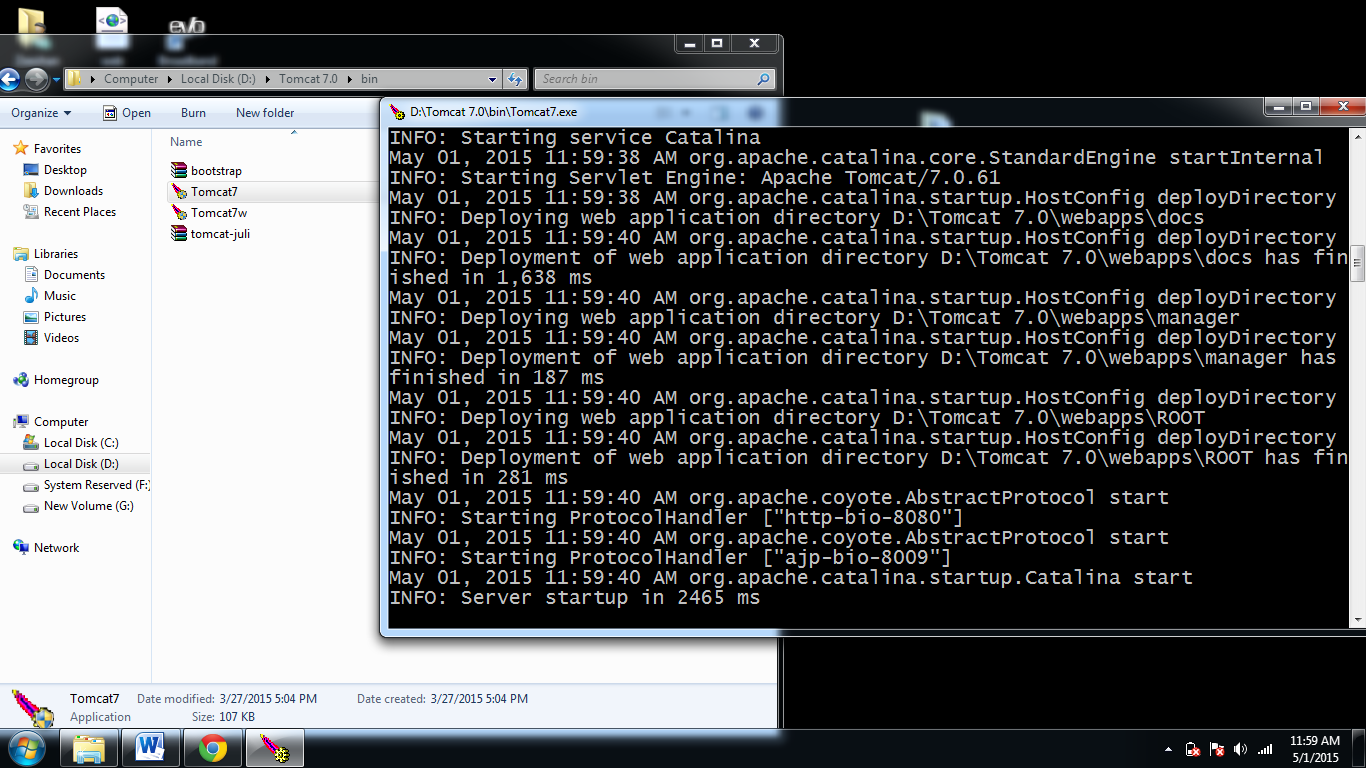
**Setting up development environment for Java Servlet and JSP applications**

Assuming you have already setup the Java Development Environment and you have Windows OS, follow the following steps to setup the Development Environment for Java Servlet and JSP applications.

* Download the apache tomcat web server from [www.tomcat.apache.org](http://www.tomcat.apache.org/). Download [32-bit/64-bit Windows Service Installer](http://mirror.sdunix.com/apache/tomcat/tomcat-7/v7.0.61/bin/apache-tomcat-7.0.61.exe)
* Install the setup in any appropriate directory of your choice. Install the setup with default settings.
* After the installation is complete, open the tomcat folder and the bin folder in it.
* To start the server, execute the Tomcat7.exe (in my case its tomcat version 7) in the bin folder. It will look like the follows,

