

# WebServer

Federico Raffoni

July 2021

## Contents

<b>1</b>	<b>Analisi</b>	<b>3</b>
1.1	Requisiti . . . . .	3
<b>2</b>	<b>design</b>	<b>4</b>
2.1	Architettura . . . . .	4
2.2	Thread . . . . .	4
2.3	Autenticazione . . . . .	5
<b>3</b>	<b>Struttura sito</b>	<b>6</b>
3.1	Richiedi una visita . . . . .	6
3.2	Area riservata . . . . .	6
<b>4</b>	<b>Avvio webServer</b>	<b>6</b>

# 1 Analisi

Si vuole sviluppare un WebServer per la gestione del sito di uno studio dentistico. Della piattaforma sarà possibile visualizzare i servizi offerti dallo studio con relativa descrizione. L'utente avrà la possibilità di richiedere una visita con il medico. Una volta compilato il relativo form l'utente attenderà la chiamata per prenotare la visita. È inoltre disponibile una sezione dedicata ai consigli del medico, dove ogni utente potrà scaricare i vari pdf allegati e consultarli a piacimento. Tramite il sito web dovrà essere possibile anche l'accesso alla sezione di admin mediante l'inserimento delle credenziali che fornirà accesso al medico per visualizzare eventuali richieste di visita.

## 1.1 Requisiti

- Il web server deve consentire l'accesso a più utenti in contemporanea
- La pagina iniziale deve consentire di visualizzare la lista dei servizi erogati dallo studio dentistico 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
- Dovrà essere possibile gestire non solo richieste di tipo GET ma anche altri tipi quali le POST.
- Possibilità di prenotazione di una visita

## 2 design

### 2.1 Architettura

Il web server si occupa essenzialmente di ricevere, gestire e rispondere a tutte le richieste in arrivo da un client, in questo caso un web browser. Il funzionamento del server 'e dettato dalla classe Handler, questa classe implementa i metodi più utili della sua classe padre (SimpleHTTPRequestHandler ), ovvero do POST() e do GET().

### 2.2 Thread

Per quanto riguarda la gestione dei threads abbiamo dal lato client il browser tramite il quale farà le richieste al web server, che considereremo come un singolo thread X, dove X indica il numero della richiesta. Il thread del client X quindi farà una richiesta HTTP al main thread del webserver, il quale si occuperà di generare un nuovo thread X il cui compito sarà quelli di gestire la richiesta del client X. Lato server abbiamo quindi un thread principale sempre attivo, mentre in presenza di N client che stanno facendo una richiesta avremo ulteriori N thread dedicati a gestire le richieste dei clients.

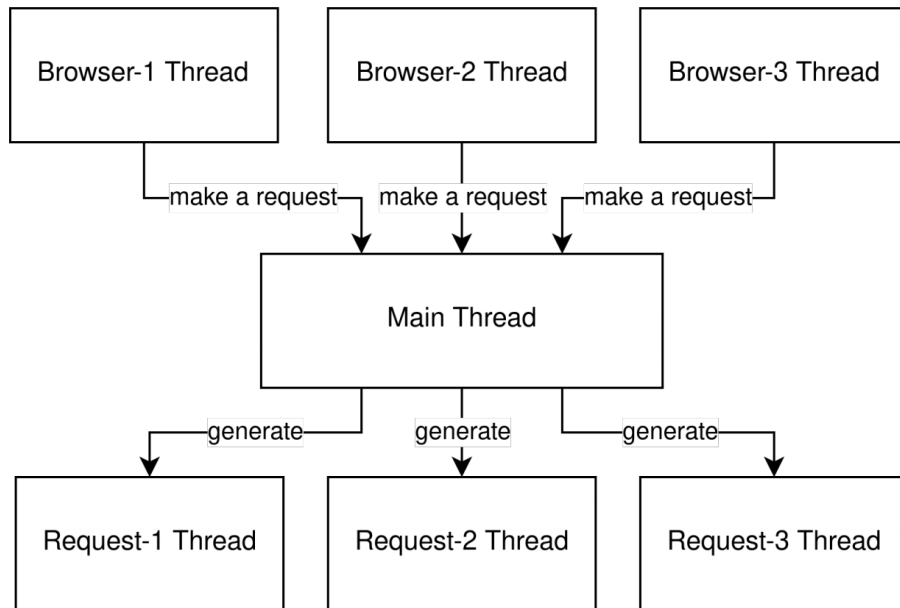


Figure 1: Schema dei threads dell'applicazione sia a livello client che a livello server

## 2.3 Autenticazione

Per quanto riguarda l'autenticazione si è deciso di utilizzare la Basic Access Authentication, avremo quindi una semplice autenticazione senza la necessità di usare cookies. Questa autenticazione sarà richiesta solo in determinate pagine dell'applicazione e una volta inserita avrà valore fino alla fine della sessione, per sloggarci dovremo quindi o cancellare la cache oppure avviare un'altra sessione del browser.

Le credenziali di accesso a queste pagine saranno quelle dell'amministratore e saranno specificate nel file 'credentials.json' in formato json in modo da rendere semplice la lettura. Nel caso specifico dell'applicazione l'autenticazione servirà per accedere alla sezione dedicata al controllo delle richieste in attesa di visita

## 3 Struttura sito

Nella Homepage troviamo tutti i riferimenti alle varie pagine del sito suddivise tra

- Servizi
- Una pagina per richiedere una visita
- Consigli del medico
- area riservata ai medici

### 3.1 Richiedi una visita

In questa pagina troviamo un form. Dopo averlo riempito e aver cliccato "invia" verrà inviata al server una richiesta POST, il server salverà tutti i campi all'interno di un file chiamato request.html

### 3.2 Area riservata

Questa è l'unica area del sito che necessita di login per accedervi. è possibile entrare nella pagina con username e password: admin admin Se si inseriscono username e password errati il server risponderà con un errore 401 e chiederà nuovamente di fare il login

## 4 Avvio webServer

Per l'avvio dell'applicazione sarà necessario digitare il comando:

```
python3 WebServer.py
```

Il server si avvierà sulla porta 8080.