

# Servlets | Servlet Tutorial



**Servlet** technology is used to create a web application (resides at server side and generates a dynamic web page).

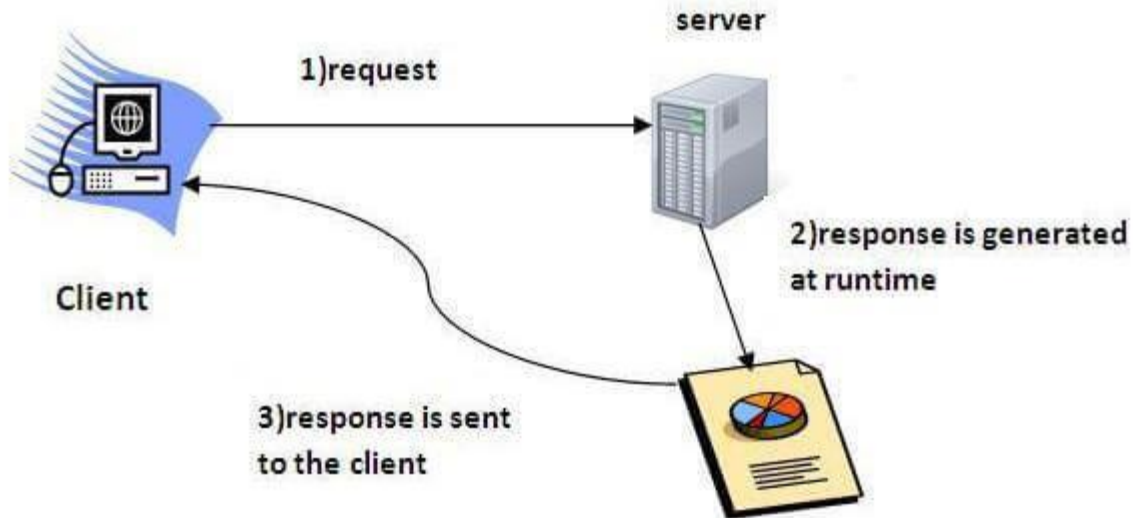
**Servlet** technology is robust and scalable because of java language. Before Servlet, CGI (Common Gateway Interface) scripting language was common as a server-side programming language. However, there were many disadvantages to this technology. We have discussed these disadvantages below.

There are many interfaces and classes in the Servlet API such as Servlet, GenericServlet, HttpServlet, ServletRequest, ServletResponse, etc.

## What is a Servlet?

Servlet can be described in many ways, depending on the context.

- Servlet is a technology which is used to create a web application.
- Servlet is an API that provides many interfaces and classes including documentation.
- Servlet is an interface that must be implemented for creating any Servlet.
- Servlet is a class that extends the capabilities of the servers and responds to the incoming requests. It can respond to any requests.
- Servlet is a web component that is deployed on the server to create a dynamic web page.



Do You Know?

- What is the web application and what is the difference between Get and Post request?
- What information is received by the web server if we request for a Servlet?
- How to run servlet in Eclipse, MyEclipse and Netbeans IDE?
- What are the ways for servlet collaboration and what is the difference between RequestDispatcher and sendRedirect() method?
- What is the difference between ServletConfig and ServletContext interface?
- How many ways can we maintain the state of a user? Which approach is mostly used in web development?
- How to count the total number of visitors and whole response time for a request using Filter?
- How to run servlet with annotation?
- How to create registration form using Servlet and Oracle database?
- How can we upload and download the file from the server?

