

Supplement IV.E: Tutorial for Tomcat 4.1.27

For Introduction to Java Programming By Y. Daniel Liang

This supplement covers the following topics:

- Obtaining and Installing Tomcat
- Starting and Stopping Tomcat
- Choosing a Different Port
- Compiling Servlets
- Running Servlets
- Running JSP
- Creating User Web Application Directories

0 Introduction

Tomcat, developed by Apache (www.apache.org), is a standard reference implementation for Java servlets and JSP. It can be used standalone as a Web server or be plugged into a Web server like Apache, Netscape Enterprise Server, or Microsoft Internet Information Server. There are many versions of Tomcat. This tutorial covers Tomcat 4.1.27 with JDK 1.5.

1 Obtaining and Installing Tomcat

You can download Tomcat 4.1.27 in a zip file named tomcat4.1.27.zip from <http://archive.apache.org/dist/jakarta/tomcat-4/tomcat-4.1.27.zip>. Unzip tomcat4.1.27.zip. tomcat4.1.27.zip is compressed file. You can use WinZip to decompress into c:\ using WinZip. WinZip is a file compression/decompression utility, which can be downloaded from www.winzip.com.

2 Starting and Stopping Tomcat

Before running the servlet, you need to start the Tomcat servlet engine. To start Tomcat, you have to first set the `JAVA_HOME` environment variable to the JDK home directory using the following command. (Please note no space before or after the = sign in the following line.)

```
set JAVA_HOME=c:\Program Files\java\jdk1.5.0
```

The JDK home directory is where your JDK is stored. On my computer, it is c:\Program Files\jdk1.5.0. You may have a different directory. You can now start Tomcat using the command **startup** from c:\jakarta-tomcat-4.1.27\bin as follows:

```
c:\jakarta-tomcat-4.1.27\bin>startup
```

NOTE: By default, Tomcat runs on port 8080. An error occurs if this port is currently being used. You can change the port number in `c:\jakarta-tomcat-4.1.27\conf\server.xml`, as discussed in the next section.

NOTE: To terminate Tomcat, use the **shutdown** command from `c:\jakarta-tomcat-4.1.27\bin`.

To prove that Tomcat is running, type the URL <http://localhost:8080> from a Web browser, as shown in Figure 1.

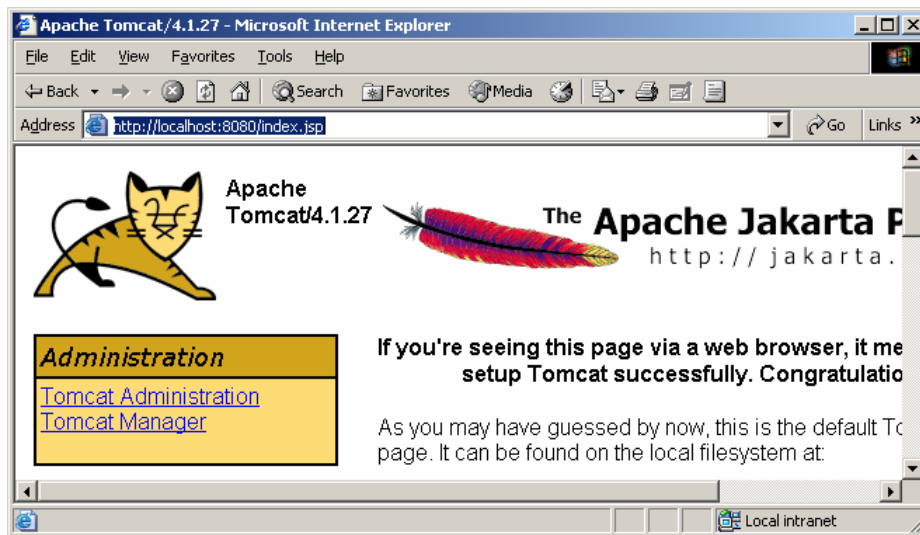


Figure 1
The default Tomcat page is displayed.

