[Тема 1. Основы ApacheMaven 4](#_Toc340432173)

[Модуль 1.1 Настройка ApacheMaven для Windows 4](#_Toc340432174)

[Задание 1 7](#_Toc340432175)

[Модуль 1.2 Создание нового проекта 8](#_Toc340432176)

[Задание 2 10](#_Toc340432177)

[Модуль 1.3 Компилирование и тестирование проекта 11](#_Toc340432178)

[Задание 3 12](#_Toc340432179)

[Модуль 1.4Описание Объектной модели проекта (POM) 14](#_Toc340432180)

[Модуль 1.5Описание жизненного цикла сборки. 16](#_Toc340432181)

[Модуль 1.6Описание профайлов сборки. 19](#_Toc340432182)

[Задание 4 21](#_Toc340432183)

[Тема 2. Инженерные техники при применении Apache Maven. 22](#_Toc340432184)

[Модуль 2.1 Автоматизация сборки 22](#_Toc340432185)

[Модуль 2.2 Модульность проекта 25](#_Toc340432186)

[Задание 5 29](#_Toc340432187)

[Модуль 2.3 Управление зависимостями 30](#_Toc340432188)

[Задание 6 35](#_Toc340432189)

[Модуль 2.4 Автоматизация установки проекта. 36](#_Toc340432190)

[Модуль 2.5 Сборка web-приложения. 40](#_Toc340432191)

[Задание 7 43](#_Toc340432192)

[Тема 3. Apache Web Сервер 44](#_Toc340432193)

[Модуль 3.1 Установка сервера Apache 44](#_Toc340432194)

[Модуль 3.2 Структура каталогов сервера Apache 53](#_Toc340432195)

[Модуль 3.3 Файл конфигурации httpd.conf 55](#_Toc340432196)

[Модуль 3.4 Разделы файла конфигурации 56](#_Toc340432197)

[Модуль 3.5 Общие директивы. Создание домашней директории пользователя, доступной при запросе http://localhost/~nik/ 58](#_Toc340432198)

[Модуль 3.6 Директивы управленияпроизводительностью 63](#_Toc340432199)

[Модуль 3.7 Директивы обеспечения постоянного соединения 65](#_Toc340432200)

[Модуль 3.8 Директивы работы с языками 66](#_Toc340432201)

[Модуль 3.9 Директивы перенаправления 68](#_Toc340432202)

[Модуль 3.10 Настройки MIME-типов 70](#_Toc340432203)

[Модуль 3.11 Файл конфигурации .htaccess. 75](#_Toc340432204)

[Модуль 3.12 Управление доступом 77](#_Toc340432205)

[Модуль 3.13 Регулярные выражения, используемые в директивах 79](#_Toc340432206)

[Модуль 3.14 Настройка сервера Apache 81](#_Toc340432207)

[Задание 8 84](#_Toc340432208)

[Тема 4. Apache Tomcat Сервер 85](#_Toc340432209)

[Модуль 4.1 Описание ApacheTomcat. 85](#_Toc340432210)

[Модуль4.2 Архитектура Tomcat 86](#_Toc340432211)

[Модуль4.3 Установка и конфигурирование Tomcat 89](#_Toc340432212)

[Модуль 4.4 Тестирование установки сервера Tomcat 95](#_Toc340432213)

[95](#_Toc340432214)

[Тема 5. Запуск сервера Tomcat 99](#_Toc340432215)

[Модуль 5.1 Структура директорий сервера Tomcat 99](#_Toc340432216)

[Задание 9 101](#_Toc340432217)

[Модуль 5.2 Выполнение скриптов сервера Tomcat 102](#_Toc340432218)

[Модуль 5.3 Передача конфигурационных опций времени выполнения скриптам Catalina. 106](#_Toc340432219)

[Тема 6. Конфигурирование сервера Tomcat 107](#_Toc340432220)

[Модуль 6.1 Установка WAR-архива 108](#_Toc340432221)

[Задание 10 111](#_Toc340432222)

[Модуль 6.2 Конфигурация хостов 112](#_Toc340432223)

[Модуль 6.3 Конфигурация контекстов 113](#_Toc340432224)

[Тема 7. Менеджер веб-приложений Tomcat 115](#_Toc340432225)

[Модуль 7.1 Что такое Менеджер веб-приложений Tomcat 115](#_Toc340432226)

[Модуль 7.2 Получение доступа к Менеджеру веб-приложений 116](#_Toc340432227)

[Модуль 7.3 Доступ к менеджеру веб-приложений, используя веб-интерфейс. 118](#_Toc340432228)

[Модуль 7.3 Прослушивание установленных веб-приложений 120](#_Toc340432229)

[Задание 11 120](#_Toc340432230)

[Модуль 7.4 Проверка статуса сервера 121](#_Toc340432231)

[Модуль 7.5 Установка нового веб-приложения 122](#_Toc340432232)

[Задание 12 123](#_Toc340432233)

[Модуль 7.6 Перезагрузка, остановка и запуск существующего веб-приложения 124](#_Toc340432234)

[Модуль 7.7 Управление сессиями веб-приложения 126](#_Toc340432235)

[Тема 8. Использование текстового интерфейса для доступа к менеджеру веб-приложений Tomcat 128](#_Toc340432236)

[Модуль 8.1 Tomcatmaven плагин 128](#_Toc340432237)

[Задание 13 131](#_Toc340432238)

[Модуль 8.2 Настройка плагина 132](#_Toc340432239)

[Тема 9. Интеграция Tomcat и Apache веб сервера 134](#_Toc340432240)

[Модуль 9.1 Настройка интеграции с помощью mod\_proxy 135](#_Toc340432241)

[Задание 14 137](#_Toc340432242)

[Модуль 9.2 Настройка интеграции с помощью mod\_jk 138](#_Toc340432243)

[Задание 15 140](#_Toc340432244)

[тема 10. Основы Subversion 141](#_Toc340432245)

[Модуль 10.1 Основные понятия управления версиями 141](#_Toc340432246)

[Модуль 10.2 Установка сервера subversion 145](#_Toc340432247)

[Задание 16 149](#_Toc340432248)

[Модуль 10.3 Установка клиента subversion и его использование 150](#_Toc340432249)

[Модуль 10.4 Сохранение своих изменений в репозиторий 155](#_Toc340432250)

[Задание 17 157](#_Toc340432251)

[Модуль 10.5 Работа с репозиторием напрямую на сервере 158](#_Toc340432252)

[Модуль 10.6 Откат изменений в рабочей копии 160](#_Toc340432253)

[Задание 18 161](#_Toc340432254)

[Модуль 10.7 Работа с утилитой LogViewer 162](#_Toc340432255)

[Тема 11. Настройка сУБД oRACLE 165](#_Toc340432256)

[Модуль 11.1 Установка сервера баз данных Oracle 11 XE 165](#_Toc340432257)

[Модуль 11.2 Соединение с Oracle 11 XE 169](#_Toc340432258)

[Задание 19. 178](#_Toc340432259)

**Тема 1.**ОсновыApacheMaven

Модуль 1.1 Настройка ApacheMaven для Windows

ApacheMaven – это программный продукт, предназначенный для упрощения сборки проектов, пришедший на смену ApacheAnt.Используя огромный потенциал Mavenа и его эффективность в решении ежедневных проблем, он стал очень популярным и широко используемым инструментом не только разработчиками, но и другими членами команд.

Официальный сайт Mavena<http://maven.apache.org>, основные его достоинства следующие:

* Делаетпроцессы сборки приложения простым
* Предоставляет унифицированную систему сборки
* Предоставляет информацию о качестве проекта
* Предоставляет описание лучших практик в разработке
* Позволяет производить прозрачную и понятную миграцию новых компонентов.

***Maven*** – это программный продукт с интерфейсом командной строки и должен быть интегрирован с переменными среды Windows. Процесс установки достаточно прост и требует установленной JavaSDK на компьютере.

Для проверки, установлена ли JavaSDK на компьютере, выполните следующую команду через командную строку Windows:

Java-version

*Результат может быть похожий на следующее:*

java version "1.6.0\_21"

Java(TM) SE Runtime Environment (build 1.6.0\_21-b07)

Java HotSpot(TM) Client VM (build 17.0-b17, mixed mode)

Если у вас не установлен **Java SDK**, тогда вы можете скачать последнюю версию на сайте http://www.oracle.com/technetwork/java/javase/downloads/index.html и установить.

|  |  |
| --- | --- |
|  | |
|  | |
|  | |
|  | |
|  | |
| **{?}** | Установлен Java SDK? |

Тогда пора установить ***apachemaven***. Скачать последнюю версию можно с сайта <http://maven.apache.org/download.html>.

После завершения скачивания Maven, распакуйте архив в любую папку. Для Windows, как правило, путь к папке не должен содержать пробелов. Я распаковал Maven на диск D: d:\opt\maven .

Для начала использования Maven необходимо настроить переменные среды Windows.**M2\_HOME** переменная должна быть установлена и **PATH**переменная должна быть модифицирована для включения папки, откуда запускается Maven.

Вы можете установить переменные среды вызовом «Настройки Системы» (System), используя «Панель Управления» (ControlPanel).

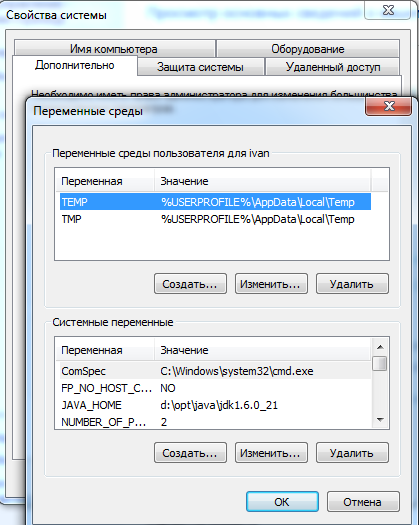


Рисунок 1.1.1Путь к изменению переменных среды.

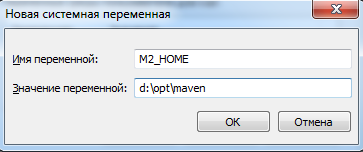


Рисунок 1.1.2Установка переменной M2\_HOME

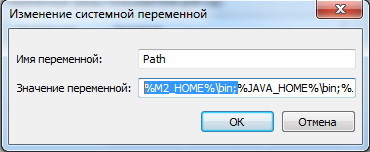


Рисунок 1.1.3 Изменение переменной PATH.

Создайте новую переменную среды M2\_HOME, указывающую путь к папке, куда был распакован Maven. Для моего компьютера значение для M2\_HOME указывает на **d:\opt\maven**.Переменная среды Path должна уже существовать. Она должна быть изменена, добавлением следующего текста в начало:

%M2\_HOME%\bin;

ApacheMaven, теперь, готов к использованию. Он, также, доступен для интеграции с IDE и другими программными средствами, предназначенными для разработки.

Прежде как использовать **Apache Maven**, вам необходимо **проверить,** как проверить установлен ли maven на вашем компьютере?

Это можно сделать с помощью следующей команды:

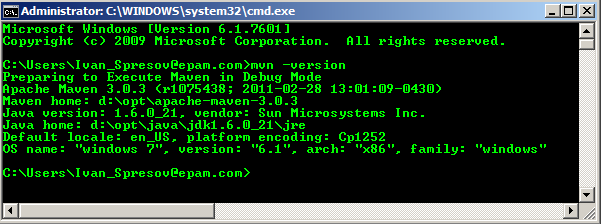


Рисунок 1.1.4. Проверка версии maven

Если результат похожий напредставленный выше, тогда вы можете быть уверенными, что maven установлен и готов к использованию.Если же ваша операционная система не может найти команду ***mvn***, тогда проверьте, что переменные среды ***path***и***M2\_HOME*** установлены корректно.

Задание 1

Возьмите установочный файл maven. Настройте environmentvariables и проверьте, что maven настроен корректно. Напишите версию Maven.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 1.2Создание нового проекта

После того, как вы установили apachemaven, вам необходимо создать новый проект. Для этого необходимо сделать следующее:

1. Запустите консоль (cmd в windows).
2. Запустите следующую команду:

mvn archetype:generate

1. Если это первый раз, когда вы запускаете эту команду, вы увидите статус скачивания в командной строке.
2. Затем вы увидите большой список архетипов, у каждого есть номер, имя, и короткое описание, описывающие, что каждый из них представляет. Выберем архетип по умолчанию. Мы предполагаем, что это архетип номер 211, с именем maven-archetype-quickstart.

[INFO] No archetype defined. Using maven-archetype-quickstart

(org.apache.maven.archetypes:maven-archetype-quickstart:1.0)

Choose archetype:

1: remote -> docbkx-quickstart-archetype (-)

….

100: remote -> maven-archetype-profiles (-)

101: remote -> maven-archetype-quickstart (An archetype which

contains a sample Maven project.)

…

375: remote -> javg-minima

Choose a number: 211:

Вас попросят выбрать **версию архетипа**. Архетип по умолчанию (default) – это последняя стабильная версия архетипа.

Затем, у вас спросят ввести параметры проекта, такие как ***groupId***, ***artifactId***, ***version*** и ***package***.

***GroupId*** параметр, используемый для определения иерархического расположения проекта в Maven***репозитории***.

В этом случае, ***репозиторий*** – это локальный репозиторий Maven, расположенный на вашем компьютере. ***GroupId*** определяет типичный каталог верхнего уровня, и это может быть использовано в нескольких проектах в организации.

***ArtifactId*** идентифицирует ваш проект и ***version*** указывает версию проекта.***ArtifactId*** используется, когда артефакты распакованные в репозиториии используется как зависимость в других проектах.

|  |  |
| --- | --- |
|  | |
| **{?}** | Установлен Java SDK? |

1. По завершению вы должны увидеть следующее:

[INFO] -------------------------------------------

[INFO] BUILD SUCCESS

[INFO] -------------------------------------------

Мы используем ***Maven***, в действительности ***Mavenplugin***, для создания или генерации нового проекта.

Новый проект***ApacheMaven*** может быть создан «вручную» cpom.xml и папками, созданными или сгенеренными в соответствии с архитектурой Maven проекта.

|  |  |
| --- | --- |
| **{?}** | Что же maven создаст для нас? |

Ответ – следующее:

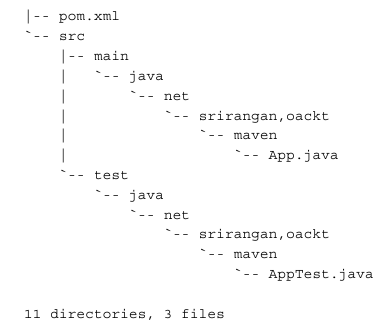


Рисунок 1.2.1 Структура файлов созданныеmaven.

Maven создал ***папку для проекта***, содержащую главный ***pom.xml*** и папку для исходных кодов ***src***, который содержит подпапки для приложения и папки для тестовых исходных кодов.

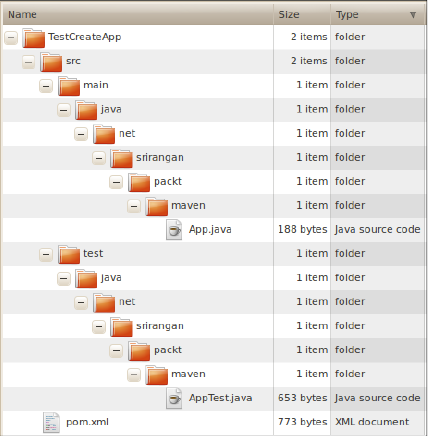


Рисунок 1.2.3. Структура файлов mavenпроекта.

Задание 2

Создайте стандартный проект maven. Настройте EclipseIDE. Напишите проблемы, которые у вас возникли.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 1.3Компилирование и тестирование проекта

Для того, что бы откомпилировать и протестировать проект с помощью maven, необходимо сделать следующее:

1. Откройте консоль ***(cmd)***
2. Перейдите в проектную папку, содержащую ***pom.xml***
3. Запустите следующую команду:

mvncompile

Apachemaven начинает скачивать зависимости (артефакты), если они не доступны в локальном репозитории, и затем компилирует проект.

[INFO] -------------------------------------------

[INFO] BUILD SUCCESS

[INFO] -------------------------------------------

Если результат-***BuildSuccess***, это значит ***mavenзавершил компиляцию*** и сборку приложения.

***TestDrivenDevelopment (TDD)*** – популярная практика, рекомендуемая большинством самых знаменитых IT-компаний с мировым именем

Maven сделал тестирование частью стандартного жизненного цикла сборки.

Что в свою сделало TDD более простым для команд, которые пытаются внедрить эту практику в процессы разработки.

Когда используется стандартное соглашение maven, то проект включает src/test папку, которая содержит все тесты для кода. Для запуска тестов, вам необходимо запустить следующую команду:

mvntest

Эта команда запускает на выполнение тесты maven, находящиеся в ***../src/test*** папки. Когда выполнение команды закончится, вы увидите отчетна консоли:

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

--------------------------------------------------------

T E S T S

--------------------------------------------------------

Running net.srirangan.packt.maven.AppTest

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.037

sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

|  |  |
| --- | --- |
| **{?}** | Как же это все работает? |

Мы запускаем следующую команду:

mvncompile

Команда запустит Java компилятор, ассоциированный с проектом. Откомпилированный код кладется в target директорию. Target директория обычно содержит откомпилированные артефакты (это может быть jar файл по умолчанию)вместе с папками для откопилированных классов и тестов.Эта папка, также, содержит pom.properties файл, а также тестовые отчеты и временные файлы.

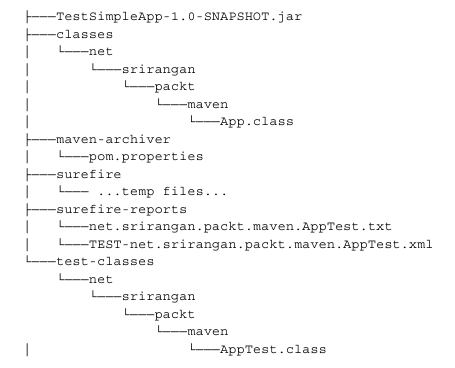


Рисунок 1.3.1Targetдиректория

|  |
| --- |
|  |

Задание 3

Измените проект, добавьте ресурсы. Скомпилируйте проект с помощью maven. Проверьте, что компиляция прошла успешна. Опишите проблемы, возникшие во время компиляции.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 1.4Описание Объектной модели проекта (POM)

Каждый ApacheMaven проект содержит файл ***pom.xml***.

***Объектная модель проекта*** – этоXML представление проекта, которая включает все метаданные, относящиеся к проекту.

***Объектная модель*** проекта содержит проектную конфигурацию, детали системы учета дефектов, проектную организацию и лицензии, проектные пути, зависимости и т.д. Структура типового стандартного файла **pom.xml** выглядит следующим образом:

**<project … >**

**<modelVersion>4.0.0</modelVersion>**

<!--Базис -->

**<groupId>...</groupId>**

**<artifactId>...</artifactId>**

**<version>...</version>**

**<packaging>...</packaging>**

**<dependencies>...</dependencies>**

**<parent>...</parent>**

**<dependencyManagement>...</dependencyManagement>**

**<modules>...</modules>**

**<properties>...</properties>**

<!—Настройкисборки -->

**<build>...</build>**

**<reporting>...</reporting>**

<!—Метаданныепроекта -->

**<name>...</name>**

**<description>...</description>**

**<url>...</url>**

**<inceptionYear>...</inceptionYear>**

**<licenses>...</licenses>**

**<organization>...</organization>**

**<developers>...</developers>**

**<contributors>...</contributors>**

<!—Средаразработки -->

**<issueManagement>...</issueManagement>**

**<ciManagement>...</ciManagement>**

**<mailingLists>...</mailingLists>**

**<scm>...</scm>**

**<prerequisites>...</prerequisites>**

**<repositories>...</repositories>**

**<pluginRepositories>...</pluginRepositories>**

**<distributionManagement>...</distributionManagement>**

**<profiles>...</profiles>**

**</project>**

Минимальноечислообязательныхполейpomфайлавключаетвсебя - groupId, artifactIdиversion. Эти три поля определяют расположение артефакта в репозитории.

Представленный выше пример ***pom файла*** включает 4 основные секции:

1. ***Базис***. Эта секция содержит поля, определяющие уникальность артефакта (groupId, artifactId и version), управление зависимостями, а также детали наследования.
2. ***Настройки сборки***. Детали сборки располагаются в этой секции.
3. ***Метаданные проекта***. Эта секция включает специфичные для проекта параметры, такие как имя, организация, разработчики, адрес сайта организации, которая разрабатывает продукт, год поставки и т.д.
4. ***Среда разработки***. Эта секция содержит всю информацию, касающуюся среды разработки, включая детали о системе контроля версий, системе управления дефектами, системы непрерывной сборки, списки рассылок, репозиториях и т.д.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 1.5Описание жизненного цикла сборки.

***Жизненный цикл сборки*** точно определяет процесс сборки, тестирования, выпуска артефактов, и это является сердцем каждого maven проекта.

Существует три встроенных жизненных циклов сборки: *стандартный, очистить исайт.*

*Стандартный жизненный цикл* обрабатывает компиляцию проекта, тестирование и установка. Он содержит более чем 20 фаз, наиболее важные из которых перечислены ниже:

1. ***Проверка (validate)***: проверяет, что вся проектная информация доступна и корректна.

|  |
| --- |
|  |
|  |
|  |

1. ***Компилирование (compile):*** компилирует исходный код.

|  |
| --- |
|  |
|  |
|  |

1. ***Тестирование (test)***: запускает модульные тесты.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

1. ***Упаковка (package)***: упаковывает откомпилированный код в соответствующий выходной формат.

|  |
| --- |
|  |
|  |
|  |

1. ***Интеграционное тестирование (integration-test)***: запускает интеграционные тесты на тестовой среде.

|  |
| --- |
|  |
|  |
|  |

1. ***Проверка упаковки (verify)***: Запускает проверки на правильность упаковки.

|  |
| --- |
|  |
|  |
|  |

1. ***Установка (Install)***: устанавливает упакованный артефакт на локальный репозиторий.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

1. ***Установка на удаленный репозиторий (deploy):*** установка финального упакованного артефакта на удаленный репозиторий.

|  |
| --- |
|  |
|  |
|  |

Как бы вы не запускали на выполнение фазу сборки, все предыдущие фазы выполняются последовательно. Например, выполняя команду

mvnintegration-test,

вы запускаете проверку, компиляцию, тестирование и упаковку, до того момента пока выполнится фаза интеграционного тестирования.

***Жизненный цикл очистки***отвечает за процесс очистки проекта от предыдущих файлов сборкии состоит из следующих фаз:

1. ***Подготовка к очистке (pre-clean):*** выполняет процессы необходимые до очистки предыдущих файлов сборки проекта.
2. ***Очистка (clean):*** удаляет все файлы полученные при предыдущей сборки.
3. ***Фаза после очистки (post-clean):*** выполняет процессы необходимые для завершения очистки файлов предыдущей сборки проекта.

***Жизненный цикл сайт*** отвечает за генерацию и установку документации проекта, состоит из следующих фаз:

1. Подготовка к генерации документации (pre-clean): выполняет процессы необходимые до генерации документации проекта.
2. Генерация документации (site): генерация проектной документации.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 1.6Описание профайлов сборки.

Проекты в maven портируемы. Портируемость достигается посредством***конфигурации POM***, обходя все особенности файловой системы, и сильно зависима от локального репозитория, в котором сохранены необходимые метаданные.

Однако, не всегда возможно достичь портируемости, благодаря внутренним взаимозависимостям конфигурации и файловой системы. В таких случаях maven представляет концепцию ***профайлов сборки***.

***Профайл сборки*** *–* это настройки, описанные в pom.xml, которые могут быть запущены когда необходимо.

Существуют следующие варианты запуска профайлов:

1. Запуск профайла через командную строку.
2. Запуск профайла в зависимости от настроек maven.
3. Запуск профайла в зависимости от среды разработки.

Профайл может быть запущен через ***командную строку*** при помощи параметра –P. Список профайлов, разделенных запятыми, которые должны быть активированы, должны быть перечислены после флага –P.

mvn install -P profile-1,profile-2

|  |
| --- |
|  |
|  |
|  |

В этом случае, только *профайлы*, указанные в команде, все же другие сконфигурированные *профайлы*, будут ***неактивны***.

Обратное действие также ***возможно***. Вы можете указать, какие профайлы не должны быть активированы.

mvn install -P !profile-1,!profile-2

|  |
| --- |
|  |
|  |
|  |
|  |

Настройки maven могут активировать профайлы, если они указанны в секции

<activeProfiles>файлаsettings.xml.

<settings>

...

<activeProfiles>

<activeProfile>profile-1</activeProfile>

<activeProfile>profile-2</activeProfile>

</activeProfiles>

...

</settings>

Профайлы, указанные в настройках maven активируются по умолчанию каждый раз и их не нужно явно указывать в командной строке.

Профайлы могут быть запущены в зависимости от среды разработки. Среда, в которой профайл должен быть активирован, определяется в секции <profiles> файла pom.xml.

<profiles>

<profile>

<activation>

<property>

<name>environment</name>

<value>dev</value>

</property>

</activation>

</profile>

</profiles>

В коде, указанном выше, профайл может быть активирован в dev среде разработке. **Пример** типичной команды maven, в которой среда разработки указана явно, выглядит так:

mvn groupId:artifactId:goal -Denvironment=dev

|  |
| --- |
|  |
|  |
|  |

Задание 4

Настройте профиль development. Настройте следующий плагин:

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-war-plugin</artifactId>

<version>2.0</version>

<configuration>

<webappDirectory>

apache-tomcat-6.0.29\webapps\example

</webappDirectory>

</configuration>

</plugin>

Ипроверьтеегоработу.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Тема 2.**Инженерные техники при применении ApacheMaven.

Модуль 2.1 Автоматизация сборки

***Автоматизация сборки*** *–* это реализованные на языке программирования задачи, которые разработчики выполняют ежедневно для запуска и тестирования написанного кода.

Эти задачи включают:

1. Компиляцию исходного кода в исполняемый код.

|  |
| --- |
|  |
|  |
|  |

1. Упаковка исполняемого кода.

|  |
| --- |
|  |
|  |
|  |

1. Запуск тестов.

|  |
| --- |
|  |
|  |
|  |

1. Установка упакованного исполняемого кода на удаленные системы.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

1. Создание документации к релизу и продукту.

|  |
| --- |
|  |
|  |
|  |

Автоматизация сборки представляет ряд преимуществ, включая ускорение сборки, ограничение неудачных сборок, стандартизация и унификация в командах и организациях, увеличение эффективности и улучшение в продуктовом качестве. Сегодня, эта практика необходима для ежедневного использования.

Если вам необходим ***готовый maven проект***. Если у вас нет готового, запустите следующую команду в командной строке, для создания простого Java проекта:

mvn archetype:generate -DgroupId=com.epam.maven

-DartifactId=MySampleApp

Команда archetype:generateсоздалApache Maven проектдлянас. Если мы выберем maven-archetype-quickstartархетип из списка, наша структура проекта будет выглядеть похожей на рисунок 2.1.1.



Рисунок 2.1.1 Структура проекта

В каждом ApacheMaven проект, включая тот, который мы сгенерировали, сборка запускает автоматически стандартный жизненный цикл сборки.

|  |
| --- |
|  |
|  |
|  |

Тоже самое вы можете выполнить вручную:

mvn validate

...

mvn compile

...

mvn package

...

mvntest

...

Вы можете включить некоторые фазы жизненного цикла сборки индивидуальной командой. Maven позволяет вам ***автоматизировать запуск всех фаз сборки в правильном порядке***.

Запустите команду mvninstall и вы запустите часть стандартного жизненного цикла сборки, включая компилирование, тестирование, упаковку и установку артефакта в локальный репозиторий.

Автоматизация сборки стандартизирует структуру проекта, что может упростить разработку и тестирование продуктов.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 2.2 Модульность проекта

Предположим, что вы собираете большое приложение, требующее взаимодействие со старой базой данных, работающее с существующими сервисами, предоставляющие современные графические интерфейсы как для web- и прикладных-составляющих, и предоставляющих API другим приложениям заказчиков. Такая сборка подразумевает разделение этого достаточно большого проекта на ***подпроекты или модули****.*

Apachemaven предоставляет верную и точную поддержку, для такой проектной организации, через ***ApacheMaven мультимодульные проекты***. Мультимодульные проекты состоят из ***«родительского проекта»***, который включает ***«дочерние проекты»*** или модули.

|  |
| --- |
|  |
|  |
|  |

Родительский pom файл содержит ссылки на все дочерние модули. Каждый модуль может быть разного типа, с разным упаковочным расширением.

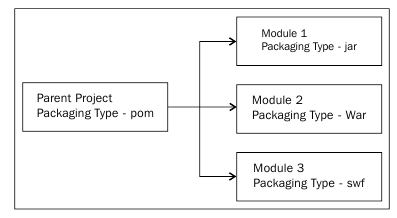


Рисунок 2.2.1 Мультимодульный проект.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |

Мы начинаем с создания родительского модуля. Не забудьте поставить значение ***packaging* в *pom***, как представлено в следующем ***примере***:

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi=<http://www.w3.org/2001/XMLSchema-instance>

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.epam.maven</groupId>

<artifactId>TestModularApp</artifactId>

<version>1.0-SNAPSHOT</version>

<packaging>pom</packaging>

<name>MyLargeModularApp</name>

</project>

Это ***базовый родительский pom-файл*** для нашего проекта MyLargeModularApp. Он не содержит никакие дочерние модули на данный момент.

|  |
| --- |
|  |
|  |
|  |

Для создания нашего ***первого дочернего модуля***, запустите консоль командной строки (cmd), перейдите в директорию, где лежит родительский pom- файл и запустите следующую команду:

mvnarchetype:generate

Данная команда покажет список всех архетипов на ваш выбор. Вы можете выбрать ***архетип maven-archetype-quickstart***, который сгенерирует базовый Java модуль. Команда *archetype:generate*требует заполнения параметров maven проекта: groupId, artifactId, package, version и др.

|  |
| --- |
|  |
|  |
|  |
|  |

После генерации модуля, просмотрите родительский pom файл, и вы увидите, что следующий блок был добавлен в секцию модулей:

<modules>

<module>moduleJar</module>

</modules>

Дочерний модуль, который мы создали, автоматически был добавлен в проект к родительскому модулю. Эта процедура работает достаточно просто – ***никакого внешнего вмешательства не требуется***.