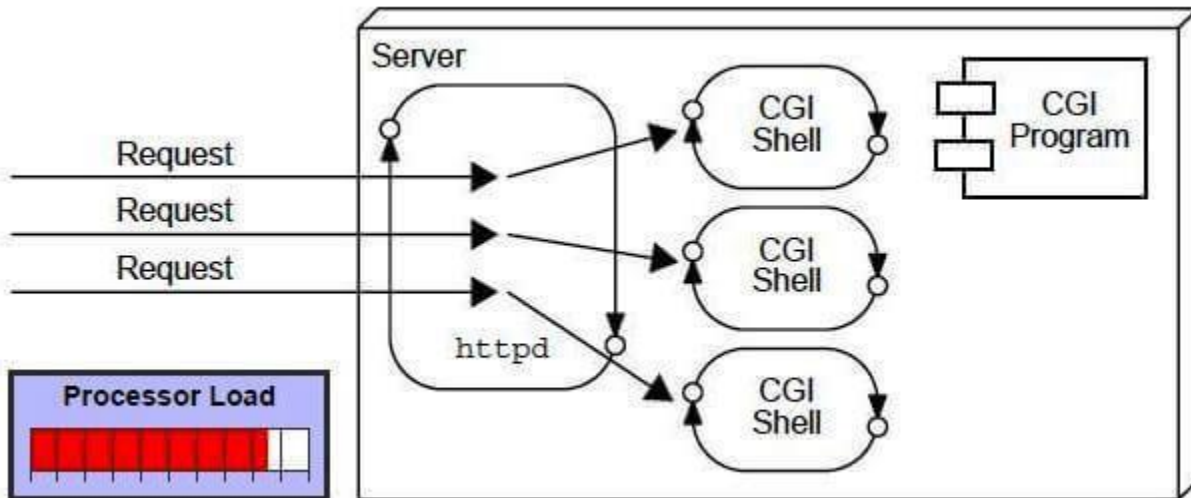
## What is a web application?

A web application is an application accessible from the web. A web application is composed of web components like Servlet, JSP, Filter, etc. and other elements such as HTML, CSS, and JavaScript. The web components typically execute in Web Server and respond to the HTTP request.

---

## CGI (Common Gateway Interface)

CGI technology enables the web server to call an external program and pass HTTP request information to the external program to process the request. For each request, it starts a new process.

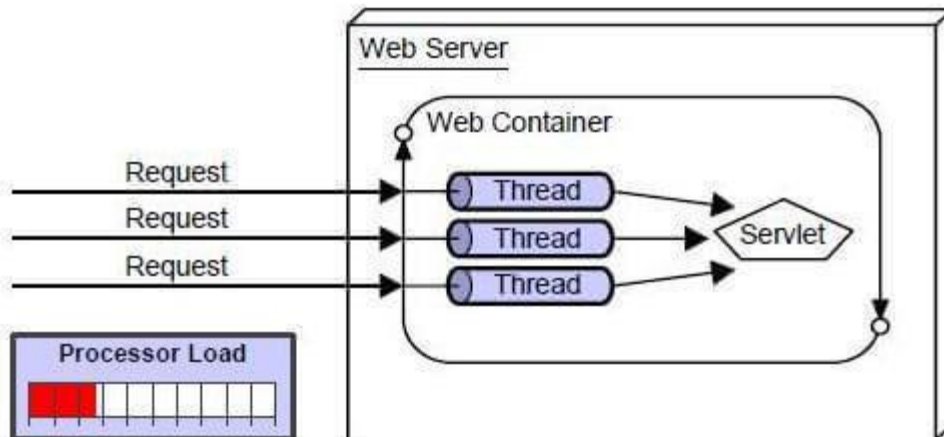


## Disadvantages of CGI

There are many problems in CGI technology:

1. If the number of clients increases, it takes more time for sending the response.
  2. For each request, it starts a process, and the web server is limited to start processes.
  3. It uses platform dependent language e.g. [C](#), [C++](#), [perl](#).
-

## Advantages of Servlet



There are many advantages of Servlet over CGI. The web container creates threads for handling the multiple requests to the Servlet. Threads have many benefits over the Processes such as they share a common memory area, lightweight, cost of communication between the threads are low. The advantages of Servlet are as follows:

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:** because it uses java language.

## Steps to create a servlet example

1. [Steps to create the servlet using Tomcat server](#)
  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 application](#)

There are given 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

The mostly used approach is by extending HttpServlet because it provides http request specific method such as doGet(), doPost(), doHead() etc.

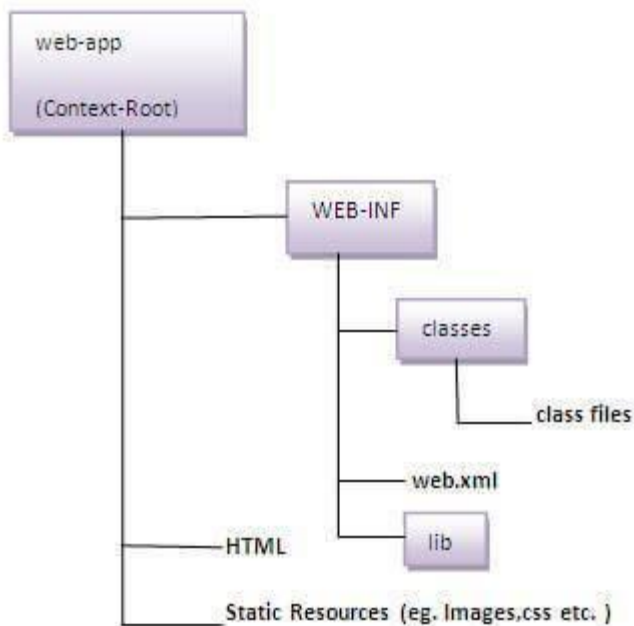
Here, we are going to use **apache tomcat server** in this example. The steps are as follows:

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

## 1) Create a directory structures

The **directory structure** defines that where to put the different types of files so that web container may get the information and respond to the client.

The Sun Microsystem defines a unique standard to be followed by all the server vendors. Let's see the directory structure that must be followed to create the servlet.



As you can see that the servlet class file must be in the classes folder. The web.xml file must be under the WEB-INF folder.

---

## 2) Create a Servlet

There are three ways to create the servlet.

1. By implementing the Servlet interface
2. By inheriting the GenericServlet class
3. By inheriting the HttpServlet class

The HttpServlet class is widely used to create the servlet because it provides methods to handle http requests doPost, doHead() etc.

In this example we are going to create a servlet that extends the HttpServlet class. In this example, we are inheriting class and providing the implementation of the doGet() method. Notice that get request is the default request.

### DemoServlet.java

1. **import** javax.servlet.http.\*;
2. **import** javax.servlet.\*;
3. **import** java.io.\*;

```

4. public class DemoServlet extends HttpServlet{
5. public void doGet(HttpServletRequest req,HttpServletResponse res)
6. throws ServletException,IOException
7. {
8. res.setContentType("text/html");//setting the content type
9. PrintWriter pw=res.getWriter();//get the stream to write the data
10.
11. //writing html in the stream
12. pw.println("<html><body>");
13. pw.println("Welcome to servlet");
14. pw.println("</body></html>");
15.
16. pw.close();//closing the stream
17. }}

```

### 3)Compile the servlet

For compiling the Servlet, jar file is required to be loaded. Different Servers provide different jar files:

Jar file	Server
1) servlet-api.jar	Apache Tomcat
2) weblogic.jar	Weblogic
3) javaee.jar	Glassfish
4) javaee.jar	JBoss

#### Two ways to load the jar file

1. set classpath

2. paste the jar file in JRE/lib/ext folder

Put the java file in any folder. After compiling the java file, paste the class file of servlet in **WEB-INF/classes** directory.

## 4) Create the deployment descriptor (web.xml file)

The **deployment descriptor** is an xml file, from which Web Container gets the information about the servlet to be invoked.

The web container uses the Parser to get the information from the web.xml file. There are many xml parsers such as SAX, DOM and Pull.

There are many elements in the web.xml file. Here is given some necessary elements to run the simple servlet program.

### web.xml file

1. **<web-app>**
- 2.
3. **<servlet>**
4. **<servlet-name>**sonoojaiswal**</servlet-name>**
5. **<servlet-class>**DemoServlet**</servlet-class>**
6. **</servlet>**
- 7.
8. **<servlet-mapping>**
9. **<servlet-name>**sonoojaiswal**</servlet-name>**
10. **<url-pattern>**/welcome**</url-pattern>**
11. **</servlet-mapping>**
- 12.
13. **</web-app>**

## Description of the elements of web.xml file

There are too many elements in the web.xml file. Here is the illustration of some elements that is used in the above web.xml file. The elements are as follows:



<**web-app**> represents the whole application.

<**servlet**> is sub element of <web-app> and represents the servlet.

<**servlet-name**> is sub element of <servlet> represents the name of the servlet.

<**servlet-class**> is sub element of <servlet> represents the class of the servlet.

<**servlet-mapping**> is sub element of <web-app>. It is used to map the servlet.

<**url-pattern**> is sub element of <servlet-mapping>. This pattern is used at client side to invoke the servlet.

