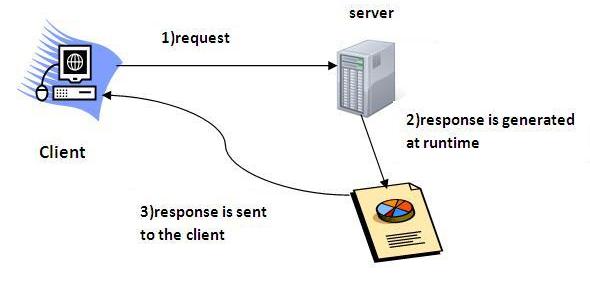
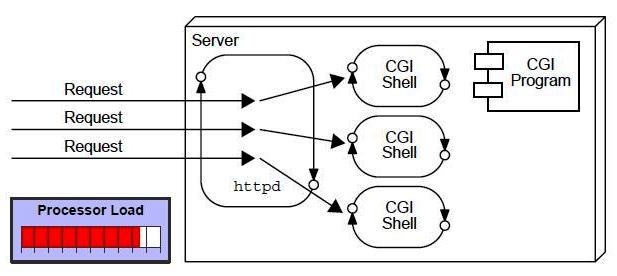
**Servlet**

* Servlet is a technology which is used to create a web application (resides at server side and generates a dynamic web pages)
* Servlet is an API, an interface, a class and a web component depending on the context.

****

CGI (Common Gateway Interface)

****

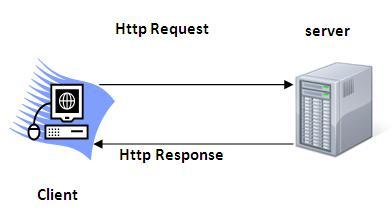
### Advantages of Servlet

1. **Better performance:** because it creates a thread for each request, not process.
2. **Portability:** because it uses Java language.
3. **Robust:** JVM manages Servlets, so we don't need to worry about the memory leak, garbage collection, etc.
4. **Secure**

### H:\servlet\3.JPG

**Hyper Text Transfer Protocol**

It is the data communication protocol used to establish communication between the client and server.

****

**Http Request Methods**

|  |  |
| --- | --- |
| **HTTP Request** | **Description** |
| **GET** | Asks to get the resource at the requested URL. |
| **POST** | Asks the server to accept the body info attached. It is like GET request with extra info sent with the request. |
| **HEAD** | Asks for only the header part of whatever a GET would return. Just like GET but with no body. |
| **TRACE** | Asks for the loopback of the request message, for testing or troubleshooting. |
| **PUT** | Says to put the enclosed info (the body) at the requested URL. |
| **DELETE** | Says to delete the resource at the requested URL. |
| **DELETE** | Asks for a list of the HTTP methods to which the thing at the request URL can respond |

**Design of Servlet class Hierarchy**

Servlet

ServletConfig

Serializable

HttpServlet

GenericServlet

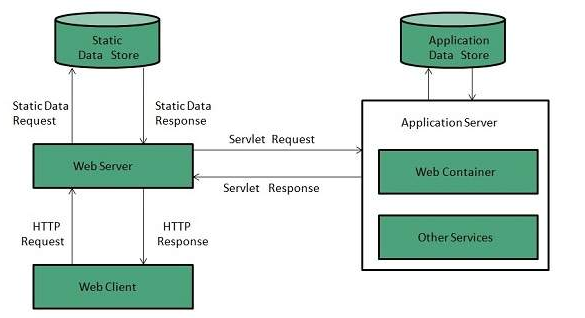
UserDefinedServlet

Interfaces

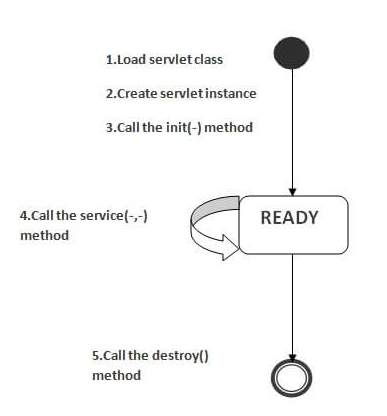
Built-In Classes

User defined class

**Web Server Working**

****

**Servlet Life Cycle**

****

**Steps to create a servlet**

Here, we are going to use **apache tomcat server** in this example. The steps are as follows:

1. Create a directory structure
2. Create a Servlet
3. Compile the Servlet
4. Create a deployment descriptor
5. Start the server and deploy the project
6. Access the servlet
7. **Create a directory structures**
8. **2** Create a Servlet
9. **Compile the servlet**
10. **Create the deployment descriptor (web.xml file)**
11. **Start the Server and deploy the project**

To start Apache Tomcat server, double click on the startup.bat file under apache-tomcat/bin directory.

### One Time Configuration for Apache Tomcat Server

You need to perform 2 tasks:

1. set JAVA\_HOME or JRE\_HOME in environment variable (It is required to start server).
2. Change the port number of tomcat (optional). It is required if another server is running on same port (8080).