Сейчас мы создадим другой дочерний модуль. Пришло время создать web-модуль, путем вызова следующей команды:

mvn archetype:generate -DarchetypeArtifactId=maven-archetype-webapp

Давайте, снова, посмотрим на родительский pom файл, и мы увидим, что оба дочерних модуля добавлены в секцию модули:

<modules>

<module>moduleJar</module>

<module>moduleWar</module>

</modules>

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Общая структура проекта (структура директорий) теперь имеет вид, как на рисунке 2.2.2.

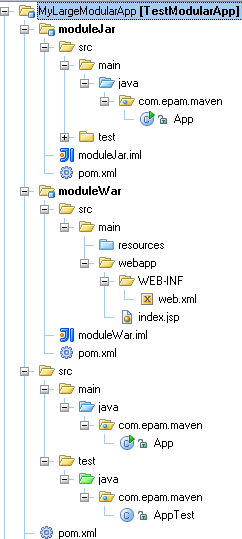


Рисунок 2.2.2 Структура директорий нашего мульти модульного проекта.

Компиляция и установка (еще бывает и распаковка на сервере) ***обоих дочерних модулей***(в правильном порядке, в случае, если дочерние модули взаимозависимы) – это ***главное*.** Данная операция может быть сделана, путем запуска командной строки, перехода в директорию, где располагается родительский pom файл, и запуска следующей команды:

mvncleaninstall

Данная команда, запуская процесс сборки для родительского модуля, так же автоматически запускает сборку всех дочерних модулей в правильном порядке.

|  |
| --- |
|  |
|  |
|  |

Вы должны увидеть результат в командной строке, подобный следующему:

---------------------------------------------------------------------------------------

[INFO] Reactor Summary:

[INFO] MyLargeModularApp ................................ SUCCESS [0.439s]

[INFO] MyModuleJar ..............................................SUCCESS [3.047s]

[INFO] MyModuleWar Maven Webapp ................. SUCCESS [0.947s]

----------------------------------------------------------------------------------------

[INFO] BUILD SUCCESS

----------------------------------------------------------------------------------------

Задание 5

Создайте многомодульный проект. Одни модуль - jar, другой модуль – war. Создайте parent модуль. Проверьте процесс сборки и опишите ваши вопросы.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 2.3 Управление зависимостями

***Управление зависимостями***заслуженно считается одной из лучших возможностей ApacheMaven. В мультимодульных проектах, где количество зависимостей достигает десятков или даже сотен, ApacheMaven отличается от других систем сборки, тем, что позволяет сохранять высокий уровень контроля и стабильности системы.

***Управление зависимостями в ApacheMaven***– это прозрачный процесс. Это значит, что maven автоматически находит артефакты, от которых зависит ваш проект. Эта возможность стала доступной, начиная с версии Maven 2, и это стало особенно удобно для многих проектов с открытым исходным кодом (opensourceprojects) и для их зависимостей, которые мы используемв сегодняшних проектах.

Зависимости в maven имеют 6 возможных областей действий:

1. Зависимость, необходимая для компилирования (***compile***): Эта область действия по умолчанию. Зависимость доступна в classpath.

|  |
| --- |
|  |
|  |
|  |

1. Зависимость, представлена во времени выполнения (***provided***): Эта область действия предполагает, что JDK или среда разработки предоставляет зависимость во времени выполнения.

|  |
| --- |
|  |
|  |
|  |

1. Зависимость, представлена во время выполнения (***runtime***): Зависимость необходима во время выполнения и определена в runtimeclasspath.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

1. Зависимость, необходима для тестирования (***Test***): Зависимости необходимы для компиляции и выполнения тестов.

|  |
| --- |
|  |
|  |
|  |

1. Системная зависимость (***System***): Зависимость всегда доступна, и в любом случае наличие JAR обеспечено.

|  |
| --- |
|  |
|  |
|  |

1. Импортрируемая зависимость (***Import***):Имортрируемая зависимость представлена в POM, включая <dependencyManagement/> секцию.

|  |
| --- |
|  |
|  |
|  |

Зависимости для ApacheMavenпроекта описаны в проектномpom-файле. Пока мы глубже начали рассматривать, как работать с зависимостями, мы рассмотрим ApacheMaven плагин управления зависимостями (***ApacheMavendependencyplugin***).

В соответствии с <http://maven.apache.org/plugins/maven-dependency-plugin/>:

*«Плагин управления зависимостями предоставляет возможность манипулирования артефактами. Он может копировать или распаковывать артефакты из локального или удаленного репозитория в указанное место на жестком диске»*

|  |
| --- |
|  |
|  |

Это плагин, который предоставляет ряд удобных задач (goals), которые заключаются в следующем:

1. ***mvndependency:analyze.*** Анализирует зависимости (используемые, неиспользуемые, декларируемые или не декларируемые зависимости).

|  |
| --- |
|  |
|  |
|  |

1. ***mvndependency:analyze-duplicate***. Определяет дублирующие друг друга зависимости.

|  |
| --- |
|  |
|  |
|  |

1. ***mvndependency:resolve***. Определить все зависимости.

|  |
| --- |
|  |
|  |
|  |

1. ***mvn dependency:resolve-plugin***. Определитьвсеплагины.

|  |
| --- |
|  |
|  |
|  |

1. ***mvndependency:tree***. Определить дерево зависимостей.

|  |
| --- |
|  |
|  |
|  |
|  |

Большинство maven проектов зависимы от других артефактов (других проектов, библиотек и инструментов). Управление и интеграция зависимостями является одной из основных особенностей maven. Зависимостипроектауказанывpom-файле.

<dependencies>

<dependency>

<groupId>...</groupId>

<artifactId>...</artifactId>

<version>...</version>

<scope>...</scope>

</dependency>

</dependencies>

В мультимодульных проектах, зависимости могут быть определены в родительскомpom-файле и могут впоследствии унаследованы в дочерних pom- файлах, когда это необходимо. Расположение всех зависимостей в одном источнике делает версионность зависимостей более простой, что делает большие проектные зависимости более управляемыми и масштабируемыми во времени.

|  |
| --- |
|  |
|  |

Следующий ***пример*** показывает мультимодульный проект, обладающий зависимостью от MySql.

Родительский pom будет содержать полное описание зависимости:

<dependencyManagement>

<dependencies>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>5.1.2</version>

</dependency>

<dependencies>

</dependencyManagement>

|  |
| --- |
|  |
|  |
|  |

Все дочерние модули требуют MySql будут включать опрощенное определение зависимости:

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

</dependency>

При таком описании не будет конфликта версий между многочисленными дочерними модулями, имеющие те же зависимости.

Область действия и тип зависимости по умолчанию – это ***compile и JAR***. Однако, они могут быть переопределены по требованию:

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.8.2</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>...</groupId>

<artifactId>...</artifactId>

<version>...</version>

<type>war</type>

</dependency>

Системные зависимости не ищутся в репозитории. Мы должны указывать путь к таким зависимостям сами:

<dependencies>

<dependency>

<groupId>sun.jdk</groupId>

<artifactId>tools</artifactId>

<version>1.5.0</version>

<scope>system</scope>

<systemPath>${java.home}/../lib/tools.jar</systemPath>

</dependency>

</dependencies>

|  |
| --- |
|  |
|  |

Однако очень ***рекомендовано избегать системных зависимостей***, потому что это противоречит общей концепции управления зависимостями ApacheMaven.

В идеале, разработчик должен иметь возможность взять код из системы контроля версий и запустить ApacheMaven командой. После этого, ApacheMaven должен отвечать за управление зависимостями.

Задание 6

Настройте зависимость многомодульного проекта на Oracledatabasedriver. Выполните сборку проекта и проверьте, что она прошла успешно.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 2.4 Автоматизация установки проекта.

***MavenDeploy плагин*** используется для добавления артефактов на удаленный репозиторий, в процессе фазы установки в жизненном цикле сборки.

Deploy плагин представляет две задачи:

1. ***deploy:deploy***: Установка проекта и всех его артефактов.

|  |
| --- |
|  |
|  |

1. ***deploy:deploy-file***: Установка одного простого артефакта.

|  |
| --- |
|  |
|  |

Установка в репозиторий означает не только копирование артефакта в директорию, но для обновления метаданных, так же как и артефактов. Это требует:

1. ***Targetrepository:*** Целевой репозиторий – это где артефакты будут установлены. Это необходимое расположение, протокол доступа (FTP, SCP, SFTP) и специфическая пользовательская информация о профиле.

|  |
| --- |
|  |
|  |

1. ***Targetartifacts***: Целевые артефакты – это артефакты, которые будут установлены. GroupId, artifactId, version, packaging, и classifier информация артефакта необходимы.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

1. ***Deployermethod***: Методустановки, с помощью которого происходит установка. Метод может быть реализован либо в кросс- платформенном, либо в платформенном виде.

|  |
| --- |
|  |
|  |

Проектный pom файл должен иметь<distributionManagement/> секцию с<repository/> элементами внутри, в которых описаны расположение и параметры удаленного репозитория.

<distributionManagement>

<repository>

<id>by.park</id>

<name>MyPrivateRepository</name>

<url>...</url>

</repository>

</distributionManagement>

Дляэтого, вамнеобходимоописатьинформациюосервереввашемsettings.xml (<USER\_HOME>/.m2/settings.xmlили<M2\_HOME>/conf/settings.xml) дляпредоставленияидентификационнойинформациидлярепозиториев.

<server>

<id>by.park</id>

<username>ivan</username>

<password>myTopSecretPassword</password>

</server>

Установка может быть выполнена из командной строки, перейдя в директорию проекта и вызвав следующую команду:

mvndeploy

ApacheMavenDeploy плагин вызывается в процессе выполнения deploy фазы жизненного цикла сборки. Как мы видели, <distributionManagement/> секция с <repository/> элементом, необходима для запуска работы плагина.

|  |
| --- |
|  |
|  |
|  |
|  |

Snapshots и releases могут быть разделены определением <snapshotRepository/> элемента.

<distributionManagement>

<repository>

<uniqueVersion>false</uniqueVersion>

<id>corp1</id>

<name>Corporate Repository</name>

<url>scp://repo/maven2</url>

<layout>default</layout>

</repository>

<snapshotRepository>

<uniqueVersion>true</uniqueVersion>

<id>propSnap</id>

<name>Propellors Snapshots</name>

<url>sftp://propellers.net/maven</url>

<layout>legacy</layout>

</snapshotRepository>

...

</distributionManagement>

Секция <distributionManagement/> наследуется всеми дочерними модулями. Действительная установка происходит на основе протоколов, определенных в репозитории. Чаще всего используют FTP и SSH протоколы. Wagon-ftp и wagon-ssh-external – основные провайдеры для этих протоколов.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Если удаленный репозиторий доступен через FTP протокол, то <build/> секция должна включать определение wagon-ftp.

<distributionManagement>

<repository>

<id>sri-ftp-repository</id>

<url>ftp://...</url>

</repository>

</distributionManagement>

<build>

<extensions>

<extension>

<groupId>org.apache.maven.wagon</groupId>

<artifactId>wagon-ftp</artifactId>

<version>1.0-beta-6</version>

</extension>

</extensions>

</build>

ЕслиудаленныйрепозиторийдоступенчерезSSH, определяемоепоartifactIdиURL, тогда<build/>секциявключаетопределениеwagon-ssh-external.

<distributionManagement>

<repository>

<id>sri-ssh-repository</id>

<url>scpexe://....</url>

</repository>

</distributionManagement>

<build>

<extensions>

<extension>

<groupId>org.apache.maven.wagon</groupId>

<artifactId>wagon-ssh-external</artifactId>

<version>1.0-beta-6</version>

</extension>

</extensions>

</build>

Идентификационная информация будет храниться в settings.xml.

|  |
| --- |
|  |

Модуль 2.5 Сборка web-приложения.

|  |  |
| --- | --- |
| **{?}** | Как же создать и собрать web-приложение с использованием maven? |

Мы начнем с концепции ***POWA*** (простое web приложение) – это web проект, состоящий из сервлетов и нескольких JSP. Шаблон, который может использо-ваться для генерации POWA называется org.sonatype.mavenbook.simpleweb.

Откройте командную строку и запустите следующую команду:

mvn archetype:generate -DarchetypeArtifactId=maven-archetype-webapp -DartifactId=testWebApp -DgroupId=com.epam.maven -Dversion=1.0-SNAPSHOT -Dpackage=com.epam.maven

Вы должны увидеть следующее в консоли:

[INFO] ------------------------------------------------------------------

[INFO] Using following parameters for creating project from Old (1.x) Archetype: maven-archetype-webapp:1.0

[INFO] ------------------------------------------------------------------

[INFO] Parameter: groupId, Value: com.epam.maven

[INFO] Parameter: packageName, Value: com.epam.maven

[INFO] Parameter: package, Value: com.epam.maven

[INFO] Parameter: artifactId, Value: testWebApp

[INFO] Parameter: basedir, Value: G:\Work\maven

[INFO] Parameter: version, Value: 1.0-SNAPSHOT

[INFO] project created from Old (1.x) Archetype in dir: G:\Work\maven\testWebApp

[INFO] ------------------------------------------------------------------

[INFO] BUILD SUCCESS

[INFO] ------------------------------------------------------------------

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Комнада archetype:generateсоздала новую проектную директорию testWebApp, которая содержит следующийpom-файл:

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.epam.maven</groupId>

<artifactId>testWebApp</artifactId>

<packaging>war</packaging>

<version>1.0-SNAPSHOT</version>

<name>testWebApp Maven Webapp</name>

<url>http://maven.apache.org</url>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>3.8.1</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<finalName>testWebApp</finalName>

</build>

</project>

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Сейчас, мы добавим buildplugin, для компилирования проекта с помощью JDK 1.6. <build/> секция pom файла должна иметь похожий вид:

..

<build>

<finalName>testWebApp</finalName>

<plugins>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<configuration>

<source>1.6</source>

<target>1.6</target>

</configuration>

</plugin>

</plugins>

</build>

..

ApacheMavenwebпроектбудетработать, какмногиедругиеmavenпроекты.Структура проекта изображена на рисунке 2.5.1.

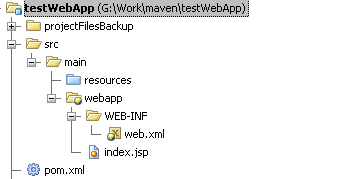


Рисунок 2.5.1 Структура maven web проекта.

Вы можете выполнить следующие команды для компилирования и тестирования проекта:

1. mvn compile
2. mvn test
3. mvn install

По окончанию установки, выходной артефакт будет упакован и установлен в локальный репозиторий, какдругие maven проекты. Он также может быть установлен на удаленный репозиторий.

Если вы посмотрите, вы можете найти проектную директорию src/main/webapp, в которой находится файл index.jsp. Вы можете изменить этот файл, как вам необходимо, например вот так:

<html>

<body>

<h2>Maven is very useful tool</h2>

</body>

</html>

Итак, давайте запустим сборку проекта - mvninstall.

В консоли мы увидем следующий результат:

[INFO] --- maven-install-plugin:2.3.1:install (default-install) @ testWebApp

[INFO] Installing G:\Work\maven\testWebApp\target\testWebApp.war to f:\opt\globoforce\repository\com\epam\maven\testWebApp\1.0-SNAPSHOT\testWebApp-1.0-SNAPSHOT.war

[INFO] Installing G:\Work\maven\testWebApp\pom.xml to f:\opt\globoforce\repository\com\epam\maven\testWebApp\1.0-SNAPSHOT\testWebApp-1.0-SNAPSHOT.pom

[INFO] ----------------------------------------------------------------------

[INFO] BUILD SUCCESS

[INFO] ----------------------------------------------------------------------

Как мы видим, после успешной сборки, ***target*** директория была создана. В target директории вы можете найти WAR файл с именем ***testWebApp.war***. Давайте посмотрим на pom файл, упакованный в данном архиве.

...

<artifactId>testWebApp</artifactId>

**<packaging>war</packaging>**

<version>1.0-SNAPSHOT</version>

<name>testWebApp Maven Webapp</name>

...

Как мы можем видеть, что packaging типустановлен в WAR и выходной артефакт отражает это свойство.

Задание 7

Создайте новое web приложение. Добавьте туда исходные файлы, которые вы делали для учебного проекта. Скомпилируйте и соберите готовое веб приложение.

|  |
| --- |
|  |
|  |
|  |
|  |

**Тема 3.**ApacheWebСервер

Модуль 3.1 Установка сервера Apache

***Web-сервер Apache*** — программное обеспечение, отвечающее за отображение документов, запрашиваемых при наборе URL-адреса в командной строке Web-браузера.

Найтидистрибутивсервера Apacheможнопоадресу http://httpd.apache.org/download.cgi.Вприведенномспискевыбираем apache\_2.2.14-win32-x86-no\_ssl.msi.

Копируемнасвойкомпьютеризапускаемфайл ***apache\_2.2.14-win32-x86-no\_ssl.msi***. В итоге отобразится окно мастера установки.

1. Нажимаем Next.

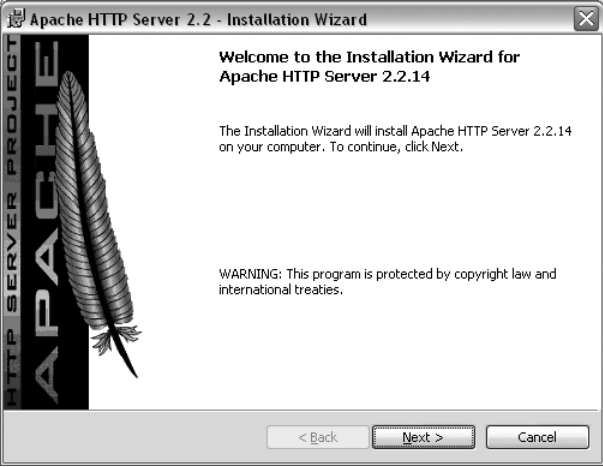


Рисунок 3.1.1Установка сервера Apache. Шаг 1

|  |
| --- |
|  |
|  |
|  |
|  |

1. Отобразитсяокнослицензионнымсоглашением.Принимаем лицензионноесоглашение.Дляэтогоустанавливаемфлажокнапротив пункта***Iacceptthetermsinthelicenseagreement*** (Я принимаю условия пользовательского соглашения). Нажимаем Next.

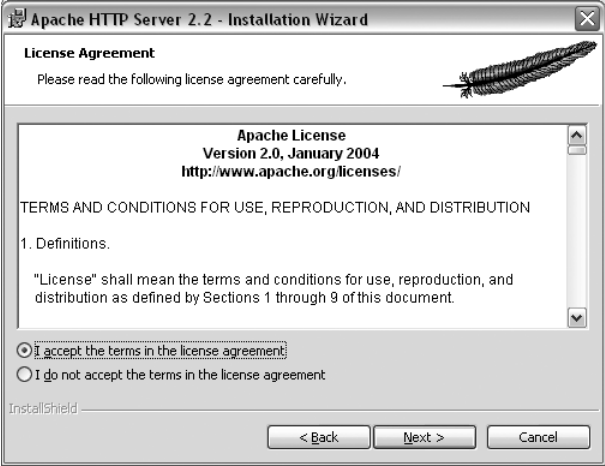


Рисунок 3.1.2. Установка сервера Apache. Шаг 2

|  |
| --- |
|  |
|  |
|  |

1. В открывшемся окне нажимаем Next.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |



Рисунок 3.1.3Установка сервера Apache. Шаг 3

1. Заполняем следующие поля:
2. ***NetworkDomain*** (Название домена) — указываем localhost;
3. ***ServerName*** (Название сервера) — указываем localhost;
4. ***Administator's Email Address*** (E-mailадминистраторасервера) —вводим любой адрес E-mail.



Рисунок 3.1.4Установка сервера Apache. Шаг 4

1. Выбираем пункт ***Typical*** и нажимаем Next.



Рисунок 3.1.5 Установка сервера Apache. Шаг 5

|  |
| --- |
|  |
|  |
|  |

1. Нажимаемкнопку***Change***иизменяемпапкудляустановкисC:\ProgramFiles\ApacheSoftwareFoundation\Apache2.2\ наC:\Apache2\ . Нажимаем Next.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |



Рисунок 3.1.6Установка сервера Apache. Шаг 6

1. Для начала установки нажимаем ***Install***.



Рисунок 3.1.7 Установка сервера Apache. Шаг 7

1. Для завершения установки нажимаем Finish.

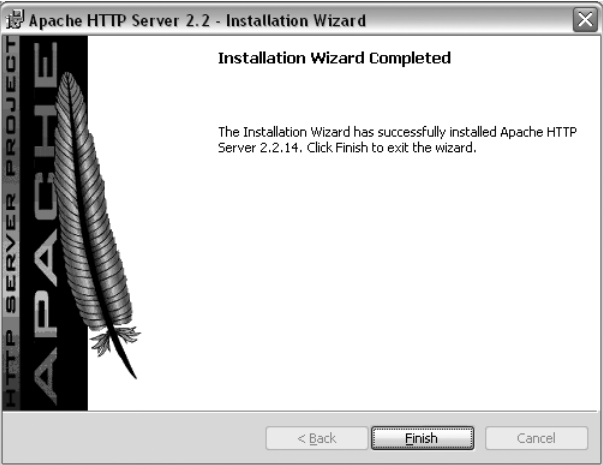


Рисунок 3.1.8 Установка сервера Apache. Шаг 8

|  |
| --- |
|  |
|  |
|  |

Еслисерверуспешноустановлен,товправомнижнемуглуэкранаотобразится перо с зеленым треугольником в центре круга.



Рисунок 3.1.9 Установка сервера Apache. Шаг 9

|  |
| --- |
|  |
|  |
|  |
|  |

Для проверки открываем Web-браузер и в адресной строке набираем: <http://localhost/>. Нажимаемклавишу <Enter>.Еслисерверустановленправильно,товокне Web-браузера отобразится стартовая страница сервера Apache.

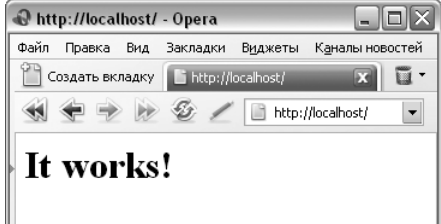


Рисунок 3.1.10 Стартовая страница сервера Apache

|  |
| --- |
|  |
|  |
|  |

Обратите внимание, если на компьютере включен Брандмауэр Windows или другой сетевой экран, то необходимо добавить Web-сервер Apache в список исключений. Обычно при первом запуске (сразу после установки) выводится запрос о блокировании программы. В этом диалоговом окне следует нажать кнопку «Разблокировать». Если диалоговое окно не отобразилось, то необходимо вручную добавить Apache в список исключений. Для этого (для Брандмауэра Windows) ***в меню Пуск выбираем Настройка | Панель управления***. Далее выбираем пункт ***Брандмауэр Windows***. В открывшемся окне выбираемвкладку Исключения. Есливспискенетпункта ApacheHTTPServer, нажимаемДобавитьпрограмму.Воткрывшемсяокненажимаемкнопку Обзор.Находим файл httpd.exe (C:\Apache2\bin\httpd.exe)инажимаемОткрыть,послечегонажимаем OK.Теперьубедимся,чтофлажокнапротив пункта ApacheHTTPServer установлен. Нажимаем OK для выхода из окна свойств Брандмауэра Windows.

|  |
| --- |
|  |
|  |
|  |

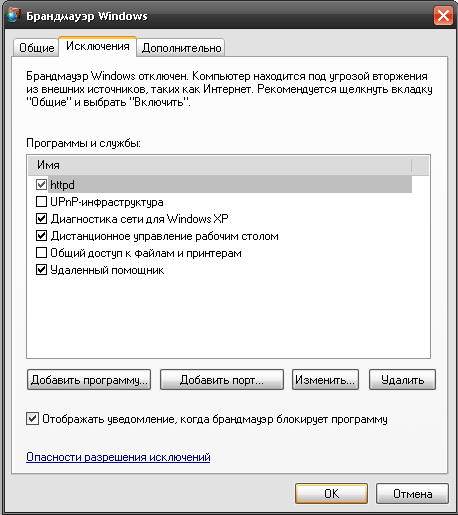


Рисунок 3.1.11 Исключение сервера Apache из списка блокируемых программ Брандмауэром Windows.

|  |
| --- |
|  |
|  |

Последобавлениянеобходимозапустить (илиперезапустить) Web-сервер. Это можно сделать следующими способами:

* в меню Пуск выбираем пункт Программы (Все программы). Находим пункт ApacheHTTPServer 2.2. Отображаем подменю, в котором находим пункт ControlApacheServer. Отображаем соответствующее подменю, в котором выбираем пункт Start или Restart.

|  |
| --- |
|  |
|  |

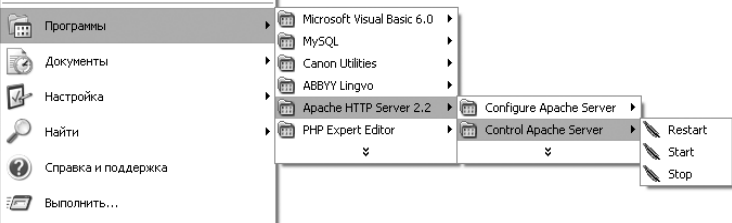


Рисунок 3.1.12 Перезапуск сервера Apache. Вариант 1

|  |
| --- |
|  |
|  |

* в правом нижнем углу окна находим пиктограмму с изображением пера. Щелкаемлевойкнопкоймыши.Выбираемпункт Apache2.2.Впоявившемся окне выбираем пункт Start или Restart.

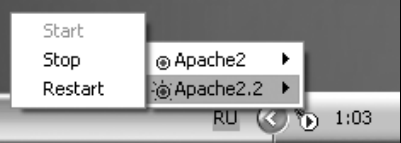


Рисунок 3.1.13 Перезапуск сервера Apache. Вариант 2

|  |
| --- |
|  |
|  |

Перезагружать Web-сервер нужно будет после ***каждого изменения в настройках***.Поэтомуспособыперезагрузкиследуетзапомнить.Вдальнейшеммы будем просто говорить "перезагрузите сервер" без явного описания способов, позволяющих это сделать.

|  |
| --- |
|  |
|  |
|  |

Модуль 3.2 Структура каталогов сервера Apache

Итак,серверустановленизапущен.Теперьдавайтерассмотримкаталоги сервера Apache, их содержание и назначение. В папке C:\Apache2\ находятся следующие каталоги:

* ***bin*** —здесьрасполагаетсяглавныйисполняемыйфайлсервера (httpd.exe);

|  |
| --- |
|  |
|  |

* ***cgi-bin*** —каталогдля CGI-программ (программ,написанныхнаязыках Perl, С и т. д.);

|  |
| --- |
|  |
|  |

* ***conf***— папка, где находятся конфигурационный файл сервера (httpd.conf) и другие файлы конфигурации сервера Apache;

|  |
| --- |
|  |
|  |

* ***error*** — каталог для файлов с сообщениями об ошибках (например, если запрашиваемый файл не найден);

|  |
| --- |
|  |
|  |

* ***htdocs***—папка,вкоторойдолжнырасполагатьсяфайлывформате HTML а также другие файлы, которые будут доступны при наборе вадреснойстроке Web-браузераhttp://localhost/ (например,изображения, файлы каскадной таблицы стилей и т. д.).

|  |
| --- |
|  |
|  |

* ***icons*** —здесьсодержитсярядизображений,используемыхвлистингах каталогов;

|  |
| --- |
|  |
|  |

* ***logs*** —вэтойпапкенаходятсяжурналырегистрациипосещений (access.log) и ошибок (error.log). Эти журналы позволяют получить подробную информацию обовсехзапросах и ошибках. Открытьэти файлы можнос помощью любоготекстового редактора (например,с помощью Блокнота);

|  |
| --- |
|  |
|  |

* ***manual*** —здесьнаходятсяфайлыдокументации.Просматриватьдокументацию следует не в этом каталоге, а набрав в командной строке Web-браузераhttp://localhost/manual/.Частьинформациидоступнанарусскомязыке. Не пытайтесь набратьэтотадрес прямо сейчас. Документация по этому адресу будет доступна после того, как мы внесем изменения в конфигурационный файл;

|  |
| --- |
|  |
|  |

* ***modules*** — этот каталог содержит подключаемые модули.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 3.3 Файл конфигурации httpd.conf

Файл ***httpd.conf*** (C:\Apache2\conf\httpd.conf)—этоосновнойфайлконфигурациисервера.Открыть и отредактировать файл можно с помощью любого текстового редактора, например,с помощью Блокнота. После любого измененияв файлеконфигурациинеобходимо***перезагрузитьсервер***. До***перезагрузки***он будет работать со ***старыми параметрами***.

Вфайле httpd.confсодержатсядирективы,влияющиенаработусервера Apache. ***Директива*** представляет собой ключевое слово, за которым следуетодноилинесколькозначений.Директивыбываютпростыми (изменяющие только односвойствосервера),а могут объединятьсяв разделы (позволяют изменять сразу несколько свойств какого-нибудь объекта).