 **Fig1: Running tomcat server**

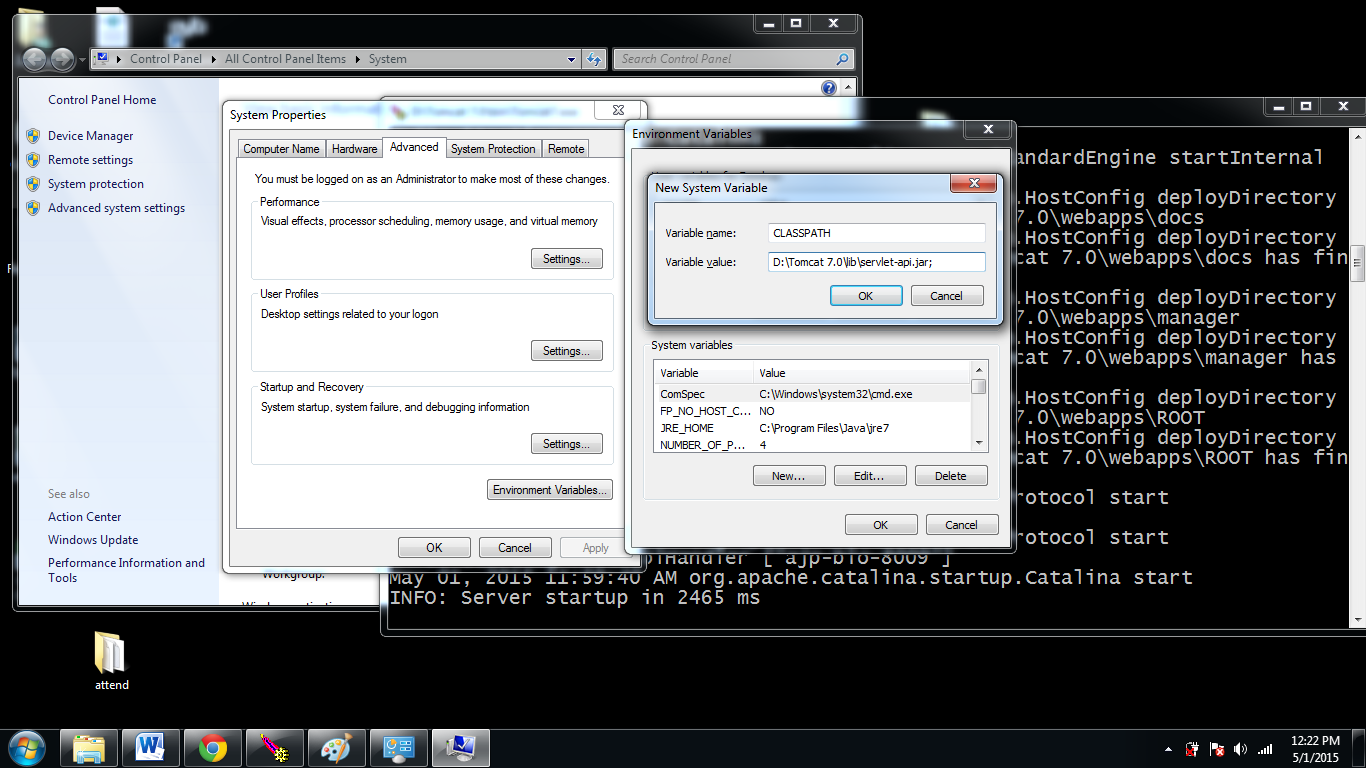
* Now check to see if the tomcat server is started with the URL [http://localhost:8080](http://localhost:8080/). If the server is successfully started, it will show the Tomcat Version and other information.

Tomcat server is now successfully started, the second step is to setup the libraries required for the Servlet and JSP applications. These libraries contain classes required for the Servlet and JSP apps and if not setup properly, the applications will not compile or run.

There are different ways to setups the libraries, but the two more convenient ways are given here (Here we are discussing about command line development).

1. **Add the libraries to the CLASSPATH environment variable**

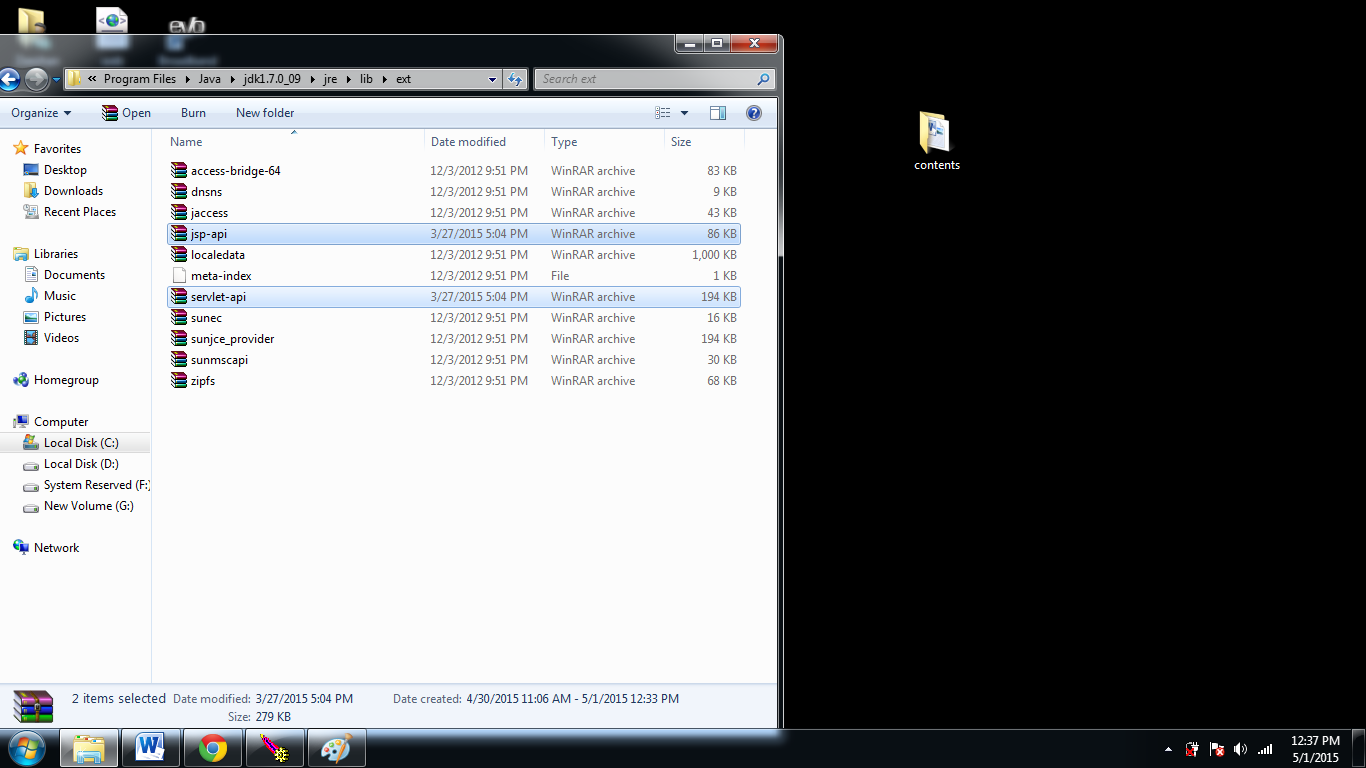
* To setup CLASSPATH environment variable, right click on my computer and open properties. Open Advanced system settings.
* Select Environment Variables…
* In the System variables, add a new system variable named CLASSPATH if it does not already exist.
* Environment variable value will point to the jar files in lib folder of Tomcat installation directory. All the jar files must be semicolon separated.
* Now add the full path of servlet-api.jar and jsp-api.jar as semicolon separated values.

**Fig2: Setting up environment variable**

* Save the environment variable and you are ready to compile the applications.

1. **Copy the jar files to jre\lib\ext folder.**

* Copy servlet-api.jar and jsp-api.jar files from tomcat lib folder to the folder <jdk\_installation\_directory>\jre\lib\ext

**Fig3: Copying jar files in the JRE**

This completes the setup for command line development of Servlet and JSP applications.

**Servlet Deployment**

**Servlet Application Structure**

A servlet application has a special format. A servlet application resides in ROOT folder of webapps in the server installation directory.

On ROOT folder, the index page and its associated files like images etc are saved and another important folder called WEB-INF is created.

WEB-INF contains the configuration file and the servlet classes. In WEB-INF folder create a web.xml configuration file and another folder called classes.

**Web.xml file:**

The web.xml file contains important information about servlet application, like servlet name, its class name, servlet URL mapping etc. Here is a sample servlet file

<web-app >

<display-name>Welcome to Servlet</display-name>

<servlet>

<servlet-name>Servlet</servlet-name>

<servlet-class>SimpleServlet</servlet-class>

</servlet>

<servlet-mapping>

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

<url-pattern>/simpleservlet</url-pattern>

</servlet-mapping>

</web-app>

The above xml sets the servlet display name, servlet name, class name and the URL mapping to access the servlet class.

**Classes folder:**

The classes folder will contain the compiled servlet classes, that is .class files. If the classes are written in packages, the format would change. Create a new folder with the package name. For example if a class named **ServletClass** is written in package com.quest, the folder structure will be classes -> com -> quest -> ServletClass.class

After Deploying the application, open web browser and access **localhost:8080**. It should open your index.html page. Now access **localhost:8080/simpleservlet** and it will execute and produce results of the SimpleServlet class.