### 1) How to set JAVA\_HOME in environment variable?

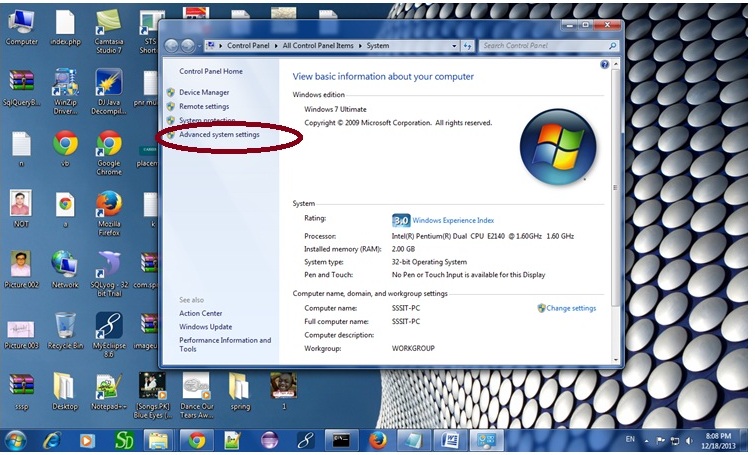
To start Apache Tomcat server JAVA\_HOME and JRE\_HOME must be set in Environment variables.

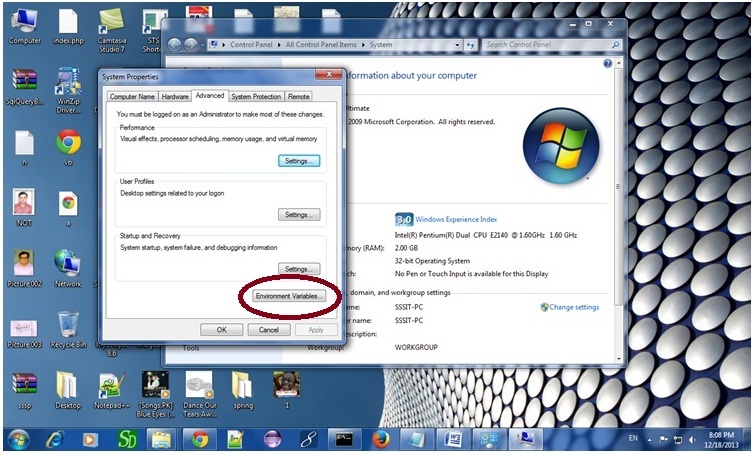
Go to My Computer properties -> Click on advanced tab then environment variables -> Click on the new tab of user variable -> Write JAVA\_HOME in variable name and paste the path of jdk folder in variable value -> ok -> ok -> ok.

Go to My Computer properties:

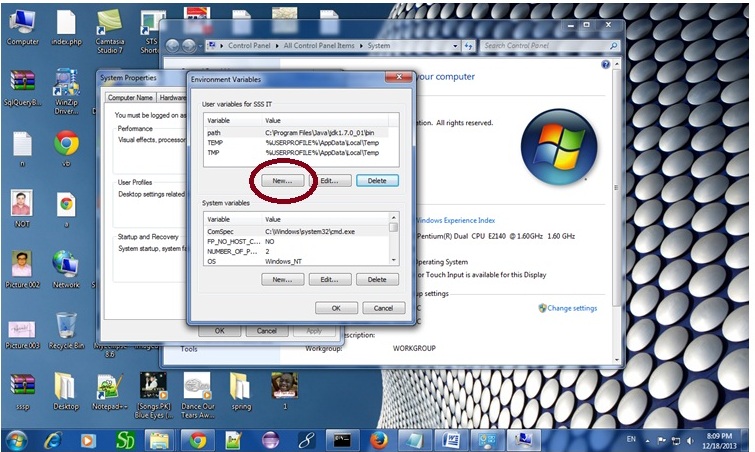


Click on advanced system settings tab then environment variables:

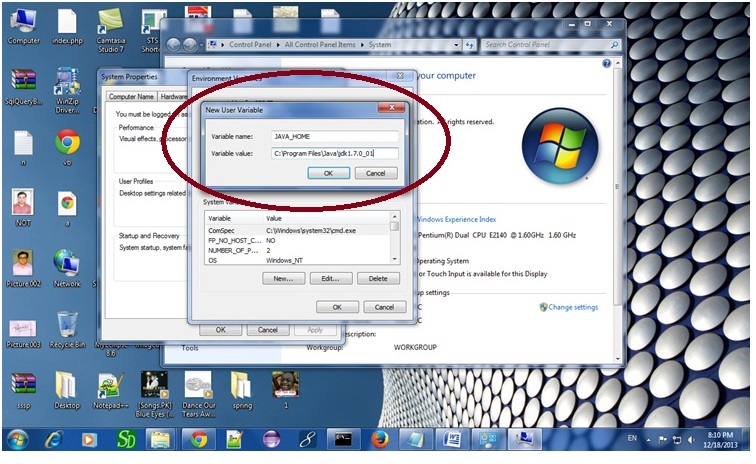




Click on the new tab of user variable or system variable:

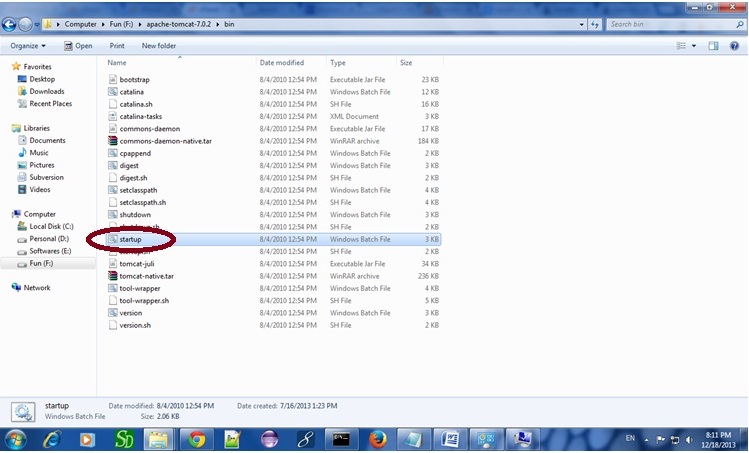


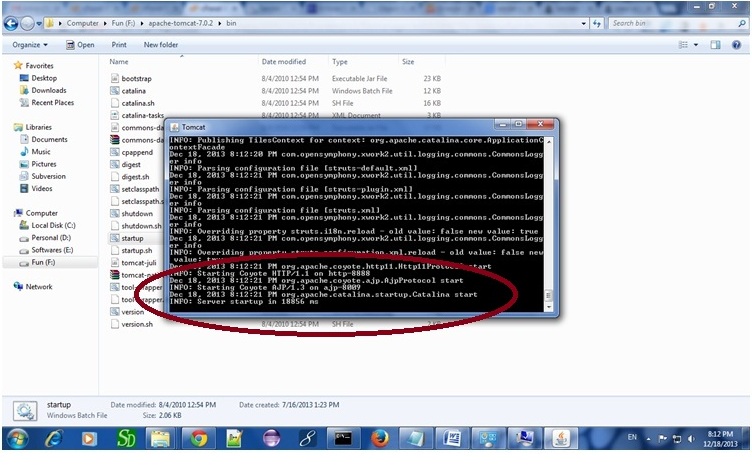
Write JAVA\_HOME in variable name and paste the path of jdk folder in variable value:



There must not be semicolon (;) at the end of the path.

|  |
| --- |
| After setting the JAVA\_HOME double click on the startup.bat file in apache tomcat/bin. |
| Note: There are two types of tomcat available:   1. Apache tomcat that needs to extract only (no need to install) 2. Apache tomcat that needs to install |
| It is the example of apache tomcat that needs to extract only. |





Now server is started successfully.

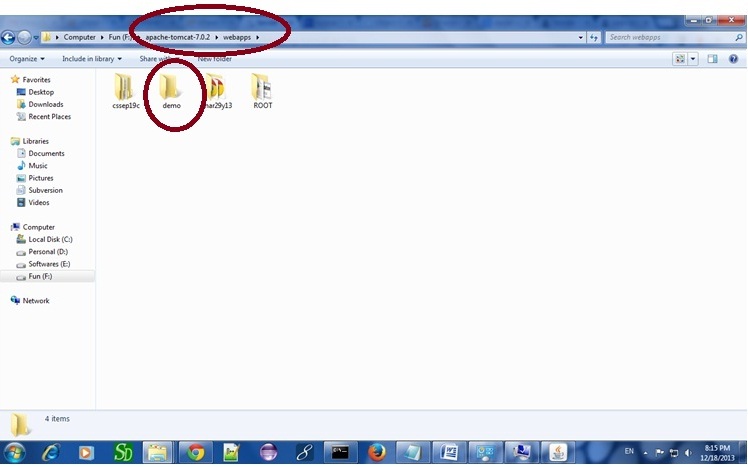
### 2) How to change port number of apache tomcat

Changing the port number is required if there is another server running on the same system with same port number.Suppose you have installed oracle, you need to change the port number of apache tomcat because both have the default port number 8080.

Open **server.xml file** in notepad. It is located inside the **apache-tomcat/conf** directory . Change the Connector port = 8080 and replace 8080 by any four digit number instead of 8080. Let us replace it by 9999 and save this file.

### 5) How to deploy the servlet project

Copy the project and paste it in the webapps folder under apache tomcat.



But there are several ways to deploy the project. They are as follows:

* By copying the context(project) folder into the webapps directory
* By copying the war folder into the webapps directory
* By selecting the folder path from the server
* By selecting the war file from the server

Here, we are using the first approach.

