

```
1:
2: import java.io.*;
3: import java.net.*;
4: import sun.net.NetworkServer;
5:
6: /**
7:  * A program that echoes to standard output an HTTP
8:  * request, whether GET or POST.
9:  *
10: * Usage: java DiagServer [port]
11: *
12: * If you spawn the server on a port already in use, you will be informed
13: * "Server failed to start: java.net.BindException: Permission denied".
14: * Note that the server's default choice of ports is 80.
15: *
16: * Excerpted from
17: * http://hotwired.lycos.com/webmonkey/99/36/index3a\_page4.html?tw=backend.
18: */
19:
20: public class DiagServer extends NetworkServer {
21:
22:     public static void main(String[] args) {
23:         int port = 80;
24:         try {
25:             port = Integer.parseInt(args[0]);
26:         }
27:         catch (Exception e) {
28:             //You might want to catch a NumberFormatException here,
29:             //or do some more sophisticated command line parsing.
30:         }
31:         try {
32:             DiagServer ds = new DiagServer();
33:             ds.startServer(port);
34:             System.err.println("DiagServer started on port " + port);
35:             System.err.println("");
36:         }
37:         catch (IOException ioe) {
38:             System.err.println("Server failed to start: " + ioe);
39:         }
40:     } //main
41:
42:     public void serviceRequest() throws IOException {
43:         int byteCount = 0;
44:         while (clientIsOpen() && (byteCount < 1000)) {
45:             byte b = (byte) clientInput.read();
```

```
46:         byteCount++;
47:         System.out.write(b);
48:         System.out.flush();
49:     }
50: } //serviceRequest
51: }
```

```
1: <?xml version="1.0" encoding="iso-8859-1"?>
2: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional
.dtd">
3:
4: <!-- a simple XHTML form that submits four parameters via GET -->
5:
6: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
7:   <head>
8:     <title>CSCI E-259</title>
9:   </head>
10:  <body>
11:    <form action="http://icel.fas.harvard.edu:8080" method="get">
12:      foo: <input type="text" name="foo" />
13:      <br />
14:      bar: 0 <input type="radio" name="bar" value="0" /> 1 <input type="radio" name="bar" value="1" />
15:      <br />
16:      baz: <input type="checkbox" name="baz" />
17:      <br />
18:      qux: <select name="qux"><option value="0">0</option><option value="1">1</option></select>
19:      <br />
20:      quux: <textarea name="quux"></textarea>
21:      <br /><br />
22:      <input type="submit" value="Submit GET Request" />
23:    </form>
24:  </body>
25: </html>
```

```
1: #!/usr/local/bin/perl
2: #
3: # http.cgi
4: #
5: # A Perl script that echoes the content of an HTTP request.
6: #
7: # Based on Steven E. Brenner's cgi-lib.pl, available at
8: # http://cgi-lib.berkeley.edu/.
9: #
10:
11: # get request's content type
12: $type = $ENV{'CONTENT_TYPE'};
13:
14: # get request's content length
15: $len = $ENV{'CONTENT_LENGTH'};
16:
17: # get request's request method
18: $meth = $ENV{'REQUEST_METHOD'};
19:
20: # read request
21: if (!defined $meth || $meth eq '') {
22:     $in = $ENV{'QUERY_STRING'};
23: } elsif($meth eq 'GET' || $meth eq 'HEAD') {
24:     $in = $ENV{'QUERY_STRING'};
25: } elsif ($meth eq 'POST') {
26:     if (($got = read(STDIN, $in, $len) != $len))
27:         { die("Short Read: wanted $len, got $got\n"); }
28: } else {
29:     die("Unknown request method: $meth\n");
30: }
31:
32: # echo content of request
33: print <<EOT;
34: Content-type: text/plain
35:
36: $in
37: EOT
```

```
1: <?xml version="1.0" encoding="iso-8859-1"?>
2: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional
.dtd">
3:
4: <!-- a simple XHTML form that submits four parameters via POST -->
5:
6: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
7:   <head>
8:     <title>CSCI E-259</title>
9:   </head>
10:  <body>
11:    <form action="http://icel.fas.harvard.edu:8080" method="post">
12:      foo: <input type="text" name="foo" />
13:      <br />
14:      bar: 0 <input type="radio" name="bar" value="0" /> 1 <input type="radio" name="bar" value="1" />
15:      <br />
16:      baz: <input type="checkbox" name="baz" />
17:      <br />
18:      qux: <select name="qux"><option value="0">0</option><option value="1">1</option></select>
19:      <br />
20:      quux: <textarea name="quux"></textarea>
21:      <br /><br />
22:      <input type="submit" value="Submit POST Request" />
23:    </form>
24:  </body>
25: </html>
```

```
1: <!-- =====
2: Configuration file for Lecture 7
3: Computer Science E-259
4:
5: Be sure to set the value of the Server element's port
6: attribute to an integer between 1024 and 65535, inclusive.
