Java Servlets

CS174

Chris Pollett

Nov 6, 2006.

Outline

- What is a servlet?
- How do I get Tomcat running?
- Example Servlet
- How do I compile/deploy the example?

What is a servlet?

- A *servlet* is a Java program which runs on the server.
- The execution of a servlet is managed by a *servlet container*. This latter may or may not run in the same process as the web server.
- If a HTTP request is received such that the web server determines that a servlet must be called, the server passes the request to the servlet container together with a request and response object.
- These objects encapsulate the system environment as well as provide streams to read from/ write to the client.
- Servlets allow Java user’s to leverage their Java knowledge to the server-side domain.
- Java has always claimed to be very reliable/maintainable.
- Servlet lifecycles do not involve forking processes so are faster than old school CGI.

How do I get Tomcat running?

- The most common free way to use servlet is to download Apache Tomcat from:

  http://tomcat.apache.org/
- For this class we should go for the most recent stable release 5.5.X
- Download the file unzip it and put it where you like.
- You need to set two environment variables in order for Tomcat to know where it is:

  CATALINA_HOME -- the directory for Tomcat

  CLASSPATH -- add the path to servlet-api.jar
- To start Tomcat you can run from the command prompt either the startup.sh (Unix) or startup.bat (Windows) file.
- To stop Tomcat you use either shutdown.sh or shutdown.bat

How can I tell if Tomcat is running?

- You can go to:

  http://localhost:8080/

  and see if the default test page shows up.
- Note you don't need Apache installed to have Tomcat installed.
- If you don't have Apache installed, you can look in $CATALINA_HOME/conf/server.xml

  and change the port to 80 if you want your servlets to serve more like web pages.
- If you are running Apache, you can download mod_jk from the Tomcat site to forward servlet requests to Tomcat.
- If you want to figure out which server is served from where one typically looks at web.xml files.
- In the default set-up the document root to serve from is $CATALINA_HOME/webapps

Remarks about Servlets

- All servlets implement the Servlet interface or extend a class that does. This class is in the package javax.servlet
- Servlet declares three methods: init, service, and destroy which are used to govern the lifecyle of a servlet.
- init and destroy are called only once at the beginning/end of this lifecycle.
- The class HttpServlet extends GenericServlet which implements the Servlet interface.
- A Servlet also declare two methods getServletConfig and getServletInfo, the former to get initialization and start-up parameters, and the latter to get information about itself such as author and version.
- The javax.servlet package also several other useful interfaces: ServletRequest, ServletResponse, ServletInputStream, ServletOutputStream.

More On Servlets

- The most common methods of HttpServlet are:
  - doGet -- to handle HTTP GET requests
  - doPost -- to handle HTTP POST requests
  - doPut -- to handle HTTP PUT request
  - doDelete -- to handle HTTP DELETE request
  - service -- generic handler for all requests
  - init -- initialize resources of the servlet
  - destroy -- to delete resources used by the servlet
  - getServletInfo -- to allow the servlet to provide information about itself.
- The prototype for doGet (service is similar) is:

  protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, java.io.IOException
- request has information about the HTTP headers, form variables sent, etc.
- response has a methods to respond. For instance setContentType, getWriter (to get a PrintWriter to client), etc.

Example Servlet

```
import java.io.*;
import java.text.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

/**
   The simplest possible servlet.

   @author C.Pollett modified from James Duncan Davidson
 */

public class Hello2 extends HttpServlet  //Notice Servlets typically extend HttpSerlvet
{
    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
        throws IOException, ServletException
    {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        out.println("<html>");
        out.println("<head><title>Hello2</title></head>");
        out.println("<body>");
        out.println("<h1>Hello2 Example</h1>");
        out.println("</body>");
        out.println("</html>");
    }
}
```

How do I compile the example?

- To compile the example, you need to have your classpath set up as described a few slides back.
- Then from the command line one could do something like:

```
javac Hello2.java
```

- This creates the class file. But Tomcat won't serve this until you tell it to.
- If you are working out of the servlet-examples directory that comes with Tomcat. Then that directory has a bunch of HTML files in it.
- It also has a WEB-INF directory. This is the directory Tomcat looks into to figure out how to serve a directory.
- Within WEB-INF, there is a file web.xml and a directory *classes*.
- You want to put the class file of your compiled servlet in this classes directory.
- Then you want to edit web.xml

What you need to add to web.xml

```
<servlet>
       <servlet-name>Hello2</servlet-name>
       <servlet-class>Hello2</servlet-class>
  </servlet>
<servlet-mapping>
       <servlet-name>Hello2</servlet-name>
       <url-pattern>/servlet/Hello2</url-pattern>
  </servlet-mapping>
```

Getting Form Data

- Suppose your servlet runs in response to a form.
- Suppose further this form has a variable bob.
- To get the value of bob on can do:

  request.getParameter(“bob”);