You can also create war file, and paste it inside the webapps directory. To do so, you need to use jar tool to create the war file. Go inside the project directory (before the WEB-INF), then write:

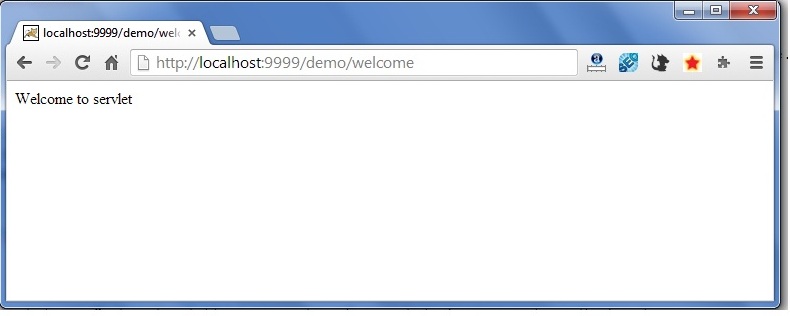
1. projectfolder> jar cvf myproject.war \*

Creating war file has an advantage that moving the project from one location to another takes less time.

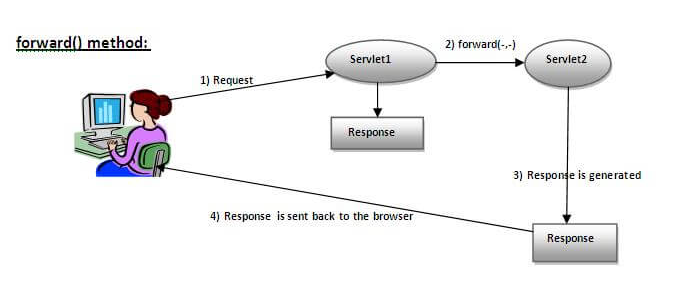
### 6) How to access the servlet

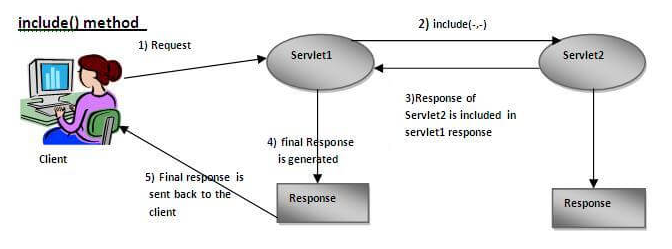
Open broser and write http://hostname:portno/contextroot/urlpatternofservlet. For example:

1. http://localhost:9999/demo/welcome

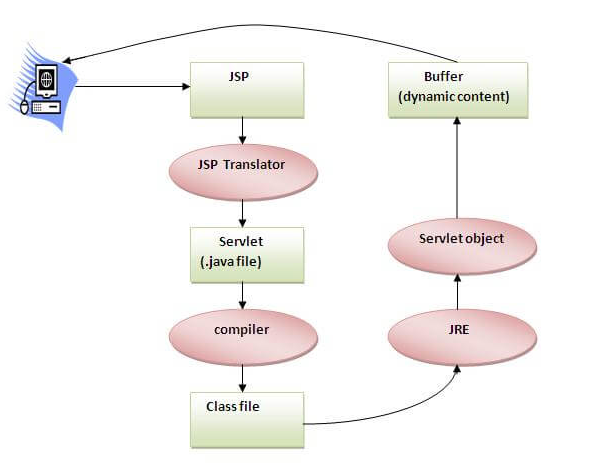


**RequestDispatcher in Servlet**

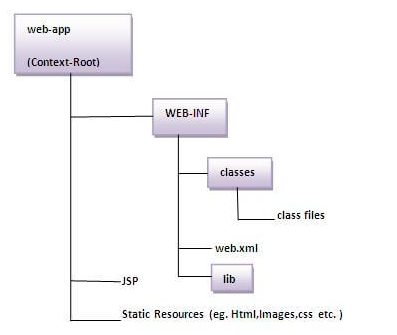
****

****

**The Lifecycle of a JSP Page**

****

**The Directory structure of JSP**

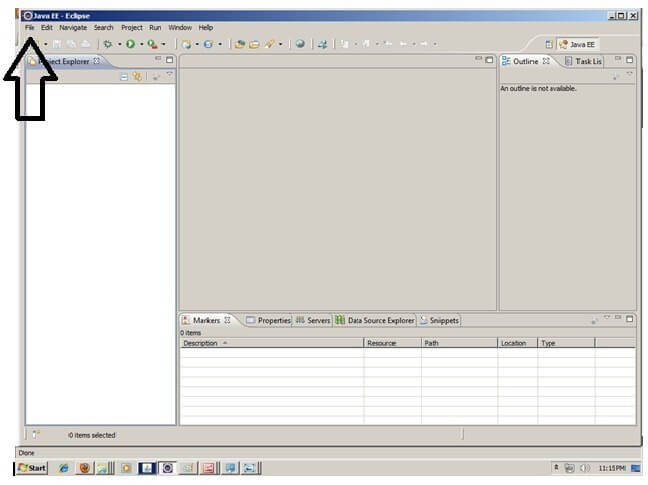
****

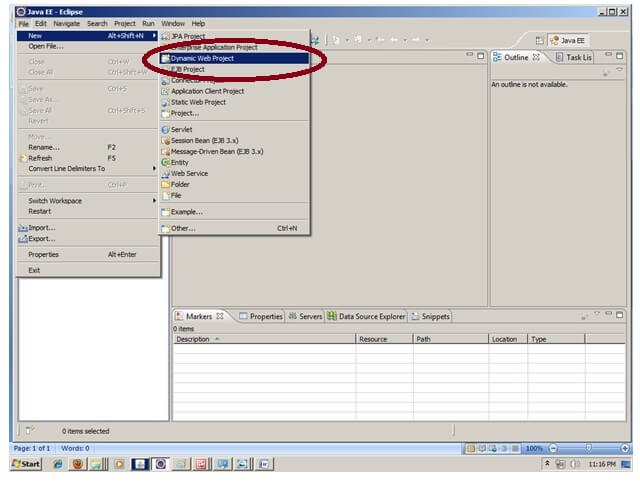
**Creating JSP in Eclipse IDE with Tomcat server**

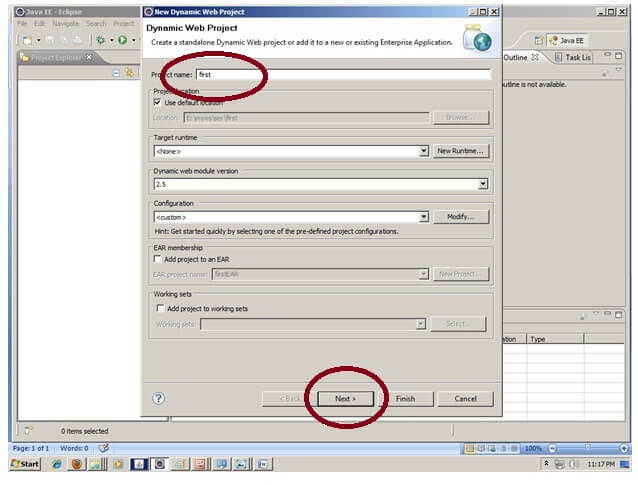
* Create a Dynamic web project
* create a jsp
* start tomcat server and deploy the project

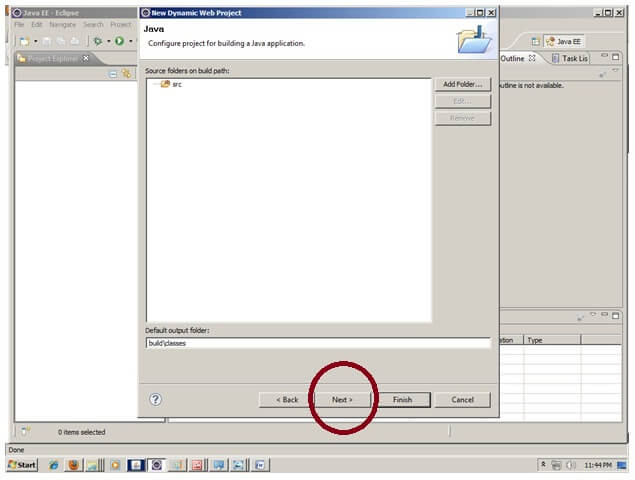
**Create the dynamic web project**

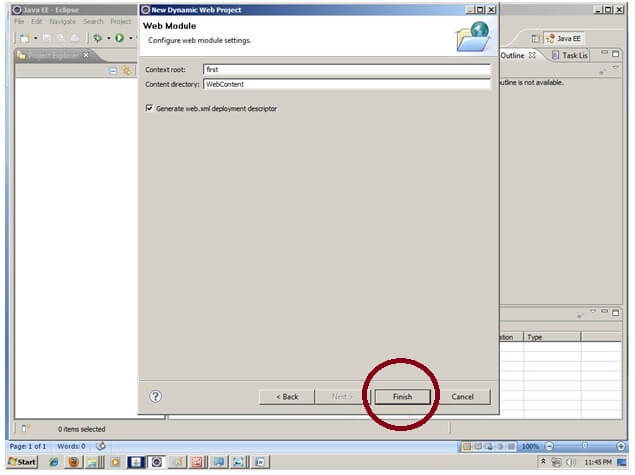
For creating a dynamic web project click on File Menu -> New -> dynamic web project -> write your project name e.g. first -> Finish.

