

Java EE

(J2EE Day 1)

Topics

Java servlets and JSP

Apache Tomcat

Apache Maven 2

Spring Framework

Spring Webflow

Spring Security

JPA (Hibernate)

JSF 2.0 (Primefaces)

RDBMS - Oracle, MySQL, Postgres

Java EE Project using Maven

Tomcat Installation

<http://tomcat.apache.org/download-60.cgi>

<http://code.google.com/p/gbif-providertoolkit/wiki/TomcatInstallationMacOSX>

Edit conf/tomcat-users.xml and add :

```
<roll rollname="manager"/>
```

```
<user username="admin" password="password" roles="standard,manager,admin"/>
```

Startup and Shutdown Tomcat

/Users/das/apache-tomcat-6.0.37/bin

==> ./shutdown.sh

Using CATALINA_BASE: /Users/das/apache-tomcat-6.0.37

Using CATALINA_HOME: /Users/das/apache-tomcat-6.0.37

Using CATALINA_TMPDIR: /Users/das/apache-tomcat-6.0.37/temp

Using JRE_HOME: /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home

Using CLASSPATH: /Users/das/apache-tomcat-6.0.37/bin/bootstrap.jar

/Users/das/apache-tomcat-6.0.37/bin

==> ./startup.sh

Using CATALINA_BASE: /Users/das/apache-tomcat-6.0.37

Using CATALINA_HOME: /Users/das/apache-tomcat-6.0.37

Using CATALINA_TMPDIR: /Users/das/apache-tomcat-6.0.37/temp

Using JRE_HOME: /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home

Using CLASSPATH: /Users/das/apache-tomcat-6.0.37/bin/bootstrap.jar

Eclipse IDE Setup

Download Eclipse IDE for Java EE developers

from

<http://www.eclipse.org/downloads/index-developer.php>

Maven Installation

Install maven build tool

- Download maven binary zip from

<http://maven.apache.org/download.cgi?Preferred=ftp://mirror.reverse.net/pub/apache/>

- New system variable MAVEN_HOME

c:/apache-maven-3.0.4

- Edit PATH variable to %MAVEN_HOME%\bin

Test maven installation

```
> mvn --version
```

Eclipse Maven Plugin

- Install new software

Update site for m2eclipse is

<http://download.eclipse.org/technology/m2e/releases>

Select m2e - Maven Integration for Eclipse

Select File -> New Project and filter for “Maven” to test that Maven is correctly installed

Maven settings folder is /Users/<username>/.m2

Install Spring Toolsuite Plugin for Eclipse

Select Eclipse Marketplace

Search for “spring”

Install the correct version of Spring Tool Suite for your version of Eclipse

Check to see if Spring IDE is included

Configure Tomcat on Eclipse

Window -> Preferences -> “server” -> Runtime Environment -> Apache Tomcat v6.0

Window -> Show View -> Servers

To create new tomcat instance

New -> Server -> Server runtime environment -> Finish

Create Project

File -> New -> Dynamic Web Project

Right-click on Project

File -> New -> Servlet

Package : com.example.test

Class name : ServletTest

Uncheck doPost, doGet and check service

Servlet code

```
PrintWriter out = response.getWriter();
```

```
out.println("Hello Java");
```

Browse to

<http://localhost:8080/ServletsJSPExample/servlettest>

Add JSP file

In WebContent

File -> New -> JSP File

Name it index.jsp and add a form

```
<form id="contactForm" action="servlettest" method="post">
    <table border="0">
        <tr>
            <td> First Name : </td> <td><input type="text" name="firstname"/></td>
        </tr>
        <tr>
            <td>Last Name : </td> <td><input type="text" name="lastname"/></td>
        </tr>
        <tr>
            <td>Submit </td> <td> <input type="submit" value="submit"/></td>
        </tr>
    </table>
</form>
```

Modifying Servlet for MVC Pattern

Add RequestDispatcher to ServletTest.java

```
if ((request.getParameter("firstname") == null)
    || (request.getParameter("lastname") == null))
{
    getServletContext().getRequestDispatcher("/index.jsp").forward(
        request, response);
}
```

Refresh

<http://localhost:8080/ServletsJSPExample/servlettest>

Create output.jsp

Right-click on WebContent

File -> New -> JSP -> output.jsp

Use session scope variable as follows :

```
<h1>Your first and last name is : </h1>
```

```
<%
```

```
String firstName = (String) session.getAttribute("firstname");
```

```
String lastName = (String) session.getAttribute("lastname");
```

```
%>
```

Call requestDispatcher in ServletTest.java again as follows :

```
request.setAttribute("firstname", firstName);
```

```
request.setAttribute("lastname", lastName);    getServletContext().getRequestDispatcher("/output.jsp").forward  
(request, response);
```

Maven Project Step1

New -> Project -> Other -> Maven -> Maven Project

Check the checkbox Create a Simple Project

Use defaults

Maven Project Step2

Group Id : com.example.j2eeapp

Artifact Id : j2eeapplication

Version : default

Packaging : war

Name : J2EE Application

Click Finish

Right-click on Project -> Select Spring Tools -> Add Spring Project Nature

Now it's a Spring Framework project

Project Structure

Spring Elements -> Contains Spring configuration

The other 4 folders are quickpaths to java folders which contain java source files

Resources folder contains additional resources and configurations

src/test/java and src/test/resources contains unit tests

We can delete the test folders to keep the folder structure clean

Change JRE System Library to 1.6 by right-clicking -> Properties -> select 1.6

Delete test under src

src/main/webapp is the public web folder

target folder will contain the compiled and packaged project

The web-archive that can be deployed on the web-server and run

pom is the project-object-model

The pom.xml can be edited

maven is more clever than ant and much more robust

Compile and package project

Right-click project -> Run as -> Maven...

Maven clean means delete target folder and all the contents with all packages

Maven install includes other maven commands such as build and package

At the end of it, we get the web-archives

Maven Clean and Maven Install

Run As -> Maven Clean

Target folder should be empty

Run As -> Maven Install

BUILD FAILURE (WEB-INF/web.xml missing)
because we didn't define web archetypes

WEB-INF/web.xml

Create folder called WEB-INF under webapp

Create web.xml under WEB-INF with content

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

```
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
  xmlns="http://java.sun.com/xml/ns/javaee" xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
```

```
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
```

```
  id="WebApp_ID" version="2.5">
```

```
    <display-name>ServletsJSPExample</display-name>
```

```
    <welcome-file-list>
```

```
      <welcome-file>index.jsp</welcome-file>
```

```
    </welcome-file-list>
```

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

Create index.jsp

Right-click on webapp

-> New -> Other - > Web -> JSP File

Modify index.jsp as follows :

```
<h1>Initial project configuration</h1>
```

Maven clean and Maven install

Maven clean

Maven install

Project should say **BUILD SUCCESS**

Upload war file from Tomcat Web Application Manager