---

## 5)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?

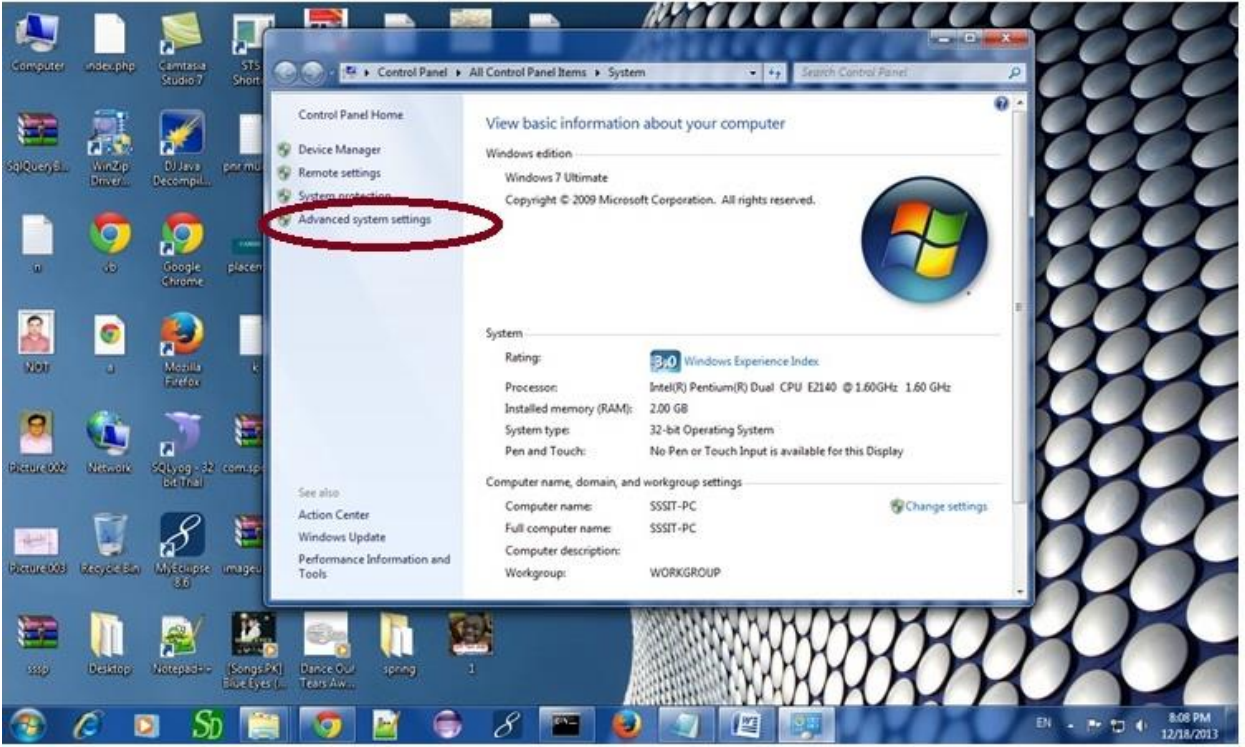
To start Apache Tomcat server JAVA\_HOME and JRE\_HOME must be set in Environment variables.

