**Servlets in Java**

|  |
| --- |
| **Servlets**: Overview, Life Cycle of Servlet, Attributes in Servlets, Interaction between Client & Servlet, Servlet demo Application development with Sessions |

Servlets are the Java programs that run on the Java-enabled web server or application server. They are used to handle the request obtained from the webserver, process the request, produce the response, then send a response back to the webserver.

**Properties of Servlets are as follows**:

* Servlets work on the server-side.
* Servlets are capable of handling complex requests obtained from the webserver.

Servlet Architecture can be depicted from the image itself as provided below as follows:

Diagram

Description automatically generated

**Execution of Servlets basically involves six basic steps: (**Interaction between Client & Servlet)

1. The clients send the request to the webserver.
2. The web server receives the request.
3. The web server passes the request to the corresponding servlet.
4. The servlet processes the request and generates the response in the form of output.
5. The servlet sends the response back to the webserver.
6. The web server sends the response back to the client and the client browser displays it on the screen

For example, we can use a Servlet to collect input from a user through an HTML form, query records from a database, and create web pages dynamically.

Servlets are under the control of another Java application called a **Servlet Container.**When an application running in a web server receives a request, the Server hands the request to the Servlet Container – which in turn passes it to the target Servlet.

**Servlets - Life Cycle**

1. Servlet class is loaded.
2. Servlet instance is created.
3. init method is invoked.
4. service method is invoked.
5. destroy method is invoked.



There are three states of a servlet: new, ready and end. The servlet is in new state if servlet instance is created. After invoking the init() method, Servlet comes in the ready state. In the ready state, servlet performs all the tasks. When the web container invokes the destroy() method, it shifts to the end state.

### **1) Servlet class is loaded**

The classloader is responsible to load the servlet class. The servlet class is loaded when the first request for the servlet is received by the web container.

### **2) Servlet instance is created**

The web container creates the instance of a servlet after loading the servlet class. The servlet instance is created only once in the servlet life cycle.

### **3) init method is invoked**

The web container calls the init method only once after creating the servlet instance. The init method is used to initialize the servlet. It is the life cycle method of the javax.servlet.Servlet interface. Syntax of the init method is given below:

1. **public** **void** init(ServletConfig config) **throws** ServletException

### **4) service method is invoked**

The web container calls the service method each time when request for the servlet is received. If servlet is not initialized, it follows the first three steps as described above then calls the service method. If servlet is initialized, it calls the service method. Notice that servlet is initialized only once. The syntax of the service method of the Servlet interface is given below:

1. **public** **void** service(ServletRequest request, ServletResponse response)

**throws** SevletException, IOException

### **5) destroy method is invoked**

The web container calls the destroy method before removing the servlet instance from the service. It gives the servlet an opportunity to clean up any resource for example memory, thread etc. The syntax of the destroy method of the Servlet interface is given below:

1. **public** **void** destroy()

**Steps to create a servlet example**

6 steps to create a **servlet example**. These steps are required for all the servers.

The servlet example can be created by three ways:

1. By implementing Servlet interface,
2. By inheriting GenericServlet class, (or)
3. By inheriting HttpServlet class

 To create a servlet the class must extend the HttpServlet class and override at least one of its methods (doGet, doPost, doDelete, doPut).

### **Methods of HttpServlet Class**

### 1**. doGet() Method**

* This method is used to handle the GET request on the server-side.
* This method also automatically supports HTTP HEAD (HEAD request is a GET request which returns nobody in response ) request.
* The GET type request is usually used to *preprocess*a request.

**Syntax:**

protected void doGet(HttpServletRequest request,HttpServletResponse response) throws ServletException,IOException

**Parameters:**

* **request –** an HttpServletRequest object that contains the request the client has made of the servlet.
* **response –** an HttpServletResponse object that contains the response the servlet sends to the client.

**Exceptions:**

* **IOException –** if an input or output error is detected when the servlet handles the GET request.
* **ServletException –** Use to handle the GET request.

### 2. **doPost() Method**

* This method is used to handle the POST request on the server-side.
* This method allows the client to send data of unlimited length to the webserver at a time.
* The POST type request is usually used to post-process a request.

**Syntax:**

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,IOException

**Parameter:**

* **request –**an HttpServletRequest object that contains the request the client has made of the servlet.
* **response –** an HttpServletResponse object that contains the response the servlet sends to the client.

**Throws:**

* **IOException –**if an input or output error is detected when the servlet handles the POST request.
* **ServletException –**Use to handle the POST request.

### 3. **doHead() Method**

* This method is overridden to handle the HEAD request.
* In this method, the response contains the only header but does not contain the message body.
* This method is used to improve performance (avoid computing response body).

**Syntax:**

protected void doHead(HttpServletRequest request, HttpServletResponse response) throws ServletException,IOException

**Parameters:**

* **request –** an HttpServletRequest object that contains the request the client has made of the servlet.
* **response –** an HttpServletResponse object that contains the response the servlet sends to the client.

**Exceptions:**

* **IOException –**if an input or output error is detected when the servlet handles the HEAD request.
* **ServletException –** Use to handle the HEAD request.

