**­434/534 Network Design**

**Project #1: Web Server Solution Guidance**

This section describes optional guidance to help you complete this project. Using an IDE and a VCS system is useful for this project. I recommend using IntelliJ IDEA[[1]](#footnote-1) and GitHub.You should use Java language level 17 (LTS) for this project.[[2]](#footnote-2)

1. Implement the skeleton server program from the slides, which will take care of the connection setup and disconnect steps of the program. Ensure that the server program can be run according to the assignment specifications (the first argument is port number, the second one is the address of the evaluationWeb provided in the assignment). java MyWebServer.java <<port\_number>> ~/evaluationWeb
2. Implement the section of the program that will receive the entire request from the socket and parse the information that you need to fulfill the GET or HEAD request. This section could be a **request parser**.
   1. You could create an HTTPRequest object that takes the string request you pull off the socket and the root path as parameters in the constructor, tokenizes the request string and takes out and stores the important information (the Command, the Path, the If-Modified-Since date (if one is specified)).
   2. At this point, you can truncate the path (if necessary), concatenate it to the end of the root directory, and add “index.html” if the path leads to a directory.
   3. Once you have this information, you move onto the next objective, which is to take some action based on the request.
   4. At this point it’s good to check for some of your HTTP errors. If the command or path is missing, or if the optional If-Modified-Since date and time is not parsable, then the error code should be changed to reflect a 400 Bad Request. If the command value isn’t “GET” or “HEAD”, the error code should be changed to reflect a command 501 Not Implemented.
3. Implement the part of the server that takes some action based upon the information within the particular HTTP request object. This could be thought of as a **request handler**.
   1. This request handler will receive the request object and will attempt find the appropriate file.
   2. Create an HTTP response object, and populate its header parameters (“Status-Line”, "Date", "Server", "Last-Modified", and "Content-Length"). After this step, you will have the header portion of your HTTP response.
   3. If the request object specified a HEAD request, you can send the response object back via the DataOutputStream with only the header fields.
   4. If the request object specified a GET request, you will *also* need to return the file via the DataOutputStream, in addition to the header.
      1. If the file specified as input was null, then we have a 404 Not Found error. If the file hasn’t been modified since the If-Modified-Since date and time, then change the error code to reflect the file 304 Not Modified. When handling the errors, you can create a basic HTML body to give the error message to the client (which will be sent instead of a file).
4. You can test your server’s functionality according to the instructions below.

**434/534 Network Design**

**Project #1: Web Server Grading**

This section describes exactly how grading will be performed. If you complete all these tests successfully, you will get full points for the project.

Java language level 17 (LTS) will be used on the grading machine. For all the tests below, replace “<<port\_number>>” with your desired port.

Tests:

1. **(5 points) Does your java code run?**

I will download your Java code and will use the following java command to run your webserver:

java MyWebServer.java <<port\_number>> ~/evaluationWeb

You get these 5 points if your code runs without additional troubleshooting.

