайте параметр инициализации с именем *schema* и значением схемы БД по умолчанию, например, **test**.
4. укажите тип аутентификации, например **BASIC** и имя области безопасности (Realm), например **JDBCRealm**. При работе со встроенным пространством пользователей, информацию по настройке области безопасности смотрите в разделе "Безопасность".
5. добавьте роль безопасности - Security Role, например **platypus_default_role**;
6. добавьте ограничение безопасности - Security Constraint. Для данного ограничения задайте имя, шаблон URL и при необходимости методы HTTP. Данное ограничение необходимо задать если предполагается осуществлять разграничение доступа.

**Важно**

Настройте ограничения безопасности для URL по которым работает Platypus Servlet.

Таблица 4.1. Диалекты - имена драйверов БД

База данных	Диалект
Oracle DB	Oracle

База данных	Диалект
MySQL	MySql
SQL Server	MsSql
PostgreSQL	PostgreSql
H2 DB	H2
IBM DB2	DB2

Platyus Client поддерживает **BASIC** аутентификацию, поэтому для его корректной работы должен быть настроен именно этот тип аутентификации.

После завершения конфигурирования разверните Web-приложение в виде папки или WAR-архива в сервлет-контейнер или J2EE сервер и укажите базовый URL для web-приложения.

Безопасность в Platypus

Платформа Platypus имеет развитые механизмы обеспечения безопасности и обеспечивает ограничение доступа к ресурсам в системе на основе ролей назначенных пользователям и их группам. Могут вводиться ограничения доступа к следующим ресурсам:

- программный модуль любого уровня - на сервере или клиенте;
- функция в программном модуле;
- форма;
- запрос к базе данных;
- отчет;
- статический ресурс;

Разграничения на доступ к ресурсам осуществляются на базе ролей. Роли вводятся на уровне приложения Platypus и являются его частью.

Для элемента приложения типа *Запрос к базе данных* могут определяться правила доступа для роли: полный доступ, только чтение (выполнение запросов **SELECT**), запись (выполнение команд **INSERT**, **DELETE** и **UPDATE**).

Помимо разграничения доступа на элементы приложения могут быть заданы ограничения на доступ с использованием встроенных средств сервлет-контейнера или J2EE сервера - при этом будет защищен URL доступа к приложению целиком.

С точки зрения аутентификации, Platypus может использовать "область безопасности". Область безопасности может быть встроенной или внешней и содержать информацию о пользователях и об их принадлежности к группам. Platypus поддерживает следующие варианты области безопасности:

- встроенная "база данных" пользователей, размещаемая в 2х таблицах БД - может использоваться при работе как по 2-х так и 3-х звенной схеме;
- внешние хранилища аутентификационных данных, например, сервер LDAP (Active Directory, OpenLDAP и т.д.), данный режим может использоваться только при работе по 3-х звенной схеме в сервлет-контейнере или J2EE сервере.

В области безопасности для пользователя определяется набор "групп" или "глобальных ролей" которым могут быть поставлены в соответствие роли на уровне приложения.

5.1. Настройка прав доступа пользователей при использовании встроенного хранилища информации о пользователях

Использование встроенного хранилища предоставляет простые инструменты для хранения информации о пользователях и их группах.

Данный режим является единственным возможным режимом при работе в 2-х звенной схеме и работе по 3-х звенной схеме в варианте работы сервера приложений Platypus вне J2EE сервера, а также может быть сконфигурирован при работе Platypus в составе J2EE сервера.

В данном режиме информация о пользователях хранится в базе данных приложения в таблице **MTD_USERS**. Каждая запись в этой таблице соответствует одному пользователю. Сведения о группах, к которым принадлежит пользователь хранятся в таблице **MTD_GROUPS**. Каждому пользователю может быть сопоставлено 0 и более групп.

Таблица **MTD_USERS** содержит следующие обязательные поля:

Поле	Описание
<i>USR_NAME</i>	Имя пользователя
<i>USR_PASSWD</i>	Хэш-сумма пароля пользователя, с использованием алгоритма MD5
<i>USR_FORM</i>	Идентификатор формы по умолчанию, например comAppFormsWelcome

Кроме того, таблица **MTD_USERS** может содержать необязательные поля, содержащие дополнительную информацию о пользователе.

Таблица **MTD_GROUPS** содержит следующие обязательные поля:

Поле	Описание
<i>USR_NAME</i>	Имя пользователя
<i>GROUP_NAME</i>	Имя группы

5.2. Использование внешнего хранилища информации о пользователях

Данный режим дает возможность использования внешних сервисов для аутентификации пользователя, например **OpenLDAP** или **Microsoft Active Directory**.

