

ShowResumePreview.java

```
1 package coreservlets;
2
3 import java.io.*;
4 import javax.servlet.*;
5 import javax.servlet.annotation.*;
6 import javax.servlet.http.*;
7 import java.util.*;
8
9 /** Servlet that handles previewing and storing resumes
10  * submitted by job applicants.
11  * <p>
12  * From <a
13  * href="http://courses.coreservlets.com/Course-Materials/">the
14  * coreservlets.com tutorials on servlets, JSP, Struts, JSF, Ajax,
15  * GWT, and Java</a>.
16  */
17 @WebServlet("/show-resume-preview")
18 public class ShowResumePreview extends HttpServlet {
19     @Override
20     public void doPost(HttpServletRequest request,
21                        HttpServletResponse response)
22         throws ServletException, IOException {
23         response.setContentType("text/html");
24         PrintWriter out = response.getWriter();
25         if (request.getParameter("previewButton") != null) {
26             showPreview(request, out);
27         } else {
28             storeResume(request);
29             showConfirmation(request, out);
30         }
31     }
32
33     /** Shows a preview of the submitted resume. Takes
34     * the font information and builds an HTML
35     * style sheet out of it, then takes the real
36     * resume information and presents it formatted with
37     * that style sheet.
38     */
39 }
```

ShowResumePreview.java

```
38
39 private void showPreview(HttpServletRequest request,
40                          PrintWriter out) {
41     String headingFont = request.getParameter("headingFont");
42     headingFont = replaceIfMissingOrDefault(headingFont, "");
43     int headingSize =
44         getSize(request.getParameter("headingSize"), 32);
45     String bodyFont = request.getParameter("bodyFont");
46     bodyFont = replaceIfMissingOrDefault(bodyFont, "");
47     int bodySize =
48         getSize(request.getParameter("bodySize"), 18);
49     String fgColor = request.getParameter("fgColor");
50     fgColor = replaceIfMissing(fgColor, "BLACK");
51     String bgColor = request.getParameter("bgColor");
52     bgColor = replaceIfMissing(bgColor, "WHITE");
53     String name = request.getParameter("name");
54     name = replaceIfMissing(name, "Lou Zer");
55     String title = request.getParameter("title");
56     title = replaceIfMissing(title, "Loser");
57     String email = request.getParameter("email");
58     email =
59         replaceIfMissing(email, "contact@hot-computer-jobs.com");
60     String languages = request.getParameter("languages");
61     languages = replaceIfMissing(languages, "<I>None</I>");
62     String skills = request.getParameter("skills");
63     skills = replaceIfMissing(skills, "Not many, obviously.");
64     String docType =
65         "<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.0 \" +
66         "Transitional//EN\">\n";
67     out.println
68         (docType +
69         "<HTML><HEAD><TITLE>Resume for " + name + "</TITLE>\n" +
70         makeStyleSheet(headingFont, headingSize,
71                        bodyFont, bodySize,
72                        fgColor, bgColor) + "\n" +
73         "</HEAD>\n" +
74         "<BODY>\n" +
75         "<CENTER>\n" +
76         "<SPAN CLASS=\"HEADING1\">" + name + "</SPAN><BR>\n" +
```

ShowResumePreview.java

```
77     "<SPAN CLASS=\"HEADING2\">" + title + "<BR>\n" +
78     "<A HREF=\"mailto:" + email + "\">" + email +
79     "</A></SPAN>\n" +
80     "</CENTER><BR><BR>\n" +
81     "<SPAN CLASS=\"HEADING3\">Programming Languages" +
82     "</SPAN>\n" +
83     makeList(languages) + "<BR><BR>\n" +
84     "<SPAN CLASS=\"HEADING3\">Skills and Experience" +
85     "</SPAN><BR><BR>\n" +
86     skills + "\n" +
87     "</BODY></HTML>");
88 }
89
90 /** Builds a cascading style sheet with information
91  * on three levels of headings and overall
92  * foreground and background cover. Also tells
93  * Internet Explorer to change color of mailto link
94  * when mouse moves over it.
95  */
96
97 private String makeStyleSheet(String headingFont,
98                               int heading1Size,
99                               String bodyFont,
100                              int bodySize,
101                              String fgColor,
102                              String bgColor) {
103     int heading2Size = heading1Size*7/10;
104     int heading3Size = heading1Size*6/10;
105     String styleSheet =
106         "<STYLE TYPE=\"text/css\">\n" +
107         "<!--\n" +
108         ".HEADING1 { font-size: " + heading1Size + "px;\n" +
109         "             font-weight: bold;\n" +
110         "             font-family: " + headingFont +
111         "             \"Arial, Helvetica, sans-serif;\n" +
112         "}\n" +
113         ".HEADING2 { font-size: " + heading2Size + "px;\n" +
114         "             font-weight: bold;\n" +
115         "             font-family: " + headingFont +
```

