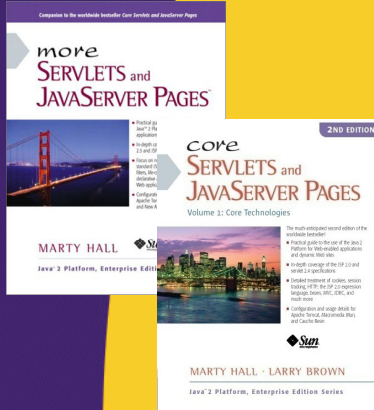




Spring Integration Part 1

Originals of Slides and Source Code for Examples:
<http://courses.coreservlets.com/Course-Materials/spring.html>

Customized Java EE Training: <http://courses.coreservlets.com/>
Servlets, JSP, Struts, JSF/MyFaces/Facelets, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.



**For live Spring & Hibernate training, see
courses at <http://courses.coreservlets.com/>.**



**Taught by the experts that brought you this tutorial.
Available at public venues, or customized versions
can be held on-site at your organization.**

- Courses developed and taught by Marty Hall
 - Java 5, Java 6, intermediate/beginning servlets/JSP, advanced servlets/JSP, Struts, JSF, Ajax, GWT, custom mix of topics
- Courses developed and taught by coreservlets.com experts (edited by Marty)
 - Spring, Hibernate/JPA, EJB3, Ruby/Rails

Contact hall@coreservlets.com for details

Topics in This Section

- **Maven 2 build system overview**
- **General build procedure**
- **Multi-module project build process**
- **Spring IoC integration**
- **Testing with Maven 2 and Spring**

4

Java EE training: <http://courses.coreservlets.com>

© 2008 coreservlets.com



Build System

Customized Java EE Training: <http://courses.coreservlets.com/>
Servlets, JSP, Struts, JSF/MyFaces/Facelets, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Conventions

- **Variables** `${KEY}`
- **Home directory** `${HOME}`
 - Linux: `$HOME` or `~`
 - Windows `%HOMEPATH%`
- **Path separator** `/`
 - Linux - `“/”`
 - Windows - `“\”`

6

Java EE training: <http://courses.coreservlets.com>

General Goals

- **Manage source and configuration files**
- **Distribute and execute in a test environment**
- **Integrate third-party libraries**
- **Package and distribute project**

7

Java EE training: <http://courses.coreservlets.com>

Maven 2 Build System

- **Implements best practices into build defaults**
- **Comprehensive build system**
 - Compile
 - Test
 - Package
 - Distribute
 - Version
 - IDE integration
 - Report generation
- **Customizable behavior**

8

Java EE training: <http://courses.coreservlets.com>

Maven 2 Architecture

- **Maven client**
 - Plugin execution framework
- **Project**
 - Source tree
 - Maven configuration
- **Repository**
 - Addressable (URL) file repository
- **Local repository cache**
 - Plugin downloads
 - Direct and transitive dependencies
 - Defaults to `${HOME}/.m2/repository`
 - Edit settings.xml for alternatives
 - `<settings><localRepository/></settings>`

9

Java EE training: <http://courses.coreservlets.com>

Client Personalization

- **Global**
 - Maven 2 installation root
 - `conf/settings.xml`
- **User**
 - `${HOME}/.m2/settings.xml`

10

Java EE training: <http://courses.coreservlets.com>

Maven 2 Installation Process

- **Verify prerequisites**
 - JDK 1.4+
 - `JAVA_HOME` set to JDK installation root
- **Download maven**
 - <http://maven.apache.org/download.html> (< 2mb)
 - Maven 2.0.9 (zip/tar.gz/tar.bz2)
- **Unpack download**
 - Extract to target installation path
- **Set Maven 2 home**
 - Set variable `M2_HOME` to the installation path
 - Contains bin, boot, conf, and lib directories
- **Add `M2_HOME` bin directory to execution path**
 - Add `${M2_HOME}/bin` to `PATH` environment setting
- **Verify repository settings**
 - Official, mirror, or corporate

