**Creating First Servlet Application using Netbeans IDE**

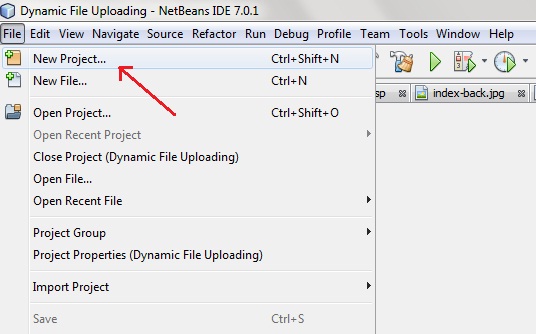
In the last lesson we created our first Servlet Application but without using any IDE. An IDE is Integrated Development Environment, and it makes creating applications a lot easier. We will learn how to create Servlet applications on NetBeans IDE and Eclipse IDE. Then you can decide which one, you want to use.

Using Intregrated Development Enviroment(IDE) is the easiest way to create Servlet Applications. An IDE is a software application that provides facilities to computer programmers for software development. **Eclipse**, **MyEcplise**, **Netbeans** are example of some popular Java IDE.

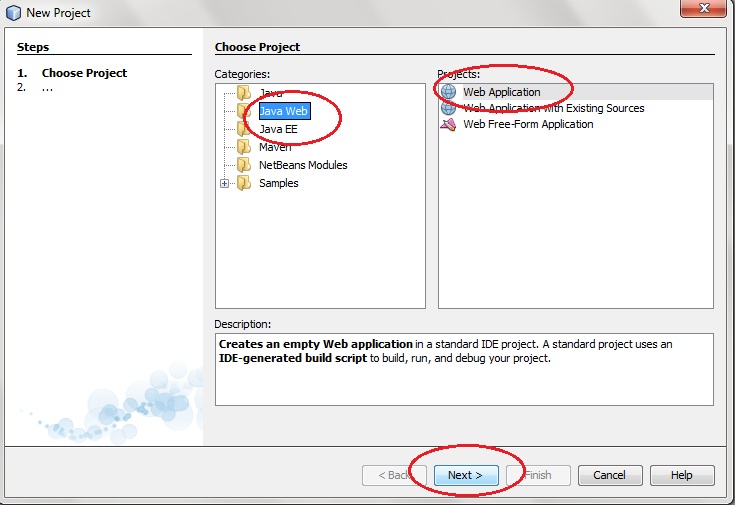
**Steps to Create Servlet Application in Netbeans IDE**

To create a servlet application in Netbeans IDE, you will need to follow the following (simple) steps :

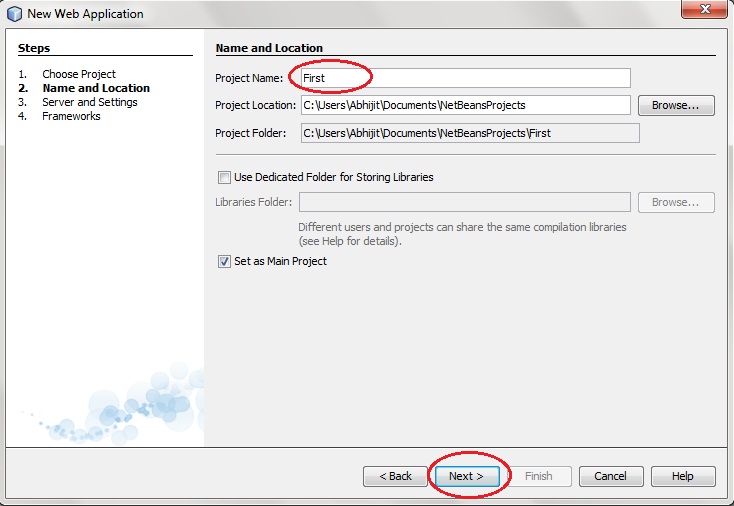
1. Open Netbeans IDE, Select **File** -> **New Project**