Если в начале строки указан символ "#", то такая строка является комментарием:

# ServerAdmin: Your address, where problems with the server should be

# e-mailed.This address appears on some server-generated pages, such

# as error documents.e.g. admin@your-domain.com

ServerAdmin admin@localhost

|  |
| --- |
|  |
|  |

Вэтом примере первыетристрокизакомментированы,ачетвертаяс помощью директивы ServerAdmin задает E-mail-адрес администратора сервера. Вставлять комментарий в середину строки нельзя.

Следуетобратитьвниманиенаиспользование***косыхчерт***впутикпапке. Путькпапке htdocsвоперационнойсистеме Windowsзаписываетсякак C:\Apache2\htdocs. А в файле конфигурации сервера Apache тот же путь будет выглядеть по-другому:

C:/Apache2/htdocs

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 3.4 Разделы файла конфигурации

Директивы могут объединяться в ***разделы,*** что позволяет ограничить область действия директив отдельным каталогом, набором файлов или набором URL. Существуют следующие разделы:

* ***DirectoryиDirectoryMatch***указывают,чтодирективыприменимык заданному каталогу и всем подкаталогам:

<Directory "C:/Apache2/htdocs">

Options -Indexes

</Directory>

***DirectoryMatch*** позволяет использовать регулярные выражения;

|  |
| --- |
|  |
|  |

* ***FilesиFilesMatch***указывают,чтодирективыприменимытолькокопределенным файлам. Символ **\*** соответствует любой последовательности символов, а символ **?** — любому одиночному символу.В качестве примера запретим доступ к текстовым файлам:

<Files \*.txt>

Deny from all

</Files>

***FilesMatch*** позволяет использовать регулярные выражения;

|  |
| --- |
|  |
|  |

* ***IfModule***указывает,чтодирективыбудутиспользованылишьвслучае загрузки указанного модуля:

<IfModule dir\_module>

DirectoryIndex index.html

</IfModule>

|  |
| --- |
|  |
|  |
|  |

* ***LimitиLimitExcept***.***Limit*** указывает,чтодирективыбудутиспользоваться,толькокогда HTTP-запросвыполненспомощьюодногоизуказанных методов (GET, POST или HEAD). ***LimitExcept*** ограничивает доступ для методов, которые не заданы;

<Limit GET POST OPTIONS PROPFIND>

Order allow,deny

Allow from all

</Limit>

|  |
| --- |
|  |
|  |

* ***Location и LocationMatch*** определяют соответствие между URL-адресом и нефайловым ресурсом или между URL-адресом и реальным файлом:

<Location /server-status>

SetHandler server-status

Order deny,allow

Deny from all

Allow from .localhost

</Location>

***LocationMatch***позволяетиспользоватьрегулярныевыражения;

|  |
| --- |
|  |
|  |

* ***VirtualHost***указывает, что директивы применимы только к документам указанноговиртуального хоста. Применяется, когда сервер обслуживает множество Web-сайтов с разными именами хостов.

<VirtualHost 192.168.0.1:80>

ServerAdmin webmaster@site.ru

DocumentRoot /www/docs/site.ru

ServerNamesite.ru

</VirtualHost>

Модуль 3.5 Общиедирективы. Создание домашней директории пользователя, доступной при запросе <http://localhost/~nik/>

Перечислим основные общие директивы сервера Apache:

* ***ServerName*** определяет имя сервера:

ServerName localhost:80

|  |
| --- |
|  |
|  |

* ***ServerAdmin*** задает E-mail администратора сервера:

ServerAdmin [admin@localhost](mailto:unicross@mail.ru)

|  |
| --- |
|  |
|  |

* ***ServerRoot***указывает местонахождение каталогов сервера:

ServerRoot "C:/Apache2"

|  |
| --- |
|  |
|  |

* ***DocumentRoot***определяетместонахождениекорневогокаталогадлядокументов на сервере:

DocumentRoot "C:/Apache2/htdocs"

|  |
| --- |
|  |
|  |

* ***UserDir*** задает имя каталога, в котором ищутся домашние каталоги пользователей при получении запроса вроде <http://localhost/~user/>:

UserDir "My Documents/My Website"

|  |
| --- |
|  |
|  |
|  |

Создадим каталог для пользователя nik. Для этого добавим в C:\Apache2 папку ***user.*** В папке ***user*** создаем папку ***nik***. Внутрь папки ***nik*** добавляем файл ***index.html*** со следующим содержанием:

<html>

<head><title>Страничка пользователя Nik</title></head>

<body>Привет всем</body>

</html>

ДалееспомощьюБлокнотаоткрываем файл httpd-userdir.conf (который находится в папке C:\Apache2\conf\extra) и изменяем значение директивы

UserDir на

UserDir ***"C:/Apache2/user"***

Находимстроку

<Directory **"C:/Documents and Settings/\*/My Documents/My Website"**>

и заменяем ее на

<Directory **"C:/Apache2/user**">

Сохраняем и закрываем файл. Теперь файл httpd-userdir.conf необходимо подключитькосновномуконфигурационномуфайлу.Открываемфайл httpd.conf и убираем символ комментария (#) перед строками

#Includeconf/extra/httpd-userdir.conf

#LoadModule userdir\_module modules/mod\_userdir.so

Сохраняемизакрываемфайл httpd.conf.Перезапускаемсервер Apache. Далееоткрываем Web-браузер ивадреснойстрокенабираем http://localhost/~nik/. В итоге в окне Web-браузера должна отобразиться надпись "Привет всем";

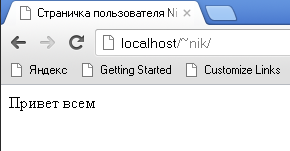


Рисунок 3.5.1 Пользовательский каталог

|  |
| --- |
|  |
|  |

* ***PidFile*** указывает местоположение файла, в котором будет регистрироваться исходный процесс сервера:

PidFile logs/httpd.pid

|  |
| --- |
|  |
|  |

* ***Listen*** связывает Apache с определенным портом и (или) IP-адресом:

Listen 80

Listen 12.34.56.78:80

|  |
| --- |
|  |
|  |

* ***Options*** позволяет включить или отключить те или иные опции в различных частях сайта. Если опция используется со знаком "+", то она добавляется к числу уже включенных опций, а если со знаком "–", то опция отключается. Если знаки "+" или "–" не указаны, то это означает, что надо выключить все установленные ранее опции и включить те, которые заданы непосредственно. Могут быть заданы следующие опции:
* ***All*** включает все опции, кроме MultiViews:

Options All

* ***None*** отключает все опции, кроме MultiViews:

Options None

|  |
| --- |
|  |
|  |

* ***ExecCGI*** позволяетвыполнять CGI-программывкаталоге,отличном от указанного в директиве ScriptAlias, например, в каталоге с обычными документами. Для правильной работы необходимо указать директиву AddHandler или SetHandler:

|  |
| --- |
|  |
|  |
|  |

<Directory "C:/Apache2/htdocs">

Options +ExecCGI

SetHandler cgi-script

</Directory>

* ***FollowSymLinks*** разрешает использование символических ссылок:

Options +FollowSymLinks

|  |
| --- |
|  |
|  |

* ***SymLinksIfOwnerMatch***разрешаетиспользованиесимволическихссылок, если ссылка указывает на объект, который принадлежит тому же пользователю, что и ссылка:

Options +SymLinksIfOwnerMatch

|  |
| --- |
|  |
|  |

* ***Includes*** разрешает использование серверных расширений (SSI):

Options +Includes

|  |
| --- |
|  |
|  |

* ***IncludesNOEXEC*** разрешает использование серверных расширений, но запрещает использование команды #exec и применение #include для загрузки CGI-программ:

Options +IncludesNOEXEC

|  |
| --- |
|  |
|  |
|  |
|  |

* ***Indexes*** — если эта опция включена и заданный по умолчанию файл не найден, то сервер генерирует листинг файлов. Если опция выключена, то вместо файла отображается сообщение об ошибке 403.

<Directory "C:/Apache2/htdocs">

Options -Indexes

</Directory>

Навиртуальномхостингеэтаопциядолжнабытьобязательновыключена, иначе пользователь будет видеть все содержимое каталога, в том числе и файлы паролей;

|  |
| --- |
|  |
|  |

* ***MultiViews***включает content-соответствие —средство,спомощью которого сервер определяет, какой документ наиболее приемлем для посетителя:

Options +MultiViews

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 3.6 Директивы управленияпроизводительностью

При увеличении нагрузки на сервер создаются новые процессы, а при снижении нагрузки эти процессы закрываются. Частые запуски и остановки порожденных процессов приводят к снижению производительности сервера.

Поэтому необходимо правильно настроить следующие директивы:

* ***StartServers*** — количество копий процесса сервера, которые будут созданы при запуске сервера;

|  |
| --- |
|  |
|  |

* ***MinSpareServers*** — минимальное число порожденных процессов;

|  |
| --- |
|  |
|  |

* ***MaxSpareServers*** — максимальное число порожденных процессов;

|  |
| --- |
|  |
|  |

* ***MaxClients*** — максимальное число возможных подключений к серверу.

Указанные директивы не применимы к платформе Windows. Вместонихиспользуются StartThreads, MinSpareThreads, MaxSpareThreads и MaxThreadsPerChild.Также применяются следующие директивы:

* ***ThreadsPerChild*** задает максимальное количество потоков, порождаемых каждым дочерним процессом сервера Apache:

ThreadsPerChild 250

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

* ***MaxRequestsPerChild***определяет,сколькозапросовможетобработать порожденный процесс за время его существования. Для снятия ограниченийнеобходимоуказать 0.Наплатформе Windowsдирективавсегда должна задавать значение 0:

MaxRequestsPerChild 0

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 3.7 Директивы обеспечения постоянного соединения

За обеспечение постоянного соединения отвечают следующие директивы:

* ***Timeout*** задает промежуток времени в секундах, в течение которого серверпродолжаетпопыткивозобновленияприостановленнойпередачи данных:

Timeout 300

|  |
| --- |
|  |
|  |

* ***KeepAlive*** разрешает постоянные соединения:

KeepAlive On

|  |
| --- |
|  |
|  |

* ***MaxKeepAliveRequests*** ограничивает число допустимых запросов на одно соединение:

MaxKeepAliveRequests 100

|  |
| --- |
|  |
|  |

* ***KeepAliveTimeout***определяет тайм-аут для постоянного соединения:

KeepAliveTimeout 15

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 3.8 Директивы работы с языками

Для работы с языками используются следующие директивы:

* ***AddDefaultCharset*** указывает язык для документов по умолчанию:

AddDefaultCharset windows-1251

|  |
| --- |
|  |
|  |

* ***AddCharset*** устанавливаетвзаимосвязьмеждукодовойтаблицейсимволов и расширением файла:

AddCharset ISO-2022-JP .jis

|  |
| --- |
|  |
|  |

* ***RemoveCharset*** удаляет взаимосвязь между кодовой таблицей символов и расширением файла:

RemoveCharset .jis

|  |
| --- |
|  |
|  |

* ***AddLanguage***устанавливаетвзаимосвязьмеждуязыкомирасширением файла:

AddLanguage ru .ru

|  |
| --- |
|  |
|  |

* ***RemoveLanguage*** удаляет все взаимосвязи между языками и расширениями файла:

RemoveLanguage .ru

|  |
| --- |
|  |
|  |

* ***DefaultLanguage*** определяет, какой язык должен быть указан в заголовке, если для расширения файла не указан определенный язык:

DefaultLanguage ru

* ***LanguagePriority***задает приоритет различных языков:

LanguagePriority ru en ca cs da de el

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 3.9 Директивы перенаправления

Перечислим основные директивы перенаправления:

* ***Alias*** и ***AliasMatch*** позволяют предоставить доступ не только к файлам, находящимсявкаталоге,указанномвдирективеDocumentRoot,ноик другим каталогам сервера. В директиве ***AliasMatch*** можно использовать регулярные выражения:

AliasMatch ^/manual(?:/(?:de|en|es|ru))?(/.\*)?$"C:/Apache2 /manual$1"

|  |
| --- |
|  |
|  |

* ***ScriptAlias***и***ScriptAliasMatch***задаютместоположениекаталогадля CGI-сценариев:

ScriptAlias /cgi-bin/ "C:/Apache2/cgi-bin/"

Директива***ScriptAliasMatch***позволяетиспользоватьрегулярныевыражения;

|  |
| --- |
|  |
|  |

* ***Redirect***и***RedirectMatch***сообщают,чтоискомыйдокументбольшене находится в данном месте, и указывают, где можно его найти. Директива RedirectMatch позволяет использоватьрегулярныевыражения. ДирективыRedirectиRedirectMatchимеютдополнительныйпараметр,указывающийсостояниепереадресации. Параметрможетприниматьследующие значения:
  + permanent — ресурс перемещен навсегда (код 301);
  + temp — ресурс перемещен временно (код 302);
  + seeother — ресурс был заменен другим ресурсом (код 303);
  + gone — ресурс удален навсегда (код 410).

**Например:**

Redirect permanent /file1.html /file2.html

RedirectMatch 301 ^/manual(?:/(de|en|es|ru)){2,}(/.\*)?$

/manual/$1$2

|  |
| --- |
|  |

* ***Обработка ошибок***

С помощью директивы ***ErrorDocument*** можно указать документ, который будет выдан Web-браузеру в случае возникновения указанной ошибки:

ErrorDocument 404 /err/error404.html

Обычноуказываютсядирективы (иразрабатываютсясоответствующиедокументы) для следующих ошибок:

* **401** — пользователь неавторизован;
* **403** —нетдоступа. ПриотсутствиииндексногофайлавкаталогеиотключеннойопцииIndexesдирективыOptionsгенерируетсяименноэта ошибка;
* **404** — ресурс не найден.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 3.10 Настройки MIME-типов

Припередаче файласерверуказывает ***MIME-типдокумента***. Это позволяет Web-браузеруправильнообработатьполучаемый файл. MIME-типуказывается в формате:

<Категория>/<Тип файла>

**Например:**

* **text/html**— для HTML-документов;
* **image/gif —**для изображений в формате GIF;
* **application/msword**— для документов в формате Word.

Конфигурации MIME-типов находятсяв файле mime.types (C:\Apache2\conf\ mime.types). Для настройки MIME-типов и смежных вопросов используются следующие директивы:

* ***AddEncoding*** устанавливает взаимосвязь между определенной кодировкой и расширением файла:

AddEncoding pkzip .zip

|  |
| --- |
|  |
|  |

* ***RemoveEncoding*** удаляет взаимосвязь между определенной кодировкой и расширением файла:

RemoveEncoding .zip

|  |
| --- |
|  |
|  |

* ***TypesConfig***указываетрасположениеконфигурационногофайласнастройками MIME-типов:

TypesConfig conf/mime.types

|  |
| --- |
|  |
|  |
|  |
|  |

* ***DefaultType***устанавливает MIME-типпоумолчанию:еслизапрашиваемый клиентом файл несоответствует ни одному из MIME-типов,то используется MIME-тип, указанный в этой директиве:

DefaultType text/plain

|  |
| --- |
|  |
|  |

* ***AddType*** позволяет добавить новый MIME-тип и связать его с определенным расширением:

AddType application/x-httpd-php .php

|  |
| --- |
|  |
|  |

* ***RemoveType*** удаляет связи между MIME-типами и расширениями:

RemoveType .cgi

|  |
| --- |
|  |
|  |

* ***ForceType*** указывает MIME-тип для набора файлов. Присваивает файлам, указанным в разделе <Directory> или <Files>, определенныйMIME-тип, не принимая во внимание расширения файлов;

|  |
| --- |
|  |
|  |

* ***AddHandler***используетсядлясвязыванияопределенногообработчикас файловым расширением:

AddHandlertype-map .var

|  |
| --- |
|  |
|  |
|  |

* ***SetHandler*** обеспечивает обработку файлов в разделах <Directory> или <Files> с помощью определенного обработчика:

<Files \*.html>

SetHandler type-map

</Files>

|  |
| --- |
|  |
|  |

* ***RemoveHandler***отменяетсвязываниеопределенногообработчикасфайловым расширением:

AddHandler server-parsed .html

RemoveHandler .html

Вдирективах ***AddHandler*** и ***SetHandler***могутбыть указаныследующие обработчики:

* default-handler — обработчик по умолчанию, который используетсядляобслуживания HTML-документов,файловизображений (то есть файлов, не требующих предварительной обработки);

|  |
| --- |
|  |
|  |

* send-as-is —посылаетфайл,содержащийвсебе HTTP-заголовки, как есть (без добавления пакетных или HTTP-заголовков). Заголовки можноуказыватьвсамом файле,отделяя ихотосновногосодержимого пустой строкой;

|  |
| --- |
|  |
|  |

* cgi-script — обрабатывает файл как CGI-скрипт;

|  |
| --- |
|  |
|  |

* imap-file — обрабатывает файл как карту-изображение;
* server-parsed — исполняет SSI-директивы в файле;

|  |
| --- |
|  |
|  |

* server-info — возвращает конфигурационную информацию сервера.

Необходимо, чтобы был подключен модуль mod\_info.so:

<Location /info>

SetHandler server-info

</Location>

|  |
| --- |
|  |
|  |

* server-status —возвращаетотчетосостояниисервера. Необходимо, чтобы был подключен модуль mod\_status.so:

<Location /status>

SetHandler server-status

</Location>

|  |
| --- |
|  |
|  |

* type-map — обрабатывает файл как файл сопоставления типов:

AddHandler type-map .var

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

В этом примере все файлы с расширением var будут использоваться как файлы сопоставления типов. Пример файла сопоставления типов:

URI: index.html.en

Content-Language: en

Content-type: text/html; charset=ISO-8859-1

URI: index.html.ru.koi8-r

Content-Language: ru

Content-type: text/html; charset=KOI8-R

* Action устанавливает соответствие между заданным названием обработчикаили MIME-типасопределенной программой,обеспечивающей механизм исполнения. Даннаядиректива позволяетсоздаватьсобственные обработчики:

Action image/gif /cgi-bin/images.cgi

Action my-file-type /cgi-bin/program.cgi

AddHandler my-file-type .xyz

|  |
| --- |
|  |
|  |

* CacheNegotiatedDocsзадаетрежимкэшированиясерверомрезультатов переговоров: если директива имеетзначение on, то документы, установленные в результате переговоров между сервером и Web-браузером о согласовании MIME-типа, языка и способа кодирования, могут быть помещены в кэш:

CacheNegotiatedDocs on

По умолчанию директива имеет значение off.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 3.11 Файл конфигурации .htaccess.

На виртуальном хостинге не предоставляется доступ к главному файлу конфигурации, так как один сервер может обслуживать множество сайтов, принадлежащихразличнымлюдям.Вэтомслучаедляконфигурированияотдельных каталогов используется файл .htaccess. При изменении этого файла нетнеобходимостиперезагружатьсервер.Файлы .htaccessанализируются при каждом запросе файла из каталога.

Если сервер в полном распоряжении, то настраивать конфигурацию необходимо в файле httpd.conf, а использование файлов .htaccess нужно запретить. Это связано с тем, что использование файлов .htaccess сильно влияет на производительностьизащиту. Файл httpd.confанализируетсятолькоодинраз (при запуске сервера), а файлы .htaccess анализируются при каждом запросе. Еслииспользованиефайлов .htaccessзапрещено,то Apacheдаженебудет искать эти файлы в каталогах.

Для настройки файлов .htaccess используются следующие директивы:

1. ***AccessFileName***задает имя файла конфигурации:

AccessFileName .htaccess

|  |
| --- |
|  |
|  |

1. ***AllowOverride*** позволяет ограничить перечень директив, которые позволеноизменятьв файлах .htaccess. Директива можетприниматьследующие значения:

* ***All*** позволяет пользователям переопределятьв файлах .htaccessглобальные параметры доступа:

AllowOverride All

|  |
| --- |
|  |
|  |

* ***None*** отключает использование файла .htaccess:

AllowOverride None

|  |
| --- |
|  |

* ***AuthConfig***позволяетиспользованиедирективавторизации (AuthName, AuthType, AuthUserFile, AuthGroupFile идр.):

AllowOverride AuthConfig

* ***FileInfo***разрешаетиспользованиедиректив,управляющихтипами документов (AddType, AddLanguage, AddEncoding, ErrorDocument, LanguagePriory и др.):

AllowOverride FileInfo

|  |
| --- |
|  |
|  |

* ***Indexes***позволяетиспользованиедиректив,управляющихиндексациейкаталога (AddIcon, DirectoryIndex, FancyIndexing, HeaderName идр.):

AllowOverride Indexes

|  |
| --- |
|  |
|  |

* ***Limit*** делает возможным использование директив, управляющих доступом к хостам (Allow, Deny и Order):

AllowOverride Limit

|  |
| --- |
|  |
|  |

* ***Options***разрешаетиспользованиедиректив,управляющихкаталогами (Options и XbitHack):

AllowOverride Options

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 3.12 Управление доступом

Директива***Order***определяетпорядокприменениядиректив***Allow*** (разрешить) и ***Deny*** (запретить). Например,с помощьюэтихдиректив можноразрешить доступ к папке только лицам, пришедшим с определенного IP-адрес или определенного домена.

Order deny, allow

Deny from all

Allow from 192.168.0.1

|  |
| --- |
|  |
|  |

Значениеallуказываетнавсеадреса.Вкачествезначенийадресаможно указать неполныйIP-адрес или неполное имя домена. Для полного доступа к папке можно использовать следующий код:

Orderallow, deny

Allow from all

|  |
| --- |
|  |
|  |

Если определено несколько критериев доступа к папке, то директива ***Satisfy*** задает, должны ли быть выполнены все условия (значение all) или хотя бы одно из условий (значение any).

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

AuthType Basic

AuthName "restricted area"

AuthUserFile "C:/Apache2/data/pass.conf"

<Limit GET POST>

require valid-user

</Limit>

Deny from all

Allow from 127.0.

Satisfyany

В этом примере, если клиент пришел с локального хоста, то он получит доступ к папке без пароля, а остальным будет выведено окно с запросом пароля.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 3.13 Регулярные выражения, используемые в директивах

Некоторыедирективыпозволяют использовать***регулярныевыражения.*** Эти выражения мало чем отличаются от регулярных выражений, используемых в JavaScript. Внихможноиспользоватьследующиеметасимволы и специальные конструкции:

* **^** — привязка к началу строки;
* **$** — привязка к концу строки;
* **[]** —позволяетуказатьсимволы,которыемогутвстречатьсянаэтом месте в строке. Можно перечислять символы подряд или указать диапазон через тире;
* **[^]** — значение можно инвертировать, если после первой скобки указать символ^.Такимобразомможноуказатьсимволы,которыхнедолжно быть на этом месте в строке.

Для использования специальных символов как обычных необходимо перед специальным символом указать символ **"\"**;

* **\d** — любая цифра;
* **\w** — любая латинская буква, цифра или знак подчеркивания;
* **\s** —любой непечатныйсимвол (пробел,табуляция, переводстраницы, новая строка или перевод каретки);
* **.** (точка) — любой символ, кроме символа перевода строки (\n);
* **\D** — не цифра;
* **\W** — не латинская буква, не цифра и не знак подчеркивания;
* **\S** — не непечатный символ;
* **\< и \>** — пустая строка перед указанным шаблоном и после него;
* **{n}** — в точности n вхождений предыдущего символа или подвыражения в строку;
* **{n,}** — n или более вхождений символа в строку;
* **{n,m}** —не менееnвхожденийсимволавстрокуинеболее m. Цифры указываются через запятую без пробела;
* **\***— ноль или большее число вхождений символа в строку;
* **+** — один или большее число вхождений символа в строку;
* ? — ноль или одно число вхождений символа в строку;
* **n|m** — один из символов n или m.

Регулярное выражение можно разбить на подвыражения с помощью круглых скобок.Каждаягруппасимволов,соответствующихподвыражению,сохраняется в памяти. В дальнейшем группу символов можно извлечь, указав после символа $ номер скобки:

**AliasMatch ^/manual(?:/(?:de|en|es|fr|ru))?(/.\*)?$"C:/Apache2/manual$1"**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 3.14 Настройка сервера Apache

Для нормальной работы необходимо изменить значения некоторых директив в конфигурационных файлах сервера Apache.

Для начала нужно настроить сервер на работу с **русским языком**. Открываем файл ***httpd-languages.conf***(расположен в папке C:\Apache2\conf\extra) и заменяем строку

# DefaultLanguage **nl**

на

DefaultLanguage**ru**

Далее находим строку

LanguagePriority en ca cs da de el eo es et fr he hr it ja ko ltz nl nn no pl pt pt-BR ru sv tr zh-CN zh-TW

и ставим русский язык (ru) на первое место:

LanguagePriority**ru**en ca cs da de el eo es et fr he hr it ja ko ltz nl nn no pl pt pt-BR sv tr zh-CN zh-TW

Вконецфайладобавляемстроку

AddDefaultCharset windows-1251

Сохраняемизакрываемфайл**httpd-languages.conf**.

Теперь внесем изменения в главный конфигурационный файл сервера Apache. Открываем файл httpd.conf и проверяем значения следующих директив:

ServerRoot "C:/Apache2"

Listen 80

DocumentRoot "C:/Apache2/htdocs"

Убираем комментарий (#) перед строкой

# ServerName localhost:80

|  |
| --- |
|  |
|  |
|  |
|  |

Чтобы иметь возможность использовать файл конфигурации .htaccess, необходимо включить его поддержку. Для этого находим раздел

<Directory "C:/Apache2/htdocs">

…

</Directory>

Внутри раздела находим строки

Options Indexes FollowSymLinks

AllowOverride None

изаменяемихна

Options **-Indexes** Includes FollowSymLinks

AllowOverride **All**

Находимстроки

<Directory />

Options FollowSymLinks

AllowOverride None

Order deny,allow

Deny from all

</Directory>

именяемихна

<Directory />

Options -Indexes Includes FollowSymLinks

AllowOverride All

Order allow,deny

Allow from all

</Directory>

|  |
| --- |
|  |
|  |
|  |

Внутри раздела <Directory "C:/Apache2/cgi-bin"> заменяем строку

AllowOverride None

на

AllowOverride **All**

Заменяемстроку

#AddHandler cgi-script .cgi

на

AddHandler cgi-script .cgi .**pl**

Убираемсимволкомментария (#) передстроками

#LoadModule rewrite\_module modules/mod\_rewrite.so

#AddType text/html .shtml

#AddOutputFilter INCLUDES .shtml

#Include conf/extra/httpd-autoindex.conf

#Include conf/extra/httpd-default.conf

#Include conf/extra/httpd-languages.conf

#Include conf/extra/httpd-manual.conf

Сохраняем файл httpd.conf и перезагружаем сервер. Теперь можно просматриватьдокументациюксерверу Apache,набраввадреснойстроке http://localhost/manual/. Установка сервера Apache закончена.

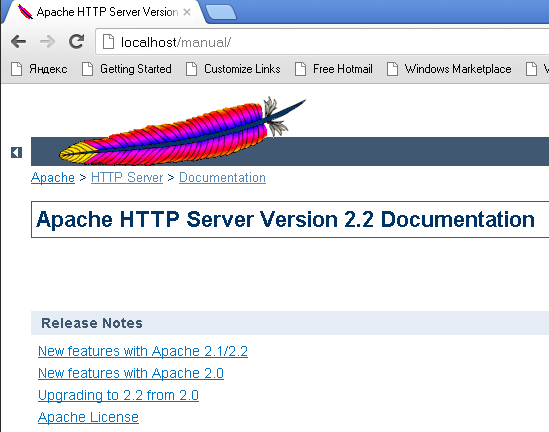


Рисунок 3.14.1Настроенный Apache

Задание 8

Установите и настройте ApacheWeb сервер. Проверьте правильность конфигурирования и работы сервера. Разместите в папке htdocs статические ресурсы (html файлы, картинки и т.д.) и проверьте их доступность.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Тема4.**ApacheTomcatСервер

Модуль 4.1 Описание ApacheTomcat.

***ApacheTomcat сервер*** – это основанный на платформе Java контейнер для веб-приложений. Сервер создан для запуска веб-приложений, основанных на сервлетах и JavaServerPages (JSP). Он был создан как подпроект ***Apache-Jakarta***; однако, благодаря своей популярности, сейчас, существует как отдельный Apache проект, который поддерживается группой волонтеров сообщества JAVA.

ApacheTomcat очень стабилен и обладает всеми достоинствами коммерческого сервера веб-приложений и выпускается под открытой лицензией Apache.

Сейчас поддерживаемыми версиями сервера Tomcat являются 5.5X, 6.0X, и 7.0X. Версии, выпущенные ранее, чем 5.5, все еще доступны для скачивания, но они не поддерживаются больше. Основные версии ApacheTomcat соответствуют версиям JavaServlet спецификации, или JavaServletAPI. Так, Tomcat 5.5X поддерживает ServletAPI 2.3, Tomcat 6.0X поддерживает ServletAPI 2.4, и последнийTomcat 7.0 поддерживает имплементацию ServletAPI 3.0. В дополнение к ServletAPI,версии Tomcat соответствуют версиям JSPAPI.

JVM совместимость так же зависит от выбранной версии сервера Tomcat.

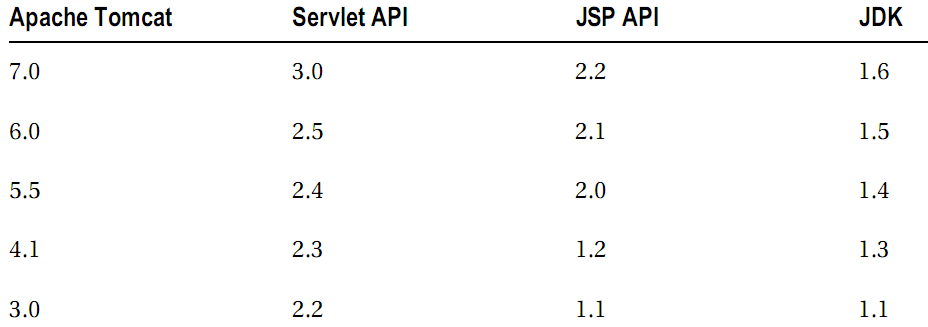


Рисунок 4.1.1 Версии Tomcatсервера и соответствующие им версии Servlet, JSPAPIи JDK.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

Модуль4.2 Архитектура Tomcat

***Сервер Tomcat*** – это высокоуровневый компонент в иерархии контейнеров Tomcat. Только одна запущенная сущность Tomcat сервера может работать в рамках JVM. Такой подход делает работу всех других Java приложений, запущенных на том же компьютере, что и сервер Tomcat, ***безопасной***, в случае краха JVM или Tomcat.

|  |
| --- |
|  |
|  |

Сервер Tomcat состоит из группы ***контейнеров приложений***, которые существуют в строго определенной иерархии. Ключевой компонент иерархии – это ***Catalina***. ***Catalina***– это Java сервлет контейнер, реализованный в соответствии со спецификацией JavaServletAPI.

