SOEN 387
WEB-BASED ENTERPRISE APPLICATIONS DESIGN

TUTORIAL 4

JSP and Servlets

Agenda

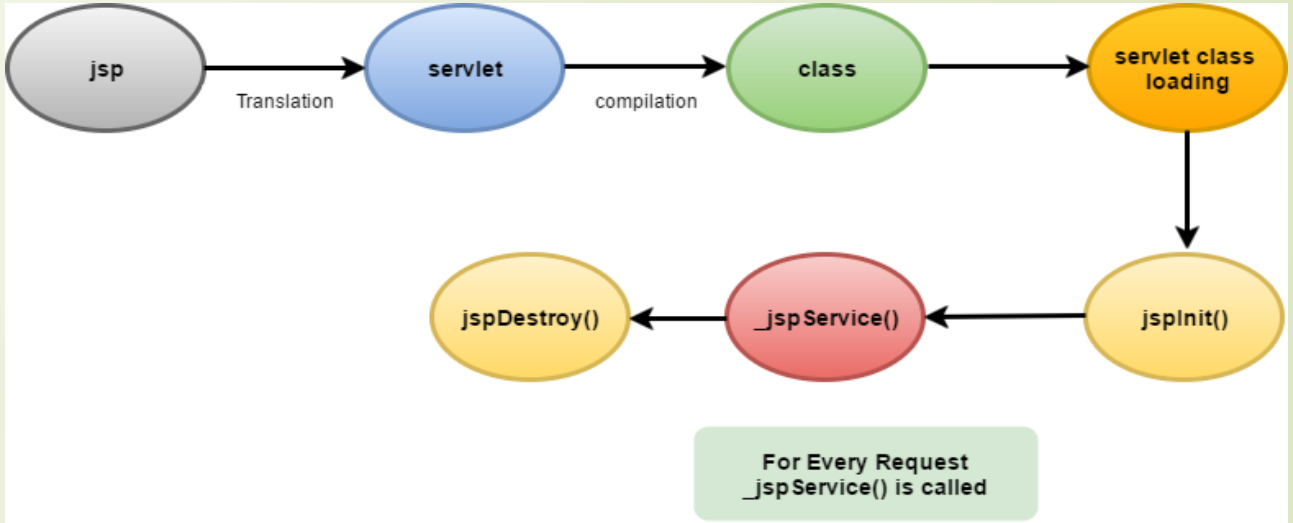
- Introduction to JSP
- JSP Lifecycle
- JSP vs Servlet
- Steps to Create JSP project in Eclipse
- Starting the server and deploying the project
- Exercises

JSP

- **Extension to Servlet** Web application just like Servlet technology
- JSP page consists of HTML tags and JSP tags.
- **Maintainable** JSP separate business logic and presentation logic
- **Less code** than Servlet
- *Reuse of components and tag libraries*:
- **Server side programming**

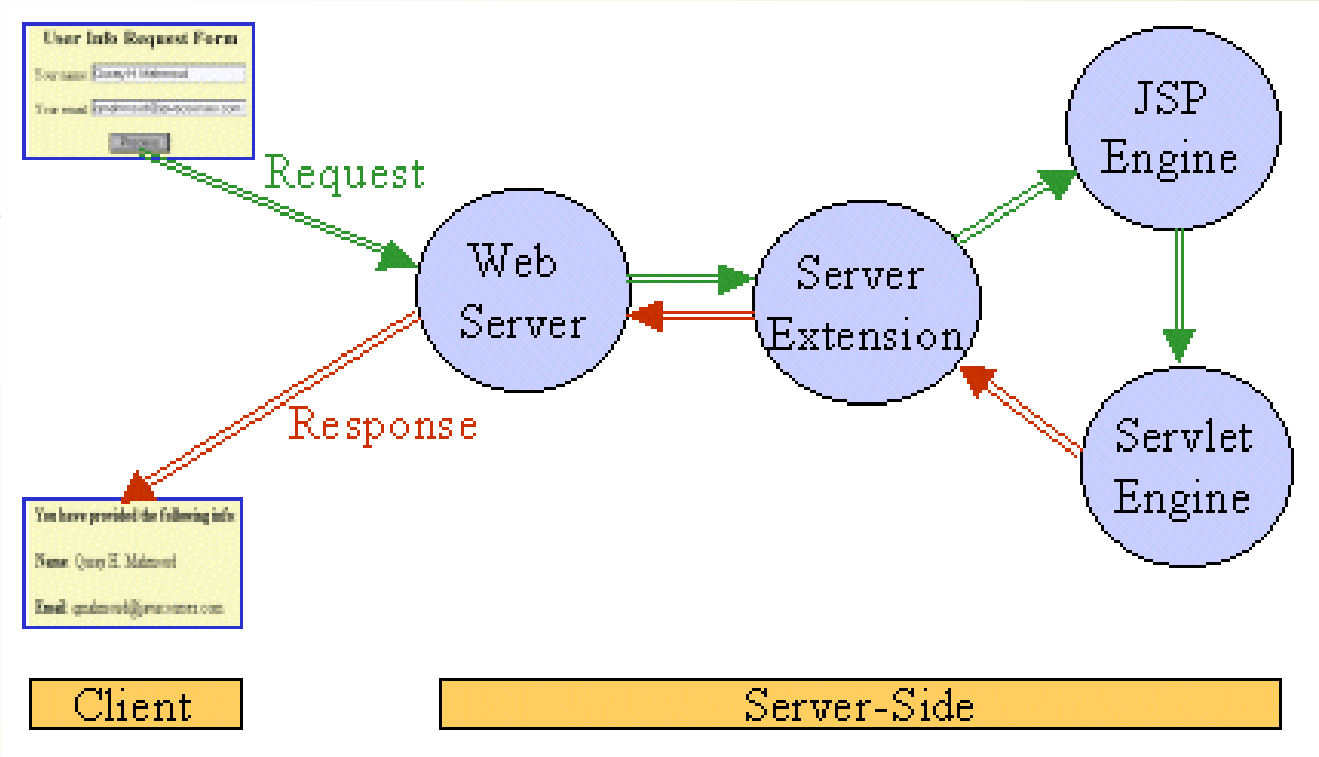
JSP Life cycle

- The JSP pages follow these phases:
  1. Translation of JSP to Servlet code.
  2. Compilation of Servlet to bytecode.
  3. Loading Servlet class.
  4. Creating servlet instance.
  5. Initialization by calling jspInit() method
  6. Request Processing by calling _jspService() method
  7. Destroying by calling jspDestroy() method



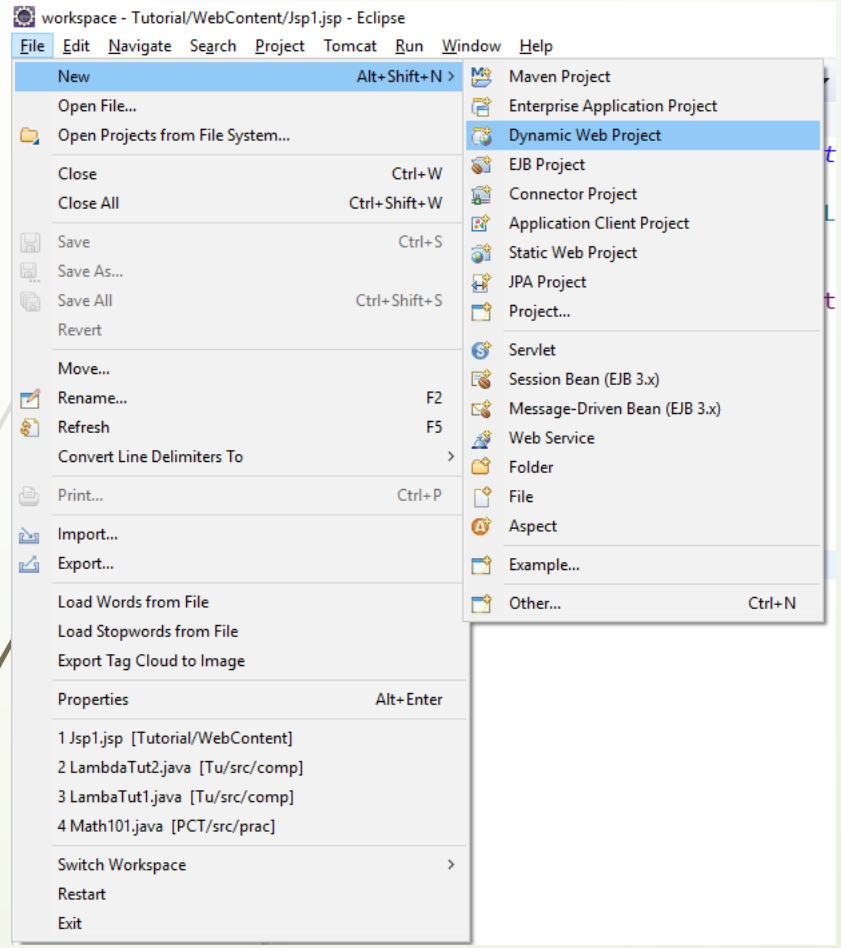
JSP vs Servlet

SERVLET	JSP
A servlet is a server-side program and written purely on Java.	JSPs are HTML pages with .jsp extension. JSP's are extension of servlets to minimize the effort of developers to write User Interfaces using Java programming.
Executes inside a Web server, such as Tomcat	A JSP program is compiled into a Java servlet before execution. Once compiled into a servlet, it's life cycle will be same as of servlet. But, JSP has it's own API for the lifecycle.
Servlets run faster than JSP	JSP runs slower because it has the transition phase for converting from JSP page to a Servlet file.

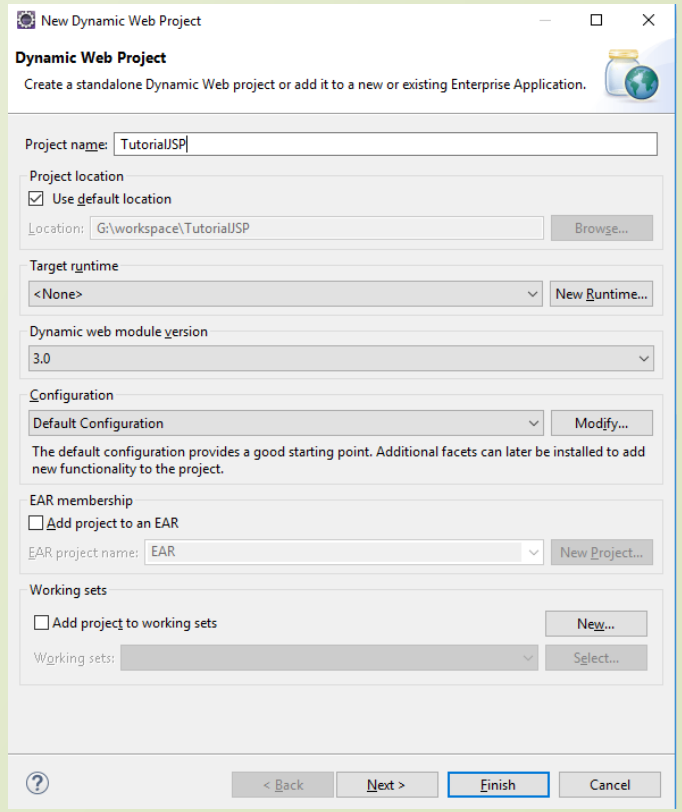


Request/Response when calling JSP

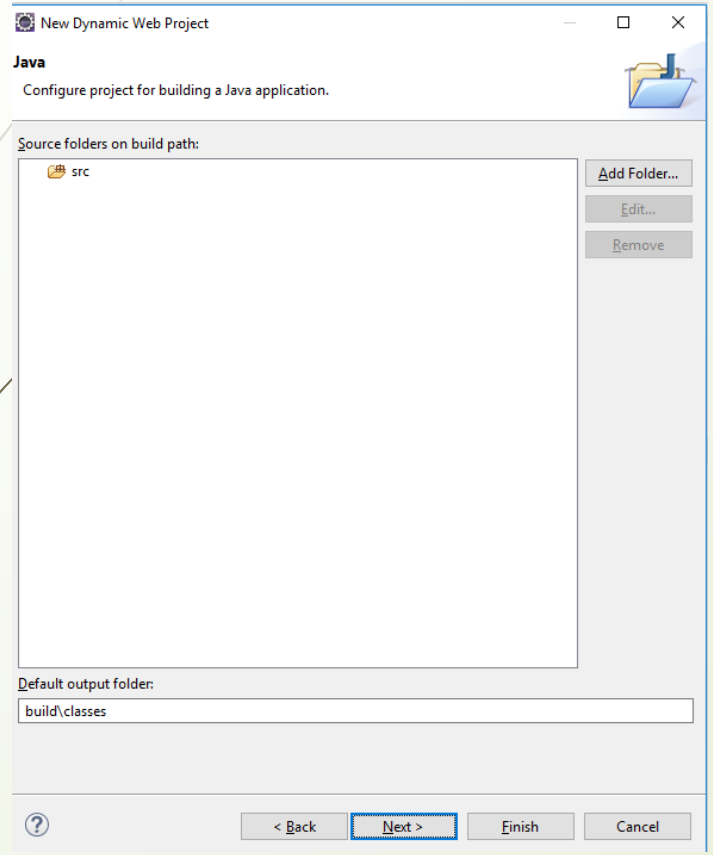
Steps to create web project



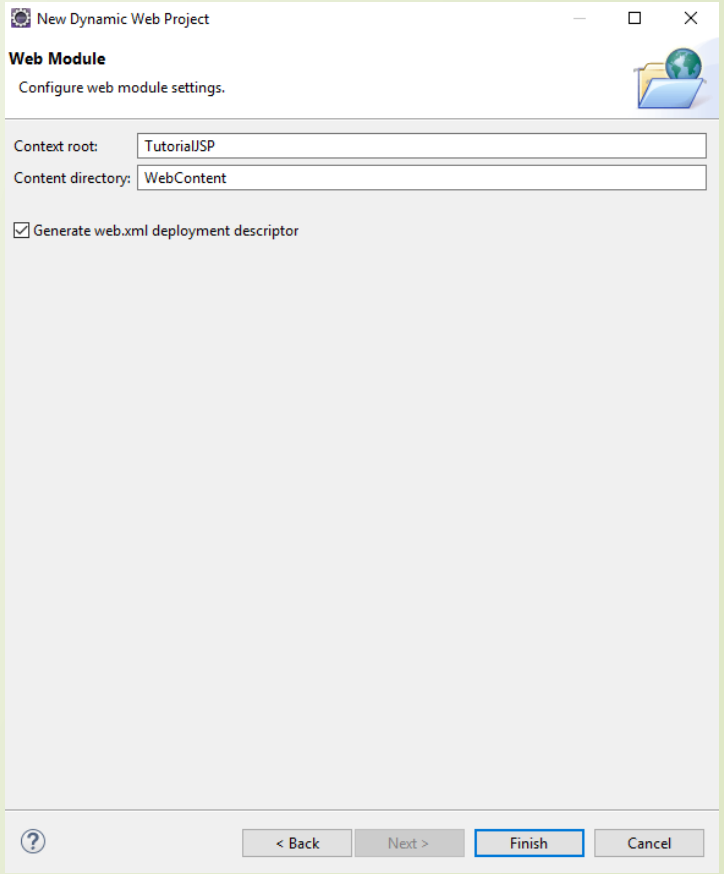
- Step 1: Select File>New>Dynamic Project
- Step 2: Enter ProjectName
- Step 3: Click Finish



- Step 4: Select Next

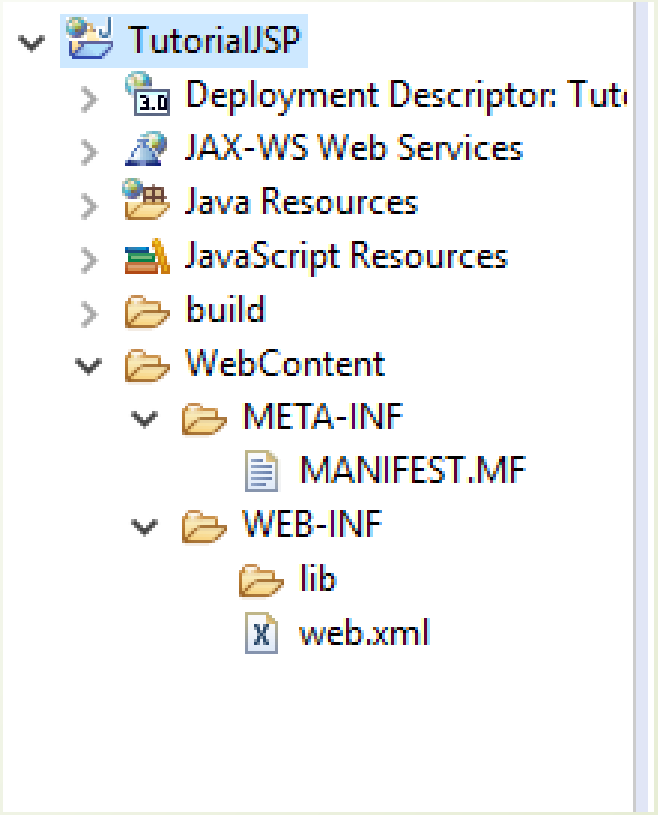


- Step 5: Select the option for Generate web.xml



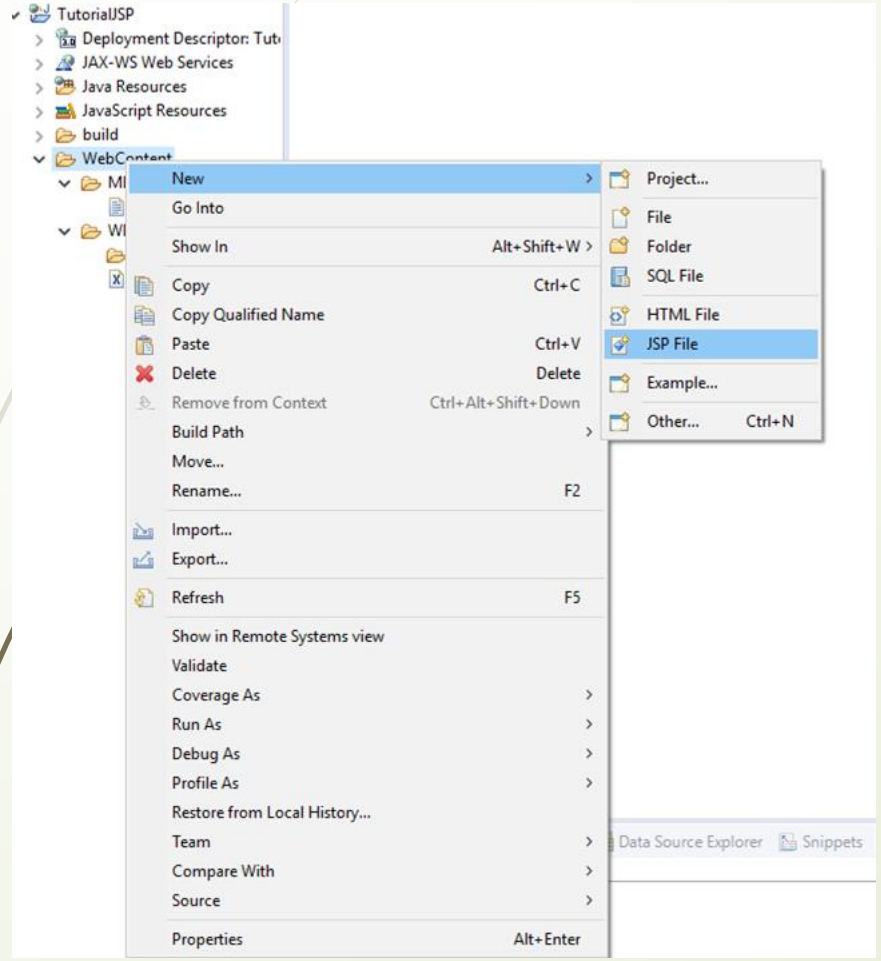
- Step 6: Select Finish

Directory Structure for web Project

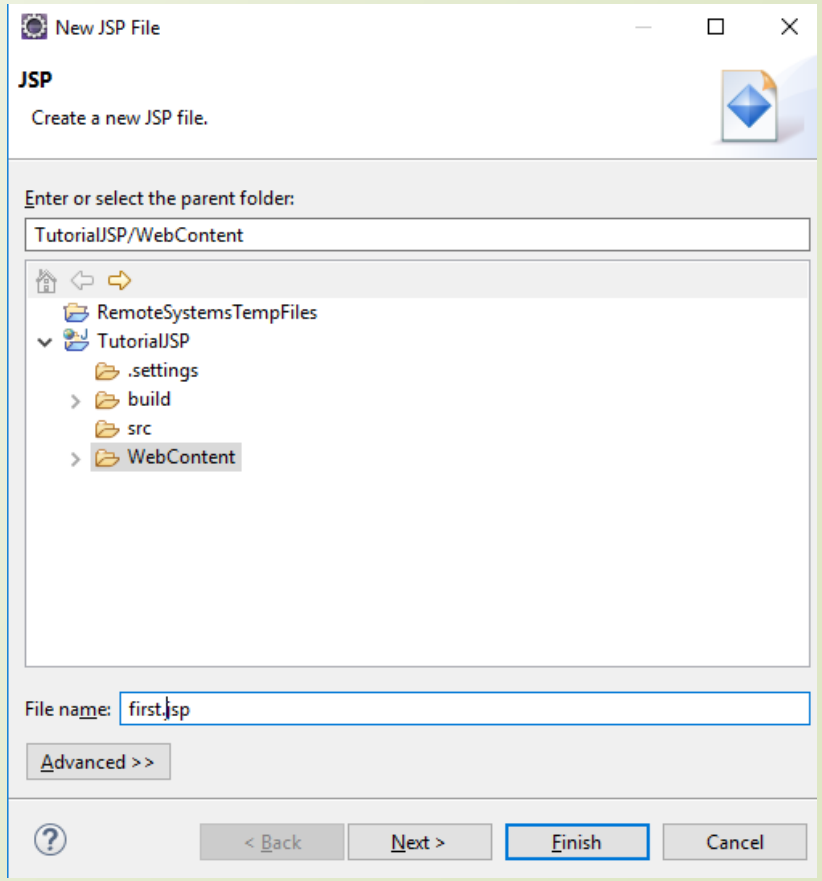


Creating a JSP file

- Select WebContent folder > New > JSP File

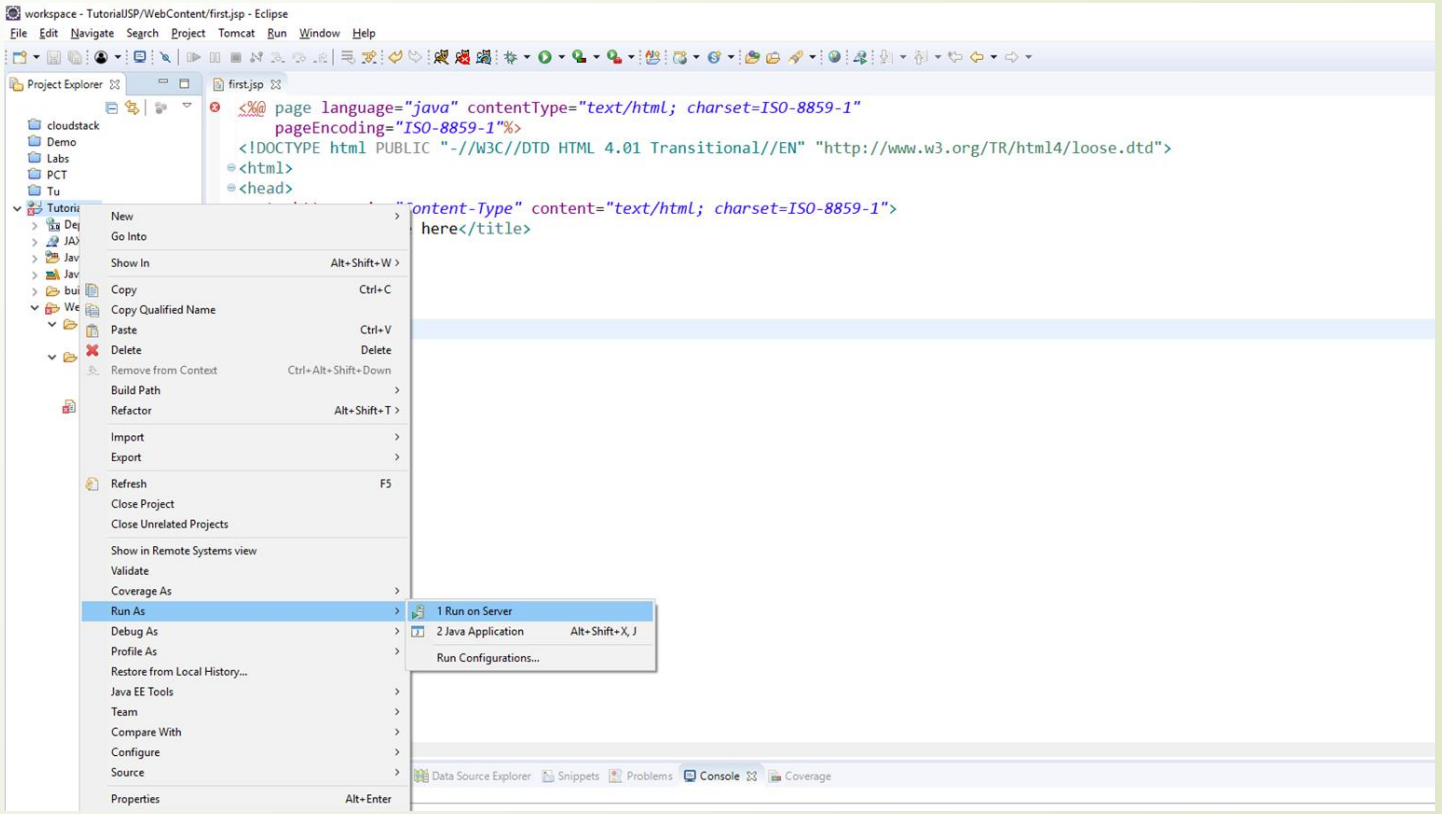


- Step 5: Select the option for Generate web.xml
- Step 6: Select Finish
- Select Finish after giving a file name

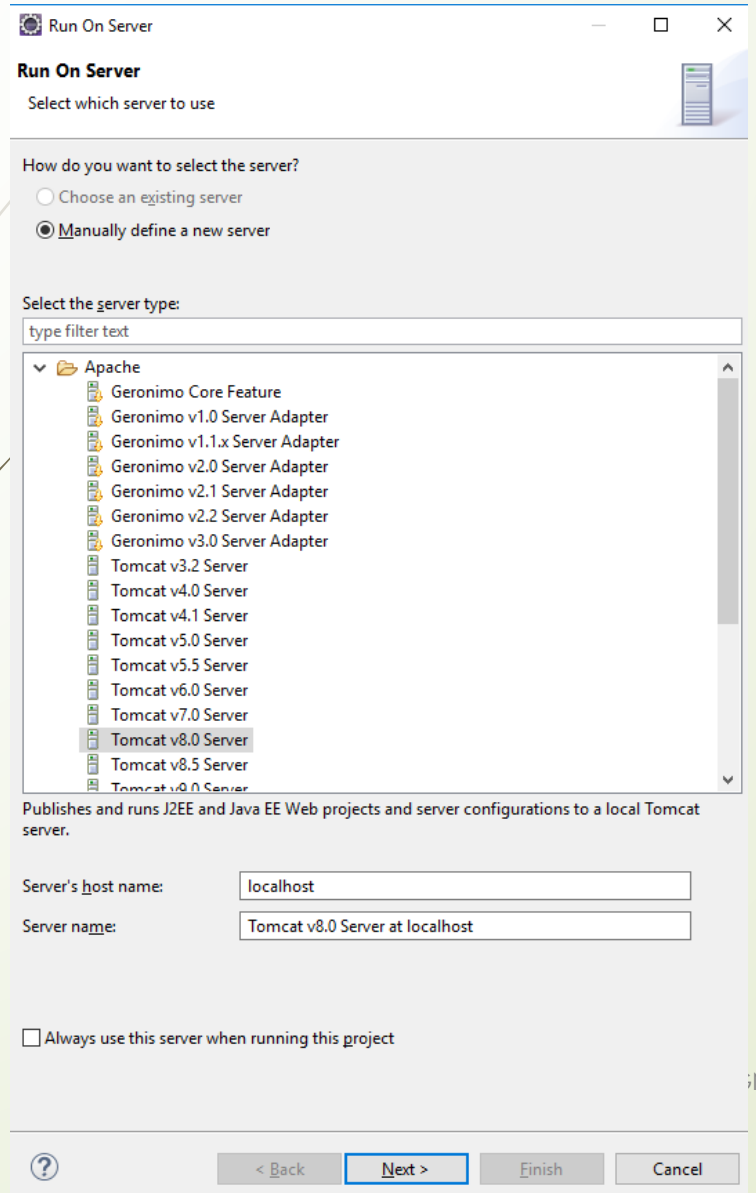


Starting the server and deploying the project

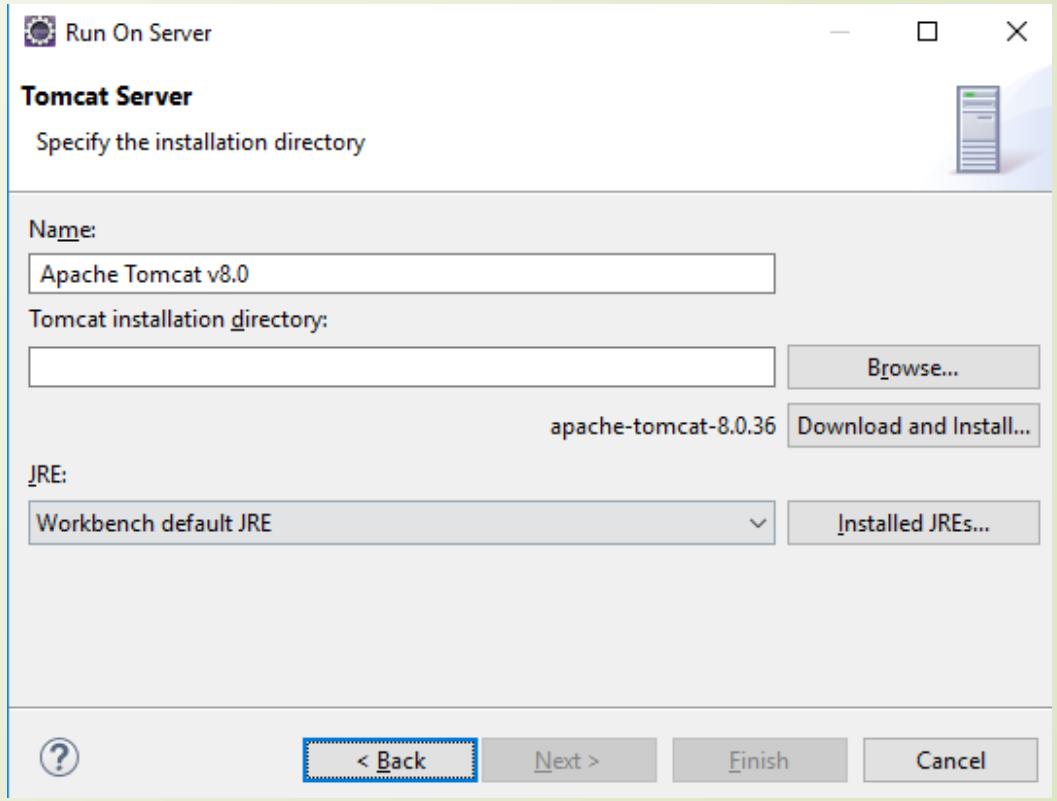
Right click on project -> Run As -> Run on Server



Run on Tomcat Server

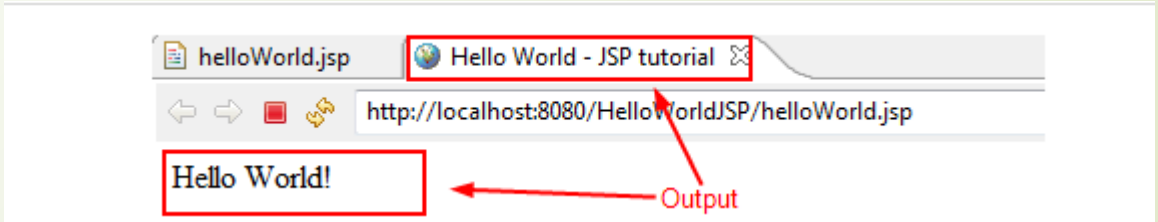


Choose Tomcat Server -> Next -> Select Download and Install -> Finish



first.jsp | web.xml

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Hello World!</title>
</head>
<body>

</body>
</html>
```

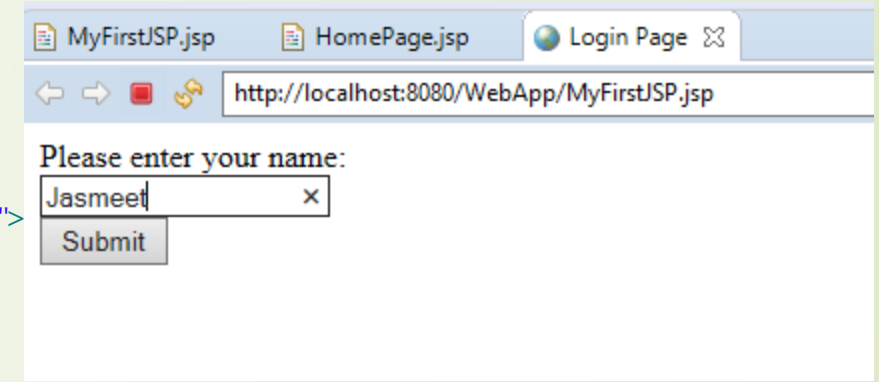


Exercise 1: Creating Form

- AIM: Posting data to the jsp
- Create a dynamic page with your name as input
- Display your name on next page

MyFirstJSP.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Login Page</title>
</head>
<body>
<form action="HomePage.jsp" method="GET">
Please enter your name: <br>
<input type="text" name="username">
<br />
<input type="submit" value="Submit" />
</form>
</body>
</html>
```



HomePage.jsp

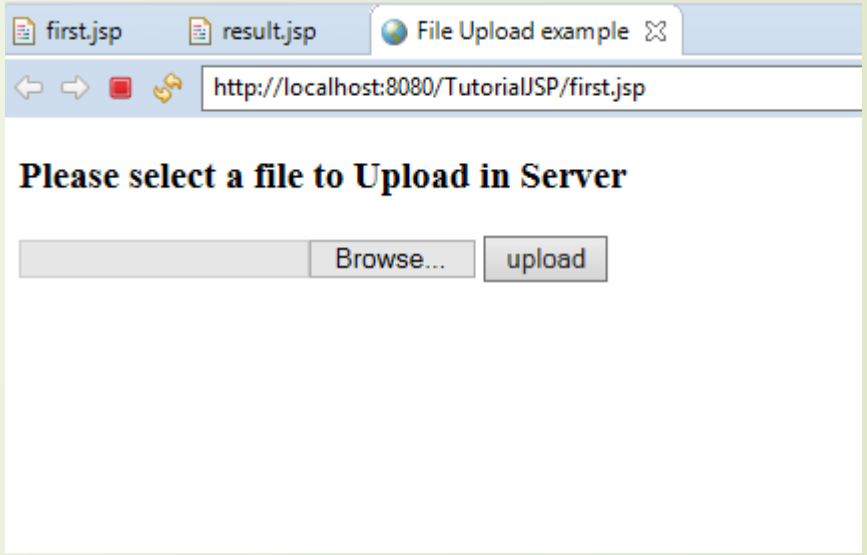
```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Welcome to Home Page</title>
</head>
<body bgcolor="yellow">
<h1>Welcome User</h1>
<h2><b><%= request.getParameter("username")%></b></h2>
</p>

</body>
</html>
```

Exercise 2: File Upload

- Use jsp for uploading a file to servlet
- Process the file
- Display file information as file name, total bytes

First.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>File Upload example</title>
</head>
<body>
<div>
<h3>Please select a file to Upload in Server</h3>
<form action="upload" method="post" enctype="multipart/form-data">
<input type="file" name="file" /> <input type="submit"
value="upload" />
</form>
</div>
</body>
</html>
```



Result.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>File successfully uploaded</title>
</head>
<body>
<div id="result">
        <h3>${requestScope["message"]}</h3>
    </div>
</body>
</html>
```

Try Exercise 2b: File Upload

- Case 1: JSP1 uploads file to a servlet, a servlet processes the file and display some output
- Case 2: JSP1 uploads file to a servlet, a servlet processes the file and then redirects (or forwards) to the second jsp
  - Option 2-1: use redirect
  - Option 2-2: use forward

File Upload servlet


```
public class FileUploadHandler extends HttpServlet {
   private final String UPLOAD_DIR = "C:/uploads/"; //hard-coded (BAD PRACTICE)
     @Override
   protected void doPost(HttpServletRequest request, HttpServletResponse response)
         throws ServletException, IOException {
      if(ServletFileUpload.isMultipartContent(request)){ //process only if its multipart content
         try {
            List<FileItem> multiparts = new ServletFileUpload(new DiskFileItemFactory()).parseRequest(request);
            for(FileItem item : multiparts){
               if(!item.isFormField()){
                  String name = new File(item.getName()).getName();
                  item.write( new File(UPLOAD_DIR + name));
               } }
         request.setAttribute("message", "File Uploaded Successfully");
         } catch (Exception ex) {
            request.setAttribute("message", "File Upload Failed due to " + ex);
         }
      }else{
         request.setAttribute("message",
                        "Sorry this Servlet only handles file upload request");
      }
      // case 2a
      // request.sendRedirect("/result.jsp");
      //return; // important
      // case 2b
      request.getRequestDispatcher("/result.jsp").forward(request, response);
      //return; // important
   }
}
```

```
import java.io.File;
import java.io.IOException;
import java.util.List;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.commons.fileupload.FileItem;
import org.apache.commons.fileupload.disk.DiskFileItemFactory;
import org.apache.commons.fileupload.servlet.ServletFileUpload;
```

References

- https://www.javatpoint.com/jsp-api
- http://jasdhir.blogspot.com/2017/02/jsp-lifecycle.html
- https://www.theserverside.com/news/1365029/Web-Application-Development-with-JSP-and-XML-Part-I-Fast-Track-JSP
- https://www3.ntu.edu.sg/home/ehchua/programming/java/JavaServerPages.html
- https://www.javacodegeeks.com/2013/08/file-upload-example-in-servlet-and-jsp.html