

```
1: // $Id: webserver.java,v 1.26 2013-08-12 12:41:45-07 - - $
2:
3: //
4: // Web server.  Responds with README.html when queried.
5: //
6:
7: import java.io.*;
8: import java.net.*;
9: import java.util.*;
10: import static java.lang.String.*;
11: import static java.lang.System.*;
12:
13: class webserver {
14:     static final String RCSID =
15:         "$Id: webserver.java,v 1.26 2013-08-12 12:41:45-07 - - $";
16:     static options opts = null;
17:     static String[] server_socket_info;
18:
19:     static String ident() {
20:         return String.format ("%s: port %d", opts.progname,
21:                                opts.portnumber);
22:     }
23:
24:     static String get_jarname() {
25:         String jarpath = getProperty ("java.class.path");
26:         int lastslash = jarpath.lastIndexOf ('/');
27:         if (lastslash < 0) return jarpath;
28:         return jarpath.substring (lastslash + 1);
29:     }
30:
31:     static class options {
32:         final String progname = get_jarname();
33:         int portnumber;
34:         options (String[] args) {
35:             try {
36:                 if (args.length != 1) throw new NumberFormatException();
37:                 portnumber = Integer.parseInt (args[0]);
38:             } catch (NumberFormatException exn) {
39:                 quit ("Usage: %s hostname portnumber%n", progname);
40:             }
41:         }
42:     }
43:
44:     static String timenow() {
45:         return format ("%1$tY %1$tb %1$td %1$ta %1$tT.%1$tL",
46:                        Calendar.getInstance());
47:     }
48: }
```

```
49:
50: static void sendline (PrintWriter writer, String line) {
51:     out.printf ("SENT: %s\n", line);
52:     writer.printf ("%s\n", line);
53: }
54:
55: static void sendfile (Socket client, String filename)
56:     throws IOException {
57:     if (filename.length() == 0) filename = "README";
58:     List<String> lines = new LinkedList<String>();
59:     PrintWriter writer = new PrintWriter (client.getOutputStream());
60:     lines.add ("<PRE>");
61:     lines.add ("Server RCS Id: " + RCSID);
62:     try {
63:         InetAddress addr = InetAddress.getLocalHost();
64:         lines.add ("From HostName: " + addr.getHostName());
65:         lines.add ("From HostAddress: " + addr.getHostAddress());
66:     }catch (UnknownHostException exn) {
67:         lines.add ("UnknownHostException: " + exn.getMessage());
68:     }
69:     for (String str: server_socket_info) lines.add (str);
70:     lines.add ("Client InetAddress: " + client.getInetAddress());
71:     lines.add ("Client LocalAddress: " + client.getLocalAddress());
72:     lines.add ("Client LocalPort: " + client.getLocalPort());
73:     lines.add ("Client Port: " + client.getPort());
74:     lines.add ("Page sent: " + timenow());
75:     lines.add ("");
76:     try {
77:         Scanner file = new Scanner (new File (filename));
78:         while (file.hasNextLine()) {
79:             String line = file.nextLine();
80:             lines.add (line);
81:         }
82:     }catch (IOException exn) {
83:         out.printf ("%s\n", exn);
84:         lines.add ("IOException: " + exn.getMessage());
85:     }
86:     int length = 0;
87:     for (String line: lines) length += line.length() + 1;
88:     sendline (writer, "HTTP/1.1 200 OK");
89:     sendline (writer, "Server: " + ident());
90:     sendline (writer, "Content-Type: text/html");
91:     sendline (writer, "Content-Length: " + length);
92:     sendline (writer, "");
93:     for (String line: lines) sendline (writer, line);
94:     writer.flush();
95: }
96:
97: static List<String> getrequest (Socket client) throws IOException {
98:     List<String> lines = new LinkedList<String>();
99:     Scanner input = new Scanner (client.getInputStream());
100:    while (input.hasNextLine()) {
101:        String line = input.nextLine();
102:        out.printf ("RECD: %s\n", line);
103:        lines.add (line);
104:        if (line.length() == 0) break;
105:    }
106:    return lines;
107: }
108:
```

```
109:
110: static class worker implements Runnable {
111:     static int worker_count = 0;
112:     Socket client;
113:     int id = ++worker_count;
114:     worker (Socket client) {
115:         this.client = client;
116:     }
117:     public void run() {
118:         Thread.currentThread().setName ("Worker " + id);
119:         out.printf ("%s: worker %d: starting%n", ident(), id);
120:         try {
121:             List<String> request = getrequest (client);
122:             String filename = request.get(0).split("\\s+")[1];
123:             sendfile (client, filename.substring(1));
124:             client.close();
125:             out.printf ("%s: worker %d: finished%n", ident(), id);
126:         } catch (IOException exn) {
127:             quit ("%s: %s%n", ident(), exn);
128:         }
129:     }
130: }
131:
132: static void quit (String format, Object... params) {
133:     err.printf (format, params);
134:     Set<Map.Entry<Thread, StackTraceElement[]>> entryset =
135:         Thread.getAllStackTraces().entrySet();
136:     for (Map.Entry<Thread, StackTraceElement[]> entry: entryset) {
137:         Thread key = entry.getKey();
138:         StackTraceElement[] value = entry.getValue();
139:         err.printf ("Thread %d: %s%n", key.getId(), key.getName());
140:         for (StackTraceElement stack: value) {
141:             err.printf ("    %s%n", stack.toString());
142:         }
143:     }
144:     exit (1);
145: }
146:
147: public static void main (String[] args) {
148:     opts = new options (args);
149:     try {
150:         ServerSocket socket = new ServerSocket (opts.portnumber);
151:         server_socket_info = new String[2];
152:         server_socket_info[0] = "Server InetAddress: "
153:             + socket.getInetAddress();
154:         server_socket_info[1] = "Server LocalPort: "
155:             + socket.getLocalPort();
156:         out.printf ("%s: waiting for client%n", ident());
157:         for (;;) {
158:             Socket client = socket.accept();
159:             out.printf ("%s: socket.accept OK%n", ident());
160:             Thread worker = new Thread (new worker (client));
161:             worker.start();
162:         }
163:     } catch (IOException exn) {
164:         quit ("%s: %s%n", ident(), exn);
165:     } catch (IllegalArgumentException exn) {
166:         quit ("%s: %s%n", ident(), exn);
167:     }
168: }
169:
```

```
170: }
```

```
1: webserver: port 8888: waiting for client
2: webserver: port 8888: socket.accept OK
3: webserver: port 8888: worker 1: starting
4: RECD: GET / HTTP/1.1
5: RECD: Host: unix1.ic.ucsc.edu:8888
6: RECD: User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:10.0.5) Gecko/20120606
Firefox/10.0.5
7: RECD: Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.
8
8: RECD: Accept-Language: en-us,en;q=0.5
9: RECD: Accept-Encoding: gzip, deflate
10: RECD: DNT: 1
11: RECD: Connection: keep-alive
12: RECD:
13: SENT: HTTP/1.1 200 OK
14: SENT: Server: webserver: port 8888
15: SENT: Content-Type: text/html
16: SENT: Content-Length: 452
17: SENT:
18: SENT: <PRE>
19: SENT: Server RCS Id: $Id: webserver.java,v 1.26 2013-08-12 12:41:45-07 - - $
20: SENT: From HostName: unix1.ic.ucsc.edu
21: SENT: From HostAddress: 128.114.108.152
22: SENT: Server InetAddress: 0.0.0.0/0.0.0.0
23: SENT: Server LocalPort: 8888
24: SENT: Client InetAddress: /128.114.108.152
25: SENT: Client LocalAddress: /128.114.108.152
26: SENT: Client LocalPort: 8888
27: SENT: Client Port: 52327
28: SENT: Page sent: 2013 Aug 13 Tue 19:16:53.993
29: SENT:
30: SENT: This is the README file in this directory.
31: SENT:
32: SENT: $Id: README,v 1.1 2012-05-29 18:09:34-07 - - $
33: webserver: port 8888: worker 1: finished
34: webserver: port 8888: socket.accept OK
35: webserver: port 8888: worker 2: starting
36: RECD: GET /favicon.ico HTTP/1.1
37: RECD: Host: unix1.ic.ucsc.edu:8888
38: RECD: User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:10.0.5) Gecko/20120606
Firefox/10.0.5
39: RECD: Accept: image/png,image/*;q=0.8,*/*;q=0.5
40: RECD: Accept-Language: en-us,en;q=0.5
41: RECD: Accept-Encoding: gzip, deflate
42: RECD: DNT: 1
43: RECD: Connection: keep-alive
44: RECD:
45: java.io.FileNotFoundException: favicon.ico (No such file or directory)
46: SENT: HTTP/1.1 200 OK
47: SENT: Server: webserver: port 8888
48: SENT: Content-Type: text/html
49: SENT: Content-Length: 414
50: SENT:
51: SENT: <PRE>
52: SENT: Server RCS Id: $Id: webserver.java,v 1.26 2013-08-12 12:41:45-07 - - $
53: SENT: From HostName: unix1.ic.ucsc.edu
54: SENT: From HostAddress: 128.114.108.152
55: SENT: Server InetAddress: 0.0.0.0/0.0.0.0
56: SENT: Server LocalPort: 8888
57: SENT: Client InetAddress: /128.114.108.152
58: SENT: Client LocalAddress: /128.114.108.152
```

```
59: SENT: Client LocalPort: 8888
60: SENT: Client Port: 52328
61: SENT: Page sent: 2013 Aug 13 Tue 19:16:54.046
62: SENT:
63: SENT: IOException: favicon.ico (No such file or directory)
64: webserver: port 8888: worker 2: finished
65: webserver: port 8888: socket.accept OK
66: webserver: port 8888: worker 3: starting
67: RECD: GET /favicon.ico HTTP/1.1
68: RECD: Host: unix1.ic.ucsc.edu:8888
69: RECD: User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:10.0.5) Gecko/20120606
Firefox/10.0.5
70: RECD: Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.
8
71: RECD: Accept-Language: en-us,en;q=0.5
72: RECD: Accept-Encoding: gzip, deflate
73: RECD: DNT: 1
74: RECD: Connection: keep-alive
75: RECD:
76: java.io.FileNotFoundException: favicon.ico (No such file or directory)
77: SENT: HTTP/1.1 200 OK
78: SENT: Server: webserver: port 8888
79: SENT: Content-Type: text/html
80: SENT: Content-Length: 414
81: SENT:
82: SENT: <PRE>
83: SENT: Server RCS Id: $Id: webserver.java,v 1.26 2013-08-12 12:41:45-07 - - $
84: SENT: From HostName: unix1.ic.ucsc.edu
85: SENT: From HostAddress: 128.114.108.152
86: SENT: Server InetAddress: 0.0.0.0/0.0.0.0
87: SENT: Server LocalPort: 8888
88: SENT: Client InetAddress: /128.114.108.152
89: SENT: Client LocalAddress: /128.114.108.152
90: SENT: Client LocalPort: 8888
91: SENT: Client Port: 52329
92: SENT: Page sent: 2013 Aug 13 Tue 19:16:54.054
93: SENT:
94: SENT: IOException: favicon.ico (No such file or directory)
95: webserver: port 8888: worker 3: finished
```