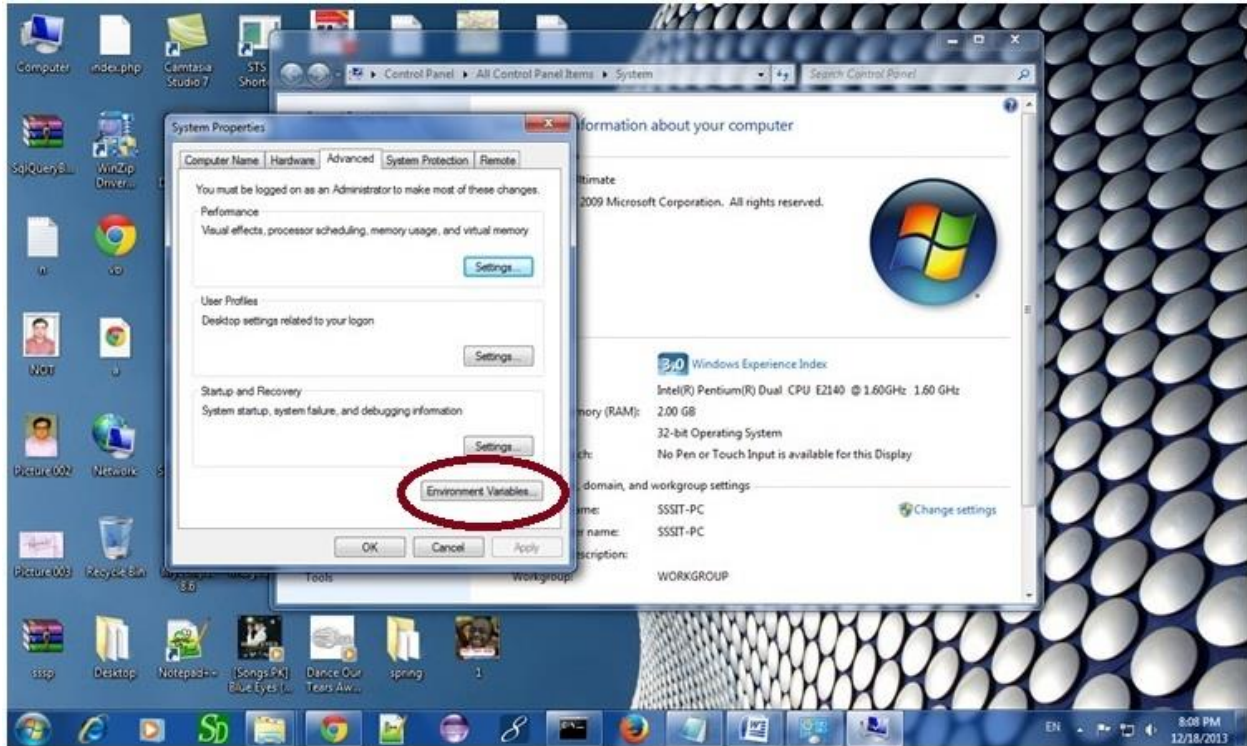
Go to My Computer properties -> Click on advanced tab then environment variables -> Click on the new tab of user variable -> Write JAVA\_HOME in variable name and paste the path of jdk folder in variable value -> ok -> ok -> ok.

Go to My Computer properties:

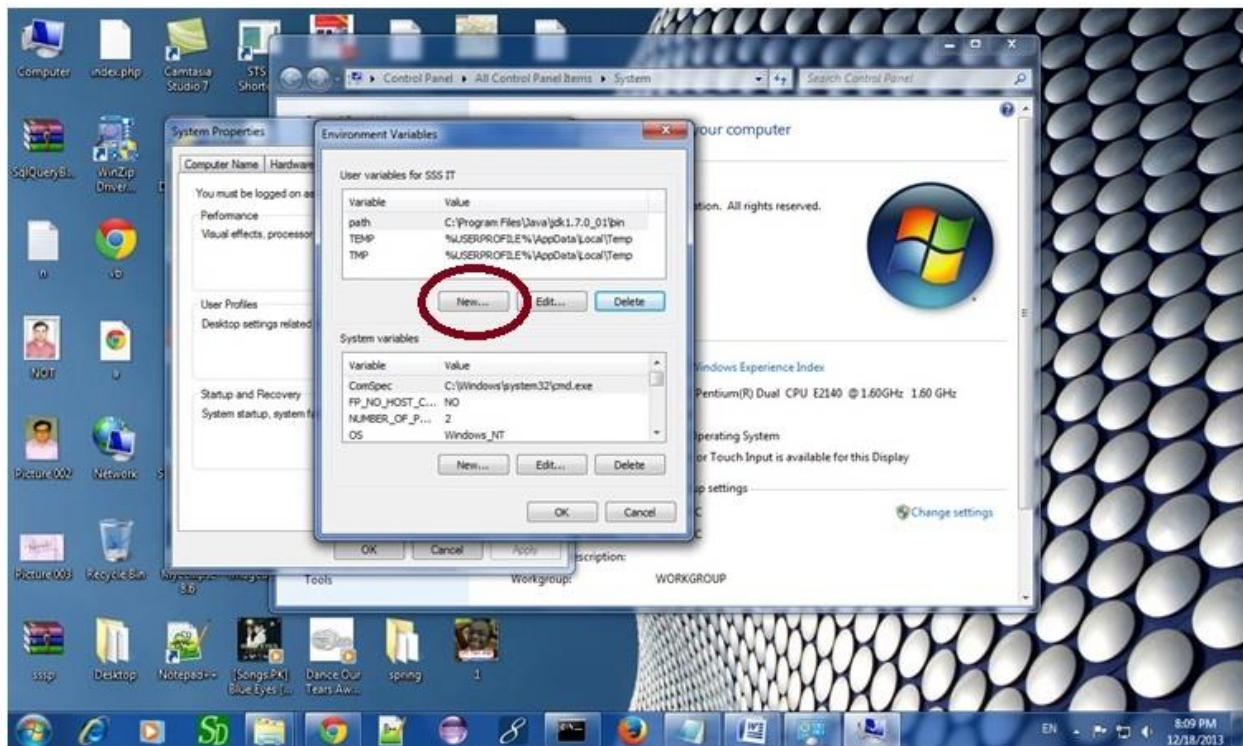


Click on advanced system settings tab then environment variables:

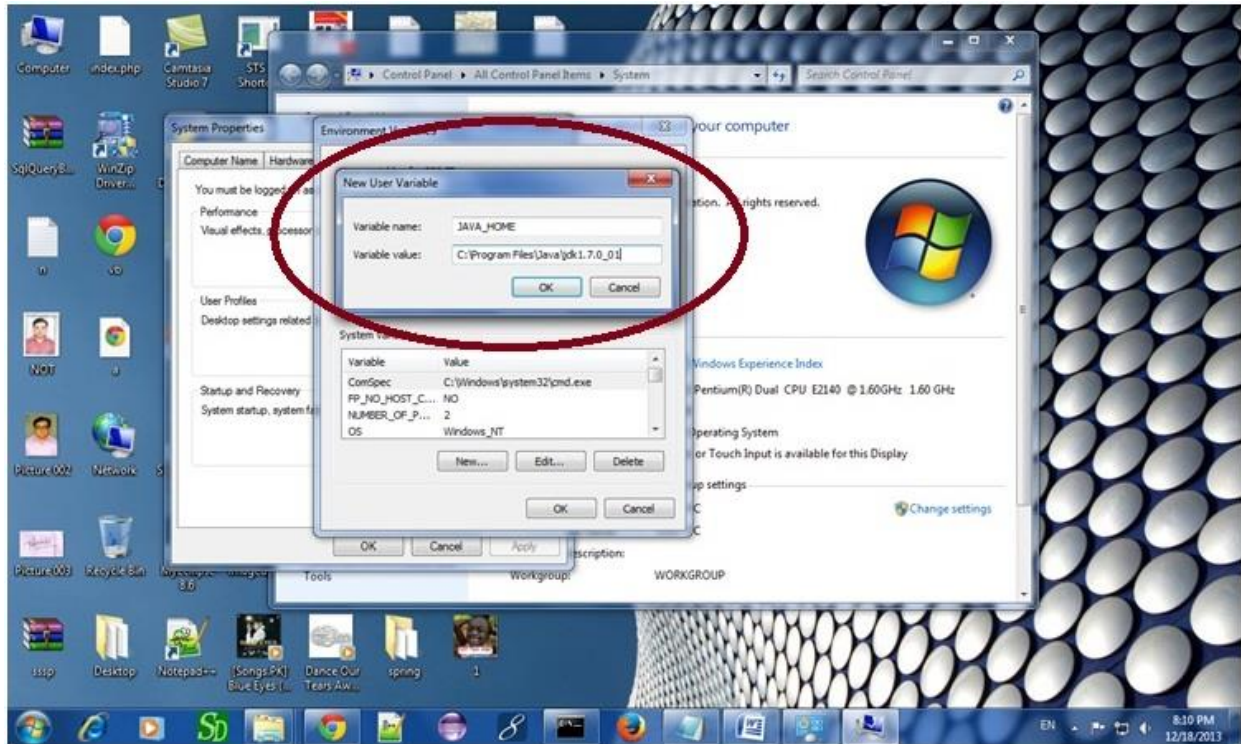




Click on the new tab of user variable or system variable:



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.