7: Of course, that port cannot already be in use. Nor can it
8: be the same value you choose for the Connector element's port.
9:
10: Also be sure to set the value of the Connector element's port
11: attribute to an integer between 1024 and 65535, inclusive.
12: Of course, that port cannot already be in use. Nor can it
13: be the same value you choose for the Server element's port.
14: ===== -->
15:
16: <Server port="" shutdown="SHUTDOWN">
17:
18:   <Listener className="org.apache.catalina.core.JasperListener" />
19:
20:   <GlobalNamingResources>
21:     <Resource name="UserDatabase" auth="Container"
22:       type="org.apache.catalina.UserDatabase"
23:       description="User database that can be updated and saved"
24:       factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
25:       pathname="conf/tomcat-users.xml" />
26:   </GlobalNamingResources>
27:
28:   <Service name="Catalina">
29:     <Connector port="" protocol="HTTP/1.1"
30:       maxThreads="150" connectionTimeout="20000" />
31:     <Engine name="Catalina" defaultHost="localhost">
32:       <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
33:         resourceName="UserDatabase"/>
34:       <Host name="localhost" appBase="webapps"
35:         unpackWARs="true" autoDeploy="true"
36:         xmlValidation="false" xmlNamespaceAware="false">
37:         <Context path="" docBase="ROOT" debug="0" reloadable="true"/>
38:       </Host>
39:     </Engine>
40:   </Service>
41: </Server>
```

```
1: <?xml version="1.0" encoding="ISO-8859-1"?>
2:
3: <!-- =====
4:     Configuration file for Lecture 7
5:     Computer Science E-259
6: ===== -->
7:
8: <web-app xmlns="http://java.sun.com/xml/ns/j2ee"
9:     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
10:     xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
11:     version="2.4">
12:
13:     <!-- default servlet -->
14:     <servlet>
15:         <servlet-name>default</servlet-name>
16:         <servlet-class>org.apache.catalina.servlets.DefaultServlet</servlet-class>
17:         <init-param>
18:             <param-name>debug</param-name>
19:             <param-value>0</param-value>
20:         </init-param>
21:         <init-param>
22:             <param-name>listings</param-name>
23:             <param-value>>true</param-value>
24:         </init-param>
25:         <load-on-startup>1</load-on-startup>
26:     </servlet>
27:
28:     <!-- JSP servlet -->
29:     <servlet>
30:         <servlet-name>jsp</servlet-name>
31:         <servlet-class>org.apache.jasper.servlet.JspServlet</servlet-class>
32:         <init-param>
33:             <param-name>fork</param-name>
34:             <param-value>>false</param-value>
35:         </init-param>
36:         <init-param>
37:             <param-name>xpoweredBy</param-name>
38:             <param-value>>false</param-value>
39:         </init-param>
40:         <load-on-startup>3</load-on-startup>
41:     </servlet>
42:
43:     <!-- default servlet's mapping -->
44:     <servlet-mapping>
45:         <servlet-name>default</servlet-name>
```

```
46:         <url-pattern>/</url-pattern>
47:     </servlet-mapping>
48:
49:     <!-- JSP servlet's mapping -->
50:     <servlet-mapping>
51:         <servlet-name>jsp</servlet-name>
52:         <url-pattern>*.jsp</url-pattern>
53:     </servlet-mapping>
54:
55:     <!-- session configuration -->
56:     <session-config>
57:         <session-timeout>0</session-timeout>
58:     </session-config>
59:
60:     <!-- MIME types -->
61:     <mime-mapping>
62:         <extension>css</extension>
63:         <mime-type>text/css</mime-type>
64:     </mime-mapping>
65:     <mime-mapping>
66:         <extension>dtd</extension>
67:         <mime-type>application/xml-dtd</mime-type>
68:     </mime-mapping>
69:     <mime-mapping>
70:         <extension>gif</extension>
71:         <mime-type>image/gif</mime-type>
72:     </mime-mapping>
73:     <mime-mapping>
74:         <extension>htm</extension>
75:         <mime-type>text/html</mime-type>
76:     </mime-mapping>
77:     <mime-mapping>
78:         <extension>html</extension>
79:         <mime-type>text/html</mime-type>
80:     </mime-mapping>
81:     <mime-mapping>
82:         <extension>jpg</extension>
83:         <mime-type>image/jpeg</mime-type>
84:     </mime-mapping>
85:     <mime-mapping>
86:         <extension>js</extension>
87:         <mime-type>text/javascript</mime-type>
88:     </mime-mapping>
89:     <mime-mapping>
90:         <extension>pdf</extension>
```



```
91:         <mime-type>application/pdf</mime-type>
92:     </mime-mapping>
93:     <mime-mapping>
94:         <extension>png</extension>
95:         <mime-type>image/png</mime-type>
96:     </mime-mapping>
97:     <mime-mapping>
98:         <extension>xml</extension>
99:         <mime-type>application/xml</mime-type>
100: </mime-mapping>
101: <mime-mapping>
102:     <extension>xsd</extension>
103:     <mime-type>application/xml</mime-type>
104: </mime-mapping>
105: <mime-mapping>
106:     <extension>xsl</extension>
107:     <mime-type>application/xml</mime-type>
108: </mime-mapping>
109: <mime-mapping>
110:     <extension>svg</extension>
111:     <mime-type>image/svg+xml</mime-type>
112: </mime-mapping>
113: <mime-mapping>
114:     <extension>svgz</extension>
115:     <mime-type>image/svg+xml</mime-type>
116: </mime-mapping>
117:
118: <!-- default filenames -->
119: <welcome-file-list>
120:     <welcome-file>index.html</welcome-file>
121:     <welcome-file>index.htm</welcome-file>
122:     <welcome-file>index.jsp</welcome-file>
123: </welcome-file-list>
124:
125: </web-app>
```

```
1: <?xml version="1.0" encoding="UTF-8"?>
2:
3: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional
.dtd">
4:
5: <!-- direct Xalan's environment to browser instead of console -->
6: <% java.io.ByteArrayOutputStream os = new java.io.ByteArrayOutputStream(); %>
7: <% java.io.PrintWriter pw = new java.io.PrintWriter(os); %>
8: <% (new org.apache.xalan.xslt.EnvironmentCheck()).checkEnvironment(pw); %>
9:
10: <html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
11:   <head>
12:     <title>Environment Happiness Page</title>
13:   </head>
14:   <body>
15:     <pre>
16: <% out.print(os); %>
17:     </pre>
18:   </body>
19: </html>
```

```
1: <?xml version="1.0" encoding="iso-8859-1"?>
2: <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional
.dtd">
3:
4: <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
5:   <head>
6:     <title><% out.print("hello, world"); %></title>
7:   </head>
8:   <body>
9:     <% out.print("hello, world"); %>
10:   </body>
11: </html>
```

```
1: <?xml version="1.0"?>
2: <jsp:root xmlns:jsp="http://java.sun.com/JSP/Page"
3:   version="1.2">
4:
5:   <jsp:directive.page contentType="text/html"/>
6:
7:   <![CDATA[<?xml version="1.0" encoding="iso-8859-1"?>]]>
8:   <![CDATA[<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-t
ransitional.dtd">]]>
9:
10:   <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
11:     <head>
12:       <title>
13:         <jsp:scriptlet>
14:           out.print("hello, world");
15:         </jsp:scriptlet>
16:       </title>
17:     </head>
18:     <body>
19:       <jsp:scriptlet>
20:         out.print("hello, world");
21:       </jsp:scriptlet>
22:     </body>
23:   </html>
24:
25: </jsp:root>
```

```
1: <?xml version="1.0"?>
2:
3: <!-- excerpted from jakarta-tomcat-5.0.19/webapps/jsp-examples/xml/ -->
4:
5: <jsp:root xmlns:jsp="http://java.sun.com/JSP/Page"
6:   version="1.2">
7: <jsp:directive.page contentType="text/html"/>
8: <jsp:directive.page import="java.util.Date, java.util.Locale"/>
9: <jsp:directive.page import="java.text.*"/>
10:
11: <jsp:declaration>
12:   String getDateTImeStr(Locale l) {
13:     DateFormat df = SimpleDateFormat.getDateTImeInstance(DateFormat.MEDIUM, DateFormat.MEDIUM, l);
14:     return df.format(new Date());
15:   }
16: </jsp:declaration>
17:
18: <html>
19: <head>
20:   <title>Example JSP in XML format</title>
21: </head>
22:
23: <body>
24: This is the output of a simple JSP using XML format.
25: <br />
26:
27: <div>Use a jsp:scriptlet to loop from 1 to 10: </div>
28: <jsp:scriptlet>
29: // Note we need to declare CDATA because we don't escape the less than symbol
30: <![CDATA[
31:   for (int i = 1; i<=10; i++) {
32:     out.println(i);
33:     if (i < 10) {
34:       out.println(", ");
35:     }
36:   }
37: ]]>
38: </jsp:scriptlet>
39:
40: <!-- Because I omit br's end tag, declare it as CDATA -->
41: <![CDATA[
42:   <br><br>
43: ]]>
44:
45: <div align="left">
```

```
46:   Use a jsp:expression to write the date and time in the browser's locale:
47:   <jsp:expression>getDateTImeStr(request.getLocale())</jsp:expression>
48: </div>
49:
50:
51: <jsp:text>
52:   &lt;p&gt;This sentence is enclosed in a jsp:text element.&lt;/p&gt;
53: </jsp:text>
54:
55: </body>
56: </html>
57: </jsp:root>
```

```
1: <?xml version="1.0" encoding="ISO-8859-1"?>
2:
3: <!-- =====
4:     Configuration file for Lecture 7
5:     Computer Science E-259
6: ===== -->
7:
8: <web-app xmlns="http://java.sun.com/xml/ns/j2ee"
9:     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
10:     xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_5.xsd"
11:     version="2.5">
12:
13:     <!-- servlets -->
14:     <servlet>
15:         <servlet-name>hello</servlet-name>
16:         <servlet-class>Hello</servlet-class>
17:     </servlet>
18:     <servlet>
19:         <servlet-name>parameters</servlet-name>
20:         <servlet-class>Parameters</servlet-class>
21:     </servlet>
22:     <servlet>
23:         <servlet-name>visits</servlet-name>
24:         <servlet-class>Visits</servlet-class>
25:     </servlet>
26:
27:     <!-- servlets' mappings -->
28:     <servlet-mapping>
29:         <servlet-name>hello</servlet-name>
30:         <url-pattern>/servlet/hello</url-pattern>
31:     </servlet-mapping>
32:     <servlet-mapping>
33:         <servlet-name>parameters</servlet-name>
34:         <url-pattern>/servlet/parameters</url-pattern>
35:     </servlet-mapping>
36:     <servlet-mapping>
37:         <servlet-name>visits</servlet-name>
38:         <url-pattern>/servlet/visits</url-pattern>
39:     </servlet-mapping>
40:
41: </web-app>
```

```
1: import java.io.*;
2: import javax.servlet.*;
3: import javax.servlet.http.*;
4:
5: /**
6:  * Outputs "hello, world" in XHTML.
7:  *
8:  * @author Computer Science E-259
9:  *
10:  * @throws IOException, ServletException
11:  */
12:
13: public class Hello extends HttpServlet {
14:
15:     public void doGet(HttpServletRequest request,
16:                       HttpServletResponse response)
17:         throws IOException, ServletException
18:     {
19:         // output XHTML
20:         response.setContentType("text/html");
21:
22:         // grab PrintWriter
23:         PrintWriter out = response.getWriter();
24:
25:         // output "hello, world" in XHTML
26:         out.println("<?xml version=\"1.0\" encoding=\"iso-8859-1\"?>");
27:         out.println("<!DOCTYPE html PUBLIC " +
28:                    "\"-//W3C//DTD XHTML 1.0 Transitional//EN\" " +
29:                    "\"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd\">");
30:         out.println("<html xmlns=\"http://www.w3.org/1999/xhtml\" " +
31:                    "xml:lang=\"en\" lang=\"en\">");
32:         out.println("<body>");
33:         out.println("<t<head>");
34:         out.println("<t<t<title>Hello World!</title>");
35:         out.println("<t</head>");
36:         out.println("<t<body>");
37:         out.println("<t<thello, world");
38:         out.println("<t</body>");
39:         out.println("</html>");
40:     }
41: }
```



```
1: import java.io.*;
2: import java.util.Enumeration;
3: import javax.servlet.*;
4: import javax.servlet.http.*;
5:
6: /**
7:  * Outputs parameters submitted to it via an HTTP GET.
8:  *
9:  * @author Computer Science E-259
10:  *
11:  * @throws IOException, ServletException
12:  */
13:
14: public class Parameters extends HttpServlet {
15:
16:     public void doGet(HttpServletRequest request,
17:                       HttpServletResponse response)
18:     throws IOException, ServletException
19:     {
20:         // output plain text
21:         response.setContentType("text/plain");
22:
23:         // grab PrintWriter
24:         PrintWriter out = response.getWriter();
25:
26:         // grab submitted parameters
27:         Enumeration params = request.getParameterNames();
28:
29:         // output each parameter's name and value
30:         while (params.hasMoreElements())
31:         {
32:             String param = (String) params.nextElement();
33:             out.println(param + ": " + request.getParameter(param));
34:         }
35:     }
36: }
```

```
1: import java.io.IOException;
2: import java.io.PrintWriter;
3: import javax.servlet.ServletException;
4: import javax.servlet.http.HttpServlet;
5: import javax.servlet.http.HttpSession;
6: import javax.servlet.http.HttpServletRequest;
7: import javax.servlet.http.HttpServletResponse;
8:
9: /**
10:  * Reports number of times a given user has visited the page in the
11:  * current session.
12:  *
13:  * @author Computer Science E-259
14:  *
15:  * @throws IOException, ServletException
16:  */
17:
18: public class Visits extends HttpServlet {
19:
20:     public void doGet(HttpServletRequest request,
21:                       HttpServletResponse response)
22:         throws IOException, ServletException
23:     {
24:         // output plain text
25:         response.setContentType("text/plain");
26:
27:         // grab PrintWriter
28:         PrintWriter out = response.getWriter();
29:
30:         // grab user parameter, else err
31:         String user = request.getParameter("user");
32:         if (user == null)
33:             throw new ServletException("missing user parameter");
34:
35:         // grab current session
36:         HttpSession sess = request.getSession();
37:
38:         // grab user's number of visits from session; if null,
39:         // initialize to 0
40:         Integer visits = (Integer) sess.getAttribute(user);
41:         if (visits == null)
42:             visits = new Integer(0);
43:
44:         // update session's count of user's number of visits
45:         sess.setAttribute(user, new Integer(visits.intValue() + 1));
```

```
46:
47:     // say hello
48:     out.println("Hey, " + user + "! " +
49:               "What's it been, " + visits.intValue() + " time(s)?");
50: }
51: }
```