### 4**. doPut() Method**

* This method is overridden to handle the PUT request.
* This method allows the client to store the information on the server(to save the image file on the server).
* This method is called by the server (via the service method) to handle a PUT request.

**Syntax:**

protected void doPut(HttpServletRequest request, HttpServletResponse response) throws ServletException,IOException

**Parameters:**

* **request –** an HttpServletRequest object that contains the request the client has made of the servlet.
* **response –**an HttpServletResponse object that contains the response the servlet sends to the client.

**Throws:**

* **IOException –**if an input or output error is detected when the servlet handles the PUT request.
* **ServletException –**Use to handle the PUT request.

### 5. **doDelete() Method**

* This method is overridden to handle the DELETE request.
* This method allows a client to remove a document or Web page from the server.
* While using this method, it may be useful to save a copy of the affected URL in temporary storage to avoid data loss.

**Syntax:**

protected void doDelete(HttpServletRequest request, HttpServletResponse response) throws ServletException,IOException

**Parameters:**

* **request –**an HttpServletRequest object that contains the request the client has made of the servlet.
* **response –** an HttpServletResponse object that contains the response the servlet sends to the client.

**Exceptions:**

* **IOException –** if an input or output error is detected when the servlet handles the DELETErequest.
* **ServletException –**Use to handle the DELETE request.

### 6**.** **service() Method**

This method receives standard HTTP requests from the public *service*method and dispatches them to the *doXXX*methods defined in this class.

**Syntax:**

protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException,IOException

**Parameters:**

* **request –** an HttpServletRequest object that contains the request the client has made of the servlet.
* **response –** an HttpServletResponse object that contains the response the servlet sends to the client.

**Exceptions:**

* **IOException –** if an input or output error is detected when the servlet handles the HTTP request.
* **ServletException –** Use to handle the HTTP request.

<https://www.javatpoint.com/creating-servlet-in-myeclipse-ide>

https://www.studytonight.com/servlet/creating-servlet-in-eclipse.php

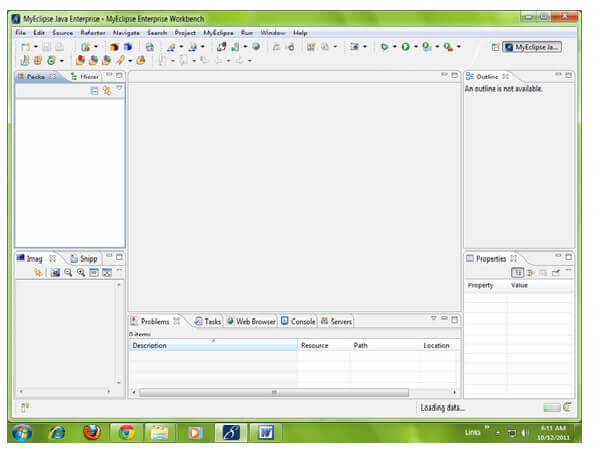
# Creating Servlet in Eclipse IDE

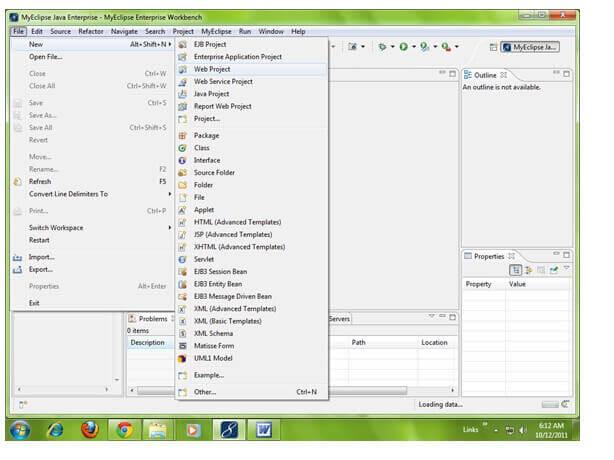
|  |
| --- |
| You need to follow the following steps to create the servlet in the Eclipse IDE. The steps are as follows: |

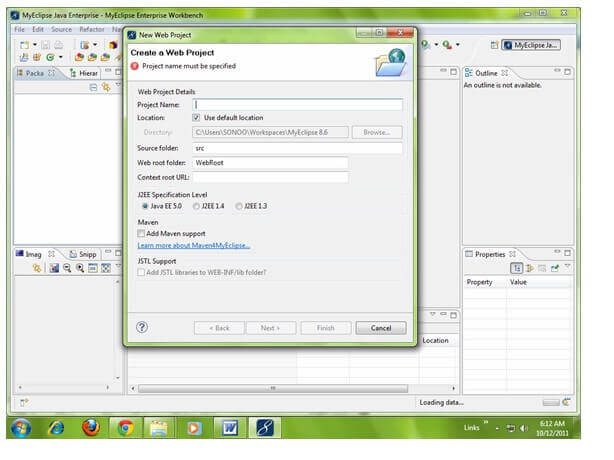
* Create a web project
* create a html file
* create a servlet
* start tomcat server and deploy project

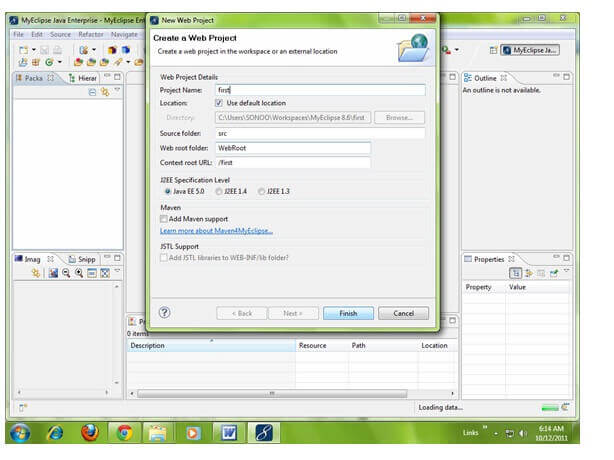
### **1) Create the web project:**

|  |
| --- |
| For creating a web project click on File Menu -> New -> web project -> write your project name e.g. first -> Finish. |



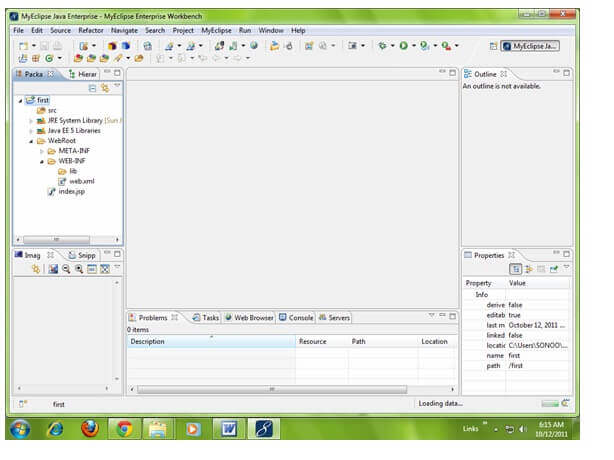




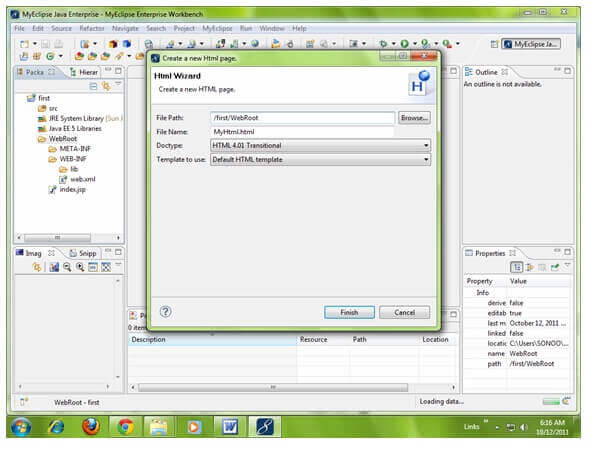


### **2) Create the html file:**

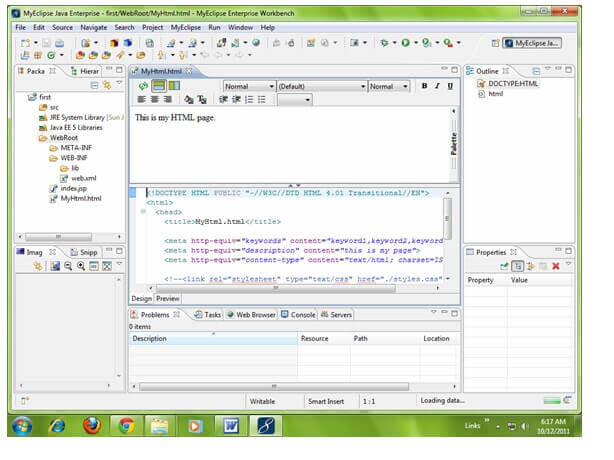
|  |
| --- |
| As you can see that a project is created named first. Now let's explore this project. |

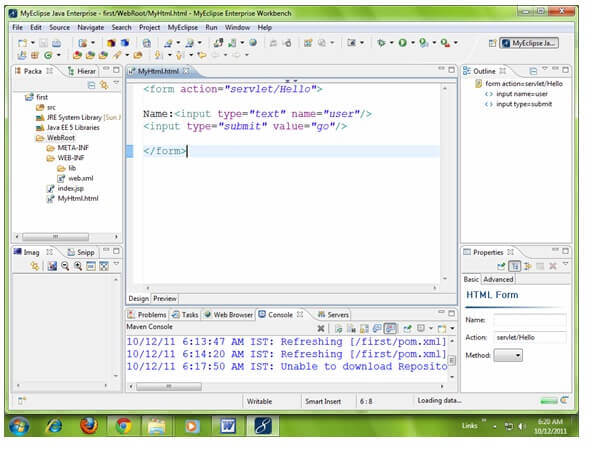


|  |
| --- |
| For creating a html file, right click on WebRoot -> New -> html -> write your html file name e.g. MyHtml.html -> Finish. |



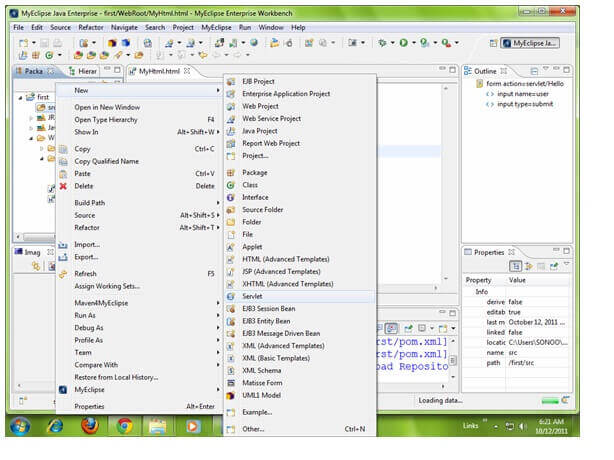
|  |
| --- |
| As you can see that a html file is created named MyHtml.html. Now let's write the html code here. |

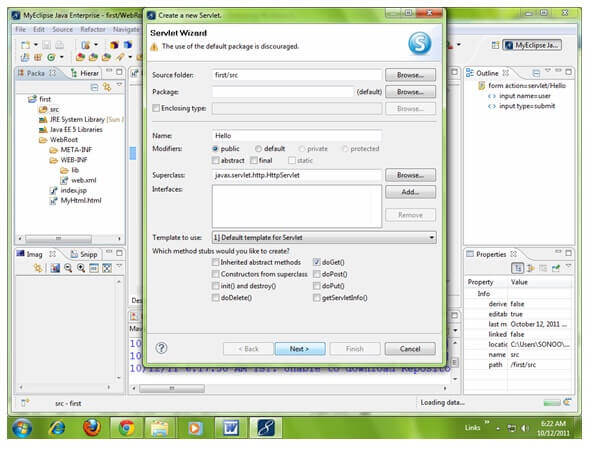


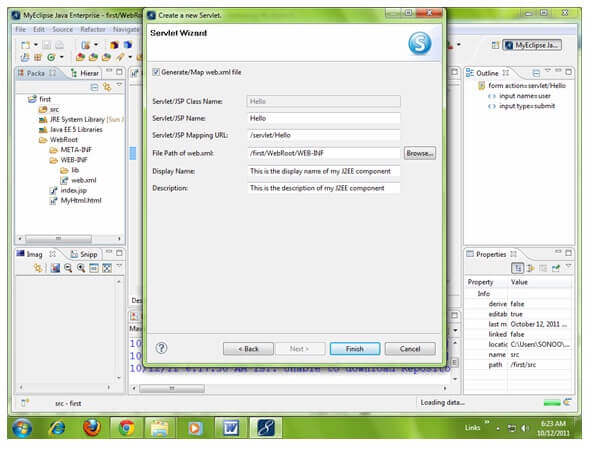


### **3) Create the servlet:**

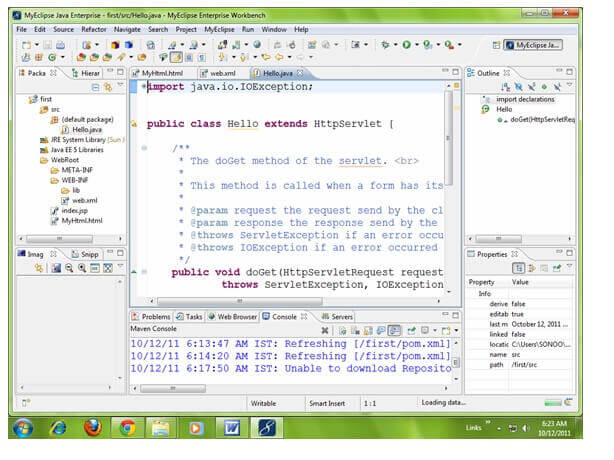
|  |
| --- |
| For creating a servlet click on File Menu -> New -> servlet -> write your servlet name e.g. Hello -> uncheck all the checkboxes except doGet() -> next -> Finish. |

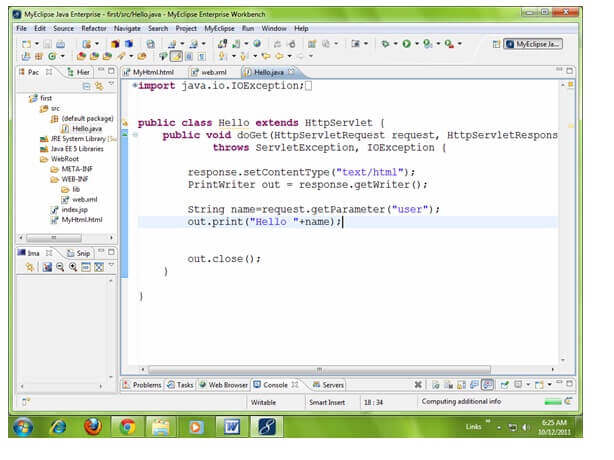




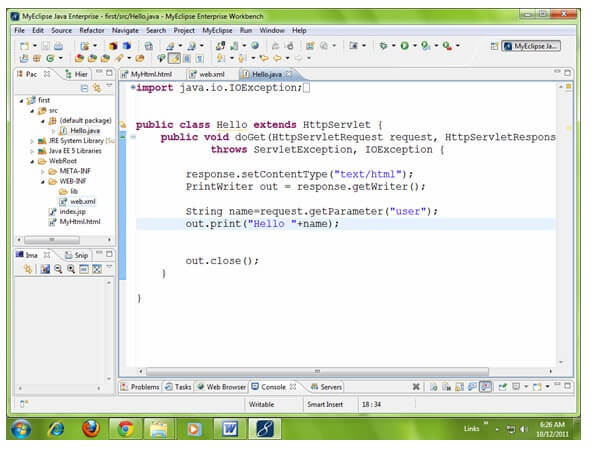


|  |
| --- |
| As you can see that a servlet file is created named Hello.java. Now let's write the servlet code here. |

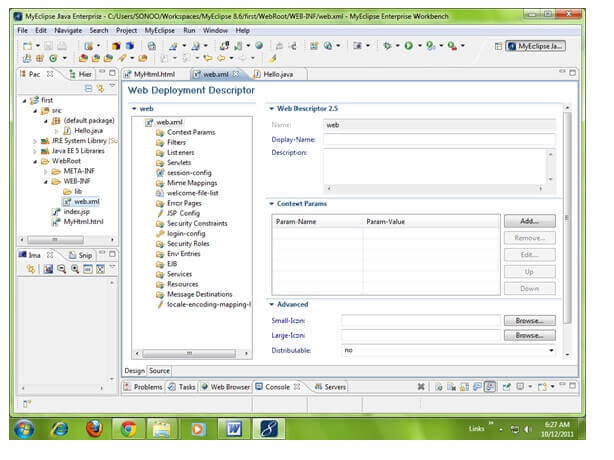




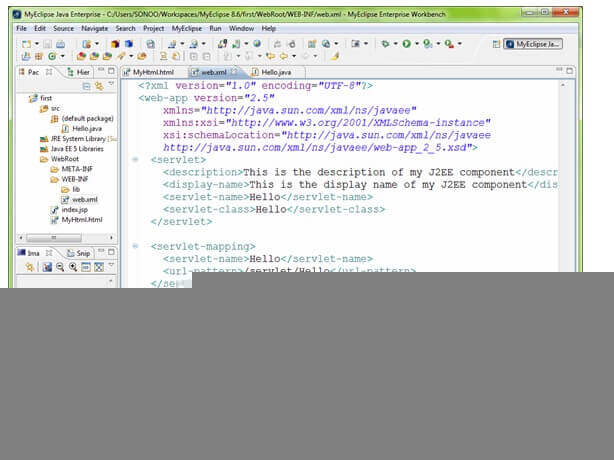
|  |
| --- |
| Now let's make the MyHtml.html file as the default page of our project. For this, open web.xml file and change the welcome file name as MyHtml.html in place of index.jsp. |

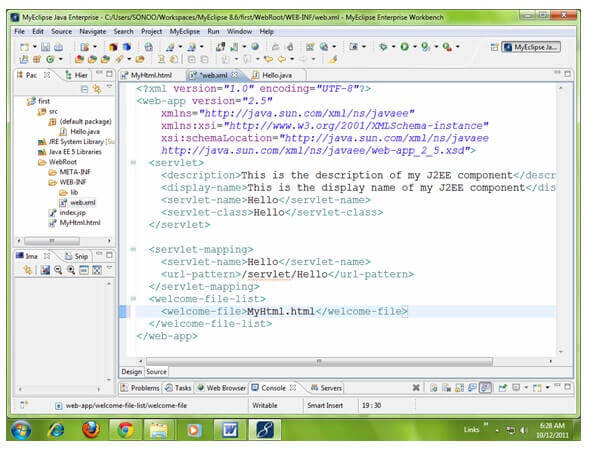


|  |
| --- |
| Click on the source tab to see the source code. |



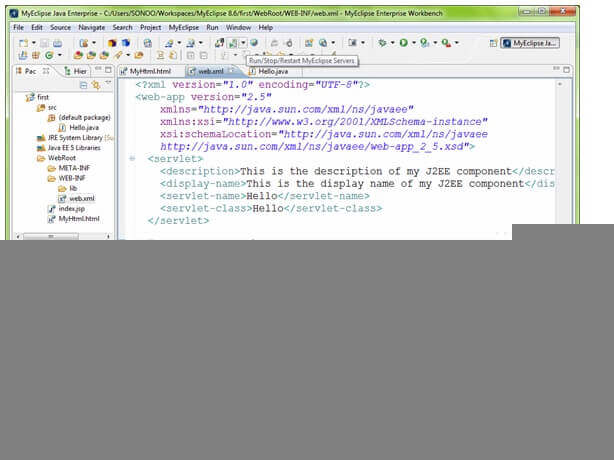
|  |
| --- |
| Now change the welcome file as MyHtml.html in place of index.jsp. |

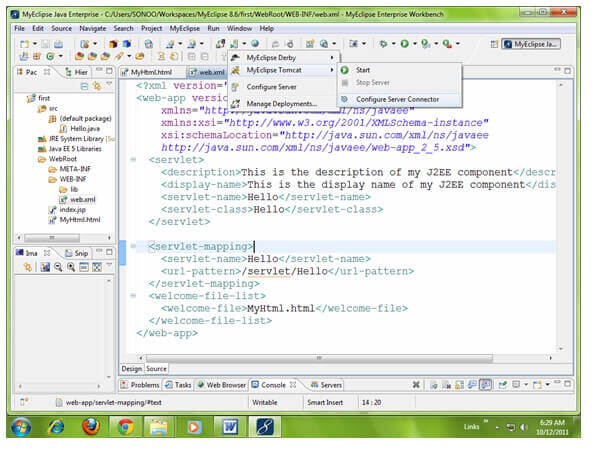




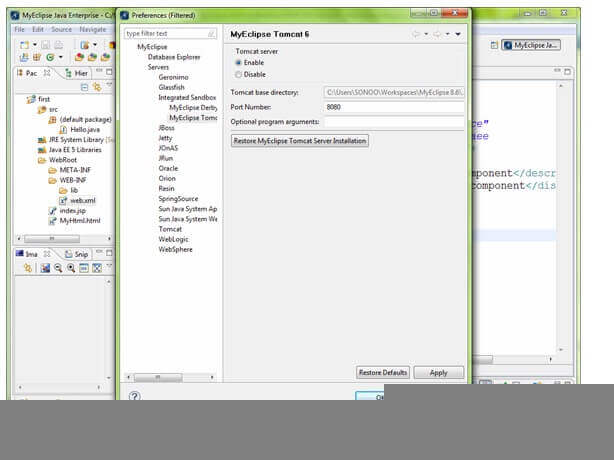
### **4) Start the server and deploy the project:**

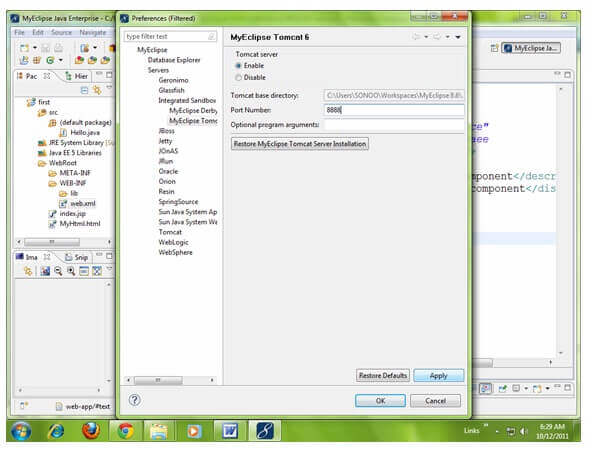
|  |
| --- |
| For starting the server and deploying the project in one step Right click on your project -> Run As -> MyEclipse server application. |
| The default port of myeclipse tomcat is 8080, if you have installed oracle on your system, the port no. will conflict so let's first change the port number of myeclipse tomcat server. For changing the port number click on the start server icon at the left hand side of browser icon -> myeclipse tomcat -> Configure server connector -> change the port number as 8888 in place of 8080 -> apply -> ok. |



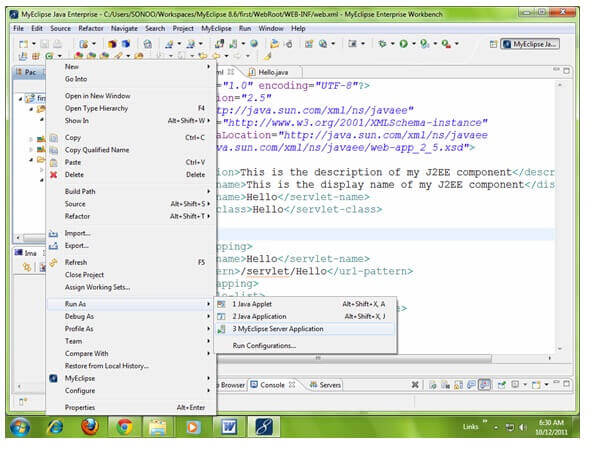


|  |
| --- |
| Now change the port number as 8888 in place of 8080 -> apply -> ok. |

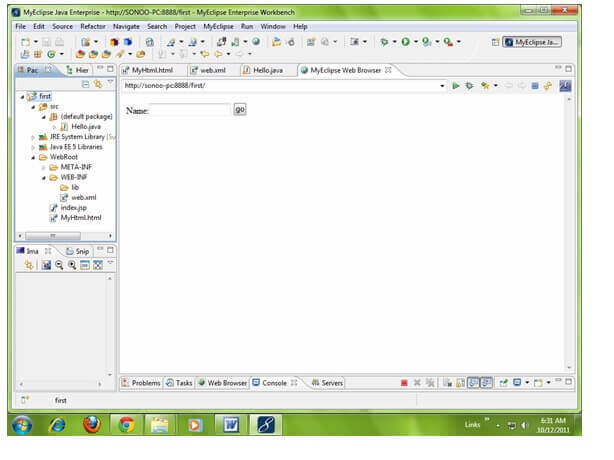


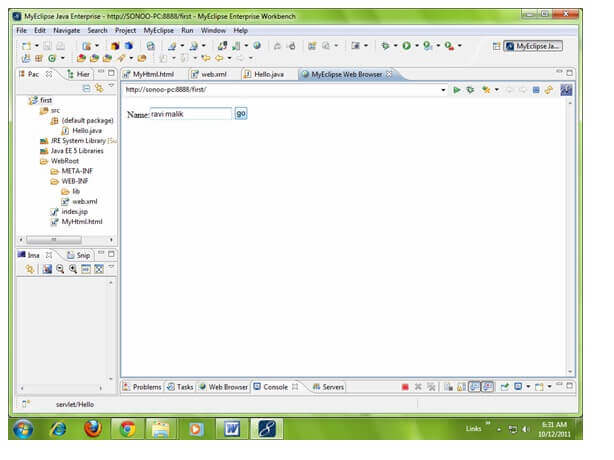


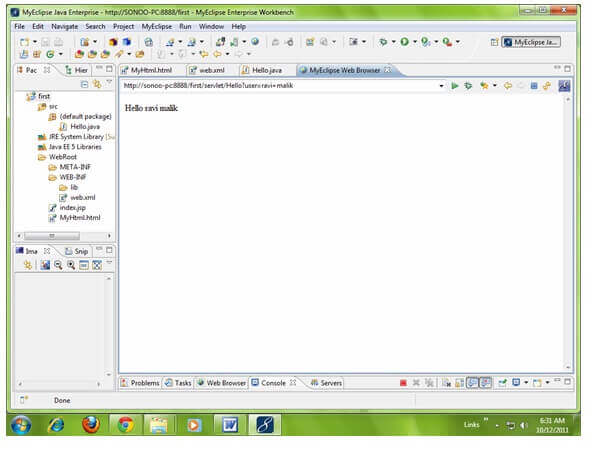
|  |
| --- |
| Now port number have been changed. For starting the server Right click on your project -> Run As -> MyEclipse server application. |



|  |
| --- |
| As you can see that default page of your project is open, write your name -> go. |







**Hello.java**

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class Hello extends HttpServlet

{

public void doGet(HttpServletRequest request, HttpServletResponse response)throws ServletException, IOException

{

response.setContentType("text/html");

PrintWriter out = response.getWriter();

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

out.print("Hello "+name);

out.close();

}

}

**MyHtml.html**

|  |  |
| --- | --- |
|  | <form action="servlet/Hello"> |
|  |  |
|  | Name:<input type="text" name="user"/> |
|  | <input type="submit" value="go"/> |
|  |  |
|  | </form> |

1. **Sevlet coding for Login page:**

**Index.html**

<html><body>

<form action="welcome" method="get" align="center">

Enter your name<input type="text" name="name"> <br>

Enter you password<input type="password" name="password"> <br>

<input type="submit" value="login">

</form>

</html>

**LoginServlet.Java**

import javax.servlet.http.\*;

import javax.servlet.\*;

import java.io.\*;

public class LoginServlet extends HttpServlet

{

public void doGet(HttpServletRequest req,HttpServletResponse res)throws ServletException,IOException

{

res.setContentType("text/html");

PrintWriter pw=res.getWriter();

String name=req.getParameter("name");

System.out.println("Name is "+name);

String password=req.getParameter("password");

System.out.println("Password is "+password);

pw.println("Welcome :-"+name +",with password:-"+ password);

}

}

**web.xml**

<web-app>

<servlet>

<servlet-name>anand</servlet-name>

<servlet-class>LoginServlet</servlet-class>

</servlet>

<servlet-mapping>

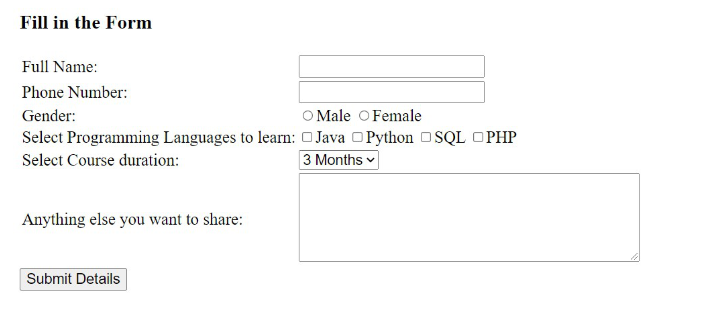
<servlet-name>anand</servlet-name>

<url-pattern>/welcome</url-pattern>

</servlet-mapping>

</web-app>

1. **Form submission :**



<!DOCTYPE html>

<**html**>

<**head**>

<**meta** charset="ISO-8859-1">

<**title**>User Details</**title**>

</**head**>

<**body** >

    <**h3**>Fill in the Form</**h3**>

    <**form** action="FormData" method="post">

        <**table**>

            <**tr**>

                <**td**>Full Name:</**td**>

                <**td**><**input** type="text" name="name" /></**td**>

            </**tr**>

            <**tr**>

                <**td**>Phone Number:</**td**>

                <**td**><**input** type="text" name="phone" /></**td**>

            </**tr**>

            <**tr**>

                <**td**>Gender:</**td**>

                <**td**><**input** type="radio" name="gender" value="male" />Male

                    <**input** type="radio" name="gender" value="female" />Female</**td**>

            </**tr**>

            <**tr**>

                <**td**>Select Programming Languages to learn:</**td**>

                <**td**><**input** type="checkbox" name="language" value="java" />Java

                    <**input** type="checkbox" name="language" value="python" />Python

                    <**input** type="checkbox" name="language" value="sql" />SQL

                    <**input** type="checkbox" name="language" value="php" />PHP</**td**>

            </**tr**>

            <**tr**>

                <**td**>Select Course duration:</**td**>

                <**td**><**select** name="duration">

                    <**option** value="3months">3 Months</**option**>

                    <**option** value="6months">6 Months</**option**>

                    <**option** value="9months">9 Months</**option**></**select**></**td**>

            </**tr**>

            <**tr**>

                <**td**>Anything else you want to share:</**td**>

                <**td**><**textarea** rows="5" cols="40" name="comment"></**textarea**></**td**>

            </**tr**>

        </**table**>

        <**input** type="submit" value="Submit Details">

    </**form**>

</**body**>

</**html**>

**import java.io.IOException;**

**import java.io.PrintWriter;**

**import javax.servlet.ServletException;**

**import javax.servlet.annotation.WebServlet;**

**import javax.servlet.http.HttpServlet;**

**import javax.servlet.http.HttpServletRequest;**

**import javax.servlet.http.HttpServletResponse;**

**// Servlet implementation class FormDataHandle**

**// Annotation to map the Servlet URL**

**@WebServlet("/FormData")**

**public class FormDataHandle extends HttpServlet {**

**private static final long serialVersionUID = 1L;**

**// Auto-generated constructor stub**

**public FormDataHandle() {**

**super();**

**}**

**// HttpServlet doPost(HttpServletRequest request, HttpServletResponse response) method**

**protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {**

**// Get the values from the request using 'getParameter'**

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

**String phNum = request.getParameter("phone");**

**String gender = request.getParameter("gender");**

**// To get all the values selected for**

**// programming language, use 'getParameterValues'**

**String progLang[] = request.getParameterValues("language");**

**// Iterate through the String array to**

**// store the selected values in form of String**

**String langSelect = "";**

**if(progLang!=null){**

**for(int i=0;i<progLang.length;i++){**

**langSelect= langSelect + progLang[i]+ ", ";**

**}**

**}**

**String courseDur = request.getParameter("duration");**

**String comment = request.getParameter("comment");**

**// set the content type of response to 'text/html'**

**response.setContentType("text/html");**

**// Get the PrintWriter object to write**

**// the response to the text-output stream**

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

**// Print the data**

**out.print("<html><body>");**

**out.print("<h3>Details Entered</h3><br/>");**

**out.print("Full Name: "+ name + "<br/>");**

**out.print("Phone Number: "+ phNum +"<br/>");**

**out.print("Gender: "+ gender +"<br/>");**

**out.print("Programming languages selected: "+ langSelect +"<br/>");**

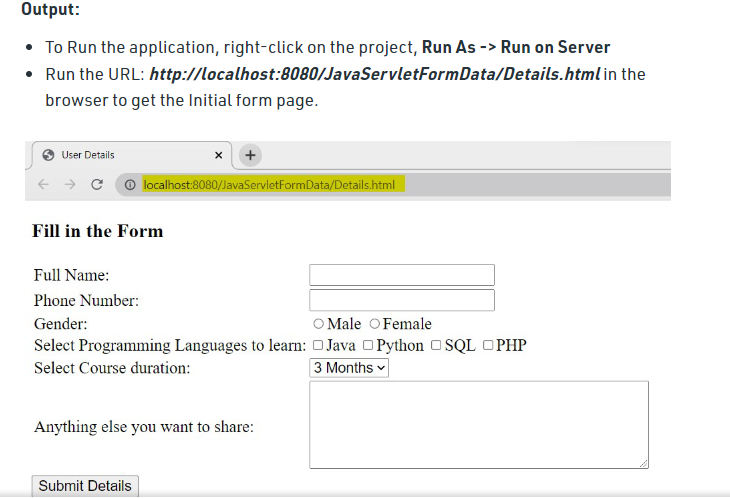
**out.print("Duration of course: "+ courseDur+"<br/>");**

**out.print("Comments: "+ comment);**

**out.print("</body></html>");**

**}**

**}**



Graphical user interface, text, application

Description automatically generated