ShowResumePreview.java

```
116         "Arial, Helvetica, sans-serif;\n" +
117     "}\n" +
118     ".HEADING3 { font-size: " + heading3Size + "px;\n" +
119     "    font-weight: bold;\n" +
120     "    font-family: " + headingFont +
121     "    \"Arial, Helvetica, sans-serif;\n" +
122     "}\n" +
123     "BODY { color: " + fgColor + ";\n" +
124     "    background-color: " + bgColor + ";\n" +
125     "    font-size: " + bodySize + "px;\n" +
126     "    font-family: " + bodyFont +
127     "    \"Times New Roman, Times, serif;\n" +
128     "}\n" +
129     "A:hover { color: red; }\n" +
130     "-->\n" +
131     "</STYLE>";
132     return(styleSheet);
133 }
134
135 /** Replaces null strings (no such parameter name) or
136  * empty strings (e.g., if textfield was blank) with
137  * the replacement. Returns the original string otherwise.
138  */
139
140 private String replaceIfMissing(String orig,
141                                String replacement) {
142     if ((orig == null) || (orig.trim().equals("")))) {
143         return(replacement);
144     } else {
145         return(orig);
146     }
147 }
148
149 // Replaces null strings, empty strings, or the string
150 // "default" with the replacement.
151 // Returns the original string otherwise.
152
153 private String replaceIfMissingOrDefault(String orig,
154                                           String replacement) {
```

ShowResumePreview.java

```
155     if ((orig == null) ||
156         (orig.trim().equals("")) ||
157         (orig.equals("default"))) {
158         return(replacement);
159     } else {
160         return(orig + ", ");
161     }
162 }
163
164 // Takes a string representing an integer and returns it
165 // as an int. Returns a default if the string is null
166 // or in an illegal format.
167
168 private int getSize(String sizeString, int defaultSize) {
169     try {
170         return(Integer.parseInt(sizeString));
171     } catch (NumberFormatException nfe) {
172         return(defaultSize);
173     }
174 }
175
176 // Given "Java,C++,Lisp", "Java C++ Lisp" or
177 // "Java, C++, Lisp", returns
178 // "<UL>
179 //   <LI>Java
180 //   <LI>C++
181 //   <LI>Lisp
182 // </UL>"
183
184 private String makeList(String listItems) {
185     StringTokenizer tokenizer =
186         new StringTokenizer(listItems, ", ");
187     String list = "<UL>\n";
188     while(tokenizer.hasMoreTokens()) {
189         list = list + "   <LI>" + tokenizer.nextToken() + "\n";
190     }
191     list = list + "</UL>";
192     return(list);
193 }
```

ShowResumePreview.java

```
194
195 /** Show a confirmation page when they press the
196  *   "Submit" button.
197  */
198
199 private void showConfirmation(HttpServletRequest request,
200                             PrintWriter out) {
201     String title = "Submission Confirmed.";
202     out.println(ServletUtilities.headWithTitle(title) +
203                "<BODY>\n" +
204                "<H1>" + title + "</H1>\n" +
205                "Your resume should appear online within\n" +
206                "24 hours. If it doesn't, try submitting\n" +
207                "again with a different email address.\n" +
208                "</BODY></HTML>");
209 }
210
211 /** Why it is bad to give your email address to
212  *   untrusted sites.
213  */
214
215 private void storeResume(HttpServletRequest request) {
216     String email = request.getParameter("email");
217     putInSpamList(email);
218 }
219
220 private void putInSpamList(String emailAddress) {
221     // Code removed to protect the guilty.
222 }
223 }
224
```