3.7 Assignment: Building a Web Server

Instructions

Estimated Effort Required: 8-10 hours

Learning Outcome: Students will understand how web-servers interact with the file system and the network by integrating networking and file creation/access to their HTTP parser.

Assignment Goal: You will write an HTTP server. This server will also serve as the basis for assignment 3, so it will be very hard to complete assignment 3 if assignment 2 is not completed.

Preface: The following words should be interpreted as per RFC 2119 for this assignment: must, must not, required, shall, shall not, should, should not, recommended, may, and optional.

Description:

In this assignment, you will adapt your HTTP parser from assignment #2 to function as an HTTP server. This will entail adding network programming so that your server will be accessible to a network and serve up files. Your server will also support HTTPS as an optional configuration.

Requirements:

1. Python is recommended for its extensive string parsing libraries.
    a. You may use any programming language..
    b. The instructor may not be able to assist with specific programming errors if languages other than C, C++, PHP, Java, or Python are chosen.

2. You must not use any libraries built for parsing HTTP requests. You may use traditional string parsing libraries and functionality.
    - a. If you have questions about whether a library is permissible, please ask.
3. Your server may be named whatever you wish.
4. Your server should have exactly four command line arguments:
    - a. The IP address to listen on.
    - b. The port to listen on.
    - c. A path to an x509 (.pem or .crt file) file to use for HTTPS encryption. If this is not provided, you may assume that HTTP is to be used rather than HTTPS.
        - a. An HTTPS request if no PEM is specified (ie: HTTPS is not enabled) should be handled gracefully. You may determine how to handle that case. One example may be to return an HTTP 500 response code.
        - b. You do not need to enforce the file extensions above
    - d. A path to the private key paired with the x509 certificate.
        - a. The server should not execute if the third argument is included but not the fourth argument.
5. You must support GET, POST, PUT, DELETE, and one other method of your choice. HEAD or CONNECT are suggested.
6. You must support the following HTTP response codes:
    - a. 200 - Sent, with the contents of the file requested, if no other error codes are returned.
    - b. 201 - Sent after a successful PUT command. The contents of the file created should be in the body of a 201 response and the file path (relative to the web root) should be returned in the location header.
    - c. 400 - Sent if the HTTP parser indicates the HTTP request was not valid.
    - d. 403 - Sent if the HTTP server does not have permission to access the requested file
    - e. 404 - Sent if the HTTP server cannot find the file requested
    - f. 411 - Sent if content-length is not set for POST requests
    - g. 500 - Sent as a default if the HTTP server or parser experiences an error from which it cannot recover while processing a request (for example, if the server experiences an exception)
    - h. 501 - Sent if the method specified in the HTTP request is not implemented in the server.

i. 505 - Sent if the HTTP server does not support the version of HTTP specified in the file (1.1 and 1.0 are permissible

6. Your server must log the first line of all valid requests into a log file.

7. Your code must be documented(commented) using the following guidelines.

a. https://medium.freecodecamp.org/code-comments-the-good-the-bad-and-the-ugly-be9cc65fbf83

8. Your code must execute on Ubuntu 18.04.

Deliverables

1. A zip file that...

a. Must include a readme text file which provides:

i. Instructions on how to install any additional dependencies your code may have, including compilers/interpreters and development environments that are necessary to run your code.

ii. Instructions for how to compile your code

iii. Instructions on how to execute your code

b. Should include a script that can be run to install any additional dependencies your code may have, if applicable.

c. May include a script that can be used to compile your code, if applicable.

d. Must include your source code

Submissions

No submissions yet. Drag and drop to upload your assignment below.

Drop files here, or click below!

Upload Choose Existing

You can upload files up to a maximum of 1 GB.

‹ ›

Activity Details

Task: Submit to complete this assignment

Due February 10 at 11:55 PM