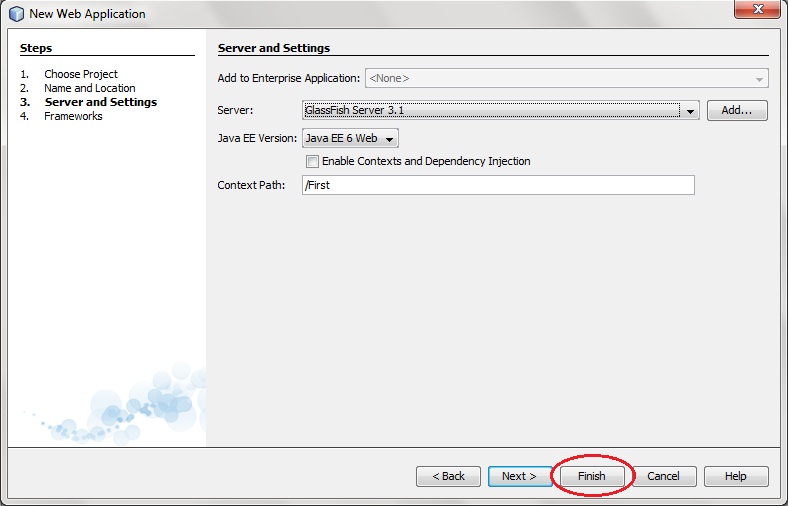
1. Select **Java Web** -> **Web Application**, then click on Next,



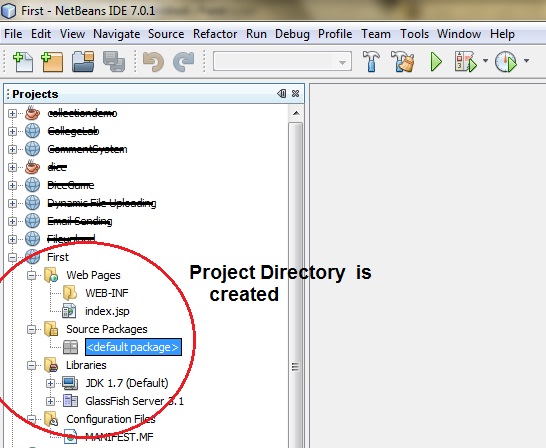
1. Give a name to your project and click on Next,



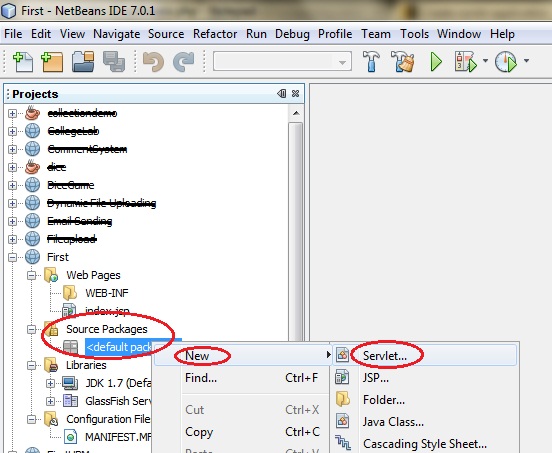
1. and then, Click **Finish**



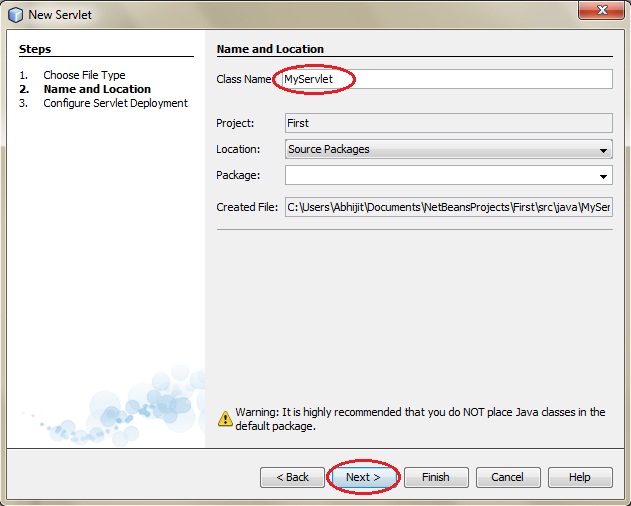
1. The complete directory structure required for the Servlet Application will be created automatically by the IDE.

