

**CSE 297 – Web Systems Programming – Fall 2014**  
**Homework Assignment 1 - Mini HTTP Server**  
**Due Monday, 9/1/14, 11:59pm**

**Objective**

To gain a better understanding of the HTTP protocol and how it might be implemented in a web server.

**Collaboration Reminder**

- \* You must submit your own work.
- \* In particular, you may not:
  - Show your code to any of your classmates.
  - Look at or copy anyone else's code.
  - Copy material found on the Internet.
  - Work together on an assignment

**Description**

The assignment is to take the Mini HTTP Server provided as a NetBeans project and add a couple of features to the server.

**Feature 1: Log file**

All industrial strength web servers keep log files that record each request to which the server responds. For example, here are some sample log entries from the Apache HTTP server:

```
127.0.0.1 - - [01/May/2012:19:12:57 -0400] "GET /xampp/splash.php HTTP/1.1" 200
1325 "-" "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:11.0) Gecko/20100101 Firefox/11.0"
127.0.0.1 - - [01/May/2012:19:12:57 -0400] "GET /xampp/xampp.css HTTP/1.1" 200 4178
"http://localhost/xampp/splash.php" "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:11.0)
Gecko/20100101 Firefox/11.0"
127.0.0.1 - - [01/May/2012:19:12:57 -0400] "GET /xampp/img/blank.gif HTTP/1.1" 200
43 "http://localhost/xampp/splash.php" "Mozilla/5.0 (Windows NT 6.1; WOW64;
rv:11.0) Gecko/20100101 Firefox/11.0"
```

Your task is to create a log file for the Mini HTTP Server:

1. When the server starts up, create a file named access.log in the server root directory – append if the file already exists.
2. Each time a request comes in add an entry to the log file containing the IP address of the client, the first line of the request and the response code and then flush the output stream:

```
127.0.0.1 - "GET /xampp/splash.php HTTP/1.1" 200
```

additional header lines to the response issued by the server. Use the following as a reference for HTTP headers:

[http://en.wikipedia.org/wiki/List\\_of\\_HTTP\\_header\\_fields](http://en.wikipedia.org/wiki/List_of_HTTP_header_fields)

**Feature 2: Header Lines**

**Additional Header #1: Content-Type header**

Implement the Content-Type response header by looking at the file extension of the resource being requested and setting the appropriate content type. For example, if the resource being requested is flower.png, then you should add the following header to the list of headers:

```
Content-Type: image/png
```

A list of content types (otherwise known as internet media types or MIME types) can be found here:

[http://en.wikipedia.org/wiki/MIME\\_type](http://en.wikipedia.org/wiki/MIME_type)

Implement this header for the following file types:

- Images in gif, jpeg, or png format (.gif, .jpg, .png extension)
- Pdf file (.pdf extension)
- Excel spreadsheet (.xls or .xlsx extension)
- HTML file (.htm or .html extension)

If the resource has none of the above extensions, then don't include a Content-Type header.

### **Additional Header #2: Last-Modified header**

Implement the Last-Modified header by looking up the modified date of the requested file and adding the Last-Modified header (with the date formatted in "HTTP-date" format!). For example:

```
Last-Modified: Tue, 15 Nov 1994 12:45:26 GMT
```

You can use the `java.io.File` class to determine the last modified date of the file.

You can use the `java.text.SimpleDateFormat` class to format the date.

(continued)

### **Procedure**

1. Download the Mini HTTP Server project from Coursesite and unzip it.
2. Rename the project folder and add your Lehigh id: Ex. MiniHTTPServer-jaf207
3. Open the project in NetBeans, make the changes and test.
4. Fill in the comment at the top of the Main.java file with your name, etc.
5. Do a Clean on the project.
6. Zip up the whole project directory and upload to Coursesite.