

ServletUtilities.java

```
1 package coreservlets;
2
3 import javax.servlet.http.*;
4
5 /** Some simple time savers. Static methods. */
6
7 public class ServletUtilities {
8     public static String headWithTitle(String title) {
9         return("<!DOCTYPE html>\n" +
10             "<html>\n" +
11             "<head><title>" + title + "</title></head>\n");
12     }
13
14     /** Read a parameter with the specified name, convert it
15      * to an int, and return it. Return the designated default
16      * value if the parameter doesn't exist or if it is an
17      * illegal integer format.
18      */
19
20     public static int getIntParameter(HttpServletRequest request,
21                                     String paramName,
22                                     int defaultValue) {
23         String paramString = request.getParameter(paramName);
24         int paramValue;
25         try {
26             paramValue = Integer.parseInt(paramString);
27         } catch (Exception nfe) { // null or bad format
28             paramValue = defaultValue;
29         }
30         return(paramValue);
31     }
32
33     /** Reads param and converts to double. Default if problem. */
34
35     public static double getDoubleParameter(HttpServletRequest
36 request,
37                                     String paramName,
38                                     double defaultValue) {
39         String paramString = request.getParameter(paramName);
```

ServletUtilities.java

```
39     double paramValue;
40     try {
41         paramValue = Double.parseDouble(paramString);
42     } catch (Exception nfe) { // null or bad format
43         paramValue = defaultValue;
44     }
45     return(paramValue);
46 }
47
48 /** Replaces characters that have special HTML meanings
49  * with their corresponding HTML character entities.
50  * Specifically, given a string, this method replaces all
51  * occurrences of
52  * {@literal
53  * '<' with '&lt;', all occurrences of '>' with
54  * '&gt;', and (to handle cases that occur inside attribute
55  * values), all occurrences of double quotes with
56  * '&quot;' and all occurrences of '&' with '&amp;'.
57  * Without such filtering, an arbitrary string
58  * could not safely be inserted in a Web page.
59  * }
60 */
61
62 public static String filter(String input) {
63     if (!hasSpecialChars(input)) {
64         return(input);
65     }
66     StringBuilder filtered = new StringBuilder(input.length());
67     char c;
68     for(int i=0; i<input.length(); i++) {
69         c = input.charAt(i);
70         switch(c) {
71             case '<': filtered.append("&lt;"); break;
72             case '>': filtered.append("&gt;"); break;
73             case '"': filtered.append("&quot;"); break;
74             case '&': filtered.append("&amp;"); break;
75             default: filtered.append(c);
76         }
77     }
```

ServletUtilities.java

```
78     return(filtered.toString());
79 }
80
81 private static boolean hasSpecialChars(String input) {
82     boolean flag = false;
83     if ((input != null) && (input.length() > 0)) {
84         char c;
85         for(int i=0; i<input.length(); i++) {
86             c = input.charAt(i);
87             switch(c) {
88                 case '<': flag = true; break;
89                 case '>': flag = true; break;
90                 case '"': flag = true; break;
91                 case '&': flag = true; break;
92             }
93         }
94     }
95     return(flag);
96 }
97
98 private ServletUtilities() {} // Uninstantiatable class
99 }
100
```