С этим режимом аутентификации могут работать Platypus Web API клиенты, HTML5 браузерный клиент и Java SE Platypus клиент при соединении с сервлет контейнером или J2EE сервером по протоколам HTTP/HTTPS. Для использования этого режима в сервлет контейнере или в J2EE сервере должна быть настроена соответствующая область безопасности - Security Realm.



Примечание

Справочную информацию по настройке области безопасности для сервера Glassfish смотрите в приложении В: "Настройка области безопасности LDAP".

Справочную информацию по настройке области безопасности для **Apache Tomcat** смотрите в документации на сервлет-контейнер **Tomcat**.

Развертывание приложений и миграции БД

Развертывание — это процесс распространения готового приложения для установки или обновления на рабочем сервере. На рабочем сервере приложение Platypus целиком располагается в базе данных, либо в виде каталога на жестком диске.

Механизм миграций обеспечивает сохранение и восстановление конфигурации и начальных данных в базе данных.

Приложения Platypus поставляются в следующих вариантах:

- на компакт или DVD диске;
- через Интернет - в виде архивного файла в формате zip.

В случае поставки приложения Platypus в виде архивного файла распакуйте его содержимое в удобное для вас место на жестком диске.

После распаковки, каталог будет иметь следующую структуру:

- **appDirectory/**
 - **src/**
 - **db/**
 - **platypus.xml**

Каталог приложения Platypus соответствует проекту приложения готового как для развертывания и исполнения, так и для модификации в **Platypus Application Designer**.

Каталог приложения содержит конфигурационный файл **platypus.xml**, каталог **src**, содержащий элементы приложения и каталог **db** в котором содержатся миграции баз данных.

База данных в которую развертывается приложение включает в себя системные таблицы и произвольное количество рабочих таблиц. При развертывании системные таблицы создаются автоматически.

К системным таблицам относятся:

- **MTD_ENTITIES** - в данной таблице хранится исполняемое приложение в виде древовидной структуры;
- **MTD_USERS** и **MTD_GROUPS** - таблицы пользователей и их групп при работе со встроенным пространством пользователей, для информации смотри раздел Безопасность.
- **MTD_VERSION** - в данной таблице хранится информация о текущей версии хранилища данных приложения.

После развертывания в БД каждый элемент приложения представляет собой одну запись в таблице **MTD_ENTITIES**.

Команда запуска утилиты развертывания и миграций БД имеет следующий вид:

```
java -jar PlatypusDeploy.jar OPTIONS
```

где *OPTIONS* - команда развертывания или миграции и набор её параметров.

Описание команд относящиеся к операции развертывания импорта и миграций смотрите в подразделах относящихся к развертыванию и миграциям БД.

Таблица 6.1. Параметры команд развертывания и миграций

Параметр	Описание
<i>-ap APP_PATH</i>	Путь к папке, содержащей готовое приложение Platypus. Если этот параметр не задан, то будет использована текущая папка.
<i>-dbuser DB_USER</i>	Имя пользователя для авторизации в БД.
<i>-dbpassword DB_PASSWORD</i>	Пароль пользователя для авторизации в БД.
<i>-dbschema DB_SCHEMA</i>	Схема БД.



Примечание

Действия по развертыванию и обновлению приложения также могут быть выполнены при помощи **Platypus Application Designer**.

6.1. Развертывание и импорт приложения



Важно

Перед выполнением обновления приложения произведите резервное копирование рабочей БД.

Во время процесса разработки приложение находится в каталоге на диске, при этом оно может запускаться для отладки и тестирования непосредственно с диска. Для промышленной эксплуатации приложение рекомендуется развертывать в базу данных.

Таблица 6.2. Команды развертывания и импорта приложения

Параметр	Описание
<i>-deploy</i>	Развертывает приложение с диска в БД.
<i>-undeploy</i>	Удаляет приложение из БД.
<i>-import</i>	Импортирует приложение из БД на диск.

В случае если параметры настройки соединения с БД или имени пользователя и пароля Platypus не заданы в опциях, информация о соединении с БД будет прочитана из файла настроек **platypus.xml** папки приложения.

