Hands-On Lab

Lab Manual

Develop a simple servlet application

**Contents**

[Develop a simple servlet application i](#_Toc439972690)

[Lab: A simple servlet application tutorial 3](#_Toc439972691)

[Lab Objectives 3](#_Toc439972692)

[Exercise 1 – Create web application 3](#_Toc439972693)

[1. Create a Dynamic Web Application 3](#_Toc439972694)

[2. Modify context path into the lowercase 4](#_Toc439972695)

[3. Configure the resource to support Unicode 5](#_Toc439972696)

[Exercise 2 - Writing Login servlet 6](#_Toc439972697)

[1. Create a new servlet by press Ctrl + N 6](#_Toc439972698)

[2. Input information for servlet 7](#_Toc439972699)

[3. Write code for LoginServlet 9](#_Toc439972700)

[4. Overview the method doGet 10](#_Toc439972701)

[5. Create the index page “index.jsp” 11](#_Toc439972702)

[6. Create the login page “login.jsp” 11](#_Toc439972703)

[7. Run the web application 12](#_Toc439972704)

[Exercise 3 – Create Login form and upgrade the LoginServlet 13](#_Toc439972705)

Lab: A simple servlet application tutorial

Estimated time to complete this lab: 45 minutes

## Lab Objectives

The objective of this lab is to introduce you how to create a simple servlet with eclipse.

* Create a dynamic web project in clipse.
* Write a jsp home page to forward to servlet
* Write a servlet
* Running the application with Tomcat.

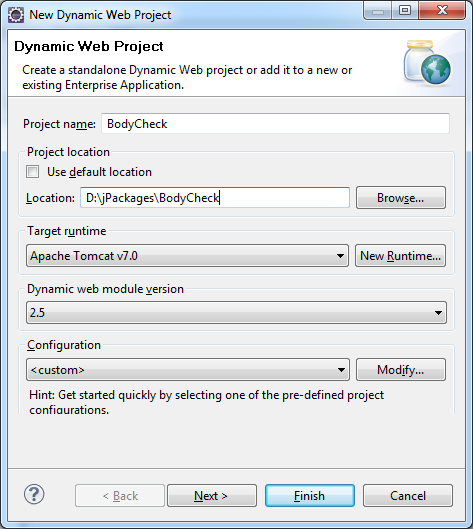
**Prerequisite**:

- Finished labs: Install JDK and Tomcat, Configure Eclipse.

## Exercise 1 – Create web application

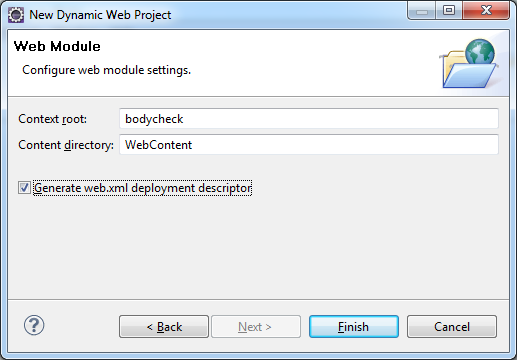
* 1. Create a Dynamic Web Application

Going to menu File > New > Dynamic Web Project. Input the Project name “BodyCheck” (without double quotes).

Select Target runtime is “Apache Tomcat v7.0”; Dynamic web module version is “2.5”.

Click on Next twice.

* 1. Modify context path into the lowercase



Click on Finish button. The structure of the project is:



The content of web application deployment are generate as below:

|  |
| --- |
| <?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>BodyCheck</display-name>  <welcome-file-list>  <welcome-file>index.html</welcome-file>  <welcome-file>index.htm</welcome-file>  <welcome-file>index.jsp</welcome-file>  <welcome-file>default.html</welcome-file>  <welcome-file>default.htm</welcome-file>  <welcome-file>default.jsp</welcome-file>  </welcome-file-list>  </web-app> |

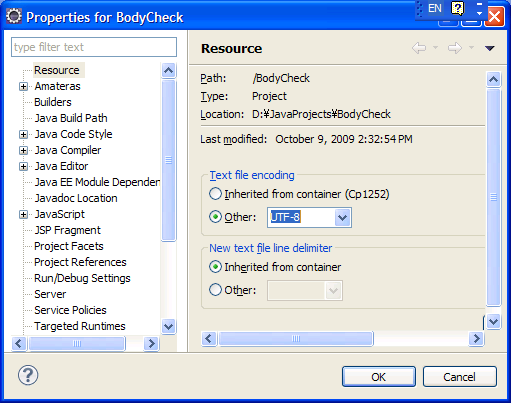
Remove all tag “welcome-file” exception tag “<welcome-file>index.jsp</welcome-file>”.

It means the default home page is index.jsp

* 1. Configure the resource to support Unicode

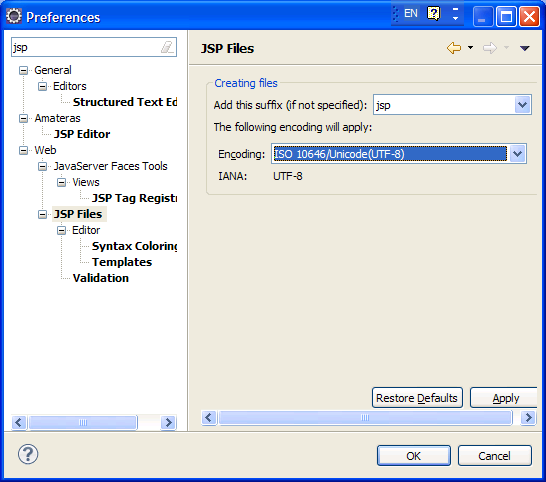
 In case you use the Open-Ones Group's eclipse, skip this step.

Right-click on the project, select Properties. The properties dialog is displayed as below



Select node “Resource”, choose value of “Text file encoding” is UTF-8.

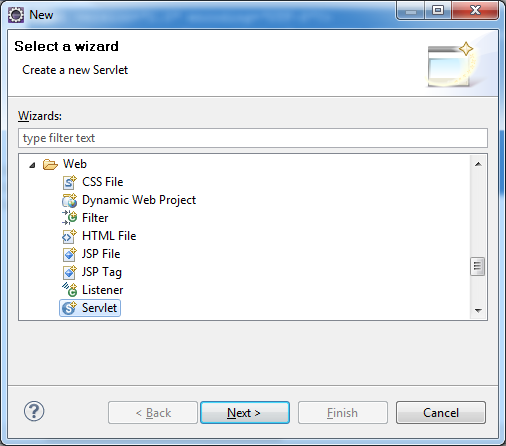
Check the default JSP encoding in eclipse by going to Window > Preferences. Expand node Web > JSP Files. Select encoding for JSP file is “ISO 10646/Unicode(UTF-8).



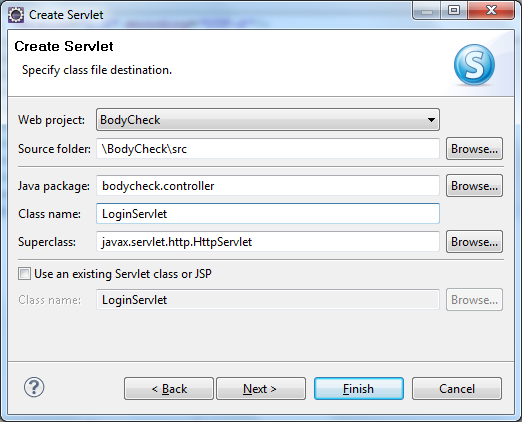
## Exercise 2 - Writing Login servlet

* 1. Create a new servlet by press Ctrl + N

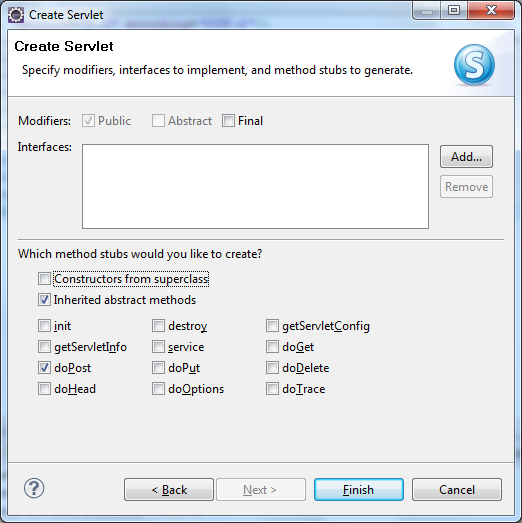
Select note “**Web** > **Servlet”**. Then click on Next button



* 1. Input information for servlet

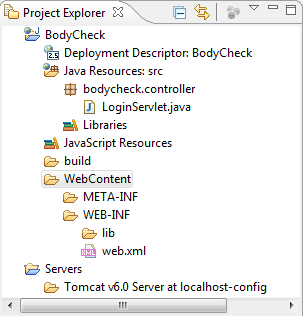


Click on Next button twice. Select below optional to generate source codes.



Click on Finish button.

Two files are generated/updates.



1. File web application deployment web.xml is updated

|  |
| --- |
| <?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>BodyCheck</display-name>  <welcome-file-list>  <welcome-file>default.jsp</welcome-file>  </welcome-file-list>  <servlet>  <description></description>  <display-name>LoginServlet</display-name>  <servlet-name>**LoginServlet**</servlet-name>  <servlet-class>**bodycheck.controller.LoginServlet**</servlet-class>  </servlet>  <servlet-mapping>  <servlet-name>**LoginServlet**</servlet-name>  <url-pattern>**/LoginServlet**</url-pattern>  </servlet-mapping>  </web-app> |

1. File LoginServlet.java is generated:

|  |
| --- |
| package bodycheck.controller;  import java.io.IOException;  import javax.servlet.ServletException;  import javax.servlet.http.HttpServlet;  import javax.servlet.http.HttpServletRequest;  import javax.servlet.http.HttpServletResponse;  /\*\*  \* Servlet implementation class LoginServlet  \*/  public class **LoginServlet extends HttpServlet** {  private static final long serialVersionUID = 1L;  /\*\*  \* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)  \*/  protected void **doPost**(HttpServletRequest request, HttpServletResponse response)  throws ServletException, IOException {  // TODO Auto-generated method stub  }  } |

* 1. Write code for LoginServlet

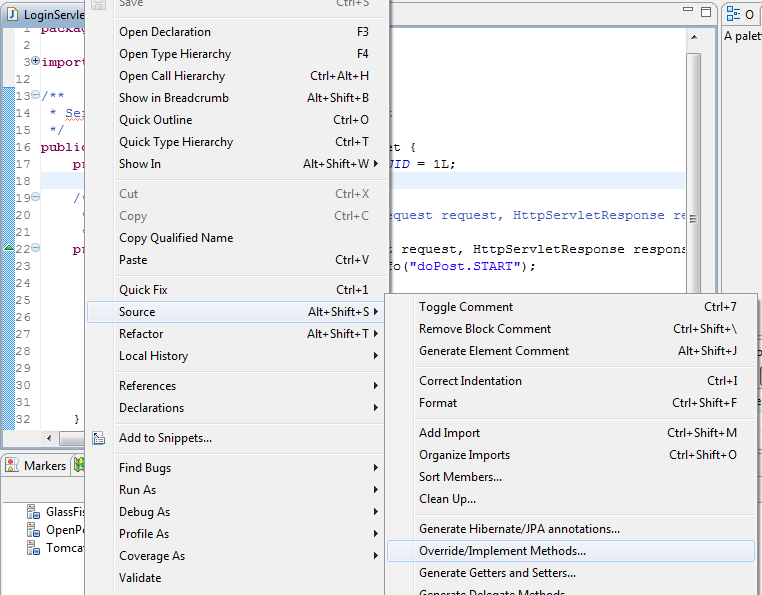
Write code for method doPost(request, response):

|  |
| --- |
| Logger.*getLogger*("LoginServlet").info("doPost.START");    Date currentDate = **new** Date();  SimpleDateFormat sdf = **new** SimpleDateFormat("EEEEE");  String today = sdf.format(currentDate);  request.getSession().setAttribute("TODAY", today);  request.getRequestDispatcher("login.jsp").forward(request, response);    Logger.*getLogger*("LoginServlet").info("doPost.START"); |

* 1. Overview the method doGet

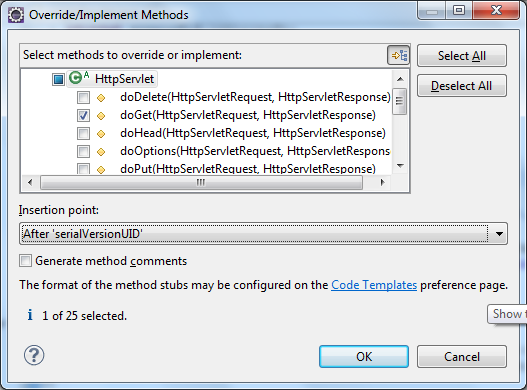
Right-click on the space of source code LoginServlet, select **Source** > **Override/Implement Methods**…

Or using Alt+Shift+N



Right-click on the space of source code

Select the method will be overrided or implemented



Then click OK button

Method doGet is generated into the LoginServlet.java. Now, add a statement to call the method doPost as below:

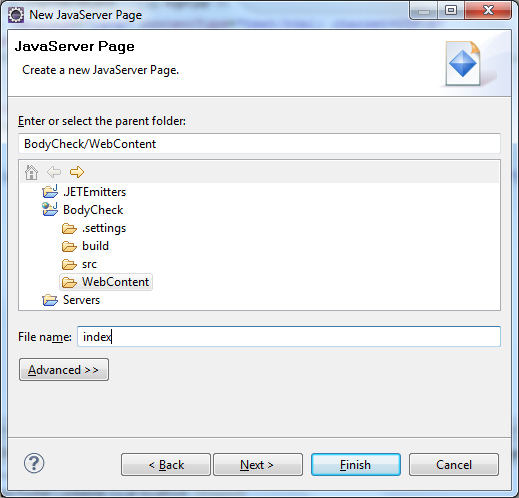
|  |
| --- |
| **protected** **void** doGet(HttpServletRequest req, HttpServletResponse resp) **throws** ServletException, IOException {  **doPost(req, resp);**  } |

* 1. Create the index page “index.jsp”

Press Ctrl + N, Select node Web > JSP. Then click on Next button.

Select “parent folder” is “BodyCheck > WebContent”.

Input File name is “index” (all characters are in lowercase).



Then click on Next and Finish button.

The index.jsp file is generated. You add statement

<jsp:forward page=*"LoginServlet"*></jsp:forward>

into the body tag. The final code of index.jsp is as below:

|  |
| --- |
| <%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*  pageEncoding=*"UTF-8"*%>  <!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=UTF-8"*>  <title>Insert title here</title>  </head>  <body>  **<jsp:forward page=*"LoginServlet"*></jsp:forward>**  </body>  </html> |

* 1. Create the login page “login.jsp”

Press Ctrl + N, Select node Web > JSP. Then click on Next button.

Select “parent folder” is “BodyCheck > WebContent”.

Input File name is “login”. Then click on Finish button.

After the login.jsp is generated. Add below code for the body tag:

This is the login page. Current time is <%= session.getAttribute("TODAY") %>

The final login.jsp is:

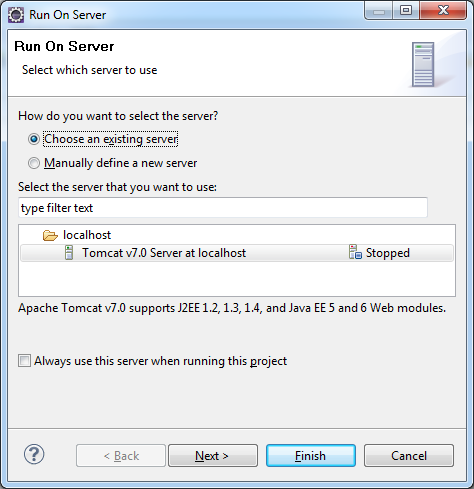
|  |
| --- |
| <%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*  pageEncoding=*"UTF-8"*%>  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">  <html>  <head>  <meta http-equiv=*"Content-Type"* content=*"text/html; charset=UTF-8"*>  <title>Login screen</title>  </head>  <body>  **This is the login page. Current time is <%= session.getAttribute("TODAY") %>**  </body>  </html> |

Save all files.

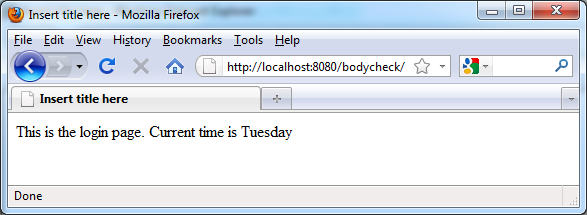
* 1. Run the web application

Right-click on the project, choose Run As > Run on Server (or using hot key Alt-Shift-X. Next press R).

Then select the server to run the project. Finally, click on Finish button.



The configured default browser is displayed as below:



Now, you have created and run the simple web application with Servlet successfully. **Congratulation!**

## Exercise 3 – Create Login form and upgrade the LoginServlet

Create the login form in login.jsp. The form contains 2 inputs “Email” and “Password” and a submit button. Click on submit button the LoginServlet is invoked to process Authentication in server side.

Please try it your-self!