LAB 3: WEB APPLICATION USING SERVLET AND JSP

Lab 3 for EBU6501 Middlware Module – Web App Development

Version: 1.5

Draft: First Draft

Date Created: Oct 2018

Author: Dr. Gokop Goteng

Affiliate: Queen Mary, University of London

To be used for the 2018 at BUPT Lab

This document is meant to guide the students on how to write web applications using Java, Servlets and JSP running in Apache Tomcat Container.

The aim of this lab is to introduce the students to web programming using Amazon Web Services (AWS) Elastic Compute Cloud (EC2) instances and:

- To enable the students put to practical use what has been learnt in the class on Web Services
- To understand how to configure URLs in Deployment Descriptor and Apache Tomcat Container
- To enable students have a practical experience in using AWS EC2 instances for web application programming and deployment

INTRODUCTION

This lab will use either AWS EC2 instances or standalone windows machines Windows machines and Apache Tomcat Container. Go to http://www.oracle.com/technetwork/java/javase/downloads/index.html to download Java J2 SE and http://www.oracle.com/technetwork/java/javaee/downloads/index.html to download Java J2 EE if they have not been downloaded already. Follow the instructions to download them at an appropriate directory.

Download Apache Tomcat version 8 at https://tomcat.apache.org/download-80.cgi (or the latest version) for Windows. Download the 64 bit if the machines in the lab are 64 bits or 32 bits if they are 32 bits machines.

Please make sure your windows machine has at least 1GB Memory, 10GB free Hard disk, 2 processors.

INSTRUCTIONS

IMPORTANT: Please take screenshots of each important step during the lab. This will be the basis to mark your performance in addition to the TA observing what you are doing in the Lab.

CREATING THE DEVELOPMENT ENVIRONMENT

The development environment is where your source codes (Java, Servlets and JSP), HTPM and forms will be located in different directories or folders. Follow these steps to do that:

- Create a folder named "MyProject" under your home directory
- Create a sub-folder under MyProject as: MyProject\MyWebApp
- Create sub-folder MyProject\MyWebApp\etc and create the DD file MyProject\MyWebApp\etc\web.xml
- Create the folder MyProject\MyWebApp\lib
- Create the folder MyProject\MyWebApp\web
- Create the sub folders MyProject\MyWebApp\src\com\example\web and create the java program MyProject\MyWebApp\src\com\example\web\ChocolateSelect.java
- Create the sub folders MyProject\MyWebApp\src\com\example\model and create the java code MyProject\MyWebApp\src\com\example\model\ChocolateExpert.java
- Create the subfolders MyProject\MyWebApp\classes\com\example\web
- Create the sub-folders MyProject\MyWebApp\classes\com\example\model
- Create the sub folders MyProject\MyWebApp\classes\com\example\web

CREATING THE DEPLOYMENT ENVIRONMENT

The deployment environment is where your compiled Java, Servlet binary will be located in different directories or folders. Follow these steps to do that:

- When you install Apache Tomcat, it will be located somewhere in your computer directory, e.g. \ApacheTomcat7\apache-tomcat-8.0\ or "\Program Files\Apache Software Foundation\Tomcat

8.0\ in Windows environment. Please locate where Tomcat is installed and continue. Let us assume that tomcat is located at \ApacheTomcat8 then:

- Under \ApacheTomcat8, you will see the folder "webapps" as \ApacheTomcat8\webapps
- Create a sub-folder as \ApacheTomcat8\webapps\MyChocolateApp
- Create sub folders \ApacheTomcat8\webapps\MyChocolateApp\WEB-INF\classes\com\example\web
- Create sub-folders \ApacheTomcat8\webapps\MyChocolateApp\WEB-INF\classes\com\example\model
- Create the subfolders \ApacheTomcat8\webapps\MyChocolateApp\WEB-INF\lib
- Create the DD file \ApacheTomcat8\webapps\MyChocolateApp\WEB-INF\web.xml
- Create the file \ApacheTomcat8\webapps\MyChocolateApp\form.html
- Create the file \ApacheTomcat8\webapps\MyChocolateApp\result.jsp

DEPLOYING AND TESTING THE WEB APP

- Create the HTML file "form.html" in the folder MyProject/MyWebApp/web

Form.html is:

```
<html><body>

<h1 align="center">Chocolate Selection Page</h1>

<form method="POST"

   action="SelectChocolate.do">

   Select chocolate characteristics<p>

   Color:

   <select name="color" size="1">

      <option value="light"> light </option>

      <option value="amber"> amber </option>

       <option value="brown"> brown </option>

       <option value="dark"> dark </option>

   </select>
```

```
  <br><br>
      <input type="SUBMIT">
 </input>
</form></body></html>
```