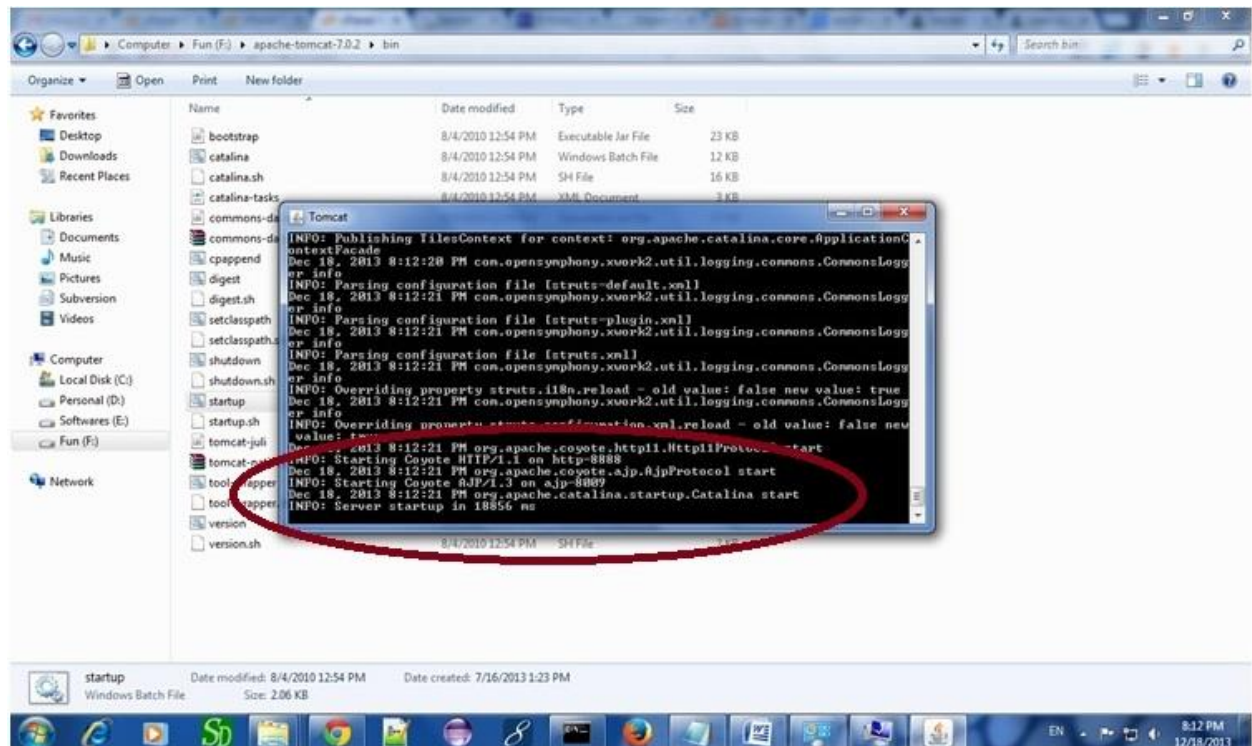
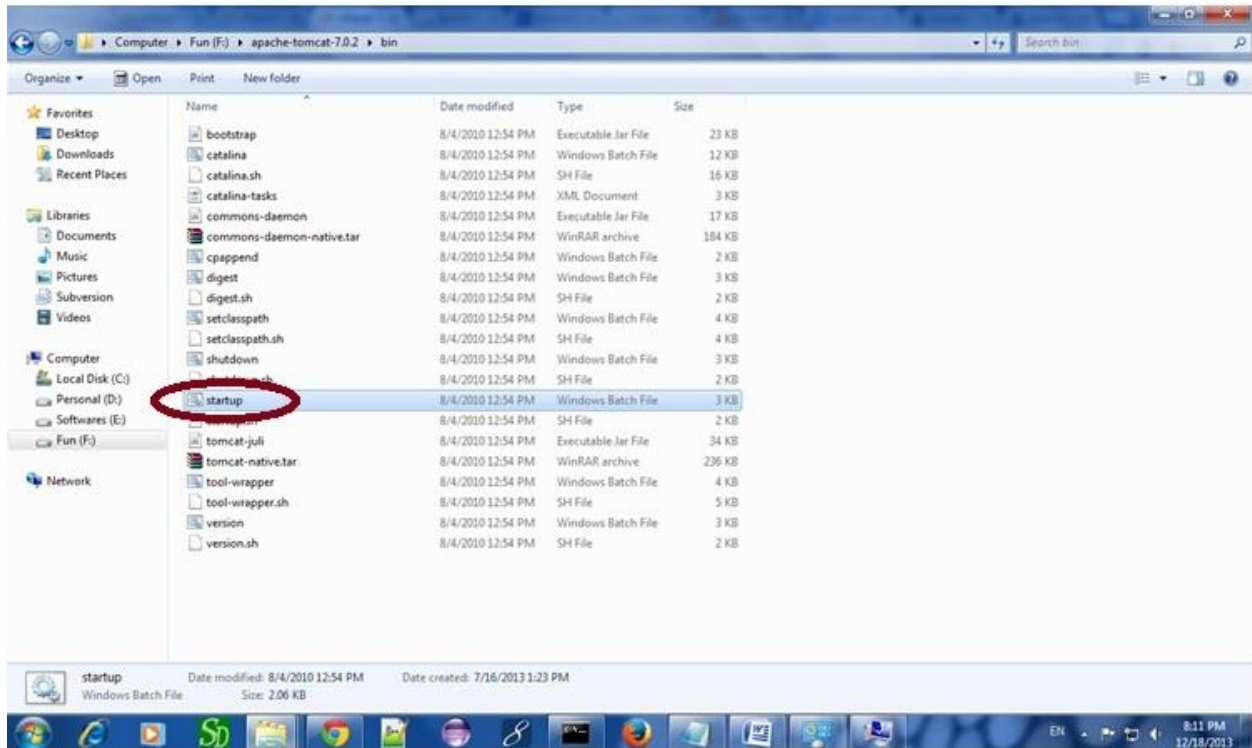
---