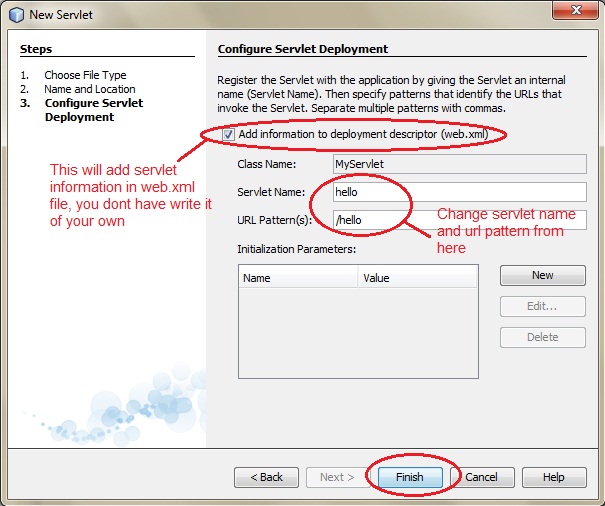


1. To create a Servlet, open **Source Package**, right click on **default packages** -> **New** -> **Servlet**.

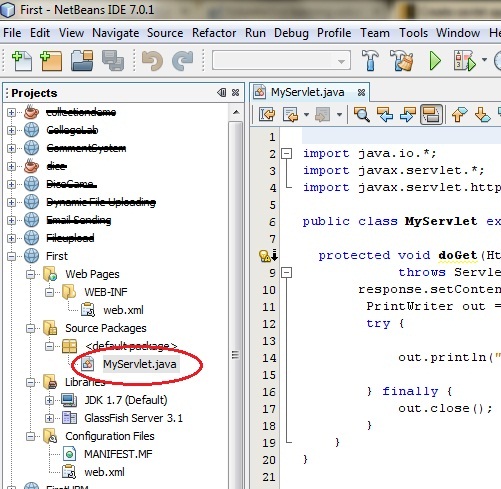


1. Give a Name to your Servlet class file,

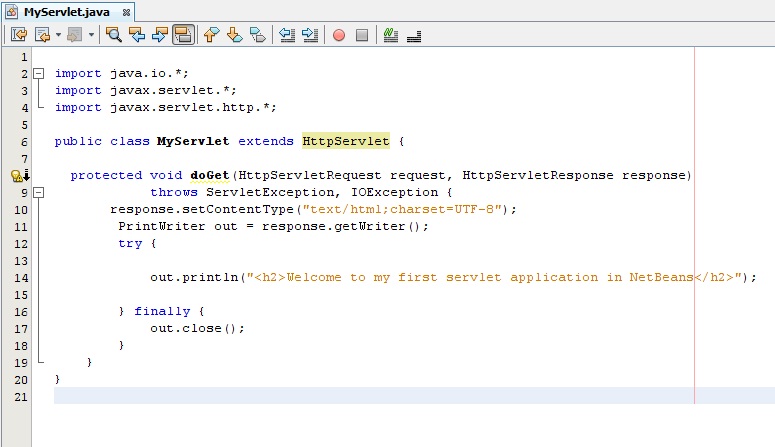




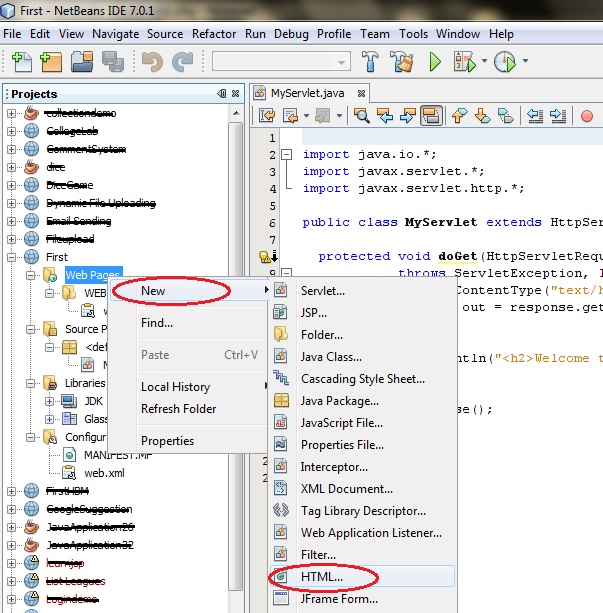
1. Now, your Servlet class is ready, and you just need to change the method definitions and you will good to go.



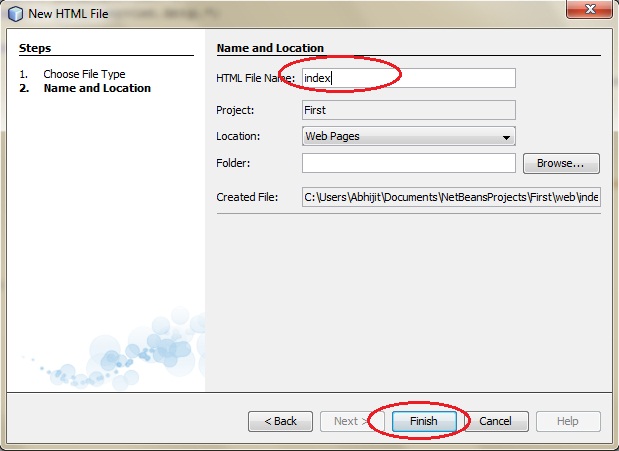
1. Write some code inside your Servlet class.



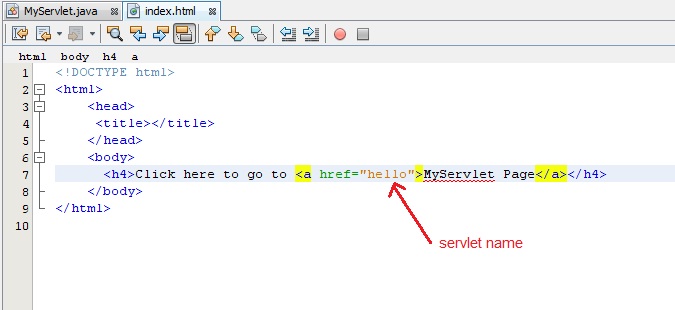
1. Create an HTML file, right click on **Web Pages** -> **New** -> **HTML**



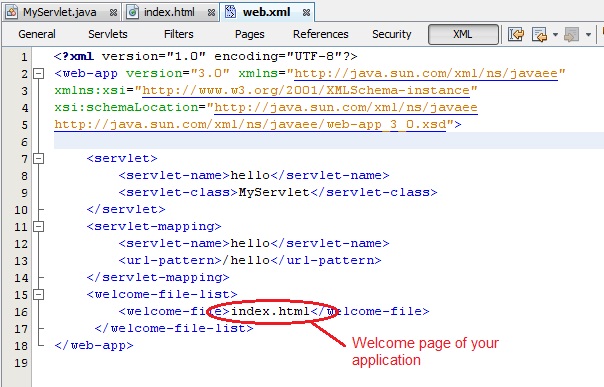
1. Give it a name. We recommend you to name it index, because browser will always pick up the index.html file automatically from a directory. Index file is read as the first page of the web application.



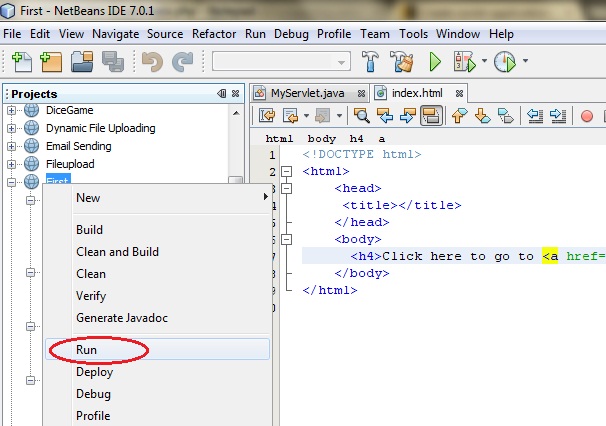
1. Write some code inside your HTML file. We have created a hyperlink to our Servlet in our HTML file.