- Copy "form.html" to deployment environment as \ApacheTomcat8\webapps\MyChocolateApp\form.html
- Create the deployment descriptor (DD) file web.xml in MyProject/MyWebApp/etc/web.xml

The DD file web.xml is:

```
<web-app xmlns="http://java.sun.com/xml/ns/j2ee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
     version="2.4">

 <servlet>
    <servlet-name>Select Chocolate</servlet-name>
   <servlet-class>com.example.web.ChocolateSelect</servlet-class>
 </servlet>

 <servlet-mapping>
    <servlet-name>Select Chocolate </servlet-name>
   <url-pattern>/SelectChocolate.do</url-pattern>
 </servlet-mapping>
</web-app>
```

- Copy the DD file web.xml to the deployment environment at \ApacheTomcat8\webapps\MyChocolateApp\WEB-INF\web.xml
- Start the Tomcate Container as:
    - cd \ApacheTomcat8\apache-tomcat-8.0\bin
    - startup
- Take the screen shot of the tomcat container running
- Test the web page by:
    - Opening the HTML page using your browser and typing:
        - http://localhost:8080/MyChocolateApp/form.html
    - Take the screenshot of the web page if there are no errors

MAPPING THE LOGICAL NAME TO A SERVLET CLASS FILE

- Create the ChocolateSelect servlet.java class in \ApacheTomcat8\webapps\MyChocolateApp\WEB-INF\classes\com\example\web as:

```
package com.example.web;


import com.example.model.*;

import javax.servlet.*;

import javax.servlet.http.*;

import java.io.*;

import java.util.*;


public class ChocolateSelect extends HttpServlet {

        public void doPost(HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException {
```

```
            String c=request.getParameter("color");
            ChocolateExpert ce=new ChocolateExpert();
            List result=ce.getBrands(c);

            response.setContentType("text/html");
            PrintWriter out=response.getWriter();
            out.println("Chocolate Selection Advice<br>");
            //String c=request.getParameter("color");
            //out.println("<br>Got chocolate color " + c);

            Iterator it=result.iterator();
            while(it.hasNext()) {
                  out.print("<br>try: " + it.next());
            }

            //request.setAttribute("styles", result);
            //RequestDispatcher view=request.getRequestDispatcher("result.jsp");
            //view.forward(request,response);

      }
}
```

- Create the ChocolateExpert.java servlet class in \ApacheTomcat8\webapps\MyChocolateApp\WEB-INF\classes\com\example\model as:

```
package com.example.model;

import java.util.*;


public class ChocolateExpert {

        public List getBrands(String color) {

                List brands = new ArrayList();

                if (color.equals("brown")) {

                        brands.add("Dark Brown");

                        brands.add("White");

                }

                else {

                        brands.add("Pale");

                        brands.add("Light Brown");

                }

                return (brands);

        }

}
```

COMPILING THE SERVLET

- Run the following commands and taking screenshots of each of the steps if there are no errors:
    - cd MyProject\MyChocolateApp
    - C:\Users\Gokop\MyProjects\MyChocolateApp>javac –classpath "\Program Files\Apache Software Foundation\Tomcat 8.0\common\lib\servlet-api.jar";classes –d classes

src\com\example\web\ChocolateSelect.java
    - cd \ApacheTomcat7\apache-tomcat-8.0\bin
    - startup
- Take a screenshot of each of the steps above

DEPLOYING AND TESTING THE WEB APP AGAIN

- Copy the servlet calss ChocolateSelect.class to deployment environment as \ApacheTomcat8\webapps\MyChocolateApp\WEB-INF\classes\com\example\web\ChocolateSelect.class
- Copy the model class ChocolateExpert.class to deployment environment as \ApacheTomcat8\webapps\MyChocolateApp\WEB-INF\classes\com\example\model\ChocolateExpert.class
- Run the following commands:
    - cd \ApacheTomcat7\apache-tomcat-8.0\bin
    - shutdown
    - startup
- Take screenshots
- Start up your web browser and type:
    - http://localhost:8080/MyChocolateApp/form.html
- Take a final screenshot if there are no errors

QUESTIONS FOR STUDENTS TO ANSWER AFTER THE LAB SESSION

- Draw the diagram of the deployment directory and explain why “web.xml”, “form.html” and “result.jsp” are located in such specific locations.
- Explain why you need a development environment and deployment environment
- Describe the parts in your lab which shows the Model (M), View (V) and Controller (C) as an example of MVC design pattern
- Submit one file with all your screenshots during the lab