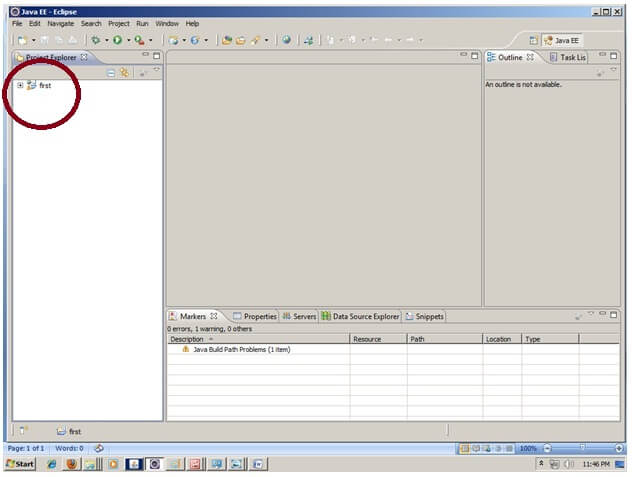






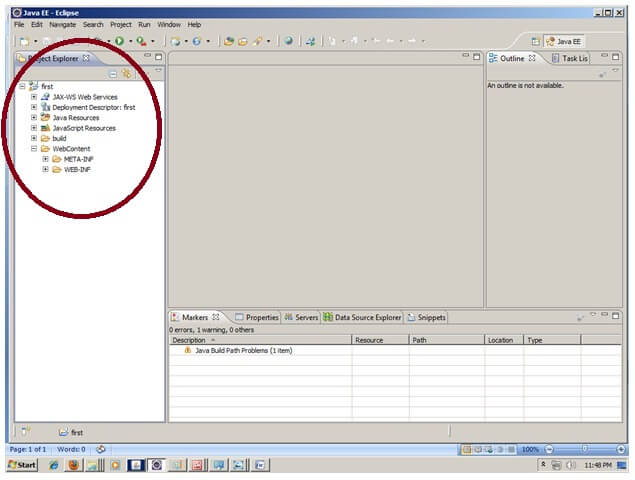


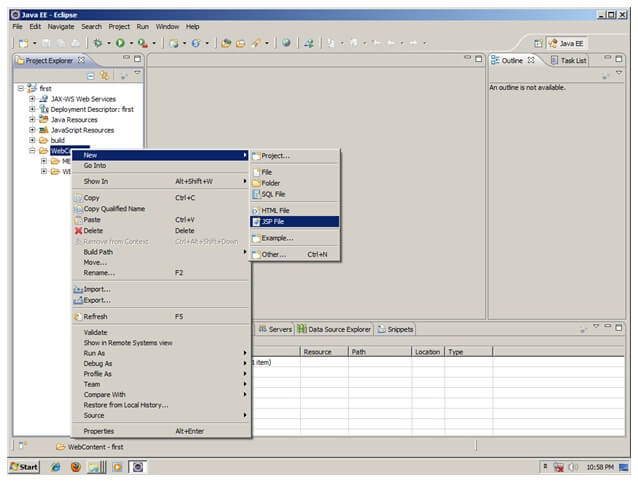


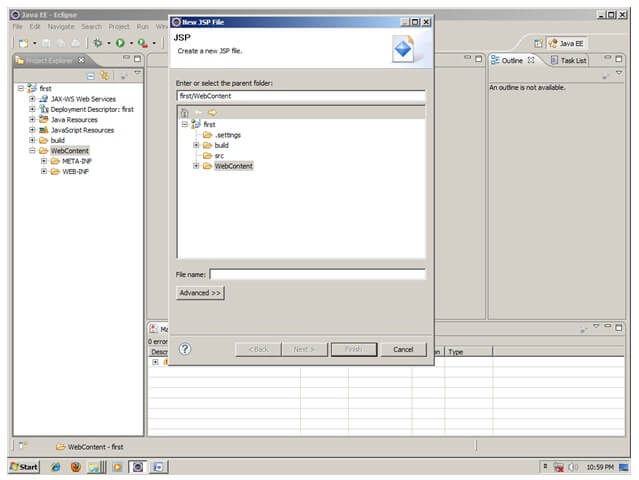


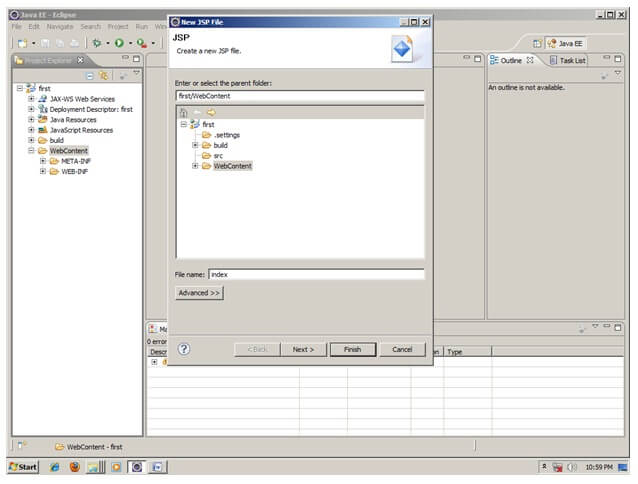
### 2) Create the JSP file in eclipse IDE