3 Choosing a Different Port (Optional)

By default, Tomcat runs on port 8080. You can change it to a different port. To do so, open `c:\jakarta-tomcat-4.1.27\conf\server.xml` using a text editor such as NotePad. Search for 8080 and change it to a desired port number such as 8100 in the following context.

```
<Connector className="org.apache.coyote.tomcat4.CoyoteConnector"  
  port="8080" minProcessors="5" maxProcessors="75"  
  enableLookups="true" redirectPort="8443"  
  acceptCount="100" debug="0" connectionTimeout="20000"  
  useURIVValidationHack="false" disableUploadTimeout="true" />
```

4 Creating a Servlet

A servlet resembles an applet in some extent. Every Java

applet is a subclass of the Applet class. You need to override appropriate methods in the Applet class to implement the applet. Every servlet is a subclass of the HttpServlet class. You need to override appropriate methods in the HttpServlet class to implement the servlet. Listed below is a simple servlet example that generates a response in HTML using the doGet method.

```
import javax.servlet.*;
import javax.servlet.http.*;

public class FirstServlet extends HttpServlet {
    /** Handle the HTTP <code>GET</code> method.
     * @param request servlet request
     * @param response servlet response
     */
    protected void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, java.io.IOException {
        response.setContentType("text/html");
        java.io.PrintWriter out = response.getWriter();
        // output your page here
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Servlet</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("Hello, Java Servlets");
        out.println("</body>");
        out.println("</html>");
        out.close();
    }
}
```

5 Compiling Servlets

Suppose you have installed Tomcat 4.1.27 at c:\jakarta-tomcat-4.1.27. To compile FirstServlet.java, you need to add c:\jakarta-tomcat-4.1.27\common\lib\servlet.jar to the classpath from DOS prompt as follows:

```
set classpath=%classpath%;c:\jakarta-tomcat-4.1.27\common\lib\servlet.jar
```

servlet.jar contains the classes and interfaces to support servlets. Use the following command to compile the servlet:

```
javac FirstServlet.java
```

Copy the resultant .class file into c:\jakarta-tomcat-4.1.27\webapps\examples\WEB-INF\classes so it can be found at runtime.

TIP: You can compile FirstServlet directly into the target directory by using the -d option in the javac command as follows:

```
javac FirstServlet.java -d targetdirectory
```

6 Running Servlets

To run the servlet FirstServlet, start a Web browser and

type ***http://localhost:8080/examples/servlet/FirstServlet*** in the URL, as shown in Figure 2.

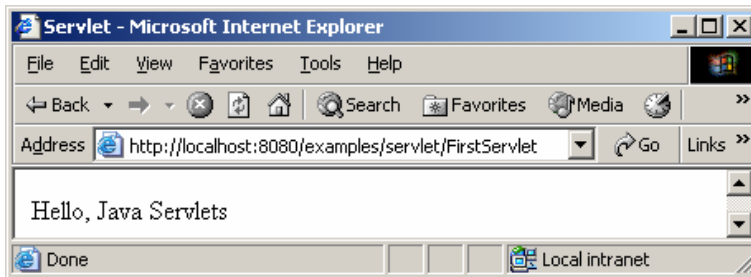


Figure 2

You can request a servlet from a Web browser.

NOTE: You can use the servlet from anywhere on the Internet if your Tomcat is running on a host machine on the Internet. Suppose the host name is *liang.armstrong.edu*, use the URL ***http://liang.armstrong.edu:8080/examples/servlet/FirstServlet*** to test the servlet.

NOTE: If you have modified the servlet, you need to shutdown and restart Tomcat.

TIP: If your servlet class uses packages (e.g., package chapter33), you have to create a directory named *chapter33* under *c:\jakarta-tomcat-4.1.27\webapps\examples\WEB-INF\classes* and copy the *.class* into the new directory. You have to use the URL ***http://localhost:8080/examples/servlet/chapter33.FirstServlet*** to invoke the servlet.

7 Running JSP

To run a JSP (e.g., *Factorial.jsp*), store it in *webapps/examples/jsp*, start a Web browser and type ***http://localhost:8080/examples/jsp/Factorial.jsp*** in the URL, as shown in Figure 3.

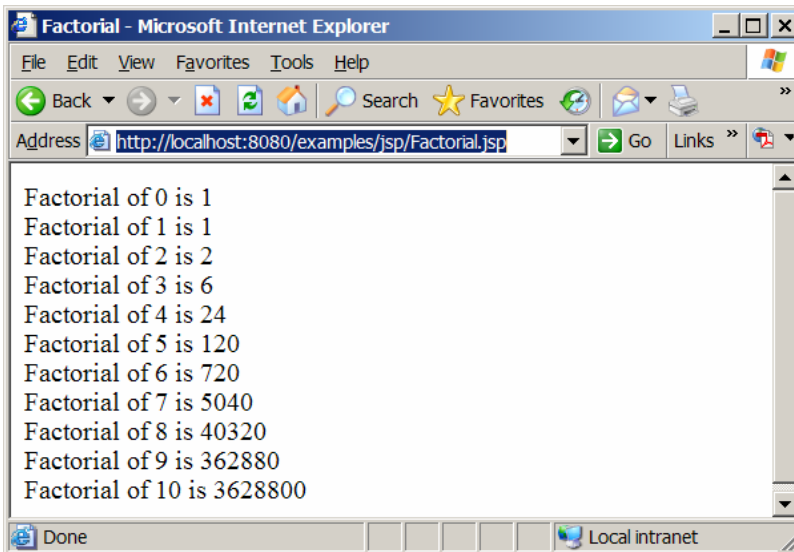


Figure 3

You can request a JSP from a Web browser.

8 Creating User Web Application Directories

The easiest way to run the servlet for novice programmers is to store all the servlet class files in `c:\Jakarta-tomcat-4.1.27\webapps\examples\WEB-INF\classes` and use the URL <http://localhost:8080/examples/servlet/ClassName> to run the servlet. You can also place the servlets in a user created directory. The user directory must be placed under `c:\Jakarta-tomcat-4.1.27\webapps`. Assume the user directory is named **smith**. You have to also create directories `images`, `jsp`, `servlets`, and `WEB-INF`, as shown in Figure 4.

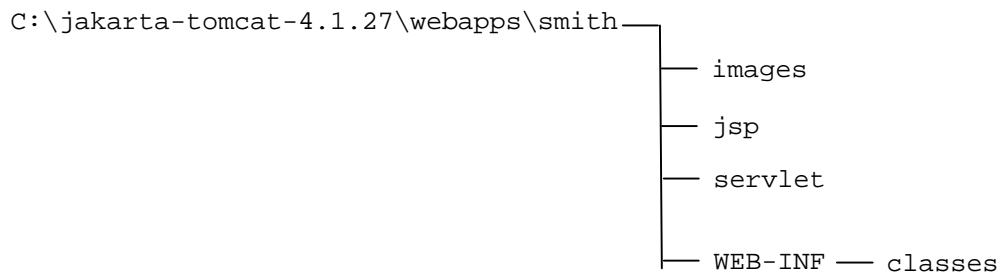


Figure 4

The user directory contains images, jsp, servlets, and WEB-INF.

The `images` directory stores image files. The `jsp` directory stores the `.JSP` files, the `servlets` direc???

```
jar cvf mywar.war WEB-INF
```

tory stores .html files, and the WEB-INF/classes directory stores .class files for servlets. Suppose you put Factorial.jsp in the jsp directory, you can view it using the URL <http://localhost:8080/smith/jsp/Factorial.jsp>, as shown in Figure 5.

