

# Progetto Reti 2021

Tommaso Patriti

29 maggio 2021

## 1 Introduzione

Web Server basato su python. La struttura di gestione delle richieste, prende spunto da "Django Framework"

## 2 Descrizione

Per avviare il server usare il comando "python3 server.py", specificare una porta come argomento, se non viene specificata quella di default sarà la 8080. Una volta avviato il server sarà possibile interrogare il sito presente su localhost.

Struttura sito e autenticazione necessaria		
/		login
/appuntamenti	basic access authentication	
/referti	basic access authentication	
/turni	basic access authentication	
/media/		None
/ambulatori		None
/info		None
/login		None
/logout		None

Sulla console sarà possibile visualizzare tutte le richieste e i relativi Handlers

## 3 Dettagli Implementativi

### 3.1 GET

Di default le richieste vengono controllate per mezzo di una whitelist. Se presenti nella lista contenuta nel file "url.py" vengono associate alla relativa funzione che si occupa di formattare il file associato (views.py). Nel caso di risorse multimediali, possono essere messe una cartella, (di default "media") che non effettua controlli e restituisce direttamente il file

### 3.2 Pattern lista url

Qui si può vedere il pattern della lista di tuple utilizzato per gestire i file

- 1) Il path che l'utente deve inserire per inviare la richiesta
- 2) Il percorso del file
- 3) La funzione associata alla richiesta

```
import views
urlpatterns = [
    ("login", "html/login.html", views.login), ... ]
```

### 3.3 POST

Le richieste vengono filtrate e accettate solamente se presenti in whitelist, nel file url, dopo di che la gestione e la manipolazione dei dati viene delegata alla funzione associata alla richiesta nel file "views.py"