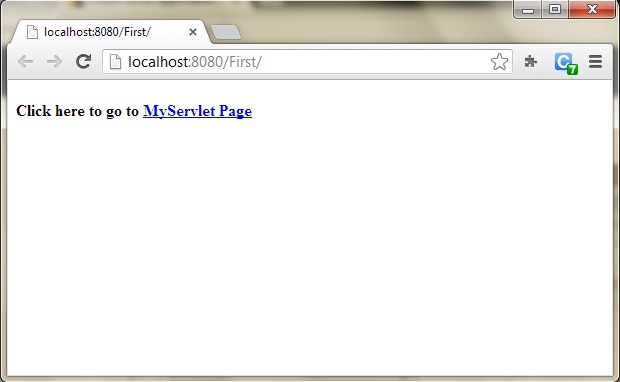
1. Edit **web.xml** file. In the web.xml file you can see, we have specified the **url-pattern** and the **servlet-name**, this means when hello url is accessed our Servlet file will be executed.



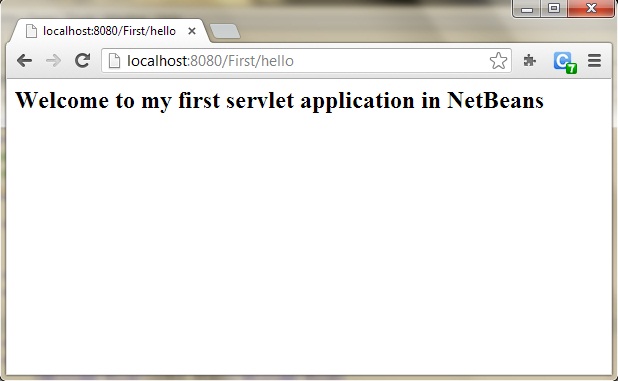
1. Run your application, right click on your Project and select **Run**



1. Click on the link created, to open your Servlet.



1. Hurray! Our First Servlet class is running.



**Steps to Create Servlet Application using tomcat server**

To create a Servlet application you need to follow the below mentioned steps. These steps are common for all the Web server. In our example we are using Apache Tomcat server. Apache Tomcat is an open source web server for testing servlets and JSP technology. Download latest version of [Tomcat Server](https://tomcat.apache.org/) and install it on your machine.

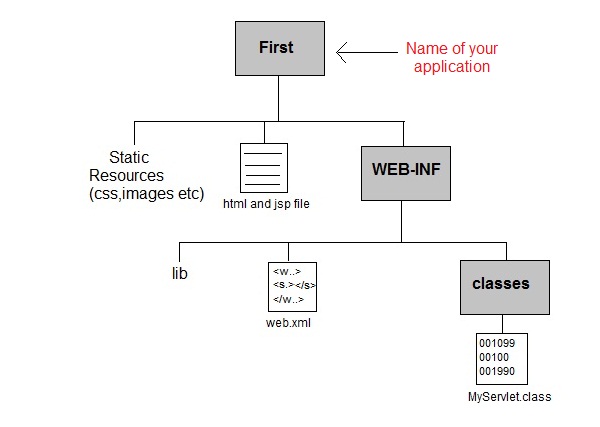
After installing Tomcat Server on your machine follow the below mentioned steps :

1. Create directory structure for your application.
2. Create a Servlet
3. Compile the Servlet
4. Create Deployement Descriptor for your application
5. Start the server and deploy the application

All these 5 steps are explained in details below, lets create our first Servlet Application.

**1. Creating the Directory Structure**

Sun Microsystem defines a unique directory structure that must be followed to create a servlet application.