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.

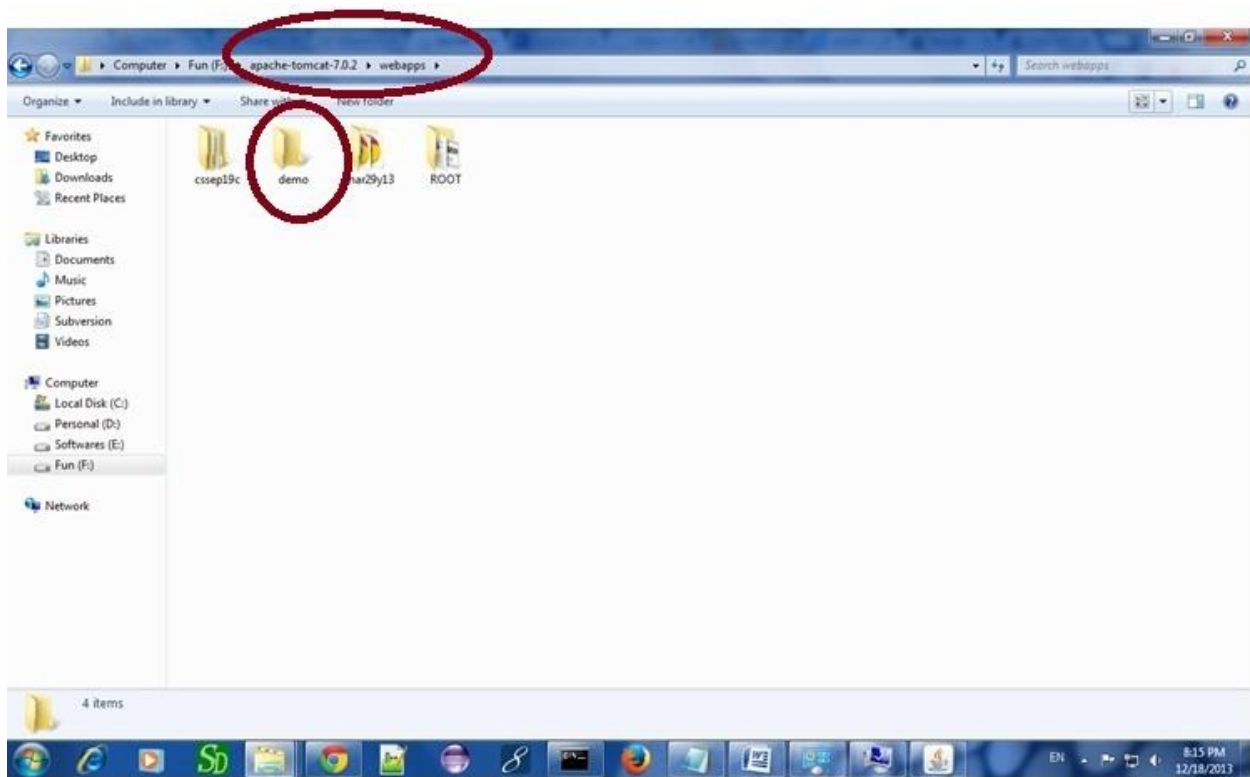
## 2) How to change port number of apache tomcat

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: