

Relazione Progetto

Traccia 2 – Python Web Server

Aldini Alessandro

0000941385

Alam MD Shokot

0000933749

Valdifiori Davide

0000941631

JULY 2021

Introduzione

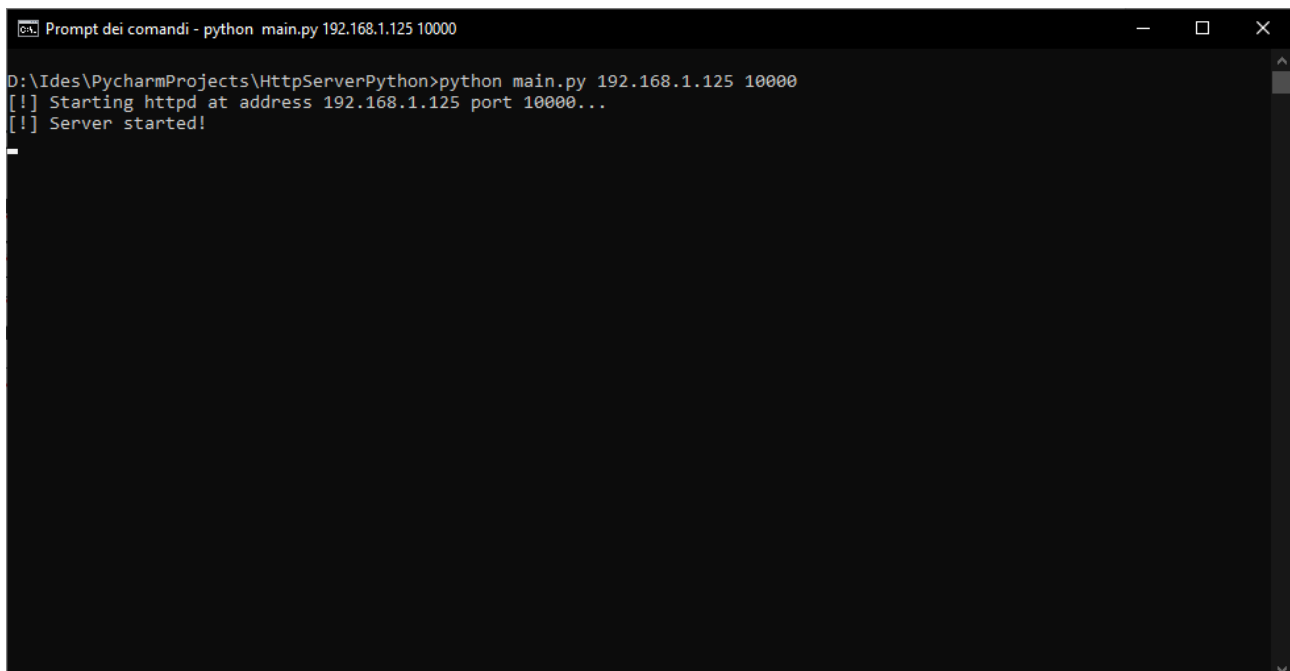
Il progetto consiste nel realizzare un server HTTP in linguaggio Python che permetta a più utenti in contemporanea di visionare il sito web di un ospedale.

La pagina principale deve mostrare la lista dei servizi erogati dall'azienda ospedaliera e dei link di riferimento alle relative pagine dedicate, a cui si potrà accedere solo dopo essersi autenticati.

Dettagli implementativi

Il programma è composto da un unico file `main.py`, che può essere avviato aggiungendo come parametri ip e porta su cui avviare il server HTTP

`python main.py 192.168.1.125 10000`

A screenshot of a Windows command prompt window. The title bar reads "Prompt dei comandi - python main.py 192.168.1.125 10000". The command prompt shows the following text:

```
D:\Ideas\PycharmProjects\HttpServerPython>python main.py 192.168.1.125 10000
[!] Starting httpd at address 192.168.1.125 port 10000...
[!] Server started!
```

Nel momento in cui il programma viene eseguito, il server viene avviato utilizzando ip e porta inseriti come argomenti, o 127.0.0.1 e 10001 come parametri di default se non è fornito nessun argomento.

Dal momento in cui il programma è avviato, il server viene aperto e resta in attesa per connessioni da parte di browser web finché non viene chiesta la terminazione da terminale tramite CTRL+C.

Tramite il controllo dell'eccezione `KeyboardInterrupt` viene correttamente gestita la terminazione del server.

```

Prompt dei comandi

D:\Ides\PycharmProjects\HttpServerPython>python main.py 192.168.1.125 10000
[!] Starting httpd at address 192.168.1.125 port 10000...
[!] Server started!
[!] Ctrl+C received, shutting down http server...
[!] Server closed! Goodbye :)

D:\Ides\PycharmProjects\HttpServerPython>

```

Nel programma sono presenti due classi

- **AuthHTTPRequestHandler**: gestisce le richieste di pagine web
- **AuthHTTPServer**: gestisce il server HTTP

Ogni richiesta di pagine web viene gestita dalla funzione `do_GET` che invierà la pagina al client se non è necessaria l'autorizzazione, altrimenti verrà richiesto di inserire le credenziali.

```

Prompt dei comandi - python main.py 192.168.1.125 10000

D:\Ides\PycharmProjects\HttpServerPython>python main.py 192.168.1.125 10000
[!] Starting httpd at address 192.168.1.125 port 10000...
[!] Server started!
[+] Client is asking for page /
Returning the page without authentication...
192.168.1.125 - - [21/Jul/2021 12:11:00] "GET / HTTP/1.1" 200 -
192.168.1.125 - - [21/Jul/2021 12:11:00] "GET /style-base.css HTTP/1.1" 200 -
192.168.1.125 - - [21/Jul/2021 12:11:00] "GET /style.css HTTP/1.1" 200 -
192.168.1.125 - - [21/Jul/2021 12:11:00] "GET /favicon.ico HTTP/1.1" 200 -
[+] Client is asking for page /radiologia.html
Receiving auth header: None
Auth is none, asking for login...
192.168.1.125 - - [21/Jul/2021 12:11:21] "GET /radiologia.html HTTP/1.1" 401 -
[+] Client is asking for page /radiologia.html
Receiving auth header: Basic YWRtaW46YWRtaW4=
Decoded auth Credentials: admin - *****
Auth is correct, returning the web page...
192.168.1.125 - - [21/Jul/2021 12:11:27] "GET /radiologia.html HTTP/1.1" 200 -

```

Invia le pagine senza login

Viene richiesto al client di inserire le credenziali

Le credenziali sono state inserite e sono corrette, viene quindi restituita la pagina



Le credenziali per l'autenticazione sono contenute all'interno del file passwords.txt che viene caricato all'inizio del programma.

Al momento del login viene controllata la password fornita dal client per vedere se corrisponde alla password dell'utente inserito.

Librerie utilizzate

- SimpleHTTPRequestHandler from http.server
- ThreadingTCPServer from socketserver
- HTTPStatus from http
- base64
- sys

Repository

<https://github.com/Valdi1111/HttpServerPython>