Create the above directory structure inside **Apache-Tomcat\webapps** directory. All HTML, static files(images, css etc) are kept directly under **Web application** folder. While all the Servlet classes are kept inside classes folder.

The web.xml (deployement descriptor) file is kept under WEB-INF folder.

**Creating a Servlet**

There are three different ways to create a servlet.

* By implementing **Servlet** interface
* By extending **GenericServlet** class
* By extending **HttpServlet** class

But mostly a servlet is created by extending **HttpServlet** abstract class. As discussed earlier **HttpServlet** gives the definition of service() method of the **Servlet** interface. The servlet class that we will create should not override service() method. Our servlet class will override only doGet() or doPost() method.

When a request comes in for the servlet, the Web Container calls the servlet's service() method and depending on the type of request the service() method calls either the doGet() or doPost()method.

**NOTE:** By default a request is **Get** request.

import javax.servlet.\*;

import javax.servlet.http.\*;

import java.io.\*;

public *MyServlet* **extends** HttpServlet

{

public void *doGet*(HttpServletRequest request,HttpServletResposne response)

**throws** ServletException

{

response.setContentType("text/html");

PrintWriter *out* = response.getWriter();

out.println("<html><body>");

out.println("<h1>Hello Readers</h1>");

out.println("</body></html>");

}

}

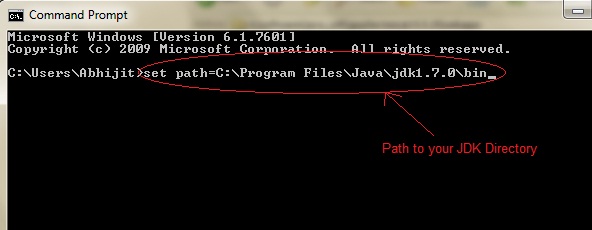
Write above code in a notepad file and save it as **MyServlet.java** anywhere on your PC. Compile it(explained in next step) from there and paste the class file into WEB-INF/classes/ directory that you have to create inside **Tomcat/webapps** directory.

**Compiling a Servlet**

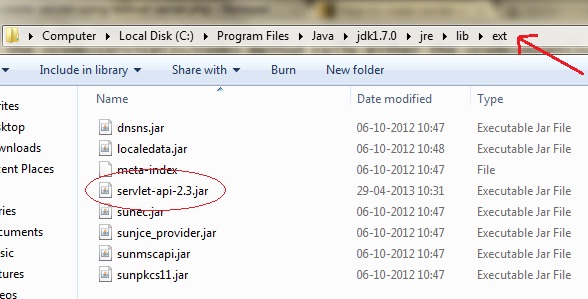
To compile a Servlet a JAR file is required. Different servers require different JAR files. In Apache Tomcat server servlet-api.jar file is required to compile a servlet class.

Steps to compile a Servlet

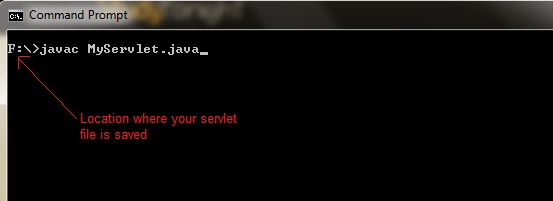
* Set the Class Path.



* Download **servlet-api.jar** file.
* Paste the servlet-api.jar file inside Java\jdk\jre\lib\ext directory.



* Compile the Servlet class.



