

RELAZIONE TECNICA

Progetto di laboratorio – Programmazione di RetiTraccia 1 – Web Server
+ Sito Web Statico

Nome: Matteo Santarelli

Email: matteo.santarelli6@studio.unibo.it

Matricola: 0001142479

1. Introduzione

Questo progetto ha come obiettivo la realizzazione di un server HTTP minimale in Python, utilizzando i socket, per la gestione e la pubblicazione di un sito web statico. La traccia prevede che il server sia in grado di rispondere a richieste GET e servire almeno tre pagine HTML, oltre a gestire risposte di errore per risorse non trovate (404).

2. Struttura del progetto

Il progetto è composto da due elementi principali:- Il file `server.py`, che contiene tutta la logica del server HTTP.- La cartella `www/`, contenente i file statici del sito web (HTML/CSS).Il server è configurato per ascoltare sulla porta `8080` e accetta connessioni su `localhost`.

3. Funzionamento del server

Il cuore del server è rappresentato dalla funzione `handle_request`, che processa la richiesta HTTP in arrivo e genera una risposta coerente. Ad esempio, il controllo del metodo supportato avviene tramite il seguente codice:

```
if method != 'GET': return "HTTP/1.1 405 Method Not  
Allowed\r\n\r\nMethod Not Allowed"
```

Questo garantisce che solo le richieste GET siano gestite, rispondendo con un errore 405 in caso contrario.

Se la risorsa richiesta è la root (`/`), il server reindirizza a `home.html`:

```
if path == '/': path = '/home.html'
```

Il server costruisce il percorso del file, controlla se esiste, e in caso positivo lo legge e lo invia come risposta. In caso contrario, viene generato un errore 404 come mostrato:

```
if not os.path.isfile(filepath): log_request(method, path,  
404) return "HTTP/1.1 404 Not Found\r\n\r\n404 Not Found"
```

4. Funzionalità realizzate

- Gestione delle richieste HTTP di tipo GET- Risposta con codice 200 per file esistenti- Risposta con codice 404 per file mancanti- Logging delle richieste (metodo, percorso, codice risposta)- Gestione corretta dei MIME type per file HTML, CSS, immagini, ecc.

5. Estensioni opzionali implementate

Sono state aggiunte alcune funzionalità opzionali tra cui:- Logging dettagliato delle richieste, con timestamp- Supporto automatico ai MIME types tramite il modulo `mimetypes`- Reindirizzamento della root verso una homepage predefinita

6. Conclusione

Il progetto soddisfa tutti i requisiti della traccia 1, inclusa la gestione delle richieste, le risposte appropriate e l'organizzazione di un sito statico funzionante. Il codice è scritto in modo modulare e documentato, facilitando l'estensione futura del server con nuove funzionalità.