

CSE 5345/7345

Assignment 1. (To-Be-Graded) Revised: Sept. 16, 2009

Due: Sep 25 -- Distance Sep 28.

A. Write a Servlet that accepts two parameters named int1 and int2 and returns XML (text) of the form:

```
<result>xxxx</result>
```

where xxxx is the sum of the two integer parameters.

The servlet should gracefully handle improper input as follows:

Missing parameter(s): return

```
<err><msg>missing parameter int1</msg></err>
```

or

```
<err><msg>missing parameter int2</msg></err>
```

or (if both are missing)

```
<err><msg>missing parameter int1</msg>  
    <msg>missing parameter int2</msg>  
</err>
```

Non-integer parameter(s): return

```
<err><msg>int1 is not an integer</msg></err>  
or  
<err><msg>int2 is not an integer</msg></err>  
or  
<err><msg>int1 is not an integer</msg>  
    <msg>int2 is not an integer</msg>  
</err>
```

B. Weather Servlet using RegExps for Screen Scraping

Write a servlet that takes one parameter called zipcode and returns XML of the form:

```
<result>
  <zipcode>75230</zipcode>
  <updated>Sep 13 10:45 a.m. PT</updated>
  <temp>72</temp>
  <humidity>55%</humidity>
</result>
```

If there is no such parameter, return:

```
<err><msg>missing parameter zipcode</msg></err>
```

If the parameter is not a valid zipcode, return:

```
<err><msg>zipcode=xxxxx is not a valid zipcode</msg></err>
```

In order to view XML in a browser, the servlet should tell the Response object to set the **Content-type** header to **text/xml**

DO:

- a) Use **regular expressions** to extract the data you need from the web page you obtain using the **URLConnection** class.
- b) Experiment with RegExps with an online tool (e.g. <http://regexpal.com/>)
- c) When you get the RegExp working the way you want it, test it with a Java program. Don't forget to escape your forward slashes.

SUBMIT/DEPLOY:

So that your servlets can be easily located, please name your project with your userid prefixed by A1. For example, in NetBeans, my project name would be A1coyle which will then create A1coyle.war when you build your project. Each project will have two servlets. Name the servlet for part a, AddServlet and the servlet for part b, WeatherServlet. Then, to test your servlets, the following should work: (replacing coyle with your userid).

```
http://neo.engr.smu.edu:8080/A1coyle/AddServlet?int1=876&int2=555
http://neo.engr.smu.edu:8080/A1coyle/WeatherServlet?zipcode=90210
```

When you have tested your servlet code from a browser, package the two source files in a zip file and a readme file that contains the links to your servlets (just in case there is a problem) and submit to Blackboard using the Assignments Link for Assignment 1. Make certain to add a JavaDoc `@author` tag with your name.

Don't forget to have your servlet set up the HTTP response to tell the browser that you are delivering text/xml. To do this add the line:

```
response.setContentType("text/xml");
```

Feel free to use the following code to access www.weather.com to obtain weather information.

```
1  package httpconnect;
2  import java.io.BufferedReader;
3  import java.io.IOException;
4  import java.io.InputStreamReader;
5  import java.io.OutputStreamWriter;
6  import java.net.HttpURLConnection;
7  import java.net.MalformedURLException;
8  import java.net.ProtocolException;
9  import java.net.URL;
10
11  public class SimpleHTTPRequest {
12
13      /**
14       * @param args
15       */
16      public static void main(String[] args) {
17          HttpURLConnection connection = null;
18          OutputStreamWriter wr = null;
19          BufferedReader rd = null;
20          StringBuilder sb = null;
21          String line = null;
22
23          URL serverAddress = null;
24
25          try {
26              String urlStr =
27 "http://www.weather.com/outlook/travel/businesstraveler/local/90210?
28 lswe=90210&lwsa=Weather36HourBusinessTravelerCommand&from=searchbox_
29 localwx";
30              serverAddress = new URL(urlStr);
31              //set up out communications stuff
32              connection = null;
33
34              //Set up the initial connection
35              connection =
36                  (HttpURLConnection)serverAddress.openConnection();
37              connection.setRequestMethod("GET");
38              connection.setDoOutput(true);
39              connection.setReadTimeout(10000);
40
```

```
41         connection.connect();
42
43         //read the result from the server
44         rd = new BufferedReader(new
45             InputStreamReader(connection.getInputStream()));
46
47         // create StringBuilder to incrementally append text
48         sb = new StringBuilder();
49
50         while ((line = rd.readLine()) != null) {
51             sb.append(line + '\n');
52         }
53
54         // we now have a string to analyze or output
55         System.out.println(sb.toString());
56
57     } catch (MalformedURLException e) {
58         e.printStackTrace();
59     } catch (ProtocolException e) {
60         e.printStackTrace();
61     } catch (IOException e) {
62         e.printStackTrace();
63     }
64     finally
65     {
66         //close the connection, set all objects to null
67         connection.disconnect();
68         rd = null;
69         sb = null;
70         wr = null;
71         connection = null;
72     }
73 }
74 }
75 }
```