

PROGETTO PROGRAMMAZIONE DI RETI

Nome cognome: Davide Denicolò

Email: davide.denicolo@studio.unibo.it

N° Matricola: 0000879677

Traccia 2: Python Web Server

Si immagini di dover realizzare un Web Server in Python per una azienda ospedaliera. I requisiti del Web Server sono i seguenti:

- Il web server deve consentire l'accesso a più utenti in contemporanea
- La pagina iniziale deve consentire di visualizzare la lista dei servizi erogati dall'azienda ospedaliera e per ogni servizio avere un link di riferimento ad una pagina dedicata.
- L'interruzione da tastiera (o da console) dell'esecuzione del web server deve essere opportunamente gestita in modo da liberare la risorsa socket.
- Nella pagina principale dovrà anche essere presente un link per il download di un file pdf da parte del browser
- Come requisito facoltativo si chiede di autenticare gli utenti nella fase iniziale della connessione.

Scopo del progetto:

Il progetto ha lo scopo di creare un web server in python dando dei collegamenti a dei servizi ospedalieri (semplici collegamenti ipertestuali), in più si è dedicata una sezione notizie relative alla salute prese dal Corriere tramite Feed RSS.

Nella home-page sono presenti collegamenti all':

- Ausl Romagna
- Ausl Bologna
- Ausl Parma
- Ausl Modena

- Ausl Ferrara
- Ausl Reggio Emilia

Inoltre sono presenti altri link per altri servizi come:

- Farmacie
- Farmacie online
- Parafarmacie
- Ambulatori privati

Inoltre è possibile rimanere aggiornati tramite il Corriere grazie a una sezione “Notizie” dedicata alla salute. E’ possibile raggiungerla cliccando in alto nell’apposito link.

Esso ha lo scopo di prendere le informazioni recenti grazie ai Feed RSS forniti dal Corriere e l’ora in cui l’aggiornamento avviene.

Per il download di un file generico basterà cliccare sull’apposito link in alto nella home-page.

Implementazione:

Il software è scritto in python (con Pycharm) che grazie a delle librerie permette di lavorare dinamicamente con file html, gestire richieste http, creare web server ecc.

Le librerie usate sono: signal, sys, http.server, feedparser e socketserver, datetime, pytz.

Datetime e Pytz: forniscono l’orario attuale.

Tutto è gestito tramite multithreading, ciò significa che è possibile aprire il server su due pagine diverse (anche browser diversi, ovviamente) e le richieste verranno gestite correttamente. Inoltre la deallocazione delle risorse avviene correttamente.

PS: il server lavora sulla porta 8081 per evitare qualsiasi disagio, magari la porta 8080 è già occupata da un altro processo.

Strutture dati utilizzate:

Le strutture dati utilizzate in questo progetto sono molto semplici, per creare le pagine html in maniera dinamica mi sono avvalso di semplici variabili che memorizzano del codice HTML preso dinamicamente con i feed rss, inoltre ho memorizzato anche il codice CSS (se pur minimale) per gestire la grafica. Stesso discorso per memorizzare il codice HTML statico (per es. il footer, è sempre quello), ho usato sempre variabili stringhe.

E' presente un handler che gestisce le richieste GET verso il server e restituisce:

404 - se il file non è stato trovato

200 - se il file è stato trovato

Utilizzo:

1) Aprire una shell, entrare nella cartella coi file e digitare:

```
python3 serv.py [PORT]
```

Il parametro [PORT] è opzionale, di default lavora nella porta 8081.

2) aprire un motore di ricerca, digitare nella barra in alto:

```
localhost:8081/
```

Di default viene aperta la pagina index.html, in caso si cerchi una pagina inesistente verrà restituito errore 404, in alternativa code 200.

3) Navigare il sito.

Qualora sia tutto funzionante il server lo notificherà nella console.

Ultime impressioni e possibili miglioramenti:

Il progetto da me svolto risulta molto minimale in termini di grafica che è quasi del tutto assente, tuttavia le richieste minime sono state rispettate. Il progetto è migliorabile di certo in termini grafici, oppure era possibile creare ulteriori sezioni notizie o incorporare la home-page dell'Ausl Romagna in una zona apposita del sito. Al di là dei possibili miglioramenti a parer mio il progetto risulta svolto correttamente, implementando le basi della programmazione di reti.

MIT License

Copyright (c) 2021 Davide Denicolò

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.