

# Session Handling

Step 1: Creating a dynamic web project

- Open Eclipse
- Go the File menu. Choose New->Dynamic Web Project
- Enter the project name as JSPSessions. Click on Next
- Enter nothing in the next screen and click on Next
- Check the checkbox Generate web.xml deployment descriptor and click on Finish
- This will create the project files in the Project Explorer.

Step 2: Creating a JSP file index.jsp

- In the Project Explorer, expand the project JSPSessions
- Expand WebContent. Right click on WebContent. Choose New->JSP File
- Enter the filename as index.jsp and click on Finish
- Enter the following code:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>JSP Sessions</title>
</head>
<body>
<%
    if (request.getParameter("error") != null)
        out.println("<b>Your session has expired or is invalid.</b>
<br>");
```

```

%>
<form action="login.jsp" method="post">
  Name <input name="name" id="name" maxlength=40><br>
  Password <input type="password" name="pwd" id="pwd"
maxlength="10"><br>
  <button>Submit</button>

</form>
</body>
</html>

```

- Click on the Save icon

### Step 3: Creating a JSP file login.jsp

- In the Project Explorer, expand the project JSPSessions
- Expand WebContent. Right click on WebContent. Choose New->JSP File
- Enter the filename as login.jsp and click on Finish
- Enter the following code:

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<%

```

```

    String name = request.getParameter("name");
    String pwd = request.getParameter("pwd");

    if (name == null || name.equals("") || pwd == null ||
pwd.equals("")) {
        response.sendRedirect("index.jsp?error=1");
    } else {

        session.setAttribute("name", name);
        response.sendRedirect("dashboard.jsp");
    }
}

```

```
}  
%>
```

- Click on the Save icon

#### Step 4: Creating a JSP file dashboard.jsp

- In the Project Explorer, expand the project JSPSessions
- Expand WebContent. Right click on WebContent. Choose New->JSP File
- Enter the filename as dashboard.jsp and click on Finish
- Enter the following code:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"  
pageEncoding="UTF-8"%>  
<!DOCTYPE html>  
<html>  
<head>  
<meta charset="UTF-8">  
<title>Dashboard</title>  
</head>  
<body>  
<%  
    if (session.getAttribute("name") == null) {  
        response.sendRedirect("index.jsp?error=1");  
    }  
%>  
<b>Hello <%= session.getAttribute("name") %></b><br>  
<a href="logout.jsp">Click here to logout</a>  
</body>  
</html>
```

- Click on the Save icon

#### Step 5: Creating a JSP file logout.jsp

- In the Project Explorer, expand the project JSPSessions
- Expand WebContent. Right click on WebContent. Choose New->JSP File
- Enter the filename as logout.jsp and click on Finish
- Enter the following code:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Session Logout</title>
</head>
<body>
<%
    session.invalidate();
%>
<b>Your session has terminated.</b><br>
<a href="index.jsp">Login again</a>
</body>
</html>
```

- Click on the Save icon

#### Step 6: Checking for servlet-api.jar

- Before building the project, we need to add servlet-api.jar to the project

- Servlet-api.jar file is already present in your practice lab. (Refer FSD: Lab Guide - Phase 2)
- To add it to the project, follow the below mentioned steps:
  - In the Project Explorer, right click on JSPSessions and choose Properties
  - Select Java Build Path from the options on the left
  - Click on Libraries tab on the right
  - Under ClassPath, expand the node that says Apache Tomcat
  - If there is an existing entry for servlet-api.jar, then click on Cancel and exit the window
  - If it is not there, then click on Classpath entry and click on Add External JARs button on the right
  - From the file list, select servlet-api.jar file and click on Ok
  - Click on Apply and Close

#### Step 7: Building the project

- From the Project menu at the top, click on Build
- If any compile errors are shown, fix them as required

#### Step 8: Publishing and starting the project

- If you do not see the Servers tab near the bottom of the IDE, go to Window menu and click Show View->Servers
- Right click on the Server entry and choose Add and Remove

- Click the Add button to move JSPSessions from the Available list to the Configured list
- Click on Finish
- Right click on the Server entry and click on Publish
- Right click on the Server entry and click on Start
- This will start the server

#### Step 9: Running the project

- To run the project, open a web browser and type:  
<http://localhost:8080/JSPSessions>

#### Step 10: Pushing the code to your GitHub repositories

- Open your command prompt and navigate to the folder where you have created your files.

`cd <folder path>`

- Initialize your repository using the following command:

`git init`

- Add all the files to your git repository using the following command:

`git add .`

- Commit the changes using the following command:

```
git commit . -m "Changes have been committed."
```

- Push the files to the folder you initially created using the following command:

```
git push -u origin master
```