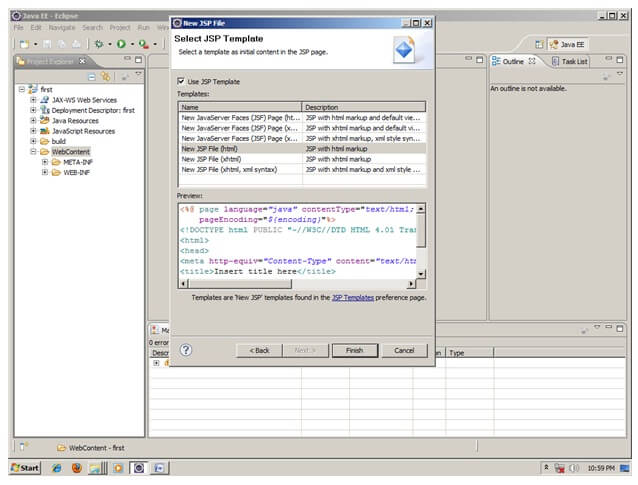
For creating a jsp file explore the project by clicking the + icon -> right click on WebContent -> New -> jsp -> write your jsp file name e.g. index -> next -> Finish.

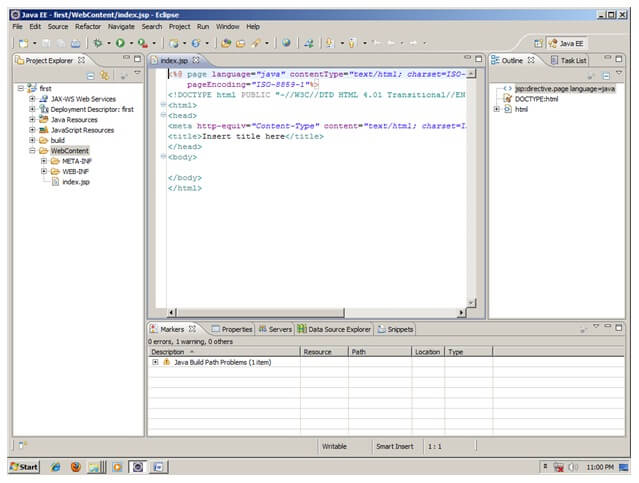




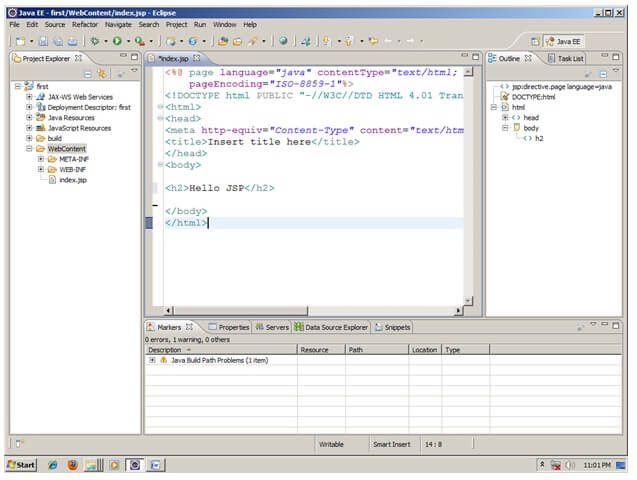








Now JSP file is created, let's write some code.



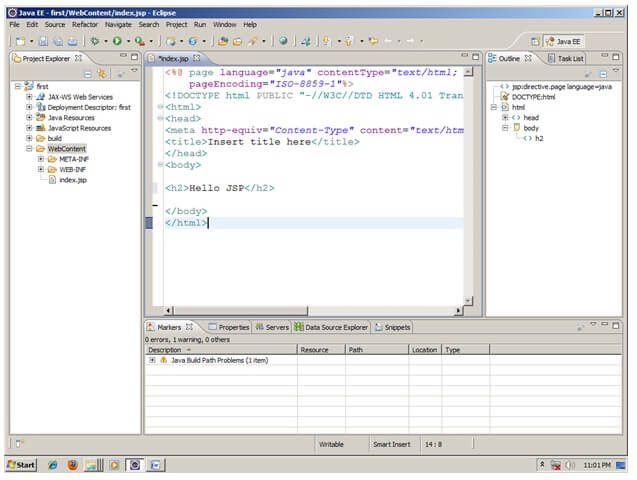
### 3) Start the server and deploy the project:

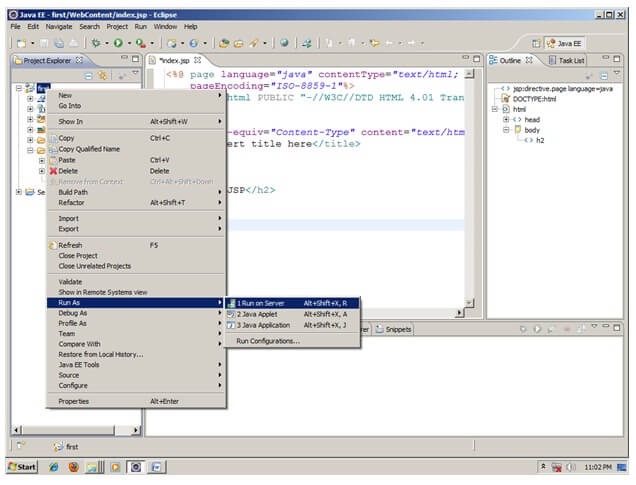
For starting the server and deploying the project in one step Right click on your project -> Run As -> Run on Server -> choose tomcat server -> next -> addAll -> finish.

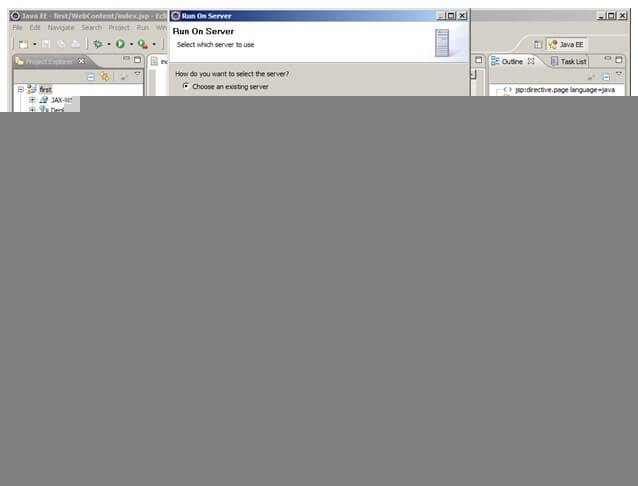
If you are using Eclipse IDE first time, you need to configure the tomcat server First. Click for [How to configure tomcat server in eclipse IDE](https://www.javatpoint.com/how-to-configure-tomcat-server-in-eclipse-ide)

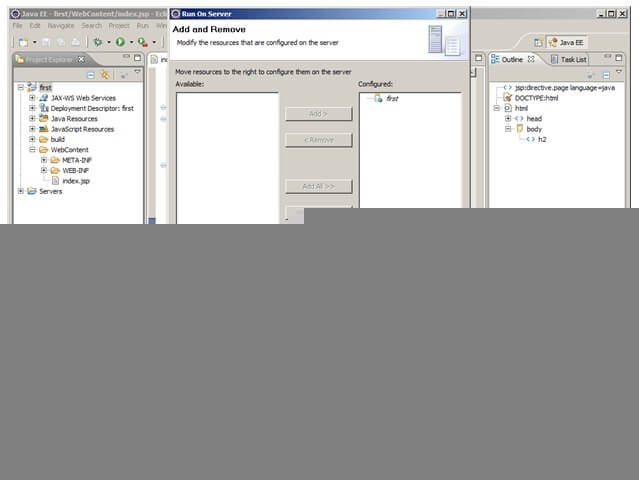
### Now start the tomcat server and deploy project

For starting the server and deploying the project in one step Right click on your project -> Run As -> Run on Server -> choose tomcat server -> next -> addAll -> finish.

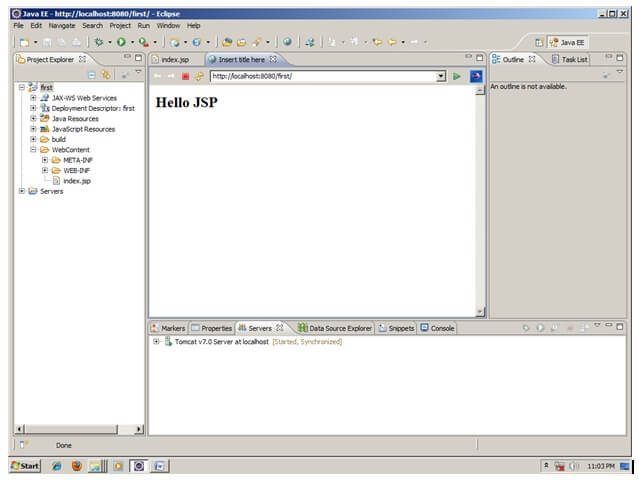




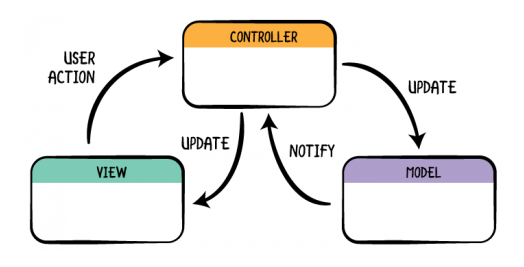




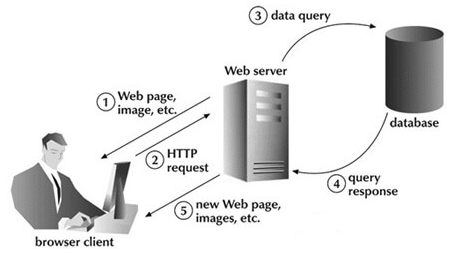
Yes, Let's see JSP is successfully running now.



## Model-View-Controller (MVC)

****

## Synchronous (Classic Web-Application Model)



## Asynchronous (AJAX Web-Application Model)