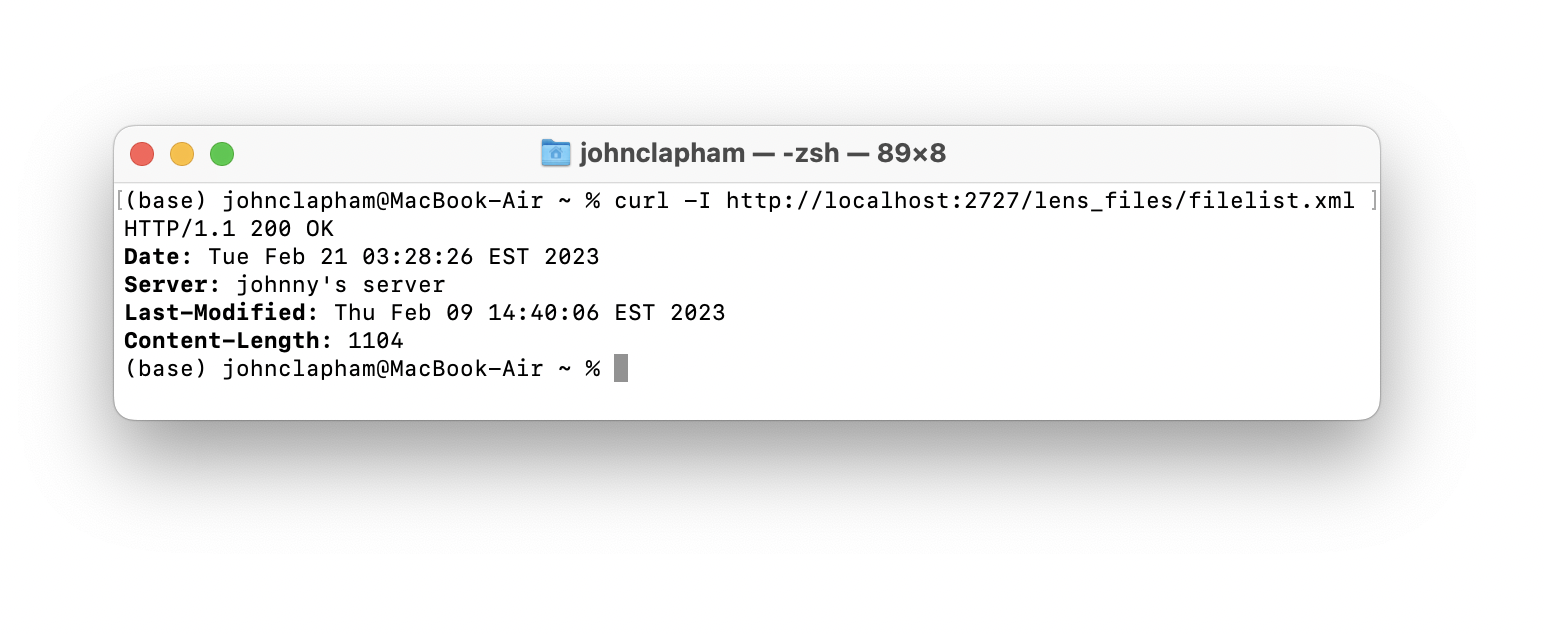
1. **(25 points) Can your webserver receive a HEAD request?**

I will use a basic terminal shell and execute the following CURL[[3]](#footnote-3) command:

To test HEAD:

curl -I http://localhost:6788/lens\_files/filelist.xml

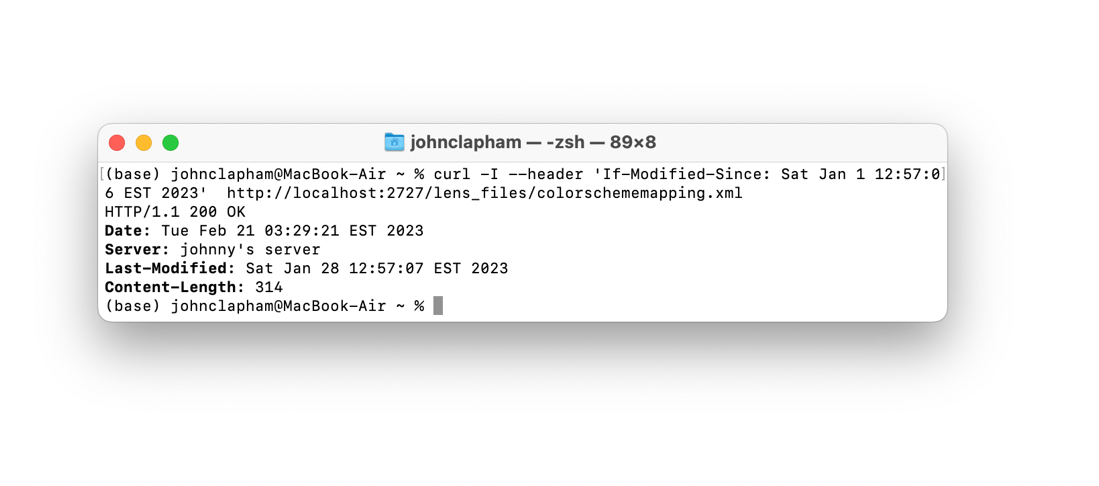
You get full points if your head request returns all headers specified in instructions (Status, Date, Server, Last-Modified, Content-Length). 5 points for each of the 5 fields.