Пример команды импорта приложения из БД:

```
java -jar PlatypusDeploy.jar -import -ap ~/apps/testApp -url  
jdbc:oracle:thin:@serverHost:1521:adb -dbuser user1 -dbpassword secret -dbschema testschema
```

Platypus Application Server и **Platypus Client** не требуют перезапуска после обновления рабочего приложения.

6.2. Миграции БД

Конфигурация базы данных приложения Platypus хранится в виде последовательных миграций, которые применяются последовательно для обеспечения корректной работы обновлений схемы и данных в БД. Каждое изменение структуры или служебных данных представляет собой файл миграции, название которого соответствует версии изменений, начиная с 1. Для каждой новой миграции имя файла получается путем прибавления единицы к максимальному номеру версии для данного приложения. Файлы миграций могут быть двух типов:

- мгновенный снимок метаданных БД, с расширением **.xdm**;
- пакет SQL команд для добавления и/или изменения служебных данных, с расширением **.batch**;

Таблица 6.3. Команды создания и применения миграций БД

Параметр	Описание
<i>-apply</i>	Применяет набор миграций к БД. Применяются те миграции, номер которых больше текущей версии БД.
<i>-undeploy</i>	Удаляет приложение из БД.
<i>-snapshot</i>	Создает новую миграцию - снимок метаданных БД, при этом версия БД устанавливается соответствующей этой миграции.
<i>-batch</i>	Создает новую пустую пакетную миграцию, при этом версия БД устанавливается соответствующей этой миграции.
<i>-clean</i>	Производит очистку каталога миграций, удаляя те миграции, которые не применяются.
<i>-getver</i>	Возвращает текущую версию БД.
<i>-setver</i>	устанавливает текущую версию БД в <i>VERSION</i> - целое, неотрицательное число.

В случае если настройка соединения с БД или имя пользователя и пароля Platypus не заданы в переметрах, информация о соединении с БД будет прочитана из файла настроек **platypus.xml** папки приложения.

При применении миграций к БД используется только последний моментальный снимок структуры БД, а также все моментальные снимки которые идут непосредственно перед пакетами SQL команд, остальные файлы миграций игнорируются и могут быть удалены командой *-clean*.

Пример применения миграций к БД:

```
java -jar PlatypusDeploy.jar -apply -ap ~/apps/testApp
```


Приложение А. Режимы протоколирования

Таблица А.1. Режимы протоколирования -loglevel

<i>OFF</i>	Протоколирование отключено
<i>SEVERE</i>	Выводятся сообщения только о серьезных проблемах, при которых невозможно нормально запустить приложение. Такой режим протоколирования полезен для разработчиков. Минимальный уровень.
<i>WARNING</i>	Выводятся сообщения о возможных проблемах. Такой режим протоколирования будет полезен для разработчиков и системных администраторов.
<i>INFO</i>	Выводятся информационные сообщения. В таком режиме протоколирования выводятся достаточно значимые сообщения, которые будут важны для конечных пользователей и системных администраторов.
<i>CONFIG</i>	Выводятся сообщения о конфигурации системы. Данный режим позволяет проводить отладку проблем, которые связаны с основной конфигурацией ПК. Например выводится информация о процессоре, глубине цветов, информация о подключенных модулях и пр.
<i>FINE</i>	Выводится отладочная информация. Данный уровень будет интересен разработчикам ПО.
<i>FINER</i>	Более глубокий уровень детализации
<i>FINEST</i>	Наиболее детальный вывод отладочной информации. Максимальный уровень.
<i>ALL</i>	Выводятся все сообщения

Приложение В. Настройка J2EE сервера Glassfish 3.1.2



Примечание

Для получения подробной информации по настройке **Glassfish** обратитесь к документации данного сервера приложений.

Настройка источника данных:

1. Скопируйте JAR-файл JDBC драйвера в каталог доступный загрузчику классов: **glassfish/domains/mydomain/lib**
2. Запустите консоль администрирования GlassFish. Для этого запустите сервер, например, используя утилиту `asadmin`. Далее в браузере перейдите по адресу **http://hostname:4848**, где `hostname` - адрес сервера приложений Glassfish, например: **http://localhost:4848**.
3. Создайте пул подключений JDBC: **Resources -> JDBC -> JDBC Connection Pools -> New**, Тип ресурса `javax.sql.ConnectionPoolDataSource`, а так же укажите параметры подключения к БД: `url`, `schema`, `username`, `password`;
4. Проверьте правильность настройки пула, нажав кнопку `Ping`;
5. Создайте JNDI ресурс для пула подключений: **Resources -> JDBC Resources-> New**; Задайте имя ресурса и укажите пул подключений JDBC.