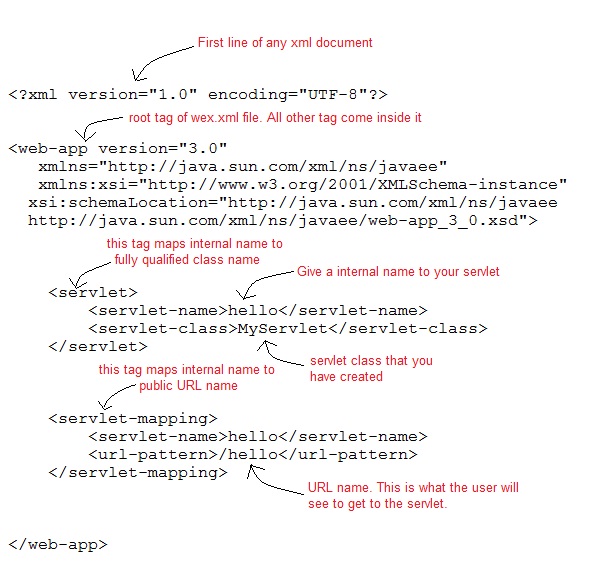
**NOTE:**After compiling your Servlet class you will have to paste the class file into WEB-INF/classes/directory.

**Create Deployment Descriptor**

**Deployment Descriptor(DD)** is an XML document that is used by Web Container to run Servlets and JSP pages. DD is used for several important purposes such as:

* Mapping URL to Servlet class.
* Initializing parameters.
* Defining Error page.
* Security roles.
* Declaring tag libraries.

We will discuss about all these in details later. Now we will see how to create a simple **web.xml** file for our web application.



**Start the Server**

Double click on the **startup.bat** file to start your Apache Tomcat Server.

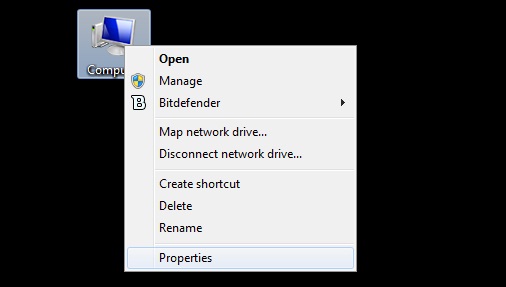
Or, execute the following command on your windows machine using RUN prompt.

C:\apache-tomcat-7.0.14\bin\startup.bat

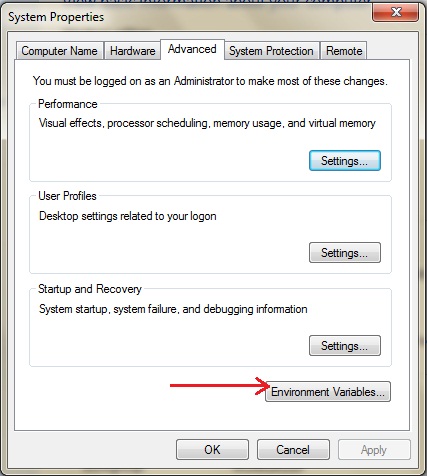
**Starting Tomcat Server for the first time**

If you are starting Tomcat Server for the first time you need to set JAVA\_HOME in the Enviroment variable. The following steps will show you how to set it.

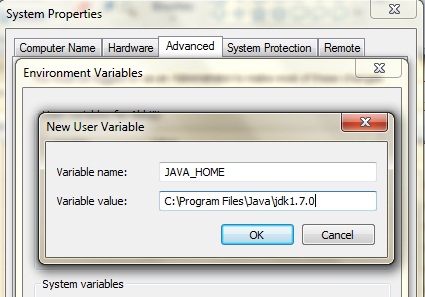
* Right Click on **My Computer**, go to **Properites**.



* Go to **Advanced** Tab and Click on **Enviroment Variables...** button.

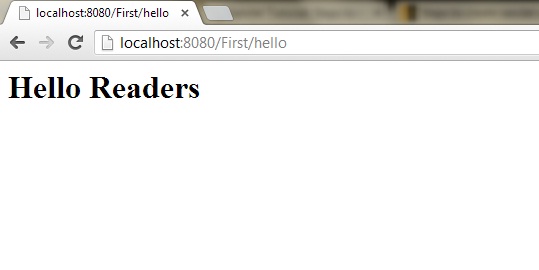


* Click on **New** button, and enter **JAVA\_HOME** inside Variable name text field and path of JDK inside Variable value text field. Click OK to save.



**Run Servlet Application**

Open Browser and type **http:localhost:8080/First/hello**



Hurray! Our first Servlet Application ran successfully.