11

Java EE training: <http://courses.coreservlets.com>

Verify Repository Settings

- **Default**
 - Official repository; a.k.a. central
 - <http://repo1.maven.org/maven2>
- **Mirror**
 - Traditional mirror
 - Redundant site
 - Corporate repository
 - Private network
 - Without internet access

12

Java EE training: <http://courses.coreservlets.com>

Repository Mirror

```
<settings>
  <mirrors>
    <mirror>
      <id>mirror</id>
      <name>mirror of central</name>
      <url>http://mirror/maven2</url>
      <mirrorOf>central</mirrorOf>
    </mirror>
  </mirrors>
</settings>
```

13

Java EE training: <http://courses.coreservlets.com>



Build Process

Customized Java EE Training: <http://courses.coreservlets.com/>
Servlets, JSP, Struts, JSF/MyFaces/Facelets, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

General Approach

- **Compile source**
 - Reference third-party libraries
 - Reference peer modules
- **Integrate project with IDE**
- **Test binaries**
- **Package source for distribution**

Process

- **Create project structure**
 - `project/src/main/java`
 - `project/src/main/resources`
 - `project/src/test/java`
 - `project/src/test/resources`
- **Create project /pom.xml**
 - Register Maven 2 NS
 - Define project meta-data
 - e.g. project identifiers
 - Plugin settings
 - Specify dependencies
 - Third-party libraries
 - Local projects

16

Java EE training: <http://courses.coreservlets.com>

Process Continued

- **Integrate with IDE**
 - Prepare for eclipse `mvn eclipse:eclipse`
 - Plugins available for NetBeans and IntelliJ
 - Refresh or import project into IDE
- **Develop source**
 - `src/main/java`
 - e.g., `coreservlets.MockCustomerDao`
- **Perform build activities**
 - Compile, test, package, generate reports, and more...
 - e.g., `mvn clean install`

17

Java EE training: <http://courses.coreservlets.com>

Create Project Structure

```
/coreservlets-api
  /src
    /main
      /java
      /resources
    /test
      /java
      /resources
```

Create pom.xml

```
/coreservlets-api
  pom.xml
  /src
    /main
      /java
      /resources
    /test
      /java
      /resources
```

Create pom.xml

- Register Maven 2 schema and namespace mapping

```
<?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/maven-v4_0_0.xsd">

</project>
```

20

Java EE training: <http://courses.coreservlets.com>

Create pom.xml

- Define project identifiers
 - Referenceable identifier from other projects

```
<?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/maven-v4_0_0.xsd">

  <modelVersion>4.0.0</modelVersion>
  <groupId>coreservlets</groupId>
  <artifactId>coreservlets-api</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>
  <name>coreservlets-api</name>
</project>
```

21

Java EE training: <http://courses.coreservlets.com>

Create pom.xml Continued

- Define compiler settings

```
<project>
...
  <build>
    <plugins><plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>2.0</version>
      <configuration>
        <source>1.5</source>
        <target>1.5</target>
      </configuration>
    </plugin></plugins>
  </build>
</project>
```

22

Java EE training: <http://courses.coreservlets.com>

Create pom.xml Continued

- Optional plugin customizations

```
<project>
...
  <build>
    <plugins><plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-surefire-plugin</artifactId>
      <configuration>
        <useFile>>false</useFile>
        <trimStackTrace>>true</trimStackTrace>
        <printSummary>>true</printSummary>
        <disableXmlReport>>true</disableXmlReport>
      </configuration>
    </plugin></plugins>
  </build>
</project>
```

23

Java EE training: <http://courses.coreservlets.com>

Create pom.xml Continued

- **Specify dependencies**

```
<dependencies>
  <dependency>
    <groupId>log4j</groupId>
    <artifactId>log4j</artifactId>
    <version>1.2.14</version>
  </dependency>
</dependencies>
```

24

Java EE training: <http://courses.coreservlets.com>

Integrate with Eclipse IDE

- **Prepare project**

- `mvn eclipse:eclipse`
 - Clean configuration with `mvn eclipse:clean`

- **Prepare Eclipse IDE**

- From the Eclipse root window menu bar
 - Open **Window / Preferences**
- From **Preferences**
 - Expand **Java / Build Path / Classpath Variables / New**
- From **New Variable Entry**
 - Add **M2_REPO** mapped to local repository path
 - e.g., `${HOME}/.m2/repository`

25

Java EE training: <http://courses.coreservlets.com>

Integrate with Eclipse IDE

- **Import projects into Eclipse workspace**
 - From the Eclipse root window menu bar
 - Open File / Import
 - From the Import wizard
 - Select / General / Existing Projects into Workspace / Next
 - Import Projects / Select root directory / Browse
 - From **Browse for Folder**
 - Drill-down into project directory
 - Select **OK**
 - Verify project is present and checked
 - Select Finish

26

Java EE training: <http://courses.coreservlets.com>

Develop Source

```
/coreservlets-api
pom.xml
/src
  /main
    /java
      /coreservlets
        Customer.java
        CustomerDao.java
    /resources
  /test
    /java
    /resources
```

27

Java EE training: <http://courses.coreservlets.com>

Perform Build Activities

- **General build tasks**
 - `mvn clean`
 - `mvn compile`
 - `mvn test`
 - `mvn package`
 - `mvn install`
- **Optionally, specify multiple build tasks**
 - `mvn clean install`

Build Artifacts

- **Project `coreservlets-api`**
 - JAR
 - `coreservlets-api-1.0-SNAPSHOT.jar`



Multi-Module Projects

Customized Java EE Training: <http://courses.coreservlets.com/>
Servlets, JSP, Struts, JSF/MyFaces/Facelets, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Multi-Module Project Structure

- **Project one**
 - `pom.xml`
 - `src/main/java`
- **Project two**
 - `pom.xml`
 - `src/main/java`
- **Project N**
 - `pom.xml`
 - `src/main/java`
- `pom.xml`

Configuration

- **Parent pom.xml**
 - Module list
 - a.k.a. child projects
 - `<modules><module/></modules>`
 - Parent settings
 - Plugin settings
 - Third-party library versions
 - Global dependencies
- **Child pom.xml**
 - Parent reference
 - `<parent/>`

32

Java EE training: <http://courses.coreservlets.com>

Parent pom.xml

- **Parent project identifier and type**

```
<?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/maven-v4_0_0.xsd">

  <modelVersion>4.0.0</modelVersion>
  <groupId>coreservlets</groupId>
  <artifactId>coreservlets-pom</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>pom</packaging>
  <name>coreservlets-pom</name>
</project>
```

33

Java EE training: <http://courses.coreservlets.com>

Parent pom.xml

- **Module list**

```
<?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/maven-v4_0_0.xsd">
  ...
  <modules>
    <module>coreservlets-api</module>
  </modules>
</project>
```

34

Java EE training: <http://courses.coreservlets.com>

Parent pom.xml

- **Plugin settings**

- Inherited by child projects

```
<project>
  ...
  <build>
    <plugins>
      ...
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>2.0</version>
        <configuration>
          <source>1.5</source>
          <target>1.5</target>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>
```

35

Java EE training: <http://courses.coreservlets.com>

Parent pom.xml

- Third-party library versions

```
<project>
  ...
  <dependencyManagement>
    <dependencies>
      <dependency>
        <groupId>log4j</groupId>
        <artifactId>log4j</artifactId>
        <version>1.2.4</version>
      </dependency>
    </dependencies>
  </dependencyManagement>
</project>
```

Parent pom.xml

- Third-party library versions

```
<project>
  ...
  <dependencies>
    <dependency>
      <groupId>log4j</groupId>
      <artifactId>log4j</artifactId>
    </dependency>
  </dependencies>
</project>
```