1. **(10 points) Does your server support the optional “If-Modified-Since” header?**

To test “If-Modified-Since” header:

curl -I --header 'If-Modified-Since: Sat Jan 1 12:57:06 EST 2023' http://localhost:6788/lens\_files/colorschememapping.xml



I will change the date to a future date (yellow highlight) to experiment whether your server can make the appropriate response and not return a file.

curl -I --header 'If-Modified-Since: Sat Dec 25 12:57:06 EST 2023' http://localhost:6788/lens\_files/colorschememapping.xml



You get 10 more points if you return the appropriate error code 304.

1. **(5 points) Does your server return 400 (bad request), 404 (file not found), 501 (not implemented) errors?**

I will attempt a 400 bad request (e.g. a malformed date, shown in yellow)

curl -I --header 'If-Modified-Since: Satsdfec 25 12:57:06 EST 2023' http://localhost:6788/lens\_files/colorschememapping.xml

I will attempt a 404 file not found request

curl -I http://localhost:6788/imaginary.file

I will attempt a 501 not implemented request

curl -X POST http://localhost:6788/

You get full points long as the errors are visible in the returned content body (not only in the header) in some way.

Examples:



1. **(15 points) Can your webserver make the correct response over Google Chrome?**

I will test that your webserver can load a sample website from Dr. Zhou. (LENS laboratory website).

I will use the following link and enter it into the browser:

http://localhost:6788/

This should load the index.html file. I will click on the “LENS LAB” button to load the lens.html file.

If your server loads both webpages without any errors/ missing images (.jpg, .gif, .png)/ missing text/ fonts/ colors/ etc., you will get full credit will not have to complete the following task 6 and automatically get the full 50 points.

Demo: <https://youtu.be/5z93AgWYFbg> (Video also available in videos/Test 5 Website Grading Tutorial.mp4)

1. **Which files can you load successfully? (35 points)**

If your server fails to load the sample website with all its images and files, I will test individual image types (.gif / .jpg / .png ), and a variety of text-based files (.txt / .html / .xml/ .css). I will use the following URL prompts:

http://localhost:<<port\_number>>/examples/penguin.jpg

http://localhost:<<port\_number>>/examples/apple\_pie.png

http://localhost:<<port\_number>>/examples/exampleGET.gif

http://localhost:<<port\_number>>/examples/text.txt

http://localhost:<<port\_number>>/examples/filelist.xml

http://localhost:<<port\_number>>/examples/mm\_health\_nutr.css

http://localhost:<<port\_number>>/examples/bbc/

You get 5 points per success

Demo: <https://youtu.be/BkOkvz_G-pI> (Video also available in videos/Test 6 Files Grading Tutorial.mp4)

1. **(5 points) Is there documentation?**

These points will be awarded for comments in code, descriptive naming of functions & variables, an organized code-base and other good practices to make it easy to analyze your code.

**Total:**

* 5% - Code runs
* 25% - HEAD request and fields
* 10% - “If-Modified-Since”
* 5% - Misc errors (400 / 404 / 501)
* 15% - Web browser test
* 35% - File test
* 5% - Documentation

1. <https://www.jetbrains.com/idea/> [↑](#footnote-ref-1)
2. You can specify this in File -> Project Structure -> Language Level. [↑](#footnote-ref-2)
3. https://curl.se/docs/httpscripting.html [↑](#footnote-ref-3)