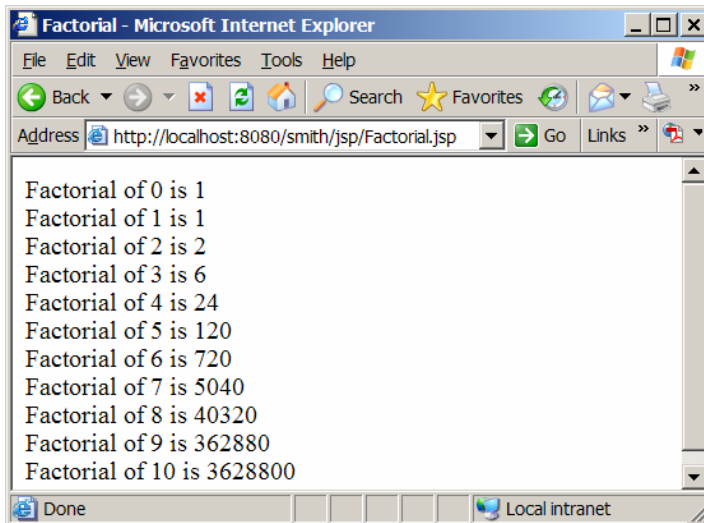


Figure 5

The JSP page is stored in the webapps/smith/jsp directory.

Suppose you put helloworld.html in the servlets directory, you can view it using the URL <http://localhost:8080/smith/servlets/helloworld.html>, as shown in Figure 6.

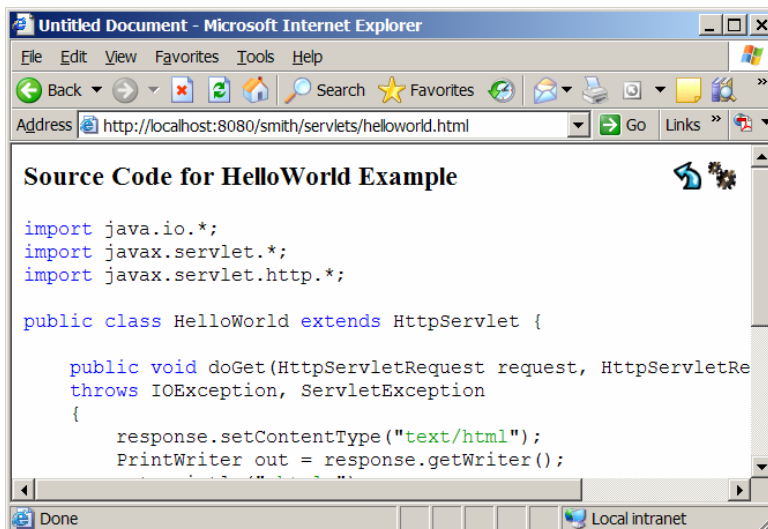


Figure 6

An HTML page is stored in the webapps/smith/servlets directory.

Suppose you put `FirstServlet.class` in the `WEB-INF/classes` directory, you can view it using the URL <http://localhost:8080/smith/servlets/FirstServlets>, as shown in Figure 7.

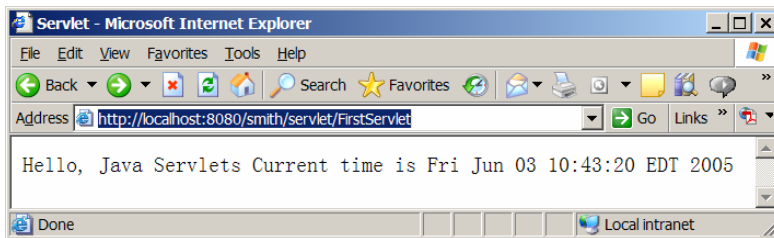


Figure 7

A servlet class file is stored in the `webapps/smith/WEB-INF/classes` directory.