Child project pom.xml

```
<?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/maven-v4_0_0.xsd">

  <modelVersion>4.0.0</modelVersion>
  <groupId>coreservlets</groupId>
  <artifactId>coreservlets-api</artifactId>
  <packaging>jar</packaging>
  <name>coreservlets-api</name>
  <version>1.0-SNAPSHOT</version>
  <parent>
    <groupId>coreservlets</groupId>
    <artifactId>coreservlets-pom</artifactId>
    <version>1.0-SNAPSHOT</version>
  </parent>
</project>
```

38

Java EE training: <http://courses.coreservlets.com>

Build Artifacts

- **Project coreservlets-pom**
 - POM
 - coreservlets-pom-1.0-SNAPSHOT.pom
- **Project coreservlets-api**
 - JAR
 - coreservlets-api-1.0-SNAPSHOT.jar

39

Java EE training: <http://courses.coreservlets.com>



Implementation Project

Customized Java EE Training: <http://courses.coreservlets.com/>
Servlets, JSP, Struts, JSF/MyFaces/Facelets, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

API Implementation Project

- **Parent configuration**
 - `pom.xml`
- **Project coreservlets-api**
- **Project coreservlets-mock**

Process

- **Create project structure for coreservlets- mock**
 - `src/main/java`
 - `src/main/resources`
 - `src/test/java`
 - `src/test/resources`
- **Create pom.xml**
- **Specify dependencies**
 - Third-party libraries
 - e.g., log4j
 - Local projects
 - e.g., coreservlets-api

42

Java EE training: <http://courses.coreservlets.com>

Process Continued

- **Integrate with IDE**
 - `mvn eclipse:eclipse`
 - Import or refresh project
- **Develop source**
 - `src/main/java`
 - `coreservlets.MockCustomerDao`
- **Perform build activities**
 - e.g., `mvn clean install`

43

Java EE training: <http://courses.coreservlets.com>

Create pom.xml

- Project identifiers
- Parent project reference

```
<project>
  <modelVersion>4.0.0</modelVersion>
  <groupId>coreservlets</groupId>
  <artifactId>coreservlets-mock</artifactId>
  <packaging>jar</packaging>
  <name>coreservlets-mock</name>
  <version>1.0-SNAPSHOT</version>

  <parent>
    <groupId>coreservlets</groupId>
    <artifactId>coreservlets-pom</artifactId>
    <version>1.0-SNAPSHOT</version>
  </parent>
</project>
```

44

Java EE training: <http://courses.coreservlets.com>

Create pom.xml

- Local project dependency

```
<project>
  ...
  <dependencies>
    <dependency>
      <groupId>${project.groupId}</groupId>
      <artifactId>coreservlets-api</artifactId>
      <version>${project.version}</version>
    </dependency>
  </dependencies></project>
</project>
```

45

Java EE training: <http://courses.coreservlets.com>

Complete Process

- **Integrate with IDE**
 - `mvn eclipse:eclipse`
 - Import or refresh project
- **Develop source**
 - `src/main/java`
 - `coreservlets.MockCustomerDao`
- **Perform build activities**
 - e.g., `mvn clean install`

46

Java EE training: <http://courses.coreservlets.com>

Build Artifacts

- **Project `coreservlets-pom`**
 - POM
 - `pom.xml`
- **Project `coreservlets-api`**
 - JAR
 - `coreservlets-api-1.0-SNAPSHOT.jar`
- **Project `coreservlets-mock`**
 - JAR
 - **`coreservlets-mock-1.0-SNAPSHOT.jar`**

47

Java EE training: <http://courses.coreservlets.com>



Deployment

Customized Java EE Training: <http://courses.coreservlets.com/>
Servlets, JSP, Struts, JSF/MyFaces/Facelets, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Deployment Project

- **Parent configuration**
 - `pom.xml`
- **Project coreservlets-api**
- **Project coreservlets-mock**
- **Project coreservlets-main**

Process

- **Create project structure**
 - `src/main/java`
 - `src/main/resources`
 - `src/test/java`
 - `src/test/resources`
- **Create `pom.xml`**
 - Register Maven 2 NS
 - Define project meta-data; e.g. project identifiers
 - Plugin settings
 - `maven-jar-plugin`
 - `maven-assembly-plugin`
 - Specify dependencies
 - Third-party libraries
 - e.g., `spring-core`, `spring-beans`, `spring-context`
 - Local projects
 - e.g., `coreservlets-api`, `coreservlets-mock`

50

Java EE training: <http://courses.coreservlets.com>

Process Continued

- **Integrate with IDE**
 - Prepare for eclipse `mvn eclipse:eclipse`
 - Plugins available for NetBeans and IntelliJ
 - Refresh or import project into IDE
- **Develop source**
 - `src/main/java`
 - e.g., `coreservlets.MockCustomerDao`
- **Perform build activities**
 - Compile, test, package, generate reports, and more...
 - e.g., `mvn clean install`

51

Java EE training: <http://courses.coreservlets.com>

Create pom.xml

```
<?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/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>coreservlets</groupId>
  <artifactId>coreservlets-main</artifactId>
  <packaging>jar</packaging>
  <name>coreservlets-main</name>
  <version>1.0-SNAPSHOT</version>

  <parent>
    <groupId>coreservlets</groupId>
    <artifactId>coreservlets-pom</artifactId>
    <version>1.0-SNAPSHOT</version>
  </parent>
</project>
```

52

Java EE training: <http://courses.coreservlets.com>

Create pom.xml

- Project identifiers
- Parent project reference

```
<project>
  <modelVersion>4.0.0</modelVersion>
  <groupId>coreservlets</groupId>
  <artifactId>coreservlets-main</artifactId>
  <packaging>jar</packaging>
  <name>coreservlets-main</name>
  <version>1.0-SNAPSHOT</version>

  <parent>
    <groupId>coreservlets</groupId>
    <artifactId>coreservlets-pom</artifactId>
    <version>1.0-SNAPSHOT</version>
  </parent>
</project>
```

53

Java EE training: <http://courses.coreservlets.com>

Create pom.xml

- Local project dependencies

```
<dependencies>
  <dependency>
    <groupId>${project.groupId}</groupId>
    <artifactId>coreservlets-api</artifactId>
    <version>${project.version}</version>
  </dependency>
  <dependency>
    <groupId>${project.groupId}</groupId>
    <artifactId>coreservlets-mock</artifactId>
    <version>${project.version}</version>
  </dependency>
</dependencies>
```

54

Java EE training: <http://courses.coreservlets.com>

Create pom.xml

- Third-party dependencies

```
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-beans</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-context</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-core</artifactId>
</dependency>
<dependency>
  <groupId>commons-logging</groupId>
  <artifactId>commons-logging</artifactId>
</dependency>
```

55

Java EE training: <http://courses.coreservlets.com>

Create pom.xml

- **Customize plugins**
 - maven-jar-plugin
 - Executable JAR
 - maven-assembly plugin
 - Aggregate artifact with dependencies

Complete Process

- **Integrate with IDE**
 - `mvn eclipse:eclipse`
 - Import or refresh project
- **Develop source**
 - `src/main/java`
 - `coreservlets.Main`
- **Perform build activities**
 - e.g., `mvn clean install`

Build Artifacts

- **Project coreservlets-pom**
 - POM
 - pom.xml
- **Project coreservlets-api**
 - JAR
 - coreservlets-api-1.0-SNAPSHOT.jar
- **Project coreservlets-mock**
 - JAR
 - **coreservlets-mock-1.0-SNAPSHOT.jar**

Build Artifacts Continued

- **Project coreservlets-main**
 - JAR
 - coreservlets-main-1.0-SNAPSHOT.jar
 - Assembly
 - coreservlets-api-1.0-SNAPSHOT.jar
 - coreservlets-mock-1.0-SNAPSHOT.jar
 - **coreservlets-main-1.0-SNAPSHOT.jar**
(Executable JAR)
 - spring-beans-2.5.5.jar
 - spring-context-2.5.5.jar
 - spring-core-2.5.5.jar
 - aopalliance-1.0.jar
 - commons-logging-1.0.4.jar
 - log4j-1.2.11.jar



Integrate Spring IoC

Customized Java EE Training: <http://courses.coreservlets.com/>
Servlets, JSP, Struts, JSF/MyFaces/Facelets, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

General Approach

- **Project coreservlets-api**
- **Project coreservlets-mock**
 - Include Spring bean definitions with build package
 - `coreservletsContext.xml`
- **Project coreservlets-main**
 - Define Spring bean definitions reference
 - Load bean definitions at runtime
 - Reference `coreservletsContext.xml`
 - Specify Spring JAR dependencies
 - Decouple mock implementation

Project Structure

- **Project coreservlets-pom**
 - pom.xml
- **Project coreservlets-api**
 - src/main/java
 - coreservlets.Customer
 - coreservlets.CustomerDao
- **Project coreservlets-mock**
 - pom.xml
 - src/main/java
 - coreservlets.MockCustomerDao
 - src/main/resources
 - coreservletsContext.xml

62

Java EE training: <http://courses.coreservlets.com>

Project Structure Continued

- **Project coreservlets-main**
 - pom.xml
 - Spring IoC JAR references
 - src/main/java
 - coreservlets.Main

63

Java EE training: <http://courses.coreservlets.com>

Project coreservlets-mock

- Add Spring bean definitions

- src/main/resources
 - coreservletsContext.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans>
  <bean id="customerDao"
        class="coreservlets.MockCustomerDao">
    <constructor-arg>
      <list>
        <bean class="coreservlets.Customer">
          <property name="id" value="jjoe" />
          <property name="name" value="Java Joe" />
        </bean>
      </list>
    </constructor-arg>
  </bean>
</beans>
```

64

Java EE training: <http://courses.coreservlets.com>

Project coreservlets-main

- Spring IoC JAR dependencies

- coreservlets-main/pom.xml

```
<project>
  ...
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-beans</artifactId>
  </dependency><dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
  </dependency><dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-core</artifactId>
  </dependency><dependency>
    <groupId>commons-logging</groupId>
    <artifactId>commons-logging</artifactId>
  </dependency>
```

65

Java EE training: <http://courses.coreservlets.com>

Project coreservlets-main

- Driver class `coreservlets.Main`

```
public class Main {  
    public static void main(String[] args){  
        System.out.println("Searching for: " + args[0]);  
  
        BeanFactory factory = new ClassPathXmlApplicationContext(  
            "/coreservletsContext.xml");  
  
        CustomerDao service =  
            (CustomerDao) factory.getBean("CustomerDao");  
  
        Customer customer = service.getCustomerByName(args[0]);  
        System.out.println("Found: " + customer);  
    }  
}
```

Standard output

```
> java -jar coreservlets-main.jar "Java Joe"  
Searching for: Java Joe  
Found: Customer id=jjoe, name=Java Joe
```

66

© 2008 coreservlets.com



Spring Library Integration

Customized Java EE Training: <http://courses.coreservlets.com/>
Servlets, JSP, Struts, JSF/MyFaces/Facelets, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

General Approach

- **Spring IoC Container**
 - Deployment contexts
 - Test contexts
- **Spring AOP Libraries**
 - Advisor code
 - Deployment contexts
 - Test contexts
- **Spring service abstractions**
 - Implementation projects

68

Java EE training: <http://courses.coreservlets.com>

Compilation Dependencies Example

- **coreservlets-api**
- **coreservlets-mock**
 - **coreservlets-api**
- **coreservlets-jdbc**
 - **spring-jdbc, spring-tx**
- **coreservlets-jpa**
 - **spring-orm, javax-persistence**
- **coreservlets-advice**
 - **aopalliance, spring-aop**
- **coreservlets-main**
 - **coreservlets-api, spring-context, spring-beans**

69

Java EE training: <http://courses.coreservlets.com>



Testing with Maven 2 and Spring

Customized Java EE Training: <http://courses.coreservlets.com/>
Servlets, JSP, Struts, JSF/MyFaces/Facelets, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Build Artifacts

- **Project coreservlets-pom**
 - POM
 - pom.xml
- **Project coreservlets-api**
 - JAR
 - coreservlets-api-1.0-SNAPSHOT.jar
- **Project coreservlets-jdbc**
 - JAR
 - **coreservlets-jdbc-1.0-SNAPSHOT.jar**

Project Structure

- **Project coreservlets-pom**
 - pom.xml
- **Project coreservlets-api**
 - src/main/java
 - coreservlets.Customer
 - coreservlets.CustomerDao
- **Project coreservlets-jdbc**
 - pom.xml
 - src/main/java
 - coreservlets.SpringJdbcCustomerDao
 - src/main/resources
 - coreservletsContext.xml
 - src/test/java
 - ?

72

Java EE training: <http://courses.coreservlets.com>

General Approach

- **Create test class**
- **Integrate Spring IoC container**
- **Inject test fixture(s)**
- **Define test cases**

73

Java EE training: <http://courses.coreservlets.com>

Process

- **Register Spring test JAR**
 - `org.springframework:spring-test` (test scope)
- **Create test class**
 - Name class with suffix `Test`
 - Place class under `src/test/java`
- **Create test resources**
 - Place test resources under `src/test/resources`
 - e.g., `applicationContext.xml` defining platform beans such as `dataSource`
- **Integrate Spring IoC container (Spring TestContext)**
 - Class annotation
 - `@RunWith(SpringJUnit4ClassRunner.class)`
 - Class annotation
 - `@ContextConfiguration(locations={"path-0", "path-N"})`

74

Java EE training: <http://courses.coreservlets.com>

Process Continued

- **Inject test fixtures**
 - `@Autowired` fields or methods
 - By type
 - Optional qualifier
 - `@Resource` fields or methods
 - By bean name
- **Define lifecycle methods (optional)**
 - `@BeforeClass`, `@Before`, `@After`, `@AfterClass`
- **Define test cases**
 - `@Test` methods
- **Execute tests**
 - `mvn test`
 - Or, a goal with a test goal dependency; e.g., `mvn install`

75

Java EE training: <http://courses.coreservlets.com>

Create Test Resources

- **src/test/resources**

- **coreservletsTestContext.xml**

```
<beans>
  <bean id="dataSource"
        class="coreservlets.EmbeddedDerbyDataSource">
    <constructor-arg value="target/ngcdb" />
    <constructor-arg>
      <list>
        <value>/setup.sql</value>
      </list>
    </constructor-arg>
  </bean>
</beans>
```

76

Java EE training: <http://courses.coreservlets.com>

Create Test Resources Continued

- **src/test/resources**

- **setup.sql**

```
create table customer (
  id varchar(36) not null,
  name varchar(32) not null,
  primary key (id),
  unique (name)
);
delete from customer;
insert into customer (id, name) values ('jjoe','Java
  Joe');
insert into customer (id, name) values ('jjohn','Java
  John');
commit;
```

77

Java EE training: <http://courses.coreservlets.com>

Create Test Class

- **src/test/java**
 - **coreservlets/CustomerDaoTransactionalTest**

```
package coreservlets;  
  
public class CustomerDaoTransactionalTest {  
  
}
```

Create Test Class Continued

- **Reference bean definition paths**
 - **src/test/resources**
 - **coreservletsContext.xml**
 - **coreservletsTestContext.xml**