Задайте параметры инициализации web-приложения `url`, `dialect` и `schema` в файле `web.xml`.

Для того, чтобы настроить Platypus для работы с внутренним хранилищем данных о пользователях или внешним сервисом аутентификации:

- Настройте J2EE сервер Glassfish для работы с областью безопасности (Realm) во внешнем LDAP сервисе;
- Настройте учетные записи пользователей.

Для того, чтобы настроить сервер Glassfish:

- Добавьте область безопасности для Glassfish - для этого измените конфигурацию сервера **Configurations -> server-config -> Security -> Realms -> New**;

Задайте имя области безопасности, выберите имя класса из списка или укажите свой класс:

- Для использования встроенного хранилища Platypus укажите имя класса `com.sun.enterprise.security.auth.realm.jdbc.JDBCRealm`; Настройте свойства, специфические для этого класса:

Таблица В.1. Свойства области безопасности JDBCRealm

<i>JAAS Context</i>	Идентификатор логин-модуля, JDBCRealm
<i>User Table</i>	Имя таблицы пользователей в БД, MTD_USERS

<i>User Name Column</i>	Имя столбца в таблице пользователей в котором хранятся имена пользователей, USR_NAME
<i>Password Column</i>	Имя столбца в таблице пользователей в котором хранятся хэши паролей, USR_PASSWD
<i>Group Table</i>	Имя таблицы групп пользователей, USR_GROUPS
<i>Group Name Column</i>	Имя столбца имен групп в таблице групп пользователей, GROUP_NAME
<i>Digest Algorithm</i>	Алгоритм хэширования пароля, MD5

- Для использования внешнего сервиса LDAP (Active Directory, OpenLDAP и т.д.) укажите имя класса com.sun.enterprise.security.auth.realm.ldap.LdapRealm; Настройте свойства, специфические для этого класса:

Таблица В.2. Основные и дополнительные свойства области безопасности LdapRealm

<i>JAAS Context</i>	Идентификатор логин-модуля, LdapRealm
<i>Directory</i>	ldap://server:389
<i>Base DN</i>	DC=ithit,DC=com
<i>Assign Groups</i>	platypus_default_role
<i>search-filter</i>	(&(objectClass=user)(sAMAccountName=%s))
<i>search-bind-password</i>	пароль сервиса LDAP
<i>group-search-filter</i>	(&(objectClass=group)(member=%d))
<i>search-bind-dn</i>	ithit\user



Примечание

Задайте значения свойств в соответствии с конфигурацией вашего LDAP-сервера. Значение свойства *Assign Groups*, группа platypus_default_role будет присвоена всем пользователям.

Сконфигурируйте JVM: Configurations -> server-config -> JVM Settings -> Add JVM Option - добавьте следующую опцию: Djava.naming.referral=follow

- Настройте режим аутентификации для web-приложения в котором будет развернут Platypus Servlet: в файле WEB-INF/web.xml для режима аутентификации укажите имя области безопасности (Realm Name), например, LdapRealm, которое было настроено на предыдущем шаге и создайте роли безопасности: добавьте следующий элемент после элемента <security-constraint> в файле web.xml, пример для BASIC аутентификации:

```
<web-app ... >
...
<login-config>
  <auth-method>BASIC</auth-method>
  <realm-name>LdapRealm</realm-name>
</login-config>
...
```

```
</web-app>
```

- Добавьте роли безопасности:

```
<web-app ... >
...
<security-role>
  <role-name>role1</role-name>
</security-role>
...
</web-app>
```

- Добавьте ограничение безопасности:

```
<web-app ... >
...
<web-resource-collection>
  <web-resource-name>platypus</web-resource-name>
  <description/>
  <url-pattern>/examples/Maps.jsp</url-pattern>
  <url-pattern>/application/*</url-pattern>
</web-resource-collection>
<auth-constraint>
  <description/>
  <role-name>platypus_default_role</role-name>
  <role-name>role1</role-name>
  <role-name>role2</role-name>
  <role-name>role3</role-name>
</auth-constraint>
</security-constraint>
...
</web-app>
```

- В файле **WEB-INF/glassfish-web.xml** свяжите роли с группами LDAP:

```
<glassfish-web-app error-url="">
...
  <context-root>/platypus</context-root>
  <security-role-mapping>
    <role-name>platypus_default_role</role-name>
    <group-name>default</group-name>
  </security-role-mapping>
  <security-role-mapping>
    <role-name>role1</role-name>
    <group-name>role1</group-name>
  </security-role-mapping>
  <security-role-mapping>
    <role-name>role2</role-name>
    <group-name>role2</group-name>
  </security-role-mapping>
  <security-role-mapping>
    <role-name>role3</role-name>
    <group-name>role3</group-name>
  </security-role-mapping>
..
</glassfish-web-app>
```

В LDAP сервиса создайте соответствующих пользователей, их группы и добавьте пользователей в эти группы.



