

Reboot HTTP

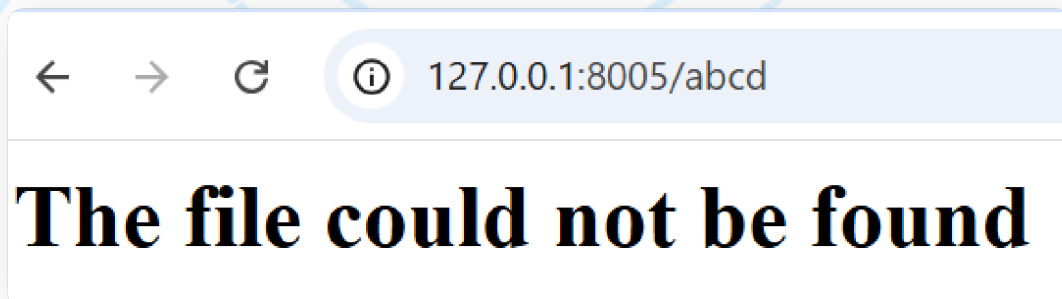
In this unit we will dive deeper into capabilities of network servers.

To do that in a familiar setting, we will re-use the HTTP and Weather servers from last year, slightly modified. This exercise will be a refresher on HTTP in general and this specific HTTP server code in general, which we will continue using in future topics.

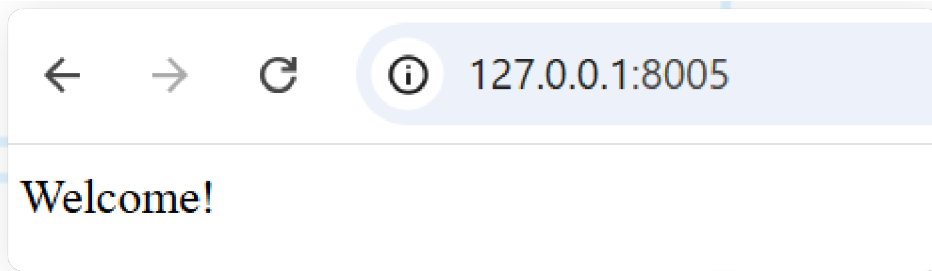
Instructions

Create a new directory (this will be the server directory) and save attached file `httpserver.py` in it. Run it and complete the following tasks; you will need to adjust the directory structure and files, and edit parts of the server code.

1. Browse with an internet browser (internet explorer, edge, etc.) to a non-existing page in the server, e.g. `http://127.0.0.1:8005/abcd`. Make sure the server listens on the correct port, 8005. The browser should display the text "The file could not be found".



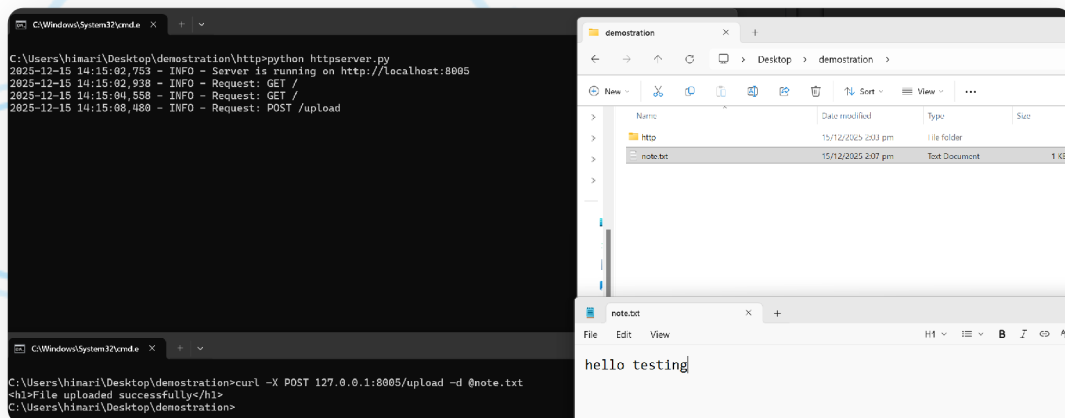
2. Browse with an internet browser (internet explorer, edge, etc.) to the server's index page in the URLs `http://127.0.0.1:8005`, `http://127.0.0.1:8005/`, `http://127.0.0.1:8005/index.html` (all should function exactly the same). Browsing to the web server will serve the file `index.html` from the static files directory, displaying the text "Welcome!" in the browser.



The code block that achieve this is as follow:

```
# serve_static_files() function
if path == '/':
    path = '/index.html'
```

3. In a temporary directory (not inside the server directory), create a text file **note.txt** and enter some text in it. Use the command line utility **curl** to upload it to the server: **curl -X POST 127.0.0.1:8005/upload -d @note.txt** Make sure the response includes "File uploaded successfully", and that the file was correctly created in the server's upload directory.

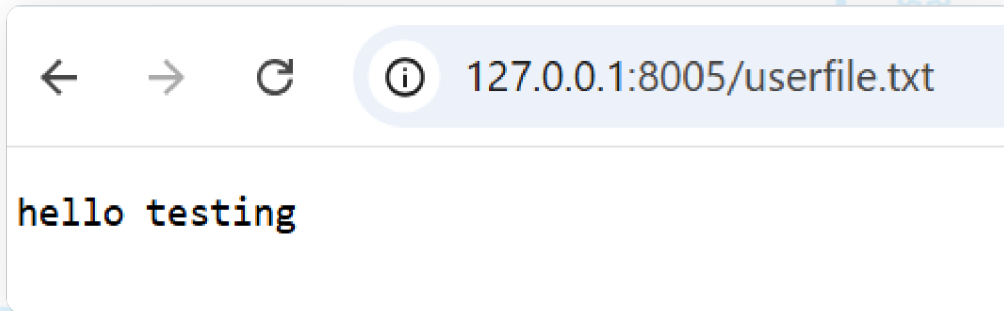


Note: The content inside **note.txt** should replace the content in **uploads/userfile.txt**.

Task 1:

- Add handling for GET requests to path **/userfile.txt**. It should return the text file you previously uploaded, from the server's upload directory. Make sure it's working by browsing to **http://127.0.0.1:8005/userfile.txt**

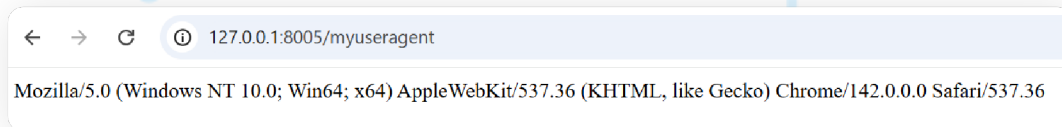
Example output:



Task 2:

- Add handling for GET requests to path `/myuseragent`. It should return plain text with the request's User-Agent string, taken from the request headers. Make sure it works by browsing to the server with different browsers and curl.

Example output:



To submit

Submit a ZIP file containing server directory. The server must be called `httpserver.py` and reside in the root of the ZIP archive, alongside any of the required files/directories.