```
@RunWith(SpringJUnit4ClassRunner.class)  
@ContextConfiguration(  
    locations={"/coreservletsContext.xml",  
              "/coreservletsTestContext.xml"})  
public class CustomerDaoIntegrationTest {  
  
}
```

Integrate Spring IoC Container

- **@RunWith**

```
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration(locations={
    "/coreservletsContext.xml",
    "/coreservletsTestContext.xml"})
public class CustomerDaoIntegrationTest {
}
```

- **AbstractJUnit4SpringContextTests**

```
@ContextConfiguration(locations={
    "/coreservletsContext.xml",
    "/coreservletsTestContext.xml"})
public class CustomerDaoIntegrationTest
extends AbstractJUnit4SpringContextTests {
}
```

80

Java EE training: <http://courses.coreservlets.com>

Inject Test Fixture with @Autowired

- **Field injection**

```
public class CustomerDaoIntegrationTest {
    @Autowired
    public CustomerDao service;
}
```

- **Setter injection**

```
@Autowired
public void setService(CustomerDao service){
    this.service = service;
}
```

- **Qualified setter injection**

```
@Autowired
public void setService(
    @Qualifier("beanName") CustomerDao service){
    this.service = service;
}
```

81

Java EE training: <http://courses.coreservlets.com>

Inject Test Fixture with @Resource

- **Field injection**

```
public class CustomerDaoIntegrationTest {  
    @Resource  
    public CustomerDao service;  
}
```

- **Setter injection**

```
@Resource  
public void setService(CustomerDao service){  
    this.service = service;  
}
```

- **Qualified setter injection**

```
@Resource(name="beanName")  
public void setService(CustomerDao service){  
    this.service = service;  
}
```

Create Lifecycle Methods (Optional)

```
public class CustomerDaoIntegrationTest {  
    ...  
    @BeforeClass  
    public static void execOnceBeforeAllTestCases(){ }  
    @Before  
    public void execBeforeEachTestCase(){ }  
  
    @Test  
    public void testMethod(){ }  
    @Test  
    public void testDemo01(){ }  
  
    @After  
    public void execAfterEachTestCase(){ }  
    @AfterClass  
    public static void execAfterAllTestCases(){ }  
}
```

Create Test Cases

```
package coreservlets;

import static org.junit.Assert.*;

public class CustomerDaoIntegrationTest {
    ...
    @Test
    public void testCase(){

        assert service != null : "Service is null";

        assertNotNull("Service is not null", service);

    }
}
```

84

Java EE training: <http://courses.coreservlets.com>

Execute Test Cases

```
-----
T E S T S
-----

Running coreservlets. CustomerDaoIntegrationTest
Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time
elapsed: 0.159 sec

Results :

Tests run: 2, Failures: 0, Errors: 0, Skipped: 0
```

85

Java EE training: <http://courses.coreservlets.com>



Wrap-up

Customized Java EE Training: <http://courses.coreservlets.com/>
Servlets, JSP, Struts, JSF/MyFaces/Facelets, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Summary

- **Maven 2**
 - Automates best practices
- **Multi-module project builds**
 - `coreservlets-api`
 - `coreservlets-mock`
 - `coreservlets-jdbc`
 - `coreservlets-main`
- **Spring library integration**
 - **Isolated to the deployment context**
 - `<dependencies><dependency/></dependency>`

Summary Continued

- **Testing with Maven 2 and Spring**
 - **Integrate container**
 - `@RunWith(SpringJUnit4ClassRunner.class)`
 - `@ContextConfiguration(locations={...})`
 - **Inject test fixtures**
 - `@Autowired` or `@Resource`
 - **Test cases**
 - `@Test`
 - **Execute tests**
 - `mvn test`

88

Java EE training: <http://courses.coreservlets.com>

© 2008 coreservlets.com



Questions?

Customized Java EE Training: <http://courses.coreservlets.com/>
Servlets, JSP, Struts, JSF/MyFaces/Facelets, Ajax, GWT, Spring, Hibernate/JPA, Java 5 & 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.