Примечание

Для получения подробной информации по настройке сервера и клиента LDAP обратитесь к документации **OpenLDAP** и **Microsoft Active Directory**.

Приложение С. Настройка Apache Tomcat 7



Примечание

Для получения подробной информации по настройке **Apache Tomcat 7** обратитесь к его документации.

Настройка источника данных:

1. Скопируйте соответствующий JAR-файл JDBC драйвера в каталог доступный загрузчику классов: **CATALINA_HOME/lib**, где **CATALINA_HOME** - каталог в который был установлен Apache Tomcat.
2. Создайте JNDI ресурс источника данных JDBC: в файле **CATALINA_HOME/conf/server.xml** в секцию **GlobalNamingResources** добавьте элемент **Resource**. В случае, если данный ресурс будет использоваться для доступа к области безопасности **DataSourceRealm** создайте глобальный ресурс:

```
<GlobalNamingResources>
...
<Resource name="jdbc/TestDB" auth="Container" type="javax.sql.DataSource"
    maxActive="100" maxIdle="30" maxWait="10000"
    username="sa" password="te$tPwD" driverClassName="org.h2.Driver"
    url="jdbc:h2:tcp://localhost/~h2db/test;schema=test"/>
...
</GlobalNamingResources>
```



Примечание

В случае, если глобальный ресурс не требуется, определите JNDI ресурс на более низком уровне контекста.

При необходимости настройте пул соединений, задав параметры удаления и очистки неиспользуемых соединений.

3. Настройте область безопасности. Для использования встроенного хранилища отредактируйте **CATALINA_HOME/conf/server.xml**, добавив область безопасности и ссылку на глобальный JNDI ресурс:

```
...
<Engine name="Catalina" defaultHost="localhost">
...
<Realm className="org.apache.catalina.realm.DataSourceRealm"
    dataSourceName="jdbc/TestDB"
    userTable="MTD_USERS" userNameCol="USR_NAME" userCredCol="USR_PASSWD"
    userRoleTable="MTD_GROUPS" roleNameCol="GROUP_NAME" digest="MD5"/>
```

```
<Context antiJARLocking="true" path="/TomcatTest" reloadable="false">
  <ResourceLink
    name="jdbc/TestDB"
    global="jdbc/TestDB"
    type="javax.sql.DataSource" />
</Context>
...
</Engine>
```

Для области безопасности DataSourceRealm задайте названия таблиц, столбцов и алгоритма хэширования для работы с таблицами **MTD_USERS** и **MTD_GROUPS**.

Для использования другого хранилища данных аутентификации, например, внешнего LDAP-сервера настройте соответствующий тип области безопасности.

4. Настройте web-приложение Playpus - в файле **WEB-INF/web.xml** установите параметру *url* значение имени JNDI ресурса источника данных. Добавьте ссылку на глобальный ресурс для контекста web-приложения:

```
...
<context-param>
  <param-name>url</param-name>
  <param-value>jdbc/TestDB</param-value>
</context-param>
<context-param>
  <param-name>dialect</param-name>
  <param-value>Oracle</param-value>
</context-param>
<context-param>
  <param-name>schema</param-name>
  <param-value>test</param-value>
</context-param>
...
<resource-ref>
  <res-ref-name>jdbc/TestDB</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
  <res-sharing-scope>Shareable</res-sharing-scope>
</resource-ref>
...
```

5. Настройте ограничения безопасности для URL Platypus Servlet:

```
...
<security-constraint>
  <display-name>Constraint1</display-name>
  <web-resource-collection>
    <web-resource-name>res0</web-resource-name>
    <description/>
    <url-pattern>/application/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>role1</role-name>
  </auth-constraint>
</security-constraint>
...
```

Приложение D. History

Издание 1-0 16.05.2012

Алексей Кашинцев, Вадим Вашкевич, Ма-
рат Гайнуллин

Первоначальна редакция

Предметный указатель

Символы

ОТЗЫВЫ

контактная информация для данного руководства, vii