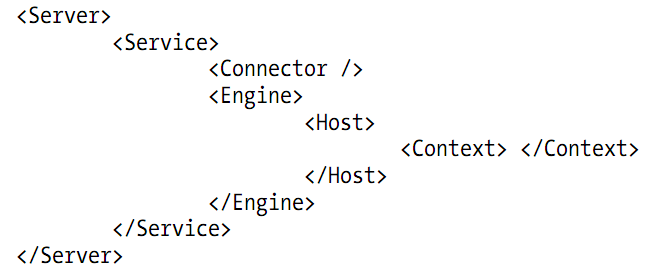


Рисунок 4.2.1 XML представление архитектуры компонентов Tomcat.

Эта сущность может быть разбита на серию ***контейнеров***, включающих *сервер, сервис, соединитель, ядро, хост, и контекст*.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

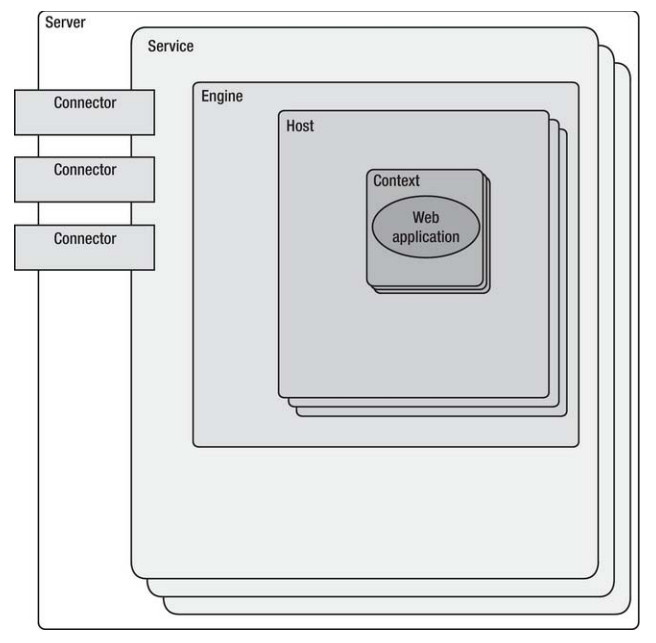


Рисунок 4.2.2 Архитектура Tomcatс основными компонентами.

По умолчанию, каждый из этих контейнеров конфигурируется, используя ***server.xml***.

***Сервер***

Первый контейнерный элемент, представленный на рисунке 4.2.1 как **<Server>.**Он представляет собой ***цельный Catalina сервер***и используется как высокоуровневый элемент в отдельной сущности Tomcat. Элемент <Server> может содержать одну или более <Service> контейнеров.

|  |
| --- |
|  |
|  |

***Сервис***

Следующий контейнерный элемент***<Service>***, который владеет коллекцией одного или более <Connector> элементов, которые делят одиночный<Engine> элемент. N-число<Service> элементов могут быть вложенными в отдельный <Server> элемент.

|  |
| --- |
|  |
|  |

***Соединитель***

Следующий тип элемента – это *<****Connector>*** элемент, который определяет класс, фактически обрабатывающий запросы и ответы от вызывающего клиентского приложения.

|  |
| --- |
|  |
|  |

***Ядро***

Третий контейнерный элемент – это ***<Engine>*** элемент. Каждый установленный <Service> может содержать тока один <Engine> элемент, и этот один <Engine> компонент обрабатывает все запросы, полученные всеми установленными <Connector> компонентами, определенные родительским сервисом.

|  |
| --- |
|  |
|  |

***Хост***

***<Host>***элемент определяет виртуальные хосты, которые содержатся в каждой сущности ***Catalina<Engine>***. Каждый<Host> может быть родительским контейнером для одного или более веб-приложений, с каждой из которых представлен <Context>компонент.

|  |
| --- |
|  |
|  |

***Контекст***

***<Context>*** элемент- это наиболее часто используемый контейнер в сущности Tomcat. Каждый <Context> элемент представляет индивидуальное веб-приложение, которое запускается с определенным <Host>. Нету ограничений, какое число контекстных элементов может быть определено в рамках <Host> элемента.

|  |
| --- |
|  |
|  |
|  |
|  |

Модуль4.3 Установка и конфигурирование Tomcat

Для установки и настройки Tomcat, прежде скачайте **установочный пакет** с http://tomcat.apache.org .

Сначала вы должны установить ***JavaDevelopmentKit (JDK)***. Так как мы будем использовать ***Tomcat 7***, то вы должны установить ***JDK 1.6*** или более поздние версии.

Послеустановки JDK, мы будем устанавливать ***apache-tomcat-7.0.29.exe***. TomcatWindows установщик очень похож на другие Windows установщики.

**Следуйте шагам, выбирайте, куда вы хотите установить сервер, и установщик распакует и скопирует файлы в корректную директорию, и сконфигурирует переменные среды и сервисные свойства.**



Рисунок 4.3.1 Установка ApacheTomcat.Шаг 1.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

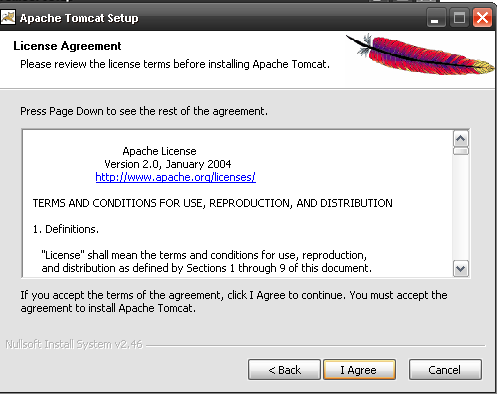


Рисунок 4.3.2 Установка ApacheTomcat.Шаг 2.

|  |
| --- |
|  |
|  |

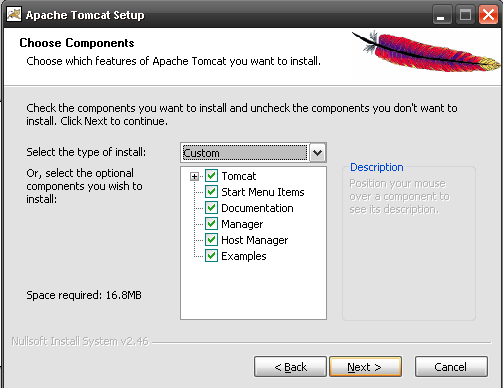


Рисунок 4.3.3 Установка ApacheTomcat.Шаг 3.

|  |
| --- |
|  |
|  |

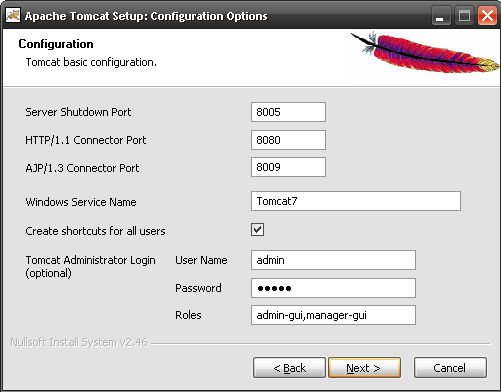


Рисунок 4.3.4 Установка ApacheTomcat.Шаг 4.

|  |
| --- |
|  |
|  |

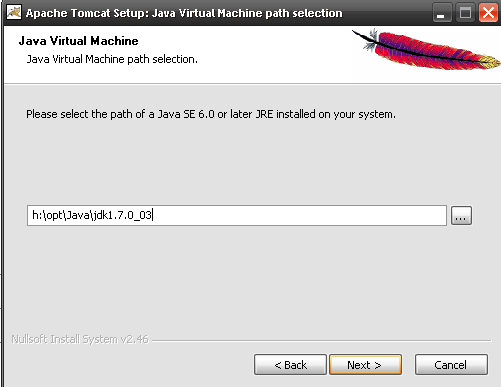


Рисунок 4.3.5 Установка ApacheTomcat.Шаг 5.

|  |
| --- |
|  |
|  |

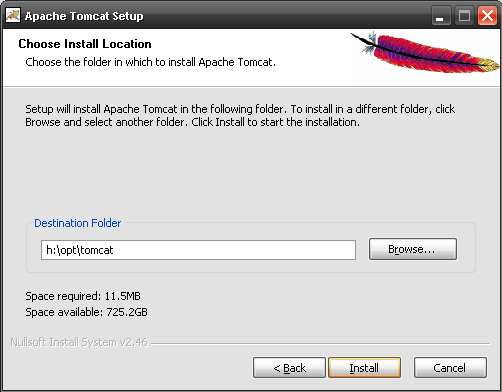


Рисунок 4.3.6 Установка ApacheTomcat.Шаг 6.

|  |
| --- |
|  |
|  |

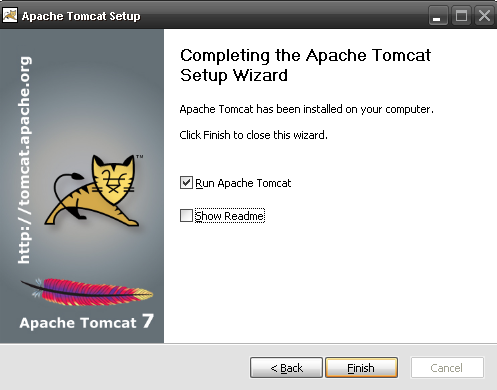


Рисунок 4.3.7 Завершение установки ApacheTomcat.Шаг 7.

После завершения установки, вы можете проверить, что переменная среды ***CATALINA\_HOME*** была установлена. Если она не была установлена, то установите ***вручную***.

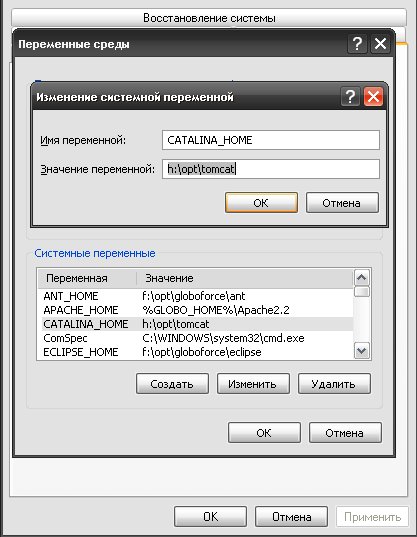


Рисунок 4.3.8 Установка переменной среды CATALINA\_HOME

Windows установщик установит Tomcat7 как Windows Сервис автоматически, так что Tomcat может быть запущен автоматически при старте windows, и работать в фоновом режиме как Windows Сервис.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

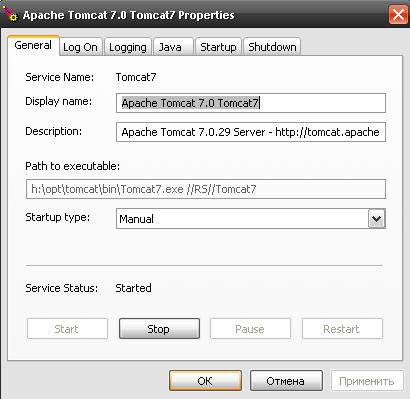


Рисунок 4.3.9Утилита для запуска и остановки ApacheTomcat, как WindowsСервиса.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 4.4 Тестирование установки сервера Tomcat

Для тестирования установки сервера Tomcat, вам необходимо для начала **запустить** его.

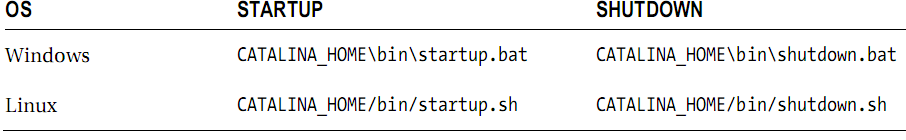


Рисунок 4.4.1 Команды запуска и остановки сервера Tomcat.

Вы также можете запускать и останавливать Tomcat через утилиту для запуска и остановки ApacheTomcat (рисунок 4.3.9), в случае использования ***установщика Windows***.

Когда Tomcat запустился, откройте в вашем браузере следующийURL:

<http://localhost:8080/>

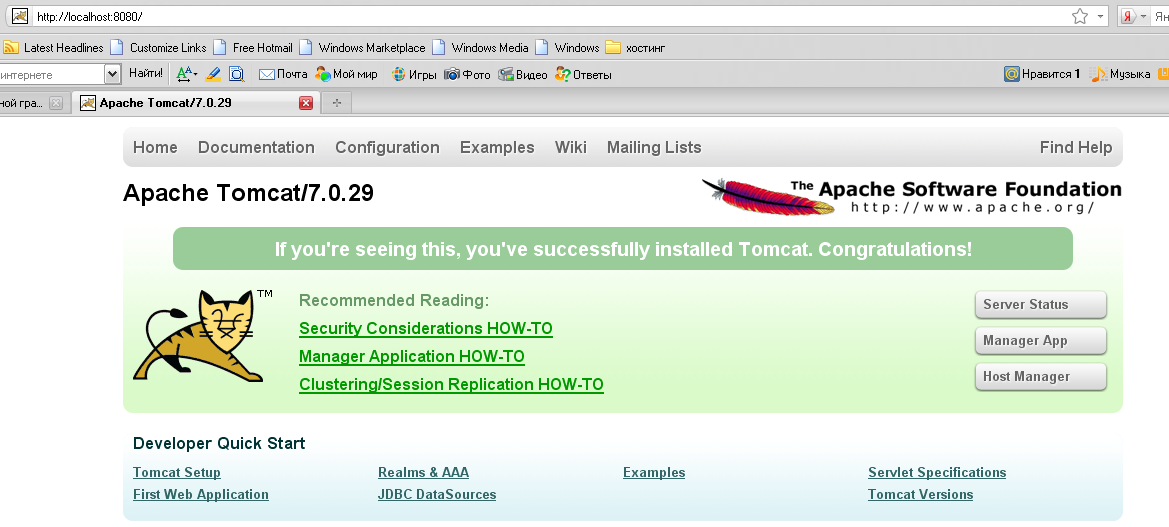


Рисунок 4.4.2Домашняя страница Tomcat сервера.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |

Tomcat по умолчанию обрабатывает все запросы на порту ***8080***. Это было сделано с целью, чтобы полностью ***разделить java сервлет контейнер от стандартного HTTP сервера***, который обрабатывает запросы на порту 80 (например, ApacheHTTP сервер). Если вы хотите обрабатывать все запросы на ***стандартном HTTP порту 80, вместо 8080***, вам необходимо сделать следующие изменения в CATALINA\_HOME/conf/server.xml файле и перезагрузить Tomcat.

Найдите следующие строки в ***server.xml*** файле:

<Connectorport="**8080**" protocol="HTTP/1.1"

connectionTimeout="20000"

redirectPort="8443" />

и замените значение порта на ***80***, как показано в следующем фрагменте кода:

<Connector port="**80**" protocol="HTTP/1.1"

connectionTimeout="20000"

redirectPort="8443" />

Сейчасвыдолжны***перезагрузитьTomcat***. Для перезагрузки сервера Tomcat, сначала остановите сервер при помощи ***shutdown***команды. Когда команда завершит свою работу, запустите сервер снова, используя ***startup***команду.

После перезагрузки, вы можете открыть ваш браузер со следующей ссылкой, и увидеть результат, похожий нарисунок 4.4.2.

<http://localhost>

У этого URL***нет номера порта***. В URL соглашении для браузеров указано, что если номер порта отсутствует в URL, то предполагается значение номера порта ***80***. Поэтому, <http://localhost:80/> будет загружать туже страницу.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

***Следующий шаг***, это проверить, что JSP страницы корректно обрабатываются сервером Tomcat. Вы можете сделать это, запустив один из JSP примеров, предоставляемый сервером Tomcat.

Для запуска примера JSP, откройте в браузере страницу <http://localhost:8080/examples/jsp/> и выбрать ***JSP пример***.

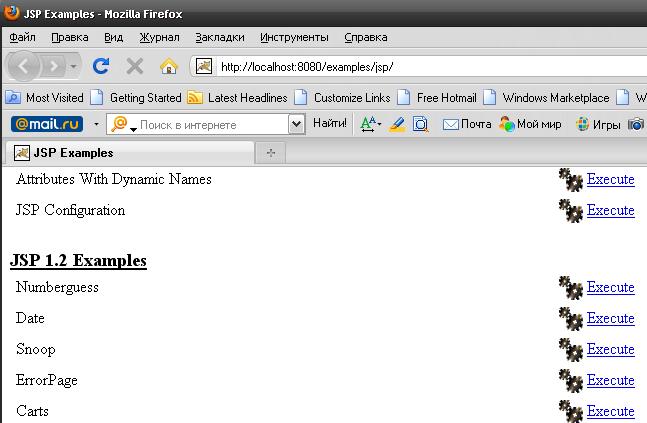


Рисунок 4.4.3 JSPпримеры сервера Tomcat.

Сейчас выберите пример ***Date*** и нажмите на ***Execute*** ссылку. Если все было установлено правильно, то вы увидите страницу с датой.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

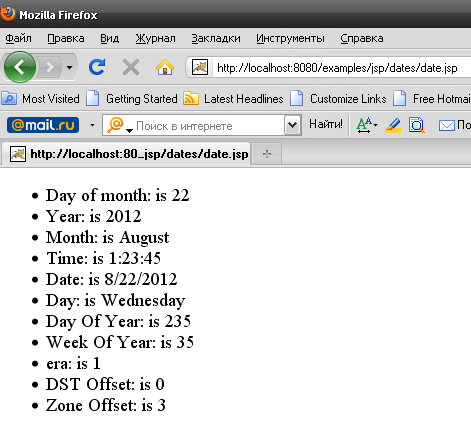


Рисунок 4.4.4 Страница с датой.

Если вы не увидели страницу с датой (рисунок 4.4.4), проверьте, что значение вашей переменной среды ***JAVA\_HOME*** совпадает с местом, где установлена ваша ***JDK***.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Тема 5**.Запуск сервера Tomcat

Сейчас, когда сервер Tomcat установлен и запущен, давайте посмотрим на ***шаги***, необходимые для ***установки веб-приложения на этот сервер***.

Для того чтобы это сделать, мы должны сначала рассмотреть ***структуру директорий сервера Tomcat***, а затем перейти к установки веб-приложения.

Модуль 5.1 Структура директорий сервера Tomcat

До того, как вы начнете создавать ваше собственное веб-приложение, вы должны ознакомиться со ***структурой директорий***Tomcat.

Вся ***структура директорий*** сервера Tomcat лежит в установочной директории ***CATALINA\_HOME***.

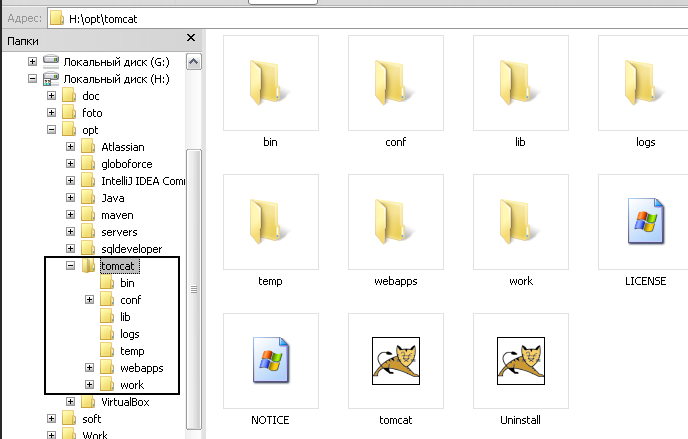


Рисунок 5.1.1 Структура директорий сервера Tomcat.

|  |
| --- |
|  |
|  |
|  |
|  |

***/bin***–Содержит скрипты запуска и остановки сервера как для ОС Windows, так и для Linux. Jar файлы с классами, необходимыми для запуска сервера, тоже находятся здесь.

|  |
| --- |
|  |
|  |

***/conf*** – Содержит основные конфигурационные файлы сервера Tomcat.Два наиболее важных конфигурационных файлов – это ***server.xml*** и ***web.xml*** (глобальный).

|  |
| --- |
|  |
|  |

***/lib*** – СодержитTomcatjava архив (jar) файлы, разделенные среди всех компонентов Tomcat. Все веб-приложения, установленные на сервер Tomcat, могут получить доступ к библиотекам, расположенным в этой директории. Она включает Servlet API иJSP API библиотеки.

|  |
| --- |
|  |
|  |

***/logs*** – Содержитlog файлы сервера Tomcat.

***/temp*** – Содержит временные файлы сервера Tomcat.

|  |
| --- |
|  |
|  |

***/webapps*** – Директория, куда все веб-приложения устанавливаются, и куда вы поместите ваш WAR файл, когда он будет готов к установке.

***/work*** -Рабочая директория сервера Tomcat, куда Tomcat копирует все сервлеты, которые были сгенерированы из JSP страниц. Если вы хотите увидеть, как существующие JSP страницы превращаются в сервлеты, загляните в эту папку.

|  |
| --- |
|  |
|  |
|  |

Задание 9

Скопируйте приложение, которое вы создали при помощи maven в директорию webapps. Запустите Tomcat и проверьте статус установки приложения в Tomcat с помощью log файлов.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 5.2 Выполнение скриптов сервера Tomcat

Все исполняемые файлы Tomcat находятся в***CATALINA\_HOME/bin*** директории.Исполняемые файлы как для Windows, так и для Linux располагаются в этой директории, поэтому вы увидите парные скрипты с одним и тем же именем и разными расширениями (\*.bat для Windows и \*.sh для Linux).

Наиболее ***важные исполняемые файлы*** в этой директории – это ***Catalina скрипты*** (catalina.sh на Linux или catalina.bat для Windows). Этот скрипт отвечает за запуск и остановку сервера Tomcat, и он принимает разные параметры командной строки для разных операций. Например, для старта сервера Tomcat, вы можете просто выполнить команду catalinastart в командной строке, иcatalinastop -для остановки сервера Tomcat.

Скрипты catalina*доступны* только, если вы скачали *zip дистрибутив* сервера Tomcat. Если вы установили Tomcat с использованием установщика сервиса Windows, то по умолчанию, вы не можете использовать скрипты Catalina. В этом случае вы можете использовать tomcat7w.exe приложение для управления сервисом Tomcat или скопировать скрипты из zip дистрибутива.

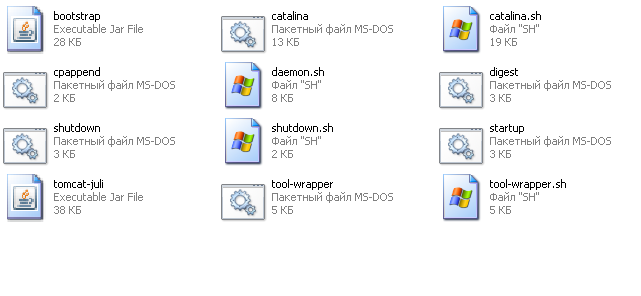
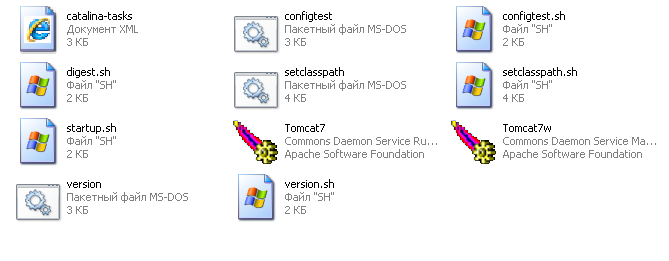


Рисунок 5.2.1 Исполняемые скрипты catalinaи сервисы запуска Tomcat.

|  |
| --- |
|  |

Итак, перечислим основные команды, которые вы можете использовать, когда запускаете скрипты Catalina.

***catalinastart –*** запускает Tomcat как новый процесс. В ОС Windows, Tomcat будет запущен в новом окне состояния командной строки.

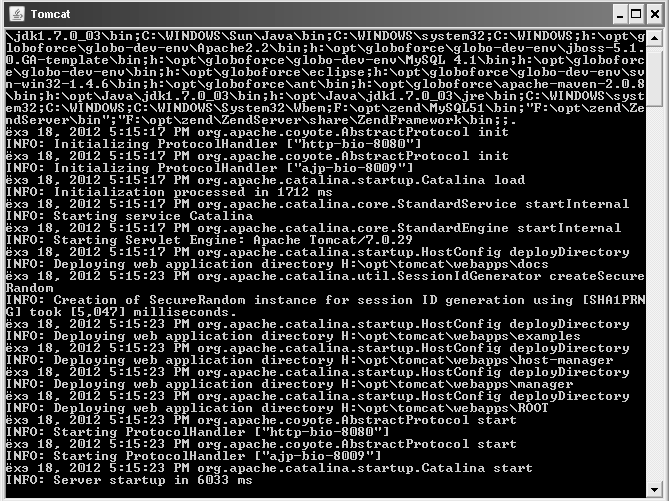


Рисунок 5.2.2 ЗапускTomcat

***catalinastop -*** останавливает Tomcat.

***catalinarun –*** запускаетTomcatвтекущемокне. В ОС Windows, выводистории запуска Tomcat будет добавлен к текущему окну состояния командной строки.

***catalinadebug –*** запуск Tomcat в режиме отладки.

***catalina version –*** выводверсииTomcat.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

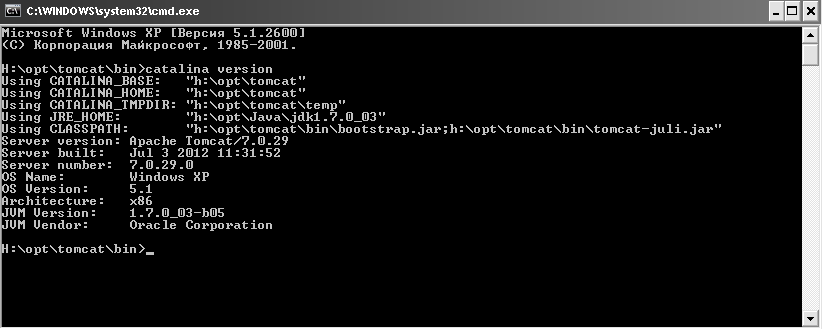


Рисунок 5.2.3 Версия Tomcat

***catalinaconfigtest –*** проверяет, что текущая конфигурация Tomcat правильна. Скрипт пытается запустить Tomcat и загрузить все конфигурационные файлы, а затем аккуратно выйти, если сервер настроен, верно. Если какие-либо файлы конфигурации настроены не верно, то скрипт выдаст отчет об ошибки.

***catalinajpdastart -***Запускает Tomcat в JavaPlatformDebuggingArchitecture (JPDA) отладочном режиме.

В дополнение, вы можете расширить опцией***–security*** любую команду скриптов catalina. С Помощью этой опции Tomcat запускается в режиме включенного менеджера безопасности (securitymanagerenabled). ***Режим включенного менеджера безопасности –*** это механизм защиты ресурсов, управляемых JVM от несанкционированного доступа. Например, если ваш код на языке Java содержит строчку System.exit(1) и этот код запускается в Tomcat, Tomcat завершит работу. Менеджер безопасности может, используя настраиваемые правила безопасности, запретить вызов специфических классов или методов. В том случае, если Tomcat запускает только известный код, запуск Tomcat в режиме включенного менеджера безопасности не требуется. Кроме того, если вы запускаете Tomcatc установленными сторонними веб-приложениями, рекомендуется, запускать его в режиме включенного менеджера безопасности.

Директория bin также содержит два подходящих скрипта для запуска и остановки работы Tomcat: ***startup.bat и shutdown.bat***. Эти скрипты являются всего лишь оболочками для вызова скриптов catalina, и они широко используются вместо вызова скриптов catalina, для запуска и остановки Tomcat, без аргументов командной строки.

|  |
| --- |
|  |
|  |
|  |

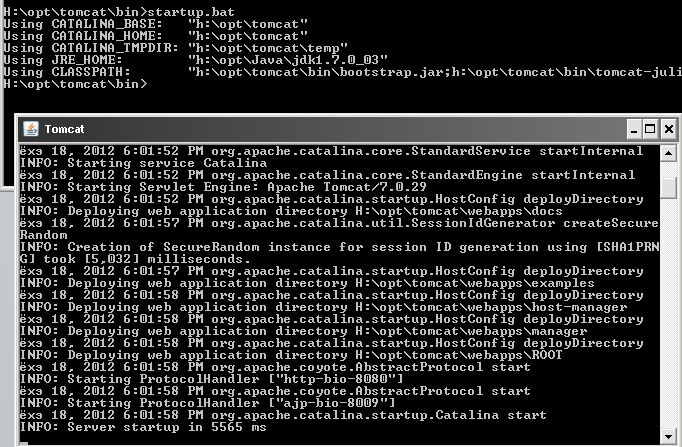


Рисунок 5.2.4 ЗапускTomcatсstartup.bat

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 5.3 Передача конфигурационных опций времени выполнения скриптам Catalina.

Когда запускаются Java приложения, то часто требуется передать различные опции *виртуальной машине java*(JavaVirtualMachine - JVM), такие как настройки памяти, настройки кодировки файлов, порты, адреса серверов и т.д. Эти опции передаются в скрипты catalina через стандартные переменные среды Java и Tomcat.

JAVA\_OPTS переменные среды содержат опции доступные всем Java процессам, запущенным на компьютере, включая Tomcat.

Для того, что бы быть уверенным, чтоJVMчитаетвсефайлыиспользуякодировкуUTF-8, мыможемдобавитьстандартныйпараметр***file.encoding***кпеременнойJAVA\_OPTS: **setJAVA\_OPTS="-Dfile.encoding=utf-8"**

Если вы запустите Tomcat с установленными JAVA\_OPTS, то это гарантирует, что JVM будет использовать UTF-8 кодировку для чтения файлов не только для Tomcat процесса, но и для всех других Java процессов, запущенных на компьютере.

Для передачи специфических параметров только для Tomcat процесса, необходимо использовать CATALINA\_OPTS.

Типичный сценарий – это добавление больше памяти Tomcatпроцессу.

set CATALINA\_OPTS=" –Xms256m -Xmx1g -XX:MaxPermSize=256m"

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Тема 6**. Конфигурирование сервера Tomcat

Директория conf содержит все конфигурационные файлы Tomcat. Основной конфигурационный файл – это ***server.xml***. Основные Tomcat элементы ядро (engines), хосты (hosts) и контексты (contexts) расположены здесь. Context.xml конфигурация содержит настройки по умолчанию, которые могут быть использованы всеми контекстами Tomcat. К дополнению к глобальным настройкам контекста, которые расположены в server.xml и context.xml, каждый контекст может иметь свой собственный конфигурационныйфайл. Расположение конфигурационного файла зависит от имени ядра и хоста, в которых контекст определен. Соглашение, что контекст с именем CONTEXT\_NAME, настроенный с хостом HOST\_NAME, имя ядра которого ENGINE\_NAMEсохраняют настройки по следующему пути:

*CATALINA\_HOME/conf/ENGINE\_NAME/HOST\_NAME/CONTEXT\_NAME.xml*

Например, если вы устанавливаете веб приложение в контекст it-academy, используя хост по умолчанию с именем localhost и стандартное ядро TomcatCatalina, конфигурационный файл будет располагаться по следующему пути:

***CATALINA\_HOME/conf/Catalina/localhost/.xml***

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 6.1 Установка WAR-архива

***WAR (Веб архив) –*** это архив, содержащий все ресурсы веб-приложение, и может быть установлен в Tomcat.

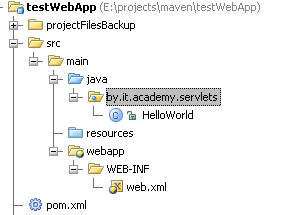


Рисунок 6.1.1 Структура web-проекта

После вызова команды mvncleaninstall происходит сборка WAR файла, который находится в target\testWebApp.war.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

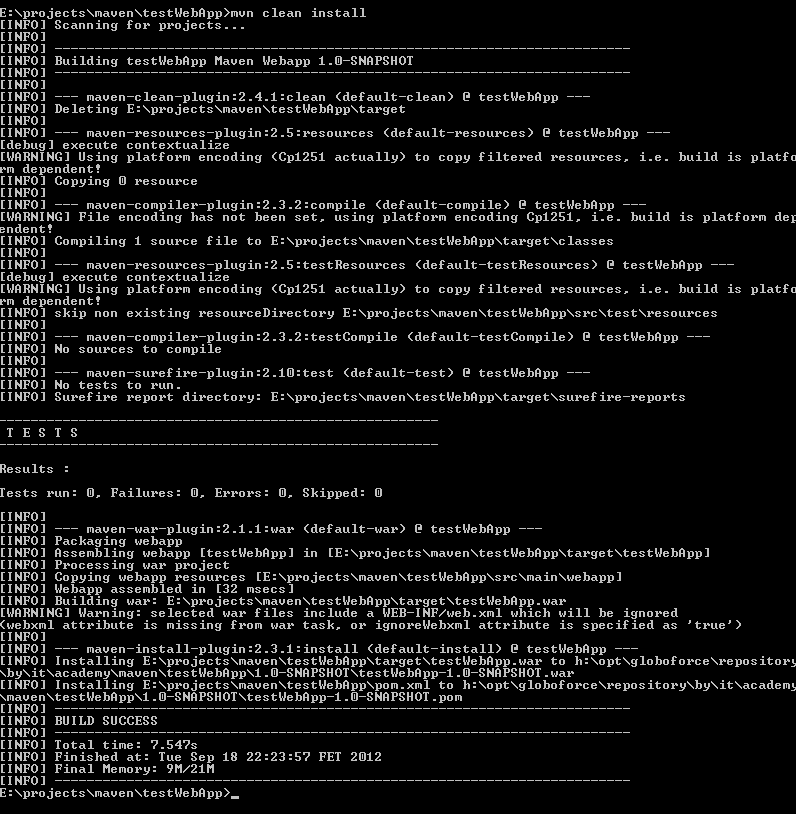


Рисунок 6.1.2 Сборка WAR

Если разархивировать testWebApp.war, то мы получим следующую структуру файлов.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |

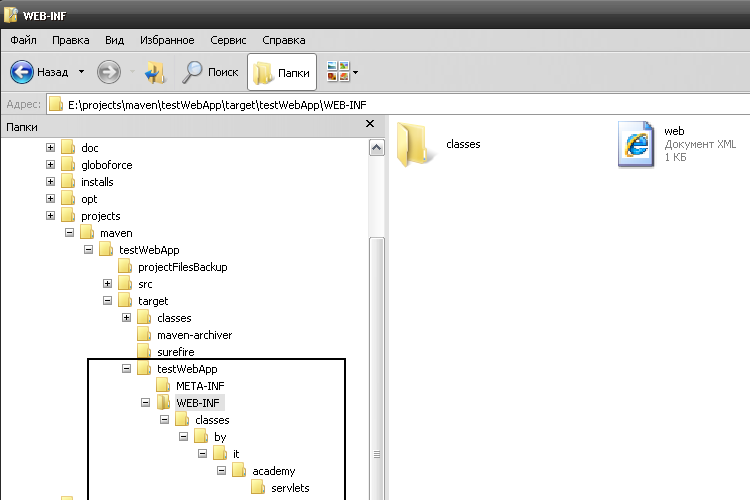


Рисунок 6.1.3Структура WAR

Для установки WAR веб-приложения, все что вы должны сделать, это скопировать WAR в ***CATALINA\_HOME/webapps***

При установке Tomcat возьмет testWebApp.war и распакует и установит приложение автоматически.

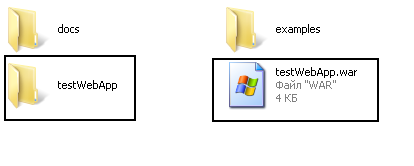


Рисунок 6.1.4Распаковка WAR после установки

После запуска Tomcat, вы можете открыть окно браузера и ввести <http://localhost:8080/testWebApp/hello>.

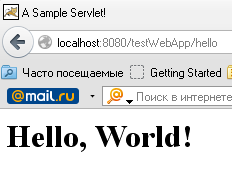


Рисунок 6.1.5Установленное веб-приложение.

Tomcat распаковывает WAR по причинам производительности, т.к. это значительно быстрее загружать файлы из файловой системы, чем распаковывать файлы из архива каждый раз, когда они необходимы.

Существуют способ настройки Tomcat, при котором он не может распаковывать архив WAR при установке. Для этого необходимо установить значение атрибута unpackWAR<Context> элемента вашего веб-приложения. Следующий пример конфигурации иллюстрирует эту возможность:

<Context path="/testWebApp" docBase="testWebApp" unpackWar="true">

Задание 10

Установите web приложение на Tomcat сервер. Настройте контекст приложения таким образом, что бы war файл автоматически распаковывался при запуске сервером приложений.Проверьте работоспособность вашего приложения.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 6.2 Конфигурация хостов

В архитектуре Tomcat, каждое установленное приложение представляет собой контекст в рамках хоста. Даже без специализированной конфигурации, у Tomcat конфигурации существуют значения по умолчанию, благодаря которым мы могли устанавливать приложения без беспокойства об хостах и контекстах.

***Хосты*** настраиваются в CATALINA\_HOME/conf/server.xml. Если вы посмотрите на этот файл, то увидите, что один хост уже настроен по умолчанию. Имяэтогохоста – ***localhost***

<Host#1

name="localhost"#2

appBase="webapps"#3

unpackWARs="true"#4

autoDeploy="true"#5

>

Хост настраивается, используя<Host>XML элемент (#1). Каждый хост должен иметь имя, настраиваемый с помощью атрибута *name* (#2). Имя хоста обычно указывает на сетевое имя компьютера (сервера), на котором запущен Tomcat, например [www.it-academy.by](http://www.it-academy.by). Хост по умолчанию имеет имя *localhost,* который указывает на локальный компьютер.

Атрибут *appBase* настраивает *базовую директорию для приложений,*из которой Tomcat загружает и устанавливает веб-приложения (#3). Путь может быть абсолютным (напримерc:\deployedapps), или относительным (без задания имени диска), когда действительная директория располагается по отношению кCATALINA\_HOME директории. Конфигурация по умолчанию указывает на webapps, что значитCATALINA\_HOME/webapps директория является базовой директорией для веб-приложений.

Следующий атрибут unpackWARs (#4)настраивает возможность автоматического разархивирования WAR при установке (само собой при запуске Tomcat).

Последний аттрибут autoDeploy (#5), по умолчанию его значение true. Этот атрибут настраивает Tomcat на периодическую проверку обновлений WAR файлов и распакованных директорий (но не их контекстов) в базовой директории для приложений, и устанавливать или переустанавливать их автоматически, если какие-либо изменения необходимы.

|  |
| --- |
|  |
|  |
|  |
|  |

Модуль 6.3 Конфигурация контекстов

Каждое установленное веб-приложение в Tomcat представляет контекст во внутренней архитектуре Tomcat. Контексты относятся к сконфигурированным хостам, и они настраиваются используя <Context>XML элемент (#1).

<Context#1

path="/testWebApp"#2

docBase="testWebApp"#3

reloadable="true"#4

/>

*Path* атрибут указывает контекстный путь (путь для запроса к серверу) для веб-приложения (#2).Каждый HTTP запрос к Tomcat, имеющий унифицированный идентификатор ресурса (URI), начинающийся с настроенного контекстного пути (после сервера и порта), будет обслуживаться этим контекстом. Этот параметр должен быть установлен если контекст настроен в server.xml; во всех остальных случаях контекстный путь берется из имени директорий и файлов, где установлено веб-приложение. Если же контекст установлен, то контекстный путь должен быть уникальным в рамках хоста.

URI – это набор символов, определяющих ресурс в Интернете. Например, [www.mycompany.com:80/testWebApp](http://www.mycompany.com:80/testWebApp)или

[http:/localhost:8080/testWebApp/index.html?page=2](http://localhost:8080/testWebApp/index.html?page=2)

Часть URIпосле имени сервера и порта используется для определения контекстного пути в Tomcat.

Атрибут *docBase* определяет корневой контекст, который является путем к контексту веб-приложения (#3). Он может содержать абсолютный путь к WAR файлу или распакованной директории, или относительный путь к корневому каталогу хоста, в рамках которого он сконфигурирован.

Последний атрибут настроенный в этом примере -это *reloadable* (#4). Если он установлен в true, Tomcat будет проверять WEB-INF/classes и WEB-INF/lib директориикорневого контекста, и переустанавливать веб-приложение, если какие-либо изменения обнаружены. Эта возможность очень полезна для разработки и тестирования, т.к. вам нет необходимости перезапускать Tomcat каждый раз при внесении изменений в приложение. Однако, с другой стороны, данная возможность ухудшает производительность, таким образом, использование данной настройки не рекомендовано на production среде. Настройка по умолчанию для данного атрибута – false.

Контекст веб-приложения может быть настроен следующим образом:

1. В файле server.xml
2. Вкаталогеконфигурациихоста:

CATALNA\_HOME/conf/ENGINE\_NAME/HOST\_NAME/CONTEXT\_PATH.xml

1. Как часть кода веб-приложения, в файле /META-INF/context.xml в WAR или распакованной директории
2. Неявно, где Tomcat создает контекст для веб-приложения, для которого ни одна из вышеупомянутых возможностей не подходит.

Когда настраиваете контекст веб-приложенияв файле server.xml,вы должны указать все атрибуты, указанные выше. Кроме того, ввиду того, что содержание файлаserver.xmlне перегружается, пока Tomcat не будет перезапущен, то любые изменения, которые вы сделали в этом файле, не будут видны до тех пор, пока вы не перезапустите Tomcat.

Вторая опция – это настройка контекста в отдельном xml файле и сохранить его в специальной директории настройки хоста в рамках CATALINA\_HOME/conf. Имя хml файла будет представлять контекстный путь, так что атрибут path устанавливать не нужно. Атрибут docBase должен быть установлен только в том случае если корневая директория контекста находится за приделами корневой директории хоста.

Для примера, конфигурационный файл должен располагаться в CATALINA\_HOME/conf/Catalina/localhost/testWebApp.xml и контекст testWebApp будет выглядеть следующим образом:

<Contextreloadable="false"/>

Мы не указывали path атрибут, так как он берется из имени xml файла. Мы также не указывали docBase атрибут, потому что он также берется из имени xml файла по отношению к корневой папке webaaps.

Третья возможность для конфигурации контекста – это настроить его в/META-INF/context.xml файле в WAR файле или распакованной директории. Этот вариант подходит лишь тогда, когда первые два варианта не настроены. Файлcontext.xml выглядит точно так же как указанный выше.

Четвертый вариант – это вариант по умолчанию. Он относится к веб-приложениям, автоматически установленным в корневом каталоге для веб-приложений ((CATALINA\_HOME/webapps директория в настройках Tomcat по умолчанию). Все распакованные каталоги, установленные в корневом каталоге для веб-приложений, обладают именем контекста, совпадающим с именем каталога, корневой контекстный путь устанавливается путем к каталогу.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Тема 7**. Менеджер веб-приложений Tomcat

Модуль 7.1 Что такое Менеджер веб-приложений Tomcat

***Менеджер веб-приложений*** – это программа, предоставляющая базовую функциональность для управления веб-приложениями, запущенными на сервере Tomcat.Некоторая из представленной функциональности включает возможностьпо установке, запуску, остановке и удалению веб-приложения. Используя менеджер веб-приложений, вы можете легко установить веб-приложение на локальный или удаленный сервер, без необходимости FTP доступа, все команды вызываются с помощью HTTP протокола – используя ваш любимый браузер или командную строку.

Для дополнительного доступа, когда безопасность критична, менеджер веб-приложений может быть доступен через ***JavaManagementExtensionproxy (JMX)***.

***JMX –*** это стандартная Java технология, созданная для управления приложений, устройств, или сетей, используястандартныйJavaAPI.

Существуют следующие контекстные пути для доступа к менеджеру веб- приложений в Tomcat:

1. Веб-приложение на основе html интерфейсе - ***/manager/html***

|  |
| --- |
|  |
|  |

1. Тестовое приложение на основе сценария - ***/manager/text***

|  |
| --- |
|  |
|  |

1. JMX proxy доступ - ***/manager/jmxproxy***

|  |
| --- |
|  |
|  |

1. Запрос статуса - ***/manager/status***

|  |
| --- |
|  |

Модуль 7.2 Получение доступа к Менеджеру веб-приложений

Прежде чем вы сможете использовать менеджер веб-приложений, вы должны создать нового пользователя с соответствующими привилегиями для доступа к веб-приложению. Если вы попробуете получить доступ к любой части веб-приложения без установки привилегий безопасности, вы будете переадресованы на страницу для ошибок Tomcatсо стандартным HTML статус-кодом 403-запрещено.

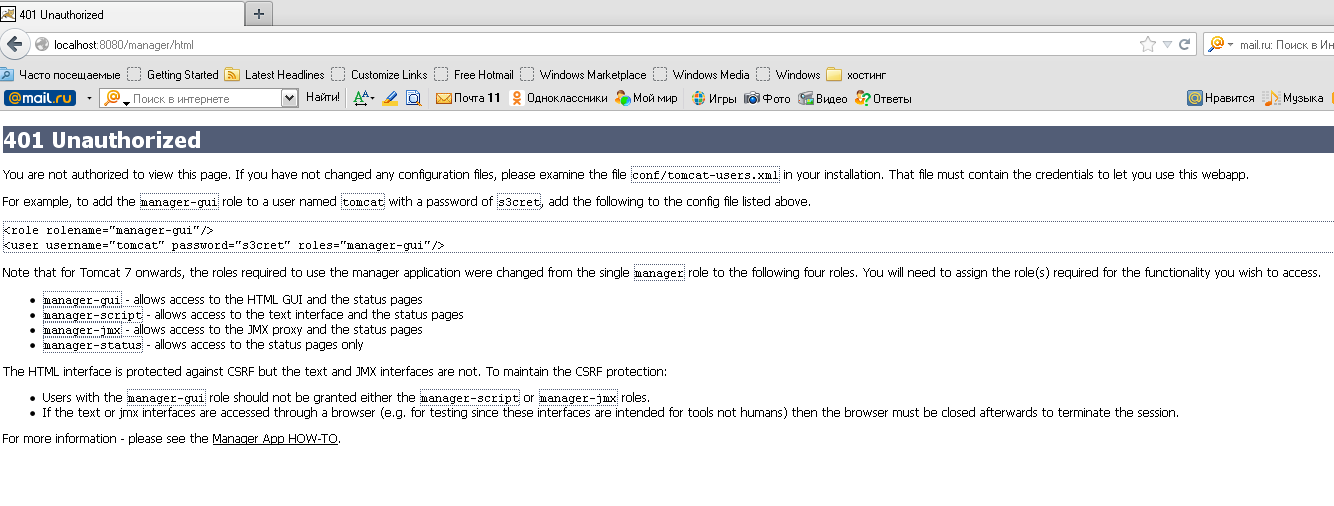


Рисунок 7.2.1 Страница для ошибок, показывающаяся, если пользователь попробует получить доступ к менеджеру веб-приложений без необходимых привилегий.

До Tomcat версии 7.0, была одна роль, необходимая для доступа к менеджеру веб-приложений, которая просто называлась «manager». С 7 версии Tomcat представляет 4 различные роли для доступа менеджера веб-приложений – одна для каждого компонента, упомянутого выше.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

|  |  |  |
| --- | --- | --- |
| ***Компонент*** | ***Требуемая пользовательская роль*** | ***Описание*** |
| Веб-приложение на основе html интерфейсе | manager-gui | Разрешаетдоступ только к HTML интерфейсу и статус странице |
| Тестовое приложение на основе сценария | manager-script | Разрешает доступ только к текстовому скриптовому движку и статус странице |
| JMX proxy доступ | manager-jmx | Разрешает доступтолько к JMX консоле и статус странице |
| Запрос статуса | manager-status | Разрешает доступ только кстатус странице |

|  |
| --- |
|  |
|  |

Сейчас нам необходимо создать несколько пользователей с необходимыми ролями, так что мы сможем продемонстрировать, что может быть сделано с помощью Tomcat менеджера веб-приложений.

Имена пользователей, пароли и роли сохранены в файле***CATALINA\_HOME/conf/tomcat-users.xml***. Если вы откроете этотфайл, вы увидите, что он *пуст*. Так мы должны добавить несколько строк вtomcat-users.xml, такмы можем создать пользователей, которые имеют доступ к менеджеру веб-приложений.

<?xml version='1.0' encoding='utf-8'?>

<user username="managerGui" password="abc123" roles="manager-gui"/>#1

<user username="managerScript" password="abc123" roles="manager-script"/>#2

</tomcat-users>

В строчке, отмеченной как #1, мы определили пользователя с именем***managerGui*** и ролью***manager-gui***. В следующей строчке (#2), мы создали пользователя***managerScript*** с ролью***manager-script***. Пользователь ***managerGui***имеет доступ к менеджеру веб-приложений через HTML интерфейс, и пользователь имеет доступ к командам, используятестовое приложение на основе сценария. Как было упомянуто выше, оба имеют доступ кстатус странице.

Модуль 7.3 Доступ к менеджеру веб-приложений, используя веб-интерфейс.

Наиболее распространённый способ доступа менеджера веб-приложений - это использования веб интерфейса. Так как у нас есть пользователь ***managerGui***, ссоответствующими привилегиями, вы можете открыть домашнюю страницу менеджера веб-приложений [***с***](http://localhost:8080/manager/html)***.***

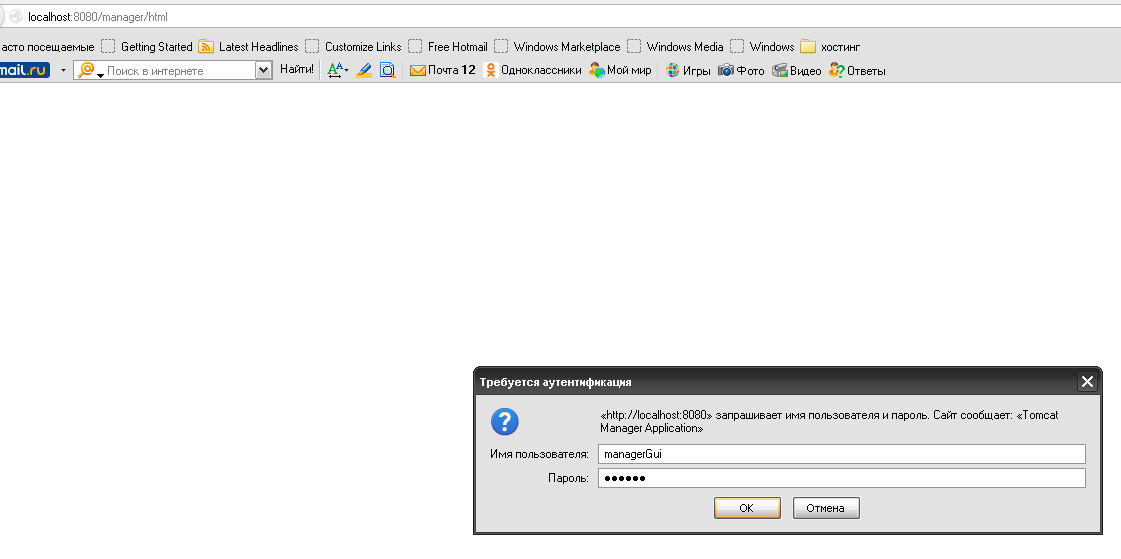


Рисунок 7.3.1 Страница ввода имени пользователя менеджера веб-приложений

Сейчас, давайте введем имя пользователя и пароль для пользователя ***managerGui***с ролью ***manager-gui***. После ввода корректных имени пользователя и пароля мы увидим домашнюю страницу менеджера веб-приложений.

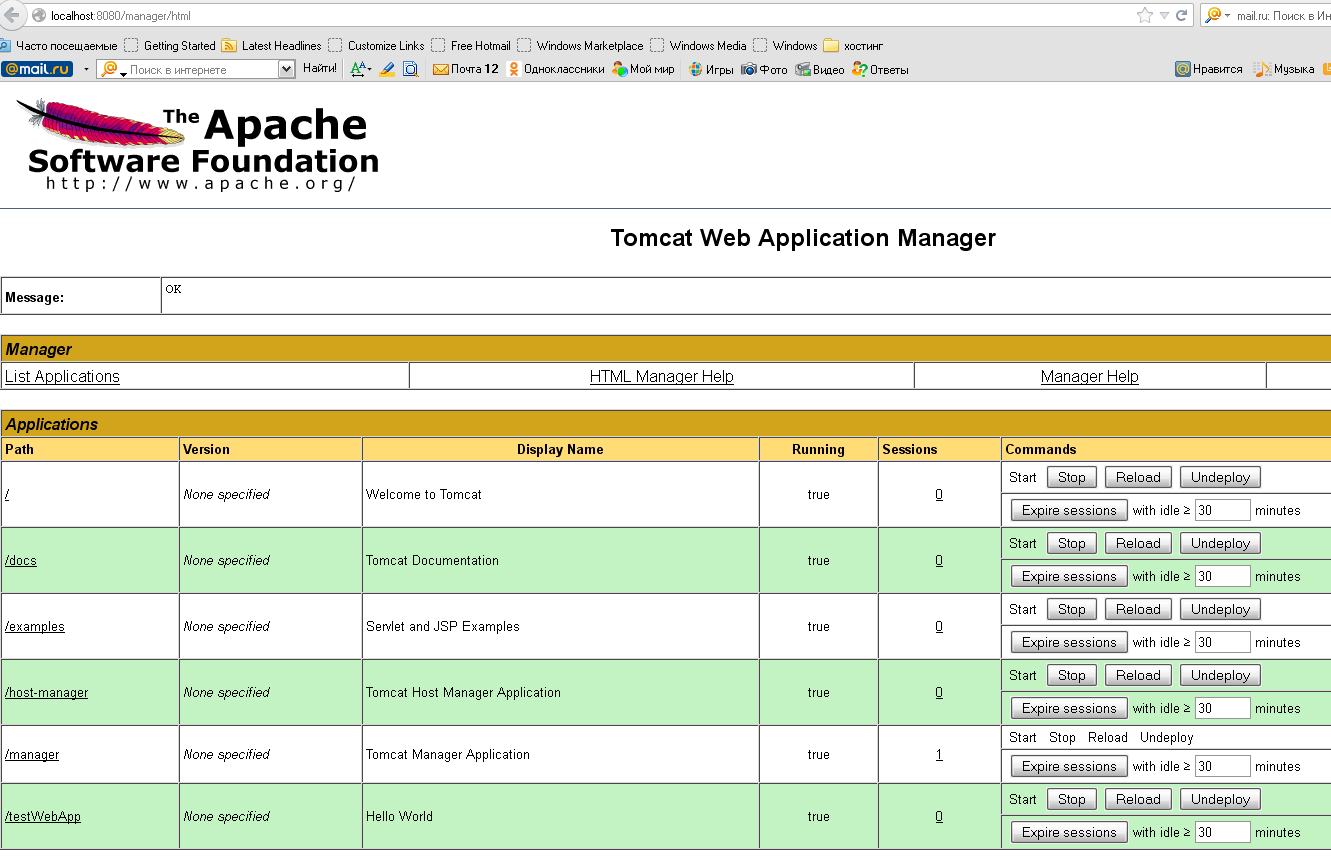


Рисунок 7.3.2 Домашняя страница менеджера веб-приложений

Чуть ниже логотипа Tomcat, находитсяполе с заголовком «Message» и текстом «ОК». В этом поле, вы можете увидеть ответ Tomcatпосле каждого действия. Каждое действие или команда, обработанная менеджером веб-приложения Tomcat,предоставитответ. Если все работает как надо, то ответ – «ОК». В случае ошибки, поле отображает сообщение об ошибки, для быстрого определения проблемы. Когда используете, менеджер веб-приложений, проверьте поле.

Под полем для сообщения, вы увидите таблицу с четырьмя ссылками: списки ссылок приложений, две ссылки на странички помощи, ссылка на страницу статуса.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 7.3 Прослушивание установленных веб-приложений

Список ссылок приложений производят список команд на Tomcat сущности. Результаты команд в списке всех приложений, установленных на сервере. Список команд вызывается по умолчанию, когда используем веб-интерфейс менеджера Tomcat. Когда мы загружаем домашнюю страничку менеджера веб-приложений, мы в действительности вызываем список команд.

Все приложения, установленные на Tomcat, показываются на домашней странице менеджера веб-приложений.На домашней странице мы видим:

1. Список веб-приложений, установленных на сервере со списком команд.
2. Проверка серверного статуса, используя команду статуса.
3. Установка нового веб-приложения в Tomcat, используя командуустановку.
4. Перезагрузка выбранного веб-приложения, используя команду перезагрузки.
5. Проверка серверных сессий для каждого веб-приложения, используя команду сессии.
6. Запустить веб-приложение, используя команду запуска.
7. Остановить запущенное веб-приложение, используя команду остановку.
8. Удаление веб-приложение из сервера, используя команду удаления.

Задание 11

Проверьте функциональность менеджера веб приложений: список веб-приложений, проверка серверного статуса, перезагрузка выбранного веб приложения, проверка серверных сессий, запустить веб-приложений и удаление веб-приложений.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 7.4 Проверка статуса сервера

Первый компонент менеджера веб-приложений, который мы собираемсяиспользовать – это ***серверный статус***. Для получения доступа к статусу ApacheTomcat, нажмите на ссылку серверного статуса в верхней области домашней страницы менеджера веб-приложений. Или вы можете ввести следующий URL в ваш браузер: <http://localhost:8080/manager/status>.

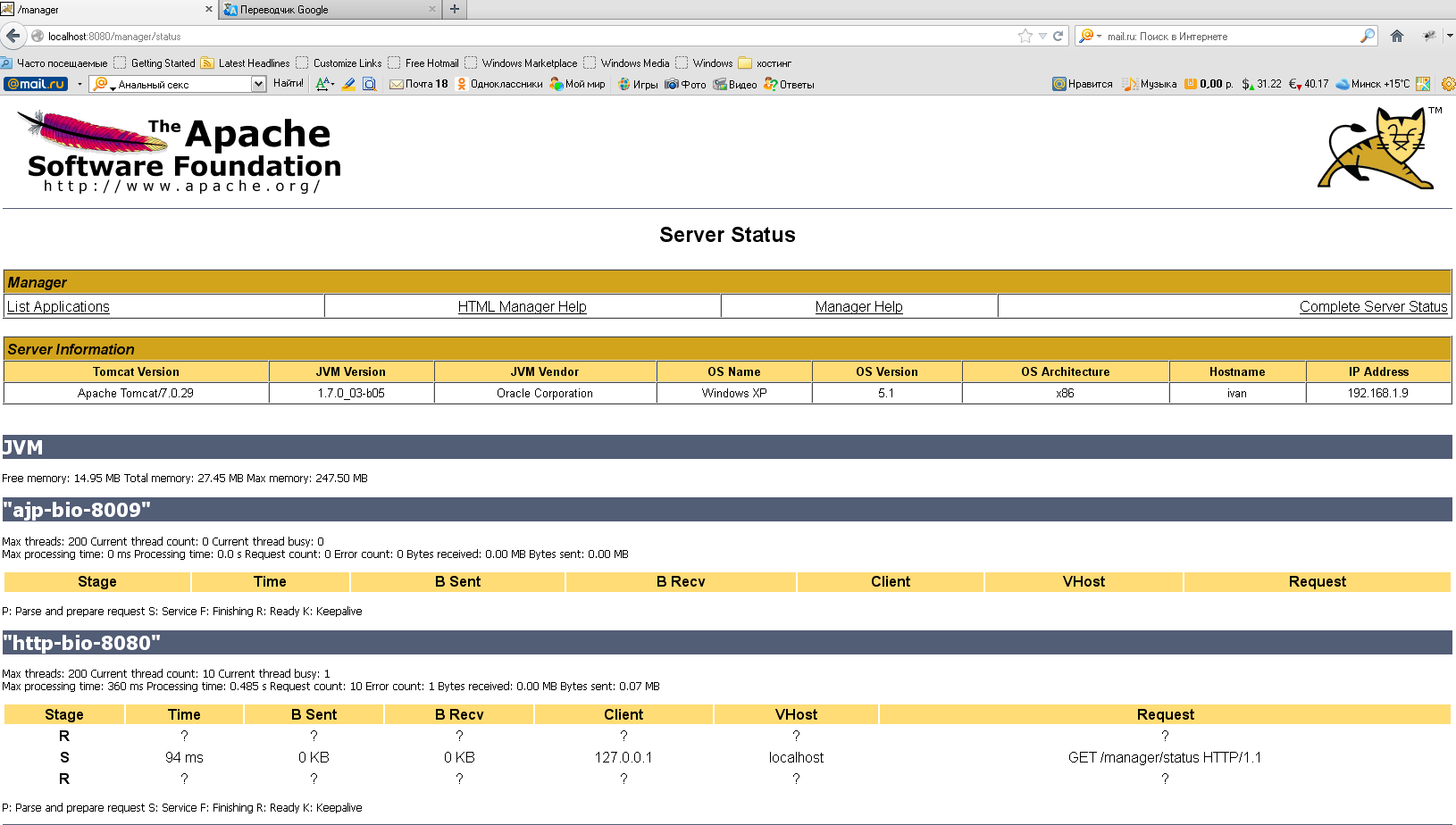


Рисунок 7.4.1 Страница статуса менеджера веб-приложений.

На странице статусавы можете увидеть детали о вашей операционной системе (имя, архитектура, и версия), ваша виртуальная машина Java (vendor и версия), и версия ApacheTomcat, которые вы запускаете.

В дополнение, внизу экрана, вы можете увидитьJVM статус памяти и статус AJP и коннекторы HTTPTomcat- статус потоков и статус активных запросов.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 7.5 Установка нового веб-приложения

Первую команду, которую мы хотим использовать – это установка, которая используется для установки нового веб-приложения. Для установки нового веб-приложения используя веб-интерфейс, вы можете использовать HTML форму домашней страницы менеджера веб-приложений. Две HTML формы доступны: одна – для установки веб-приложения для файла или директории, уже доступной на сервере Tomcat, и другой, позволяющий вам установить веб-приложение из WAR файла, расположенного на вашем локальном компьютере.

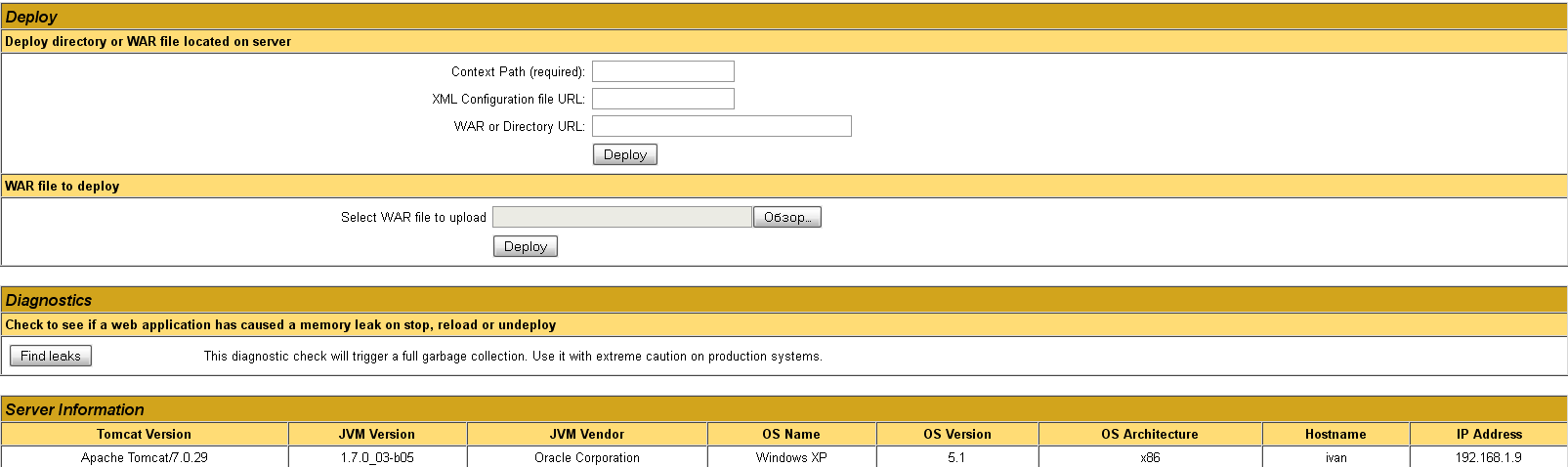


Рисунок 7.5.1 Используя HTMLформу для вызова команду установки в менеджере веб-приложений.

Наиболее частое использование менеджера веб-приложений для установки на Tomcat – это удаленная установка, где располагается ваше веб-приложение, упакованное в WAR файл, которое вы хотите установить на удаленный сервер Для того, что бы достичь этого, мы будем использовать вторую форму в секции установки на домашней странице менеджера веб-приложений, обозначенной «WAR файл на установку». Все что вы должны сделать – это выбрать файл на вашем диске (нажимая на «Обзор…»), и нажать на кнопку установки под полем выбора. Веб-приложение будет установлено по контекстному пути, соответствующему имени WAR файла, например testWebApp.WAR, контекстный путь - /testWebApp .

Когда выполнение команды завершится, то страница менеджера веб-приложений перезагрузится, и вы увидите сообщение, что операция завершена успешна (просто будет написано «ОК») в поле для сообщений. В дополнение, вы увидите новое, установленное сообщение в списке приложений.

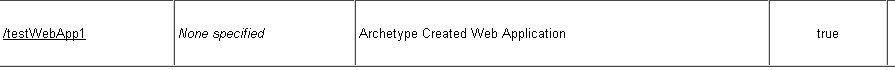


Рисунок 7.5.2 Установленное веб-приложение

|  |
| --- |
|  |
|  |
|  |

Вы можете также использовать менеджер веб-приложений для установки веб-приложения, расположенного на файловой системе сервера.Это может быть сделано с помощью верхней части формы установки, которая содержит три поля: контекстный путь, адрес (URL) конфигурационного файла XML, адрес (URL) WAR или директории. Первый параметр, контекстный путь, это имя, рекомендуемое для установки. Если вы знаете расположение WAR или адрес директории на вашей локальной системе, то вы должны внести адрес в «WAR или адрес директории» и оставить поля «адрес XML конфигурационного файла XML». Если же у вас есть адрес конфигурационного файла XML, то вы должны сделать наоборот.



Рисунок 7.5.3 Установка с помощью адреса веб-приложения на файловой системе сервера.

Задание 12

Установите ваше веб приложение, собранное maven, на Tomcat сервер при помощи менеджера веб приложений. Проверьте статус установки при помощи менеджера верб приложений и Log файлов.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 7.6 Перезагрузка, остановка и запуск существующего веб-приложения

Для перезагрузки существующего приложения, необходимо вызвать *reload*команду. Данная команда перегружает все веб-компоненты, включая сервлеты, JSP, и зависимые классы, ассоциированные с именем веб приложения.



Рисунок 7.6.1 Перезагрузка веб-приложения.

После вызова команды ***reload***, в строке сообщений появится *«OK–Reloadedapplicationatcontextpath /testWebApp»*, что обозначает, что приложение было перегружено успешно.



Рисунок 7.6.2 Сообщение об успешной перезагрузки.

Вlog-файлахдолжнопоявитсяследующаязапись: *«INFO: Reloading Context with name [/testWebApp] is completed»*.

Для того чтобы ***остановить*** веб-приложение необходимо вызвать команду ***stop***.

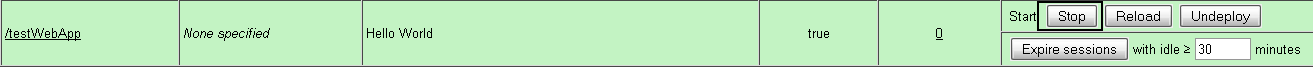


Рисунок 7.6.3 Остановка существующего веб-приложения.

В результате вызова этой команды, в строке сообщения появится *«OK - Stoppedapplicationatcontextpath /testWebApp»*.



Рисунок 7.6.3 Сообщение об успешной остановки.

Если вы попробуете открыть ваше приложение в браузере, например <http://localhost:8080/testWebApp/>, то увидите 404 ошибку (страница не найдена).

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

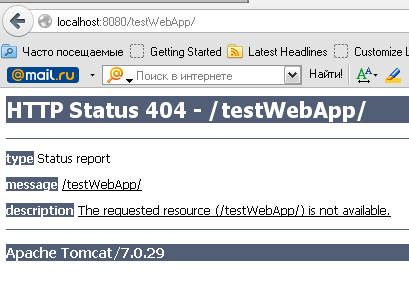


Рисунок 7.6.4 Приложение не доступно, если оно остановлено.

Для ***старта*** веб-приложения необходимо вызвать команду ***start***.



Рисунок 7.6.5 Запусквеб-приложения

Послезапускаприложения, встрокесообщенияпоявится*«OK - Started application at context path /testWebApp»*.



Рисунок 7.6.6 Сообщение об успешном запуске веб-приложения.

Если вы откроете веб-приложение, то увидите, что оно запустилось успешно.

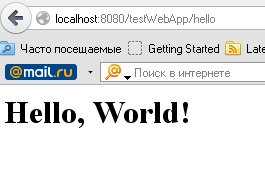


Рисунок 7.6.6 Приложение запустилось успешно.

|  |
| --- |
|  |
|  |
|  |
|  |

Модуль 7.7 Управление сессиями веб-приложения

HTTP протокол, который мы используем для связи с сервером веб-приложений,не сохраняет состояния. Это означает, что протокол обрабатывает каждый запрос независимо, без информации о предыдущих запросах от того же пользователя или от того же интернет адреса. Контейнер сервлетов, такой как Tomcat, использует пользовательские сессии для сохранения состояния между запросами. Tomcat связывает каждого клиента с ***sessionid***, и сохраняет параметры в сессии.

Вы можете увидеть число связанных пользовательских сессий для каждого веб-приложения на домашней странице менеджера веб-приложений. Эта информация представлена числом в колонке сессии списка установленных веб-приложений.

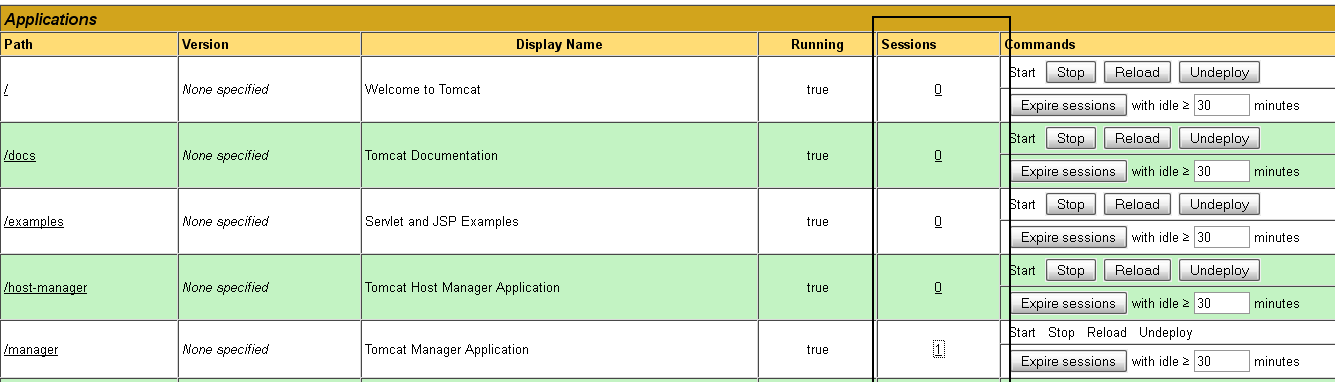


Рисунок 7.7.1 Колонка сессий веб-приложений.

Вы можете изменить время, в течении которого сессия «жива», установив значение в поле рядом с кнопкой «Expiresessions».

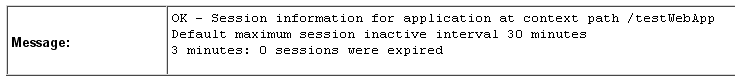


Рисунок 7.7.2 Изменение времени деактивации сессии.

Вы можете получит детальную информацию о каждой сессии, нажав на ссылку (количество сессий) в колонке сессий.



Рисунок 7.7.3 Детальная информация о сессиях.

Если же вы хотите посмотреть полную информацию о конкретной сессии, нажмите на ссылку с id сессии.

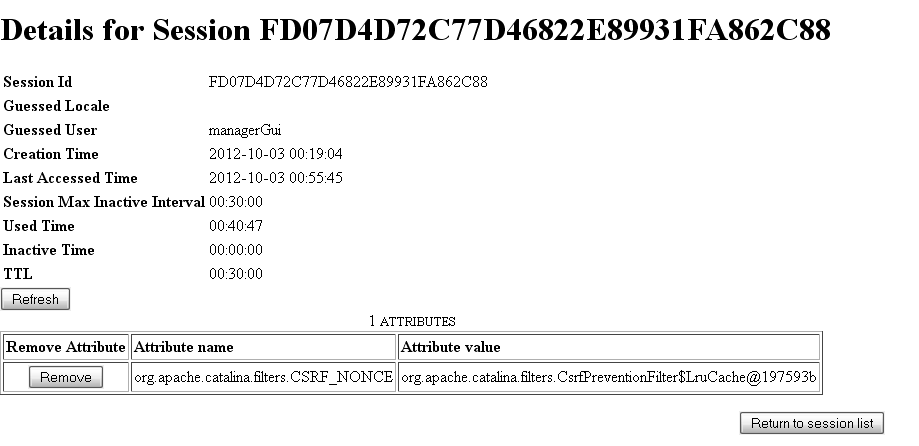


Рисунок 7.7.4 Детальная информация о конкретной сессии.

На странице детальной информации о сессии, находится следующая информация:

1. ***SessionId*** – идентификатор сессии, уникальная строка для сущности Tomcat.
2. ***GuessedLocale –*** пользовательская локаль, если она может быть определена из атрибута запроса.
3. ***GuessedUsername –*** Имя вошедшего в систему пользователя, если он может быть взят из запроса.
4. ***CreationTime –*** Дата и время, когда сессия создана.
5. ***LastAccessedTime –*** Дата и время последнего доступа сессии.
6. ***UsedTime –*** Период времени между созданием и временем последнего использования.
7. ***Inactive –*** Период времени с момента последнего обращения.
8. ***TTL–***Время жизни сессии.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |

**Тема 8**. Использование текстового интерфейса для доступа к менеджеру веб-приложений Tomcat

Менеджерский текстовый интерфейс может быть вызван напрямую из браузера, путем ввода команды в форме URL, и получения текстового ответа прямо в браузере. Однако, на практике, текстовый доступ к менеджеру веб-приложений Tomcat осуществляется с помощью ***maven***.

Модуль 8.1 Tomcatmaven плагин

Tomcatmaven плагин предоставляет возможности для установки и удаления веб-приложения на сервер Tomcat при помощи текстового интерфейса доступа к менеджеру веб-приложений.

Существуют следующие команды работы с этим плагином:

1. ***Переустановка веб-приложения***

Для того, что бы переустановить веб-приложение, вам необходимо вызвать

***mvn tomcat:redeploy***.

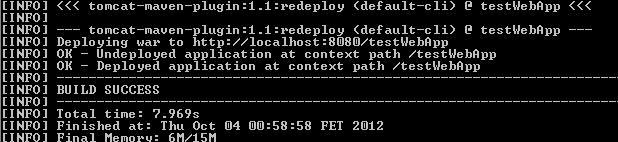


Рисунок 8.1.1 Результат выполнения команды переустановки

Как видно из рисунка 8.1.1, в результате выполнения команды ***redeploy***maven сначала удаляет предыдущий вариант веб-приложения из Tomcat, а затем устанавливает новый.

|  |
| --- |
|  |
|  |

1. ***Удаление веб-приложения из сервера Tomcat***

Для того, что бы удалить веб-приложение, вам необходимо вызвать

***mvn tomcat:undeploy***

|  |
| --- |
|  |
|  |



Рисунок 8.1.2 Результат выполнения команды удаления

Как видно из рисунка 8.1.2, в результате выполнения команды ***undeploy***mavenудалит веб-приложение из сервера Tomcat.

1. ***Запуск веб-приложения на сервере Tomcat***

Для запуска веб-приложения на сервере Tomcat, вам необходимо вызвать команду ***mvntomcat:start***.



Рисунок 8.1.3 Результат выполнения команды запуска веб-приложения.

Для того, что бы команда ***start*** выполнилась успешно, необходимо, чтобы веб-приложение было установлено на сервере и остановлено.

|  |
| --- |
|  |
|  |

1. ***Остановка веб-приложения на сервере Tomcat***

Для остановки веб-приложения на сервере Tomcat, вам необходимо вызвать команды ***mvntomcat:stop***.



Рисунок 8.1.4 Результат выполнения команды остановки веб-приложения

Для того, что бы команда ***stop***выполнилась успешно, необходимо, чтобы веб-приложение было установлено на сервере.

|  |
| --- |
|  |
|  |

1. ***Вывод всех существующих сессий веб-приложения***

Для вывода всех сессий веб-приложения, вам необходимо вызвать команду ***mvntomcat:sessions***.

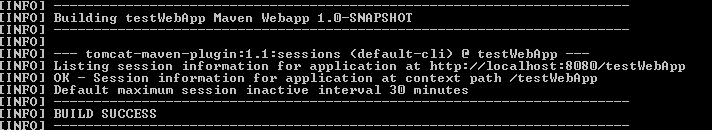


Рисунок 8.1.5 Результат выполнения команды вывода всех сессий веб-приложения

Как видно на рисунке 8.1.5, в результате мы получили статистику сессий, установленного веб-приложения.

|  |
| --- |
|  |
|  |

1. ***Вывод статики об установленных веб-приложениях на серверTomcat***

Для того, что бы вывести список установленных приложений вам необходимо вызвать команду ***mvntomcat:list***.

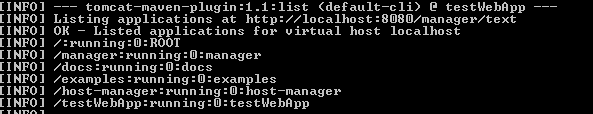


Рисунок 8.1.6 Результат выполнения команды вывода списка всех веб-приложений

Как видно на рисунке 8.1.6, в результате мы получили список всех установленных веб-приложений на сервере Tomcat.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |

1. ***Вывод информации о сервере Tomcat***

Для того, что бы вывести информацию о сервере Tomcat вам необходимо вызвать команду ***mvntomcat:info***

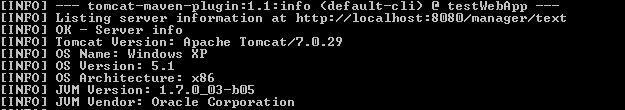


Рисунок 8.1.7 Информация о сервере Tomcat

При вызове вышеупомянутой команды, в консоль будут выведены информация о сервере Tomcat.

Задание 13

Соберите ваше веб приложение при помощи maven. Установите, переустановите, посмотрите статус сервера при помощи Tomcatmavenplugin. Опишите проблемы, возникшие при использовании, и попробуйте их решить.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 8.2Настройка плагина

Для настройки плагина и установки веб-приложения вам необходимо открыть pom.xmlвашего веб-приложения.

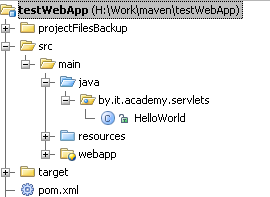


Рисунок 8.2.1 Веб-приложение

В pom.xml добавьте в секцию plugins следующую запись:



Рисунок 8.2.2 Настройка плагина для доступа к менеджеру веб-приложения по текстовому интерфейсу

После того, как вы настроили плагин при вызове команды mvninstall, плагин откроет доступ к текстовому интерфейсу менеджеравеб-приложений, и установит веб-приложение на сервер Tomcat.

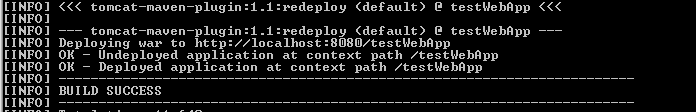


Рисунок 8.2.3 Установка приложения на сервер Tomcat

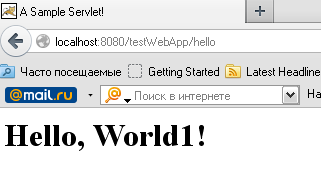


Рисунок 8.2.4 Работающее веб приложение.

Как вы можете убедиться, использование maven плагина для доступа к текстовому интерфейсу менеджера веб-приложений, не так удобно как к веб-интерфейсу, однако это гораздо проще выполнять установку веб-приложения как часть сборки проекта.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Тема 9.**Интеграция Tomcatи ApacheWEB сервера

Существует несколько причин, почему необходимо интегрировать Tomcat и Apache веб сервер вместо того, чтобы использовать Tomcat как контейнер веб-приложений и HTTP сервер.

Исторически, первая причина была ***производительность***. Apache –это высокопроизводительный веб сервер, который обрабатывает статический контент, такой как html файлы, картинки и видео, значительно быстрее чем Tomcat. Однако, последние релизы Tomcat, особенно Tomcat 7, по производительности не уступают Apache, а иногда даже превосходят.

Вторая причина – это настраиваемость Apache веб сервера. За время разработки Apache веб сервера (более 15-ти лет), Tomcat просто не может предоставить такое же количество возможностей по настройке сервера, как Apache веб сервер. Диапазон модулей простирается от специфических для конкретных языков программирования модулей (PHP, Perl, Python и другие)и функций безопасности (SSL и аунтификация). Также существуют прокси модуль,***mod\_proxy***, модуль переадресации URL,***mod\_rewrite***,поддерживающий пользовательское логирование и фильтрацию запросов, и т.д.Для большинства веб-приложений необходима некоторая из этой функциональности, и даже если использовать Tomcat, как контейнер сервлетов, Apache веб сервер будет общедоступным веб сервером для вашего нового веб-приложения.

В конечном итоге, Apache веб сервер – очень мощный и высокомаштабируемый, поэтому он необходим для production среды. Apache веб сервер обладает хорошей способностью к обработке ресурсов, минимальными потерями памяти, удобным запуском и остановкой.А также, можно часто увидеть, работающий месяцами без перезагрузки Apache веб сервер.

Итак, когда необходима интеграция Apache и Tomcat, существуют некоторые задачи, которые необходимо решить для их успешной интеграции:

1. Настройка Apache веб сервера для перенаправления выбранных запросов на Tomcat.
2. Настройка Apache веб сервера и Tomcat связи одного с другим по известному протоколу.
3. Настройка Tomcat для понимания запросов, направленных ему из Apache веб сервера, и их соответствующая обработка.

|  |
| --- |
|  |
|  |
|  |

Модуль 9.1 Настройка интеграции с помощью mod\_proxy

Как говорит имя модуля mod\_proxy – это расширение Apache веб сервера, для перенаправления веб траффика. При определенной настройке mod\_proxy модуля, вы можете перенаправлять некоторые запросы на обработку Tomcat серверу.

Существуют два вида mod\_proxy модулей: mod\_proxy\_http, используемый для перенаправления трафика по HTTP протоколу, и mod\_proxy\_ajp, использующий AJP протокол для связи.

Для того, что бы настроить интеграцию Apache и Tomcat с помощью mod\_proxy вам необходимо сделать следующее:

1. Включить загрузку модуля mod\_proxy в http.conf (Apache2.2\conf\httpd.conf)

LoadModule proxy\_module modules/mod\_proxy.so

LoadModule proxy\_http\_module modules/mod\_proxy\_http.so

1. Настроить виртуальный хост для переадресации запросов по контекстному пути /testWebApp на <http://localhost:8080/testWebApp>( адрес веб-приложения tomcat). Для этого необходимо сделать следующее:
   1. Включаем модуль mod\_vhost\_alias.soв http.conf (Apache2.2\conf\httpd.conf)

LoadModule vhost\_alias\_module modules/mod\_vhost\_alias.so

Include conf/extra/httpd-vhosts.conf

* 1. Настраиваем виртуальный хост в Apache2.2\conf\extra\httpd-vhosts.conf

<VirtualHost \*:80>

ServerAdmin admin@localhost

DocumentRoot "H:/opt/Apache2.2/htdocs

ServerName localhost

ErrorLog "logs/localhost-error.log"

CustomLog "logs/localhost-access.log" common

ProxyRequests off

ProxyPreserveHost on

ProxyPass /testWebApp http://localhost:8080/testWebApp

ProxyPass /examples http://localhost:8080/examples

ProxyPassReverse /testWebApp http://localhost:8080/testWebApp

ProxyPassReverse /examples http://localhost:8080/examples

</VirtualHost>

***ProxyRequest*** – директива, позволяющая перенаправлять стандартный прокси-запрос

|  |
| --- |
|  |
|  |
|  |

***ProxyPreserveHost***– использует входящий HTTP запрос заголовка на запрос прокси.

|  |
| --- |
|  |
|  |

***ProxyPass***– настраивает соответствие одного удаленного сервера для вызова другого.

|  |
| --- |
|  |
|  |

***ProxyPassReverse***– директива, маскирующая внутренние перенаправления между страницами веб-приложения, для того, что бы не было видно, что веб-приложение вызывается не из apacheweb сервера, а из Tomcat.

|  |
| --- |
|  |
|  |

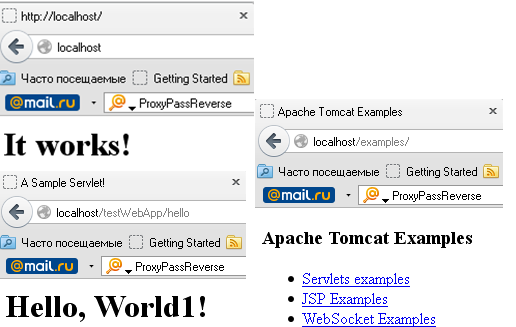


Рисунок 9.1.1 Результат настройки интеграции

|  |
| --- |
|  |
|  |

Задание 14

Настройте интеграцию ApacheWebсервера и ApacheTomcatcпомощью mod\_proxy.Настройте доступность вашего веб приложения через вызов localhost. Запустите ApacheWeb сервер и Tomcat. Проверьте работоспособность приложения. Проконтролируйте статус загрузки и работы приложения через log-файлы и менеджер веб-приложений.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 9.2 Настройка интеграции с помощью mod\_jk

***AJP –*** это ApacheJServ протокол.AJP протокол передает туже информацию, что и HTTP, но в оптимизированном, бинарном формате.Apache предоставляет реализацию AJP протокола как часть подпроекта TomcatConnectors (или mod\_jk), который является набором библиотек, содержащих реализацию AJP протокола для интеграции между Apache веб-сервером и Tomcat.

Для того чтобы настроить интеграцию нужно сделать следующее:

1. Скачать бинарную версию модуля ***mod\_jk.so***для конкретной версии apache и скопировать ее в Apache2.2\modules\

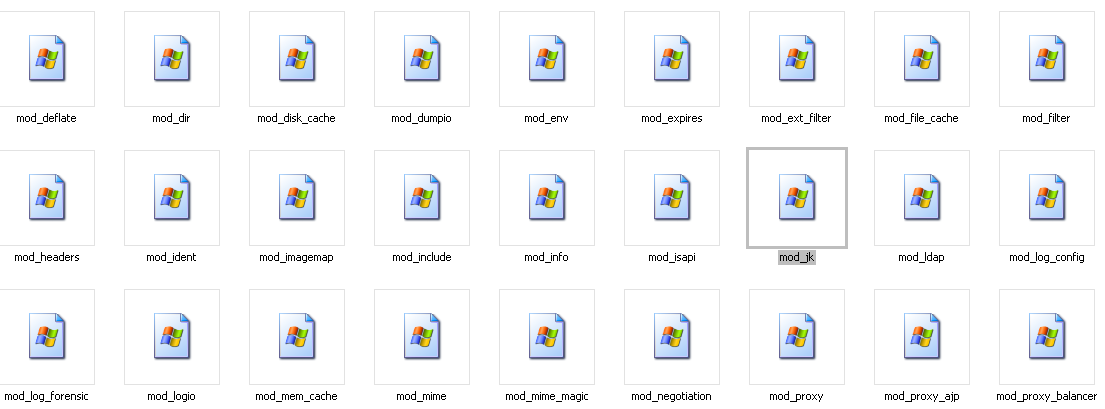


Рисунок 9.2.1 Модуль mod\_jk

1. Включить загрузку модуля mod\_jk и настроить включение конфигруационного файла mod\_jk.conf в http.conf (Apache2.2\conf\httpd.conf)

LoadModulejk\_modulemodules/mod\_jk.so

Include conf/extra/mod\_jk.conf

|  |
| --- |
|  |
|  |

1. Создатьконфигурационныйфайлmod\_jk.conf (Apache2.2\conf\extra\mod\_jk.conf)

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |

# mod\_jk config

<IfModule mod\_jk.c>

JkworkersFile conf/extra/workers.properties

# Tomcat examples context

JkMount /examples/\* mytomcat

#<Directory /home/\*/public\_html>

#AllowOverride FileInfo AuthConfig Limit Indexes

#Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec

#<Limit GET POST OPTIONS PROPFIND>

#Order allow,deny

#Allow from all

#</Limit>

#<LimitExcept GET POST OPTIONS PROPFIND>

#Order deny,allow

#Deny from all

#</LimitExcept>

#</Directory>

</IfModule>

|  |
| --- |
|  |
|  |

1. Создатьконфигурационныйфайлworkers.properties (conf/extra/workers.properties)

worker.list=mytomcat

worker.ajp13.port=8009

worker.ajp13.host=localhost

worker.ajp13.type=ajp13

|  |
| --- |
|  |
|  |
|  |

1. Настроить tomcat\conf\server.xml

<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />

|  |
| --- |
|  |
|  |
|  |

В результате мы получим результат, аналогичный результату полученному на рисунке 9.1.1.

Задание 15

Настройте интеграцию ApacheWeb сервера и ApacheTomcatc помощью mod\_jk. Настройте доступность вашего веб приложения через вызов localhost. Запустите ApacheWeb сервер и Tomcat. Проверьте работоспособность приложения. Проконтролируйте статус загрузки и работы приложения через log-файлы и менеджер веб-приложений.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**тема 10.** Основы Subversion

Модуль 10.1 Основные понятия управления версиями

***Subversion*** - это централизованная система для совместной работы. В её основе лежит ***хранилище***, которое содержит данные в форме ***дерева файловой системы*** - обычной иерархии файлов и папок.

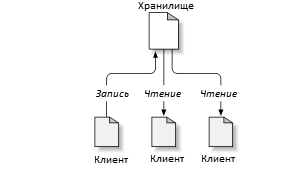


Рисунок 10.1.1 Система клиент-сервер

|  |
| --- |
|  |
|  |

*Subversionзапоминает* каждое внесённое изменение, когда-либо записанное в хранилище: любое изменение любого файла, и даже изменения в самом дереве каталогов, такие как добавление, удаление и перемещение файлов и каталогов.Когда клиент просто читает данные из хранилища, он получает *последнюю версию* дерева файловой системы. Но, помимо этого, клиент имеет возможность посмотреть *предыдущие* состояния файловой системы.

Существуют различные модели обмена информацией между пользователями, не мешая друг другу.

***Проблема совместного использования файлов***. Два пользователя редактируют один и тот же файл из репозитория и затем сохраняют его. В результате, история изменений сохранится и в последней версии этого файла в репозитории будут изменения пользователя, который сохранял изменения последним.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

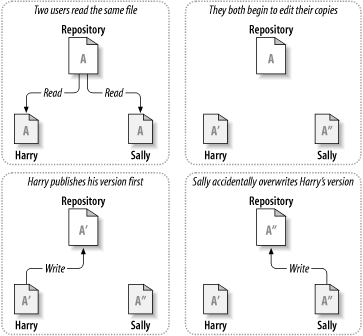


Рисунок 10.1.2 Проблема совместного использования файлов.

Для того что бы избежать данной проблемы, были разработаны следующие модели:

1. ***Модель Блокирование-Изменение-Разблокирование.***

Эта модель реализована следующим образом. Если два пользователя пытаются внести изменения в один и тот же файл в хранилище, то хранилище разрешает вносить изменения только одному человеку за раз. Для того, чтобы пользователь мог внести изменения в хранилище, он должен сначала его ***заблокировать***. При блокировки другой пользователь не сможет внести изменение в хранилище, пока первый не закончит правку. После завершения работы первого пользователя, блокировка снимается и файл доступен для редактирования вторым пользователем (и соответственно его блокировке).

Проблема с моделью блокирование-изменение-разблокирование в ее жестокости, она может создать следующие неудобства пользователям:

1. Блокирование может вызвать административные проблемы.
2. Блокирование может вызвать излишнюю поочерёдность.
3. Блокирование может вызвать ложное чувство безопасности.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

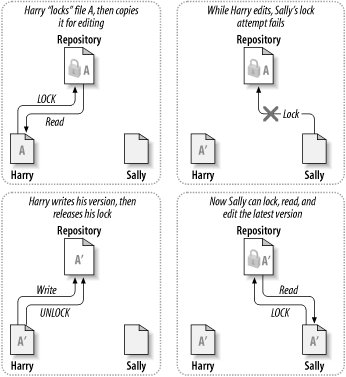


Рисунок 10.1.3 Модель Блокирование-Изменение-Разблокирование

|  |
| --- |
|  |
|  |

1. ***Модель Копирование-Изменение-Слияние.***

Subversion, CVS и другие системы управления версиями используют модель *копирование-изменение-слияние* вместо *блокирования*. В этой модели клиент каждого пользователя считывает из хранилища проект и создаёт персональную рабочую копию - локальное отражение файлов и каталогов хранилища. После этого пользователи работают, одновременно изменяя свои личные копии. В конце концов, личные копии сливаются в новую, финальную версию. Обычно система управления версиями выполняет слияние автоматически, но в общем случае необходимо присутствие человека.

***Вот пример:*** Два пользователя создали свои рабочие копии одного и того же проекта. Они работают одновременно, и вносят изменения в файл A в своих рабочих копиях. Первый пользователь сохраняет свои изменения в хранилище первым. Затем, когда второй пытается сохранить свои, хранилище информирует его, что его файл А устарел. Другими словами, файл А в хранилище был изменён с тех пор, как второй пользователь получил его. Тогда второй пользователь выполняет ***слияние (merge)*** любых изменений хранилища со своей рабочей копией. Вероятно, что изменения первого пользователя не пересекаются с его собственными, и, поскольку теперь его рабочая копия содержит оба набора изменений, он записывает её обратно в хранилище.

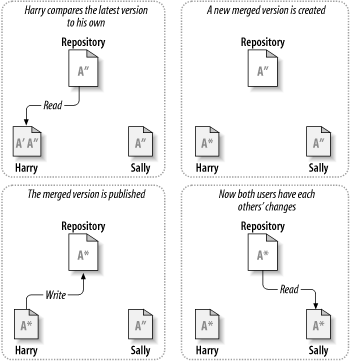
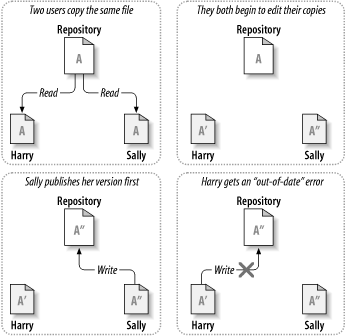


Рисунок 10.1.4 Модель Копирование-Изменение-Слияние

Но что будет, если изменения первого пользователя всё-таки пересекаются с изменениями второго? Эта ситуация, называемая ***конфликтом***, обычно не большая проблема. Когда второй пользователь просит объединить свои изменения с изменениями из хранилища, его копия файла А помечается как находящаяся в ***состоянии конфликта***: он видит оба набора конфликтующих изменений, и вручную выбирает между ними. ***Обратите внимание***, *программа не может автоматически разрешать конфликты*, только человек способен понять и сделать необходимый осмысленный выбор. Когда второй пользователь разрешит пересекающиеся изменения, он может безопасно сохранить объединённый файл обратно в хранилище.

Всё сводится к одному решающему фактору: ***взаимодействию пользователей***. При плохом взаимодействии пользователей, увеличивается количество и смысловых, и синтаксических конфликтов.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 10.2 Установка сервера subversion

Для того, что бы установить сервер subversion, необходимо сделать следующее:

1. Скачиваем файл VisualSVN-Server.msi и запускаем установку

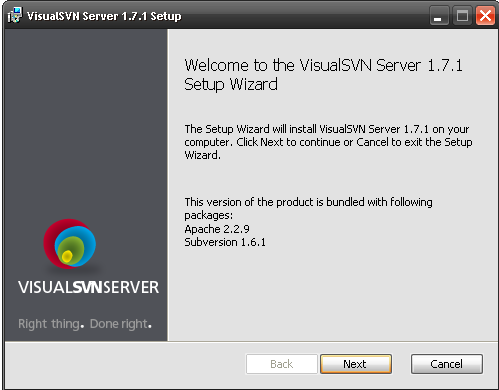


Рисунок 10.2.1 Начало установки

1. В мастере установки указываем, использовать ли для доступа HTTPS, либо просто HTTP. Указываем порт для прослушивания по выбранному протоколу и способ аутентификации. Так же указываем каталог, в котором будут храниться репозитории.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

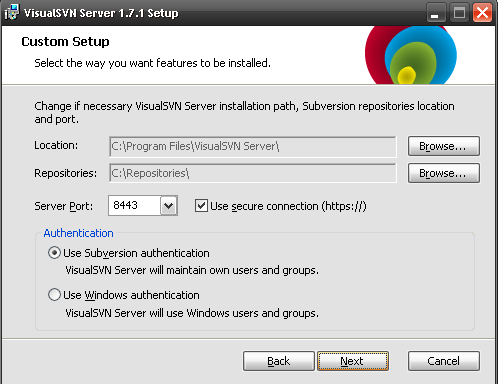


Рисунок 10.2.2 Установка порта для прослушивания и поддержки https

1. После установки открываем ManagementConsole (через Пуск, например) и создаем пользователей и репозитории.

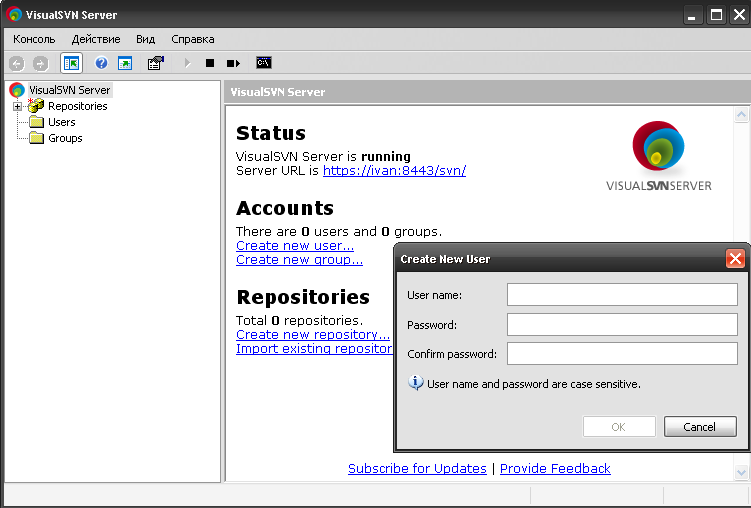


Рисунок 10.2.3 Создание пользователя в VisualSVN

Для управления репозиторием и svn аккаунтами можно применить ***VisualSVNManagementConsole***.

Давайте создадим нового пользователя. Для этого выберем папку **Users** в левой части окна менеджера. При нажатии на правую кнопку мыши появится контекстное меню, в котором можно выбрать пункт «создать пользователя».

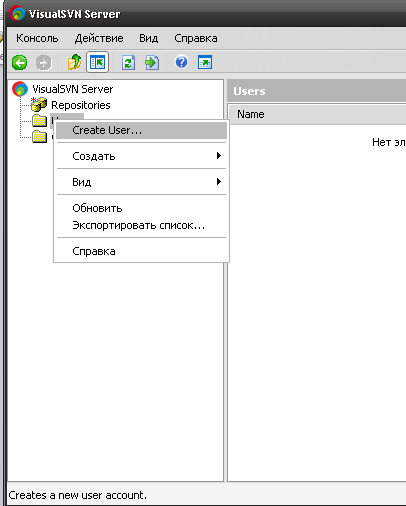


Рисунок 10.2.4 Меню пользователей

Далее вы вводите username, password и confirmpassword.

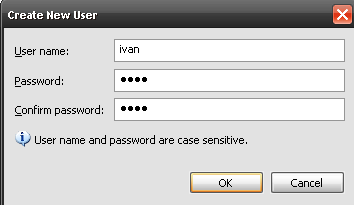


Рисунок 10.2.4 Создание пользователя

Для создания репозитория необходимо в левой части окна менеджера выбрать пункт ***«Repositories»*** и нажать на правую кнопку мыши. В появившемся контекстном меню нужно выбрать ***«Create New Repository»***.

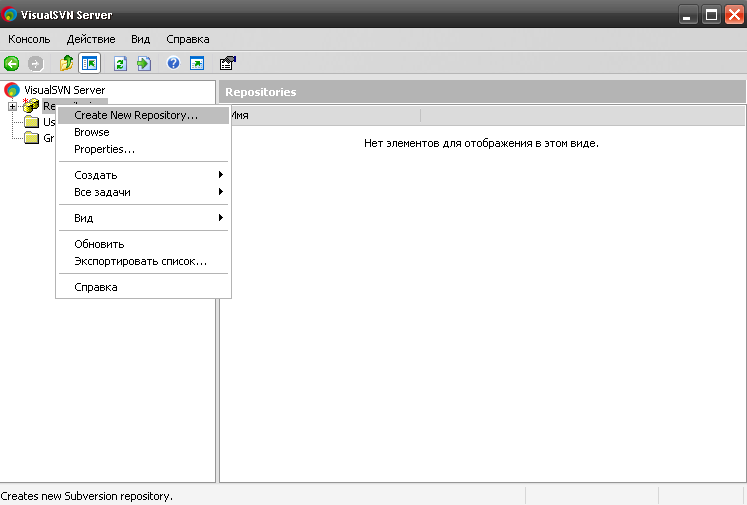


Рисунок 10.2.5 Контекстное меню репозитория

Для создания репозитория нужно ввести имя нового репозитория и установить галочку на пункте «создать структуру по умолчанию» для стандартизации структуры проекта с общепринятым подходом.

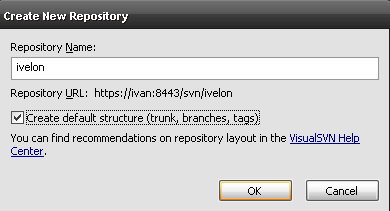


Рисунок 10.2.6 Создание нового репозитория

После завершения операции, репозиторий будет доступен либо через svn-клиент, либо через браузер.

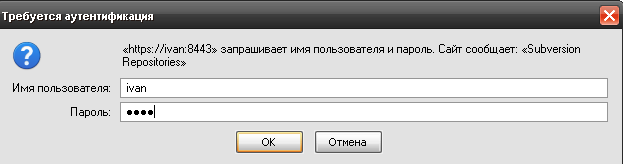


Рисунок 10.2.7 Ввод имени пользователя и пароля для доступа к репозиторию

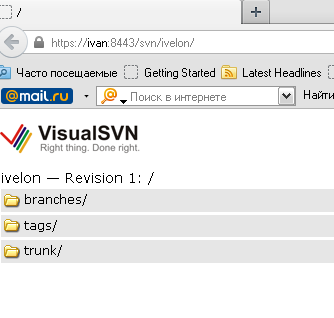


Рисунок 10.2.8 Структура репозитория

Задание 16

Установите subversion сервер. Создайте новое хранилище. Создайте пользователей. Проверьте доступ к хранилищу через ваш браузер.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 10.3 Установка клиента subversion и его использование

В качестве клиента subversion мы будем использовать TortoiseSVN. ***TortoiseSVN*** — это бесплатный Windows-клиент с открытыми исходным кодом для системы управления версиями ApacheSubversion. То есть TortoiseSVN управляет файлами и директориями во времени.

TortoiseSVN поставляется в виде простого в использовании установочного файла.Для запуска установки необходимо нажать на этот файл.

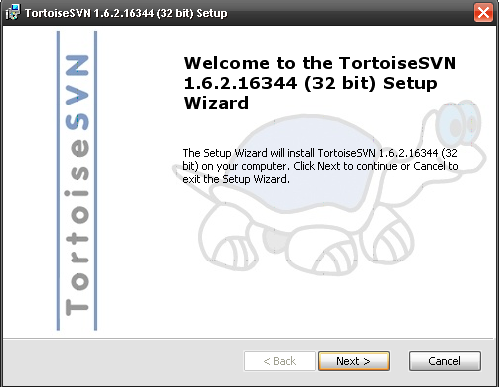


Рисунок 10.3.1 Установка TortoiseSVN

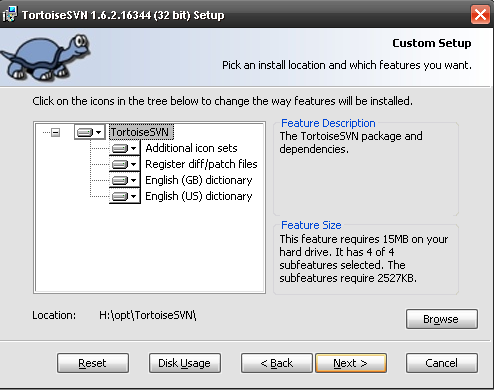


Рисунок 10.3.2 Задание пути установки TortoiseSVN

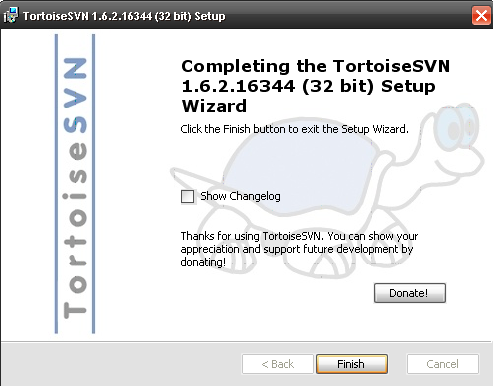


Рисунок 10.3.2 Окончание установки TortoiseSVN

После завершения установки вам **необходимо** будет ***перезагрузить*** ваш компьютер.

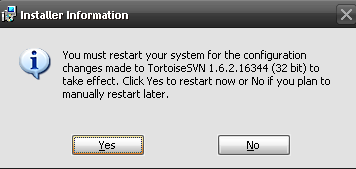


Рисунок 10.3.3 Перезагрузка компьютера

Для начала работы с клиентом вам необходимо открыть проводник Windows, а затем, нажимая правую клавишу мыши, откроется контекстное меню, в котором вы можете работать с функциями клиента TortoiseSVN.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

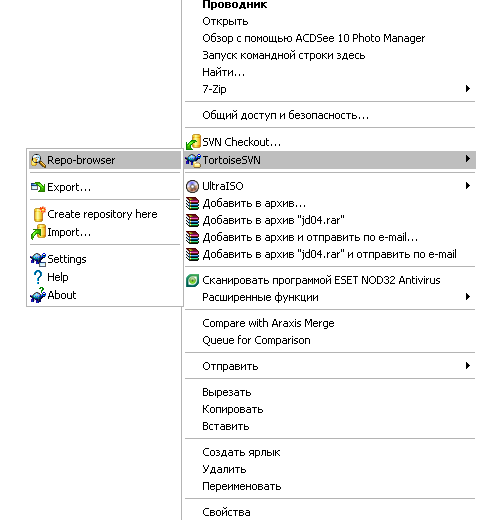


Рисунок 10.3.4 Доступ к функциям клиента.

У нас есть хранилище <https://ivan:8443/svn/ivelon/trunk/>, но оно совершенно пустое в данный момент. У нас есть проект testWebApp, который мы хотим поместить в это хранилище для совместного пользования.

|  |
| --- |
|  |
|  |

Что нужно, что бы поместить testWebAppпод управление subversion?

Перейдите к папке testWebApp в проводнике и вызовите ***TortoiseSVN>Import***.

|  |
| --- |
|  |
|  |

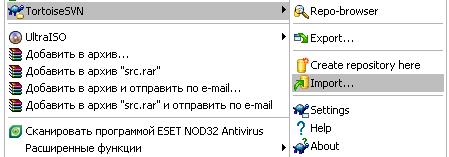


Рисунок 10.3.5 Функция импорта кода в репозиторий.

Далее откроется диалог импорта, в котором нужно в строке адреса репозитория внести адрес используемого хранилища <https://ivan:8443/svn/ivelon/trunk/>, а в строке сообщения – поясняющее сообщение, что бы потом понять для чего вносились файлы в хранилище.

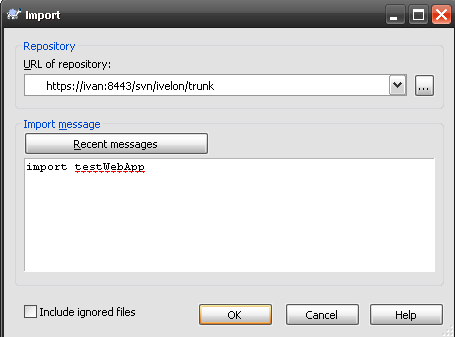


Рисунок 10.3.6 Диалог импорта.

При проведении операции, клиент спросит имя и пароль к хранилищу. Вам необходимо будет ввести имя пользователя и пароль, а также поставить галочку напротив пункта меню *«сохранить пароль»*.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

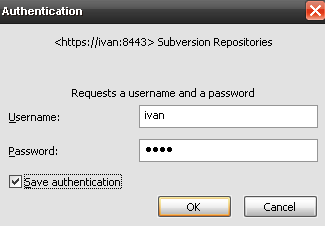


Рисунок 10.3.7 Диалог ввода имени пользователя и пароля.

В результате вы получите сообщение о помещенных файлах в репозиторий.

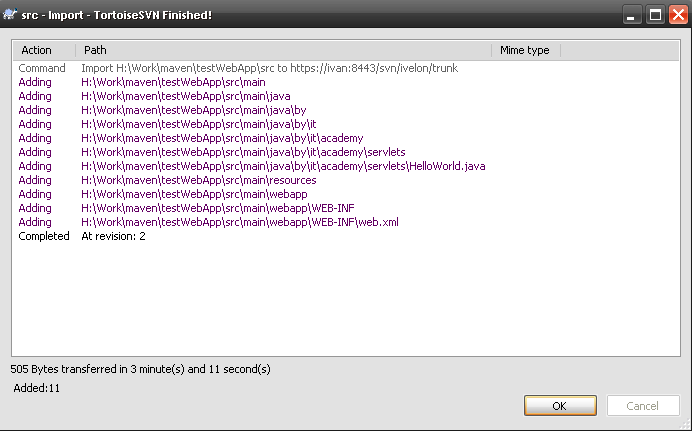


Рисунок 10.3.7 Отчет о помещенных файлах.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 10.4 Сохранение своих изменений в репозиторий

Если Вы хотите внести свои изменения в скачанный репозиторий и у Вас есть на это право (право коммита в репозиторий), то просто начинайте работать с локальной рабочей копией как с обычной папкой: создавайте, редактируйте, удаляйте файлы и/или папки.

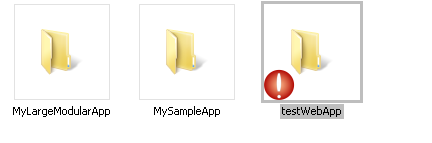


Рисунок 10.4.1 Измененные файлы.

Теперь чтобы внести изменения в хранилище, нужно:

1. Заходим в папку, в которой находятся файлы скачанной рабочей копии в Проводнике Windows. Внимание! Рабочая копия должна быть скачана от логина пользователя, у которого есть право на коммит изменений (коммит-флаг).
2. Нажимаем правой кнопкой по свободному месту в папке, выбираем «SVNCommit…».

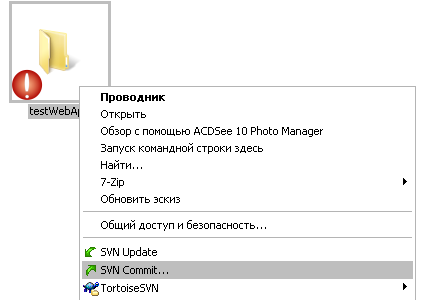


Рисунок 10.4.2Команда SVN Commit…

1. В открывшемся окне «Commit» в секции «Message» вводим краткое описание своих изменений, но описание можно и оставить пустым (не рекомендуется, т.к. другие разработчики или пользователи не смогут понять что именно было изменено).

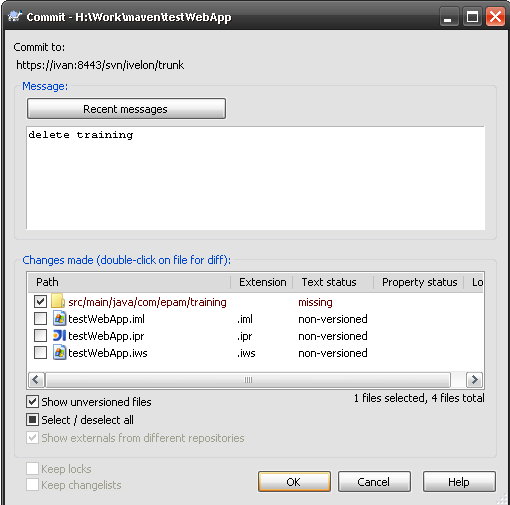


Рисунок 10.4.3 Команда Commit и комментарии.

1. В секции «Changesmade (double-clickonfilefordiff)» Вы увидите список всех изменённых, добавленных и удалённых Вами файлов локальной рабочей копии. У изменённых файлов будут автоматически стоять флажки, у остальных — нет.
2. После выделения нужных файлов или всех сразу («Select / deselectall»), нажмите кнопку «OK» и дождитесь окончания загрузки изменений на сервер. Будет открыто окно со списком добавленных, изменённых и/или удалённых файлов, а также указан текущий номер ревизии.

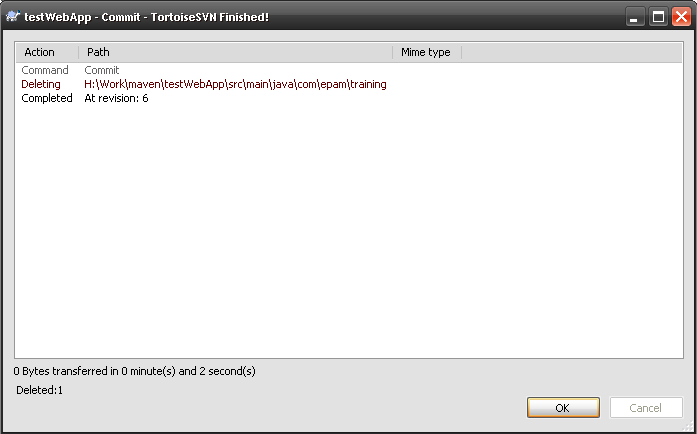


Рисунок 10.4.4 Результат операции «commit».

Задание 17

Скопируйте веб-проект в хранилище, созданное ранее. Затем скачайте (checkout) изменения в отдельный каталог. Попробуйте изменить что-либо в вашем проекте ипосмотрите на статус папок в проводнике Windows.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 10.5 Работа с репозиторием напрямую на сервере

Иногда требуется работать с svn-репозиторием напрямую на сервере не скачивая его на локальный компьютер. Для этих целей существует утилита «SVNRepo-browser», также входящая в состав TortoiseSVN.

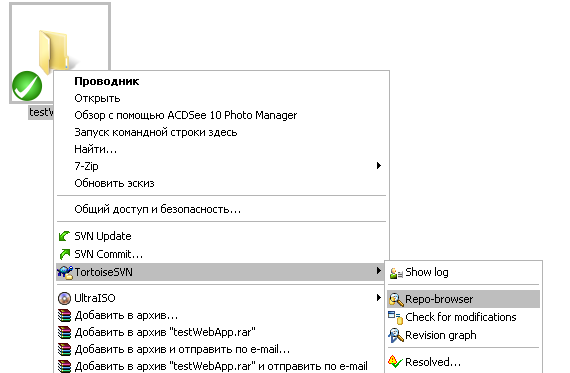


Рисунок 10.5.1 Операция «Repo-browser»

1. Запускаем проводник Windows, заходим в любую папку, нажимаем правой кнопкой по свободному месту и выбираем «SVNRepo-browser».
2. В открывшемся окне «URL» вводим полныйURL нужного нам svn-репозитория и нажимаем «OK».

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

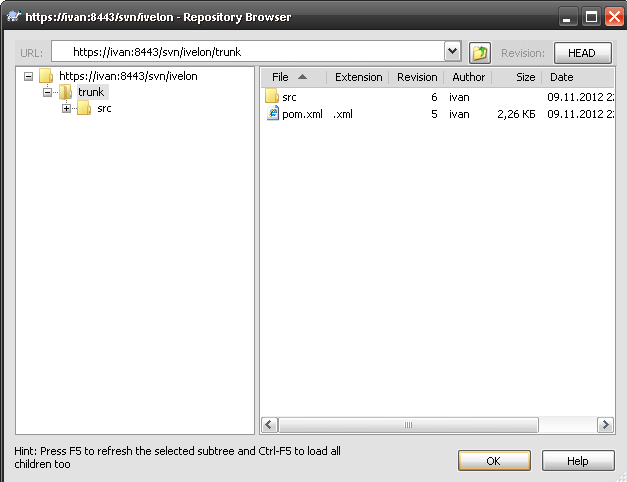


Рисунок 10.5.2 Просмотр хранилища на сервере.

Работать в этом окне можно так же, как в Проводнике Windows. Разрешено перетаскивание объектов (DragandDrop). Через контекстное меню файла/папки можно:

1. удалять «Delete…», переименовывать «Rename» файлы и каталоги (папки);
2. добавить файл («Addfile…») или папку («Addfolder…») в репозиторий;
3. создать пустую папку («Createfolder») в репозитории;
4. скачать («Checkout…») репозиторий;
5. извлечь файл или папку из репозитория без служебных каталогов («Export…»)

Извлечённая папка отличается от рабочей копии тем, что её нельзя обновлять и она не содержит служебных скрытых каталогов .svn;

1. создать ветку или тег («Copyto…»);
2. изменять свойства файла или папки («Showproperties»).

Любое изменение требует ввести текст комментария (разрешается не вводить и оставить поле пустым) и увеличивает номер ревизии на единицу. Изменения применяются немедленно.

1. Нажмите кнопку «OK» для выхода из RepositoryBrowser.

Модуль 10.6 Откат изменений в рабочей копии

Если работая с локальным репозиторием Вы совершили какую-либо ошибку, например, повредили какой-то файл, неправильно отредактировали и т.д., то это легко исправить:

1. Заходим в папку, в которой находятся файлы скачанной рабочей копии в Проводнике Windows.
2. Нажимаем правой кнопкой по свободному месту и выбираем «TortoiseSVN» — «Revert... »

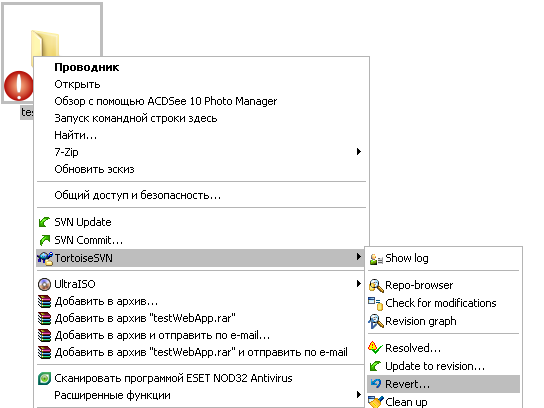


Рисунок 10.6.1 Команда «Revert».

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

1. В открывшемся окне Вы увидите список всех изменённых файлов Вашей рабочей копии. Установите флажок у того файла, который Вы хотите откатить до версии из репозитория, либо выберите сразу все файлы («Select / deselectall»). Двойной щелчок по файлу покажет его отличия от оригинального файла.

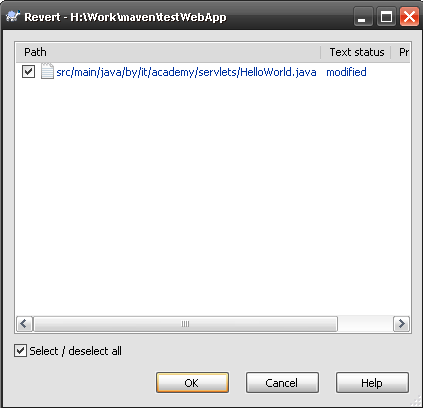


Рисунок 10.6.2 Откат изменений.

1. Нажмите «OK«. Выбранные файлы будут восстановлены, а изменённые будут отправлены в Корзину Windows.

Задание 18

Внесите изменения в ваш веб проект, например, измените текст на jsp. И затем выполните откат изменений и проверьте статус выполнения операции.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 10.7 Работа с утилитой LogViewer

Утилита LogViewer, запускаемая после нажатия на кнопку «Showlog…», либо при нажатии правой кнопкой по свободному месту в локальной рабочей копии — «TortoiseSVN» — «Showlog». По умолчанию утилита показывает последние 50 ревизий и их описаний (если во время коммита они были введены). Все основные действия выполняются из контекстного меню ревизии:

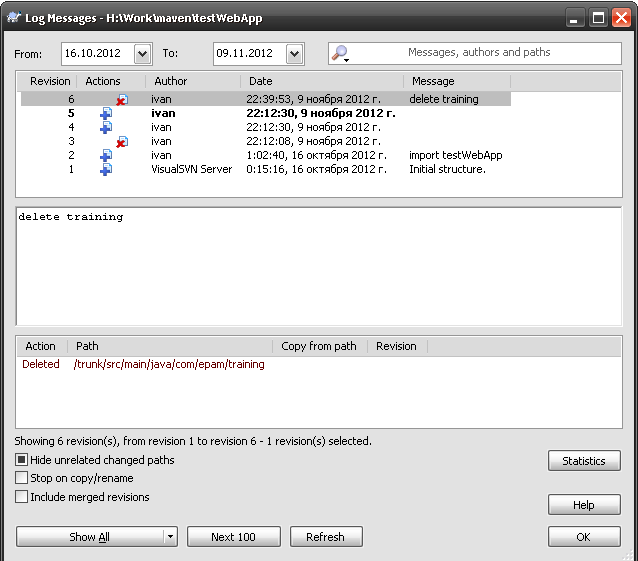
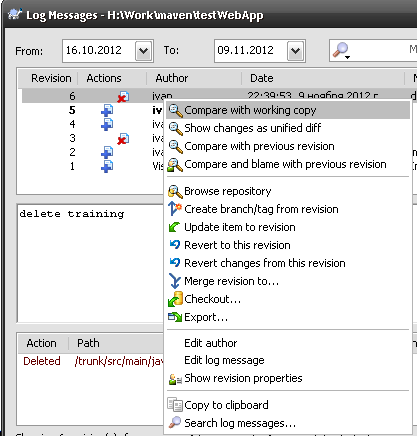


Рисунок 10.6.2 Log Viewer

1. «Comparewithwokingcopy» — сравнитьтекущую(выделенную) ревизию с локальной рабочей копией. Список изменений будет показан в TortoiseMerge.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |



1. «Showchangesasunifieddiff» — показать изменения выделенных ревизий как унифицированный патч-файл (unifieddiff). Если выделена только одна ревизия, то она будет сравниваться с предыдущей.
2. «Comparewithpreviousrevision» — показать изменения выделенной и предыдующей ревизии. Будет отображено окно с различающимися файлами, двойной щелчок по файлу откроет его в TortoiseMerge для просмотра изменений.
3. «Compareandblamewithpreviousrevision» — то же, что и предыдущее, но здесь отображается кто и когда внёс какое-либо изменение в файл построчно.
4. «Browserepository» — открывает описанный выше RepositoryBrowser для прямой работы с репозиторием.
5. «Createbranch/tagfromrevision» — позволяет создать ветку или тег из выделенной ревизии (см. выше).
6. «Updateitemtorevision» — обновляет до выделенной ревизии.
7. «Reverttothisrevision» — позволяет откатиться до выделенной ревизии.
8. «Revertchangesfromthisrevision» — позволяет откатить все изменения, сделанные в выделенной ревизии.
9. «Mergerevisionsto…» — позволяет объединить несколько ревизий в рабочую копию.
10. «Checkout…» — скачать выделенную ревизию.
11. «Export…» — извлечь выделенную ревизию без служебных каталогов.
12. «Editauthor» — редактировать имя автора выделенной ревизии (нужны права администратора).
13. «Editlogmessage» — позволяет редактировать текст описания выделенной ревизии (нужны права администратора).
14. «Showrevisionproperties» — показать свойства выделенной ревизии.
15. «Copytoclipboard» — скопировать номер выделенной ревизии, автора, описание и список изменённых файлов в буфер обмена Windows.
16. «Searchlogmessages» — запустить поиск в описаниях ревизий.

Кнопка «Statistics» отображает различную статистическую информацию: наиболее активный и наименее активный автор, количество изменений в репозитории в неделю, общее количество изменений и т.д.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Тема 11.**Настройка сУБД oRACLE

Модуль 11.1 Установка сервера баз данных Oracle 11 XE

Версия XE может использоваться бесплатно в коммерческих организациях. (Ограничения: работа с 1 ядром, 1 GB оперативной памяти и данных не более 4 GB для 10 версии и 11 GB для 11 версии).

Для установки OracleXE 11 вам необходимо сделать следующее:

1. Скачать дистрибутивOracleXE112\_Win32.zip c сайта производителя <http://www.oracle.com>
2. Распаковать zipфайл в какую-либо папку

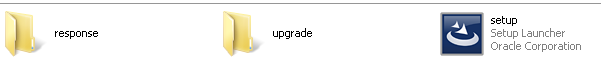


Рисунок 11.1.0 Распакованный архив OracleXE112\_Win32.zip

1. Запустить setup.exe

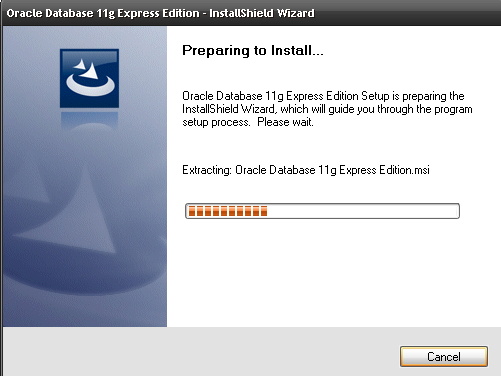


Рисунок 11.1.1 Начало установки.

|  |
| --- |
|  |
|  |
|  |
|  |

1. Подтвердить условия лицензии.

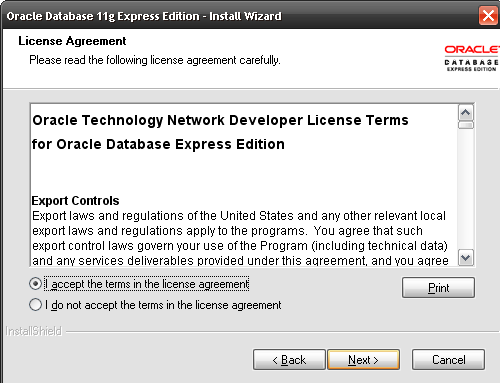


Рисунок 11.1.2 Подтверждение условия лицензии.

1. Выбрать путь установки.

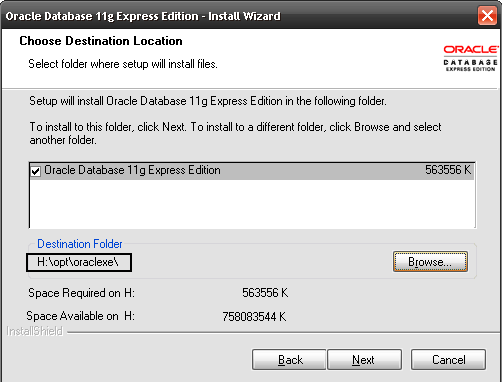


Рисунок 11.1.2Выбор пути установки.

* 1. Задать пароль для SYSTEMиSYSадминистраторов сервера.

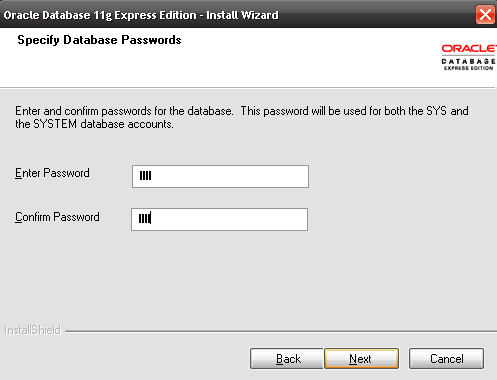


Рисунок 11.1.3Задание пароля для SYSTEMиSYS

1. Откроется окно с описанием выбранных параметров установки

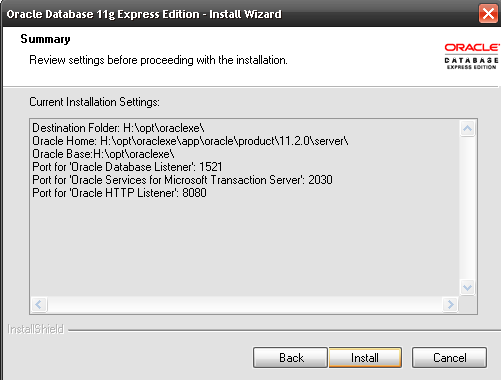


Рисунок 11.1.4Окно с описанием выбранных параметров установки

1. После завершения установки вы увидите в меню пуск папку OracleDatabase 11gExpressEdition

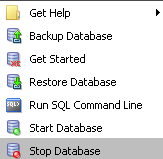


Рисунок 11.1.5Команды запуска и остановки сервера.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Модуль 11.2 Соединение с Oracle 11 XE

Для наблюдения за статусом базы данных можно использовать DatabaseHomePage.

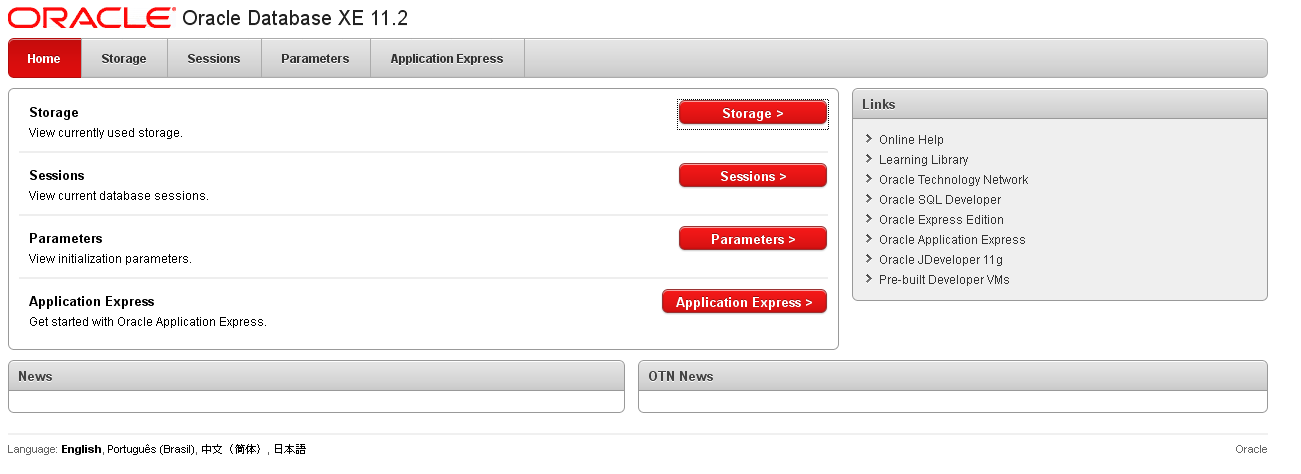


Рисунок 11.2.1Домашняя страница OracleXE.

Для того, чтобы получить детальную информацию об интересующей вас категории (например,Storage), нажмите на ссылку или на элемент табуляции. Если вы будете перенаправлены на страницу логина, где вы можете ввести ваши имя пользователя и пароль (SYSTEM/<ваш пароль)>.

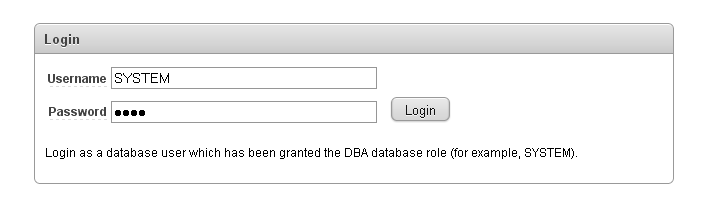


Рисунок 11.2.2Страница логина.

Для создания пользователя на сервере OracleXE существует несколько способов, самым простым из которых является использование SQLкомандной строки.

Для того чтобы использовать SQLкомандную строку, необходимо открыть ПУСК🡪Все программы 🡪 Oracle Database 11g Express Edition 🡪 Run SQL Command Line.

|  |
| --- |
|  |
|  |
|  |

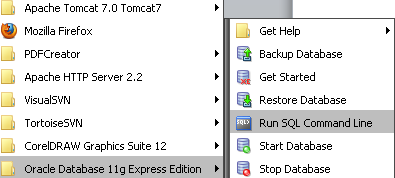


Рисунок 11.2.3Вызов SQLкомандной строки

Для подключения к серверу, вам необходимо в SQL командной строке вызвать команду Connect SYSTEM/<ваш пароль>.

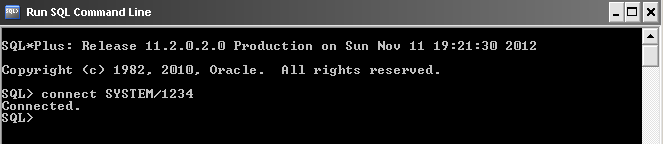


Рисунок 11.2.4Соединение с SQL-сервером.

После того, как вы подключились к системе, настало время создания пользователя. Для этого необходимо выполнить следующую команду:

create user <user> identified by <password-for-user>;

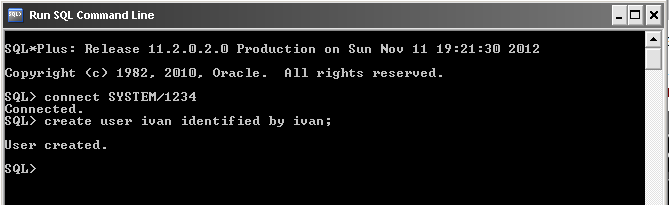


Рисунок 11.2.5Создание пользователя

Далее необходимо добавить нужные привилегии созданному пользователю. При общем подходе имеется три базовых привилегии системы:

1. CONNECT(Подключить)
2. RESOURCE (Ресурс)
3. DBA(Администратор Базы Данных)

|  |
| --- |
|  |

Проще, можно сказать, что CONNECT состоит из права зарегистрироваться и права создавать представления и синонимы. RESOURCE состоит из права создавать базовые таблицы. DBA - это привилегия супер пользователя, дающая пользователю высокие полномочии к базе данных.

Для задания вышеперечисленных привилегий мы будем использовать следующую команду:

grant connect,resource,dba to<user>;



Рисунок 11.2.6Добавление привилегий.

Вторым, более распространённым способом настройки соединения и использования сервера баз данных OracleXE 11g, является OracleSqlDeveloper.

Давайте разберемся как же его установить и настроить на локальном компьютере. Для того, что бы установить sqldeveloper, необходимо его скачать с сайта производителя:

http://www.oracle.com/technetwork/developer-tools/sql-eveloper/downloads/index.html



Рисунок 11.2.7Сайт для скачивания дистрибутива sqldeveloper.

|  |
| --- |
|  |

После завершения скачивания, необходимо распаковать дистрибутив в любую удобную для вас директорию и вызвать sqldeveloper.exe.

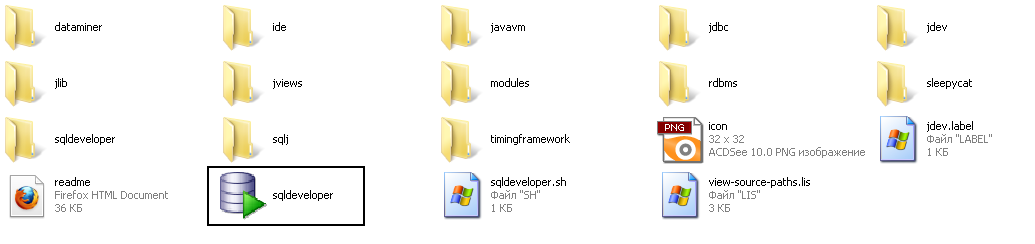


Рисунок 11.2.8Директория с утановленнымsqldevelopером.

Если вас просят ввести полный путь к java.exe, нажмите кнопку Обзор и найдите java.exe. Например, в системе Windows путь может быть похож на C: \ Program Files \ Java \ jdk1.6.0\_21 \ bin \ java.exe.

После этого откроется главное окно sqldeveloper.

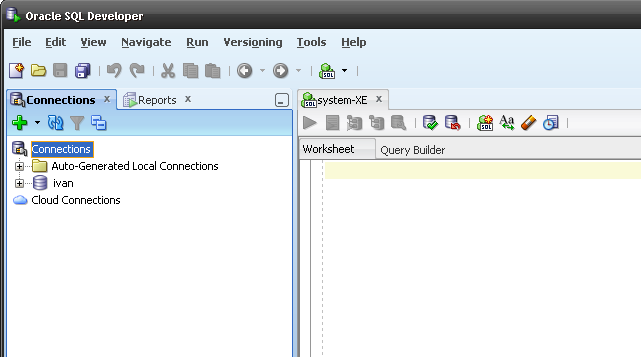


Рисунок 11.2.9Главное окно sqldeveloper.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |

Следующий этап – это создание соединения с сервером баз данных OracleXE. Для начала, давайте, рассмотрим вариант автоматической генерации локальных соединений. Для этого необходимо в левой части окна нажать правой кнопкой мыши на «Сonnections» и выбрать пункт меню «Create Local Connections».

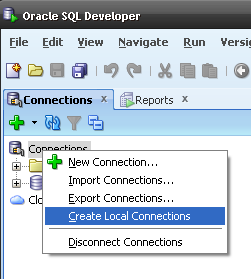


Рисунок 11.2.10Создание локальных соединений

После этого вы можете зайти в директорию Auto-Generated Local Connections и выбрать нужное вам соединение.

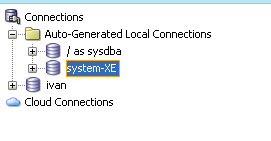


Рисунок 11.2.11Выбор соединения

Существует очень частая проблема, когда пользователь, у которого sql developer запускается на русскоязычной операционной системе Windows пытается соединиться с базой (например с system-XE), то появляется ошибка «Can not access NLS data files or invalid environment specified..», т.е. существует проблема с кодировками между русскоязычной windows и англоязычной установкой sqldeveloper. Для решения данной проблемы, зайдите в **sqldeveloper\sqldeveloper\bin\**, а там найдите файл **sqldeveloper.conf**. В этот файл добавьте следующую строку:

AddVMOption -Duser.region=US

|  |
| --- |
|  |
|  |

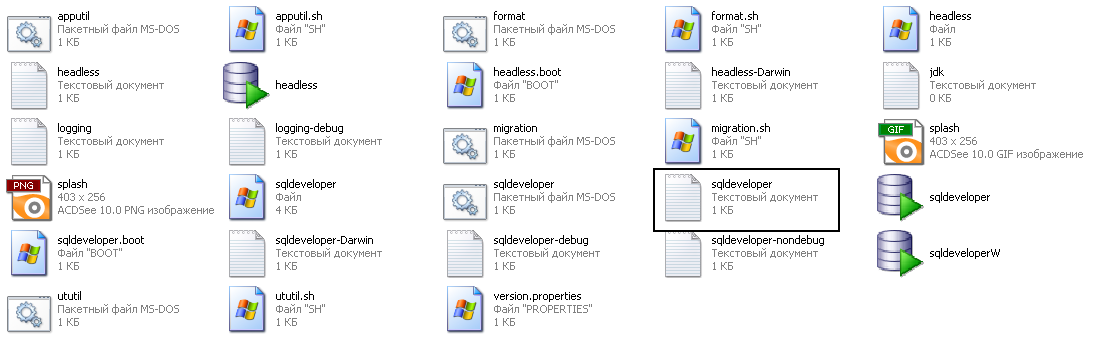


Рисунок 11.2.12Файлsqldeveloper.conf

Пример файла sqldeveloper.conf может быть следуюущим:

IncludeConfFile ../../ide/bin/ide.conf

SetJavaHome H:\opt\Java\jdk1.7.0\_03

AddVMOption-Doracle.ide.util.AddinPolicyUtils.OVERRIDE\_FLAG=true

AddVMOption -Dsun.java2d.ddoffscreen=false

AddVMOption -Dwindows.shell.font.languages=

AddVMOption-XX:MaxPermSize=256M

AddVMOption -Doracle.jdbc.mapDateToTimestamp=false

**AddVMOption -Duser.region=US**

IncludeConfFilesqldeveloper-nondebug.conf

После этого перезагрузите sqldeveloper и проверьте, что соединение с сервером успешно.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

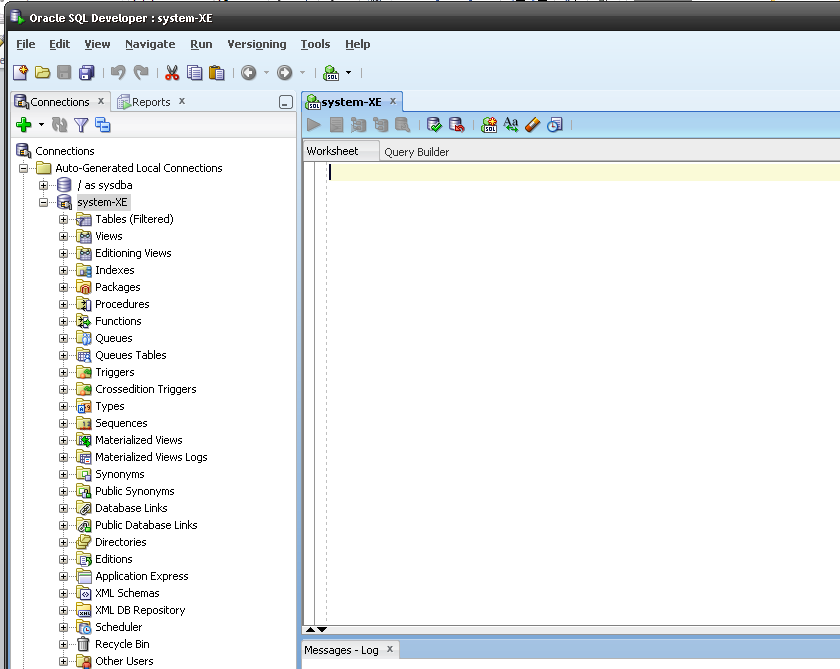


Рисунок 11.2.13Успешное соединение с system-XE.

А как соединиться со схемой, созданной с помощью SQL командной строкой?

В этом нам поможет ручное создание соединения. Для того, чтобы это сделать, вам, необходимо нажать правой кнопкой на «Connections» и выбрать пункт меню «New Connection».

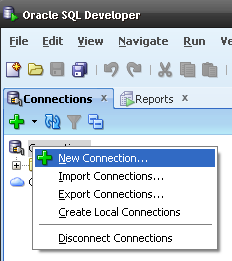


Рисунок 11.2.14Выбор ручного способа создания нового соединения.

|  |
| --- |
|  |

Далее откроется окно создания нового соединения.

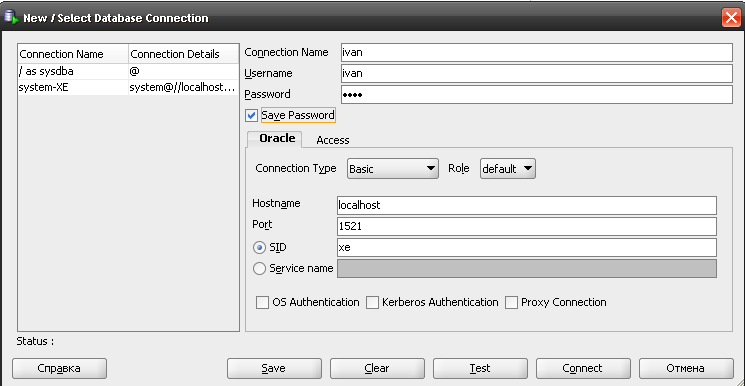


Рисунок 11.2.15Окно создания нового соединения.

В данном окне необходимо ввести следующие параметры:

1. *Connection Name -* Псевдоним для подключения к базе данных, используя информацию, которую вы вводите. Пример: ivan
2. *Username -* Имя пользователя базы данных. Этот пользователь должен иметь достаточно прав для выполнения задач, которые необходимо выполнить при подключении к базе данных. Пример: ivan
3. *Password -* Пароль, связанный с указанным пользователем базы данных
4. *Save Password -* Укажите эту опцию, если вы хотите, чтобы пароль сохранялся в информации о подключении. Вам не будет предложено вводить пароль при последующей попытке подключиться с помощью этого соединения
5. *Hostname –* Хост системы для базы данных Oracle. Для вашей базы данных Express Edition, имя хоста–localhost
6. *Port –* Порт на котором работает ваш сервер. Порт по умолчанию 1521 для OracleСУБД.
7. *SID –* Имя базы данных. Значение по умолчанию для ExpressEdition–xe.

|  |
| --- |
|  |
|  |
|  |

Для проверки соединения вы можете нажать на кнопку «Test».

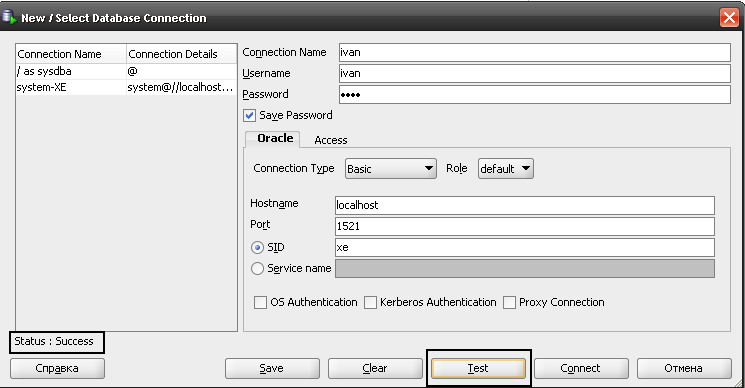


Рисунок 11.2.16Проверка соединения

Далее вам остается только нажать на кнопку «Save». После чего вы увидите ваше соединение в списке ваших соединений.



Рисунок 11.2.17Ваше соединение в списке соединений

После того, как вы подключились к базе данных, далее начинается реальное использование возможностей oracle и sqldeveloper или других клиентов СУБД Oracle. Более глубокое использования их является материалом, выходящим за границы данного курса, поэтому я предлагаю вам ознакомится с ними самостоятельно.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Задание 19.

Установить и настроить oraclexe 11g и sqldeveloper на локальный компьютер. Создать нового пользователя. Создать несколько связанных таблиц для данной схемы данных. Ваше web-приложение должно уметь работать с таблицами базы данных: вносить, удалять, выбирать и обновлять записи. Ваше веб приложение должно выкачиваться из subversion хранилища, собираться и устанавливаться на сервер с помощью maven. А после запуска сервера, пользователь должен иметь возможность проверить работоспособность приложения с помощью браузера.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |