

Assignment #9

Java Source Code Lister

Filenames: ListFiles.java, ListSource.java

Write a servlet which will display a directory listing of all of the .java files in your classes folder (on mislab). When the user clicks one of the listed file names, display the contents of that .java file.

1. In Eclipse, you can use your Servlets project, and simply create two new classes in that project. You have already added servlet.jar to that project, so this will save you a step.
2. Create a new class named ListFiles. It should be a subclass of HttpServlet.
3. You will need two import statement. You need to import

```
javax.servlet.http.*
```

```
java.io.*
```

4. You will need to write a `doGet()` method.
5. Your `doGet()` method should set the contentType of your response to text/html.
6. Since you will want to generate some output to the browser, you need to call your response object's `getWriter()` method:

```
PrintWriter out=response.getWriter();
```

7. You can instantiate a File object to get a listing of the files in the current directory. Once you get that File object, you can call its `listFiles()` method to get an array of file names.

```
File dir=new File(".");
```

```
File[] filesList=dir.listFiles();
```

"." indicates "the current directory".

8. Now use a for loop to step through the elements of the filesList array. Process each element in the array (each element is a File object).

You can determine whether a File object is a file or a directory by calling its `isFile()` method.

You can get a File object's filename by calling its `getName()` method.

Once you have a filename (a String object), you can use its `matches()` method to see if the filename ends with `.java`.

```
if(filename.matches(".*\\.java"))
```

If the filename ends in `.java`, generate a line of HTML that includes an anchor tag. Clicking on the anchor tag should submit a request for your ListSource servlet, passing it the current filename. You want to generate something like:

```
<a href='abc123.ListSource?name=Test.java'>Test</a>
```

9. Update `web.xml`, in your WEB-INF folder, to provide support for both ListFiles and ListSource.

10. When you get this servlet running, it will display a list of the `.java` files in your classes folder, displaying them as anchor tags.

11. Now write `ListSource.java`. It should also be a subclass of `HttpServlet`.

12. `ListSource` should set the `contentType` of your response to `text/plain` (not `text/html`, because you don't want the browser to try to render this Java source code).

13. Retrieve the submitted file name.

14. Instantiate a `FileReader` object, then a `BufferedReader` object.

15. Loop, reading a record from the `.java` file, and displaying it to the browser (as plain text).

16. In your `index.html`, change the href in the anchor tag for this assignment to request your ListFiles servlet.

Revision #2

Created 9 January 2019 23:59:37 by Aaron Kimbrell

Updated 10 April 2019 17:23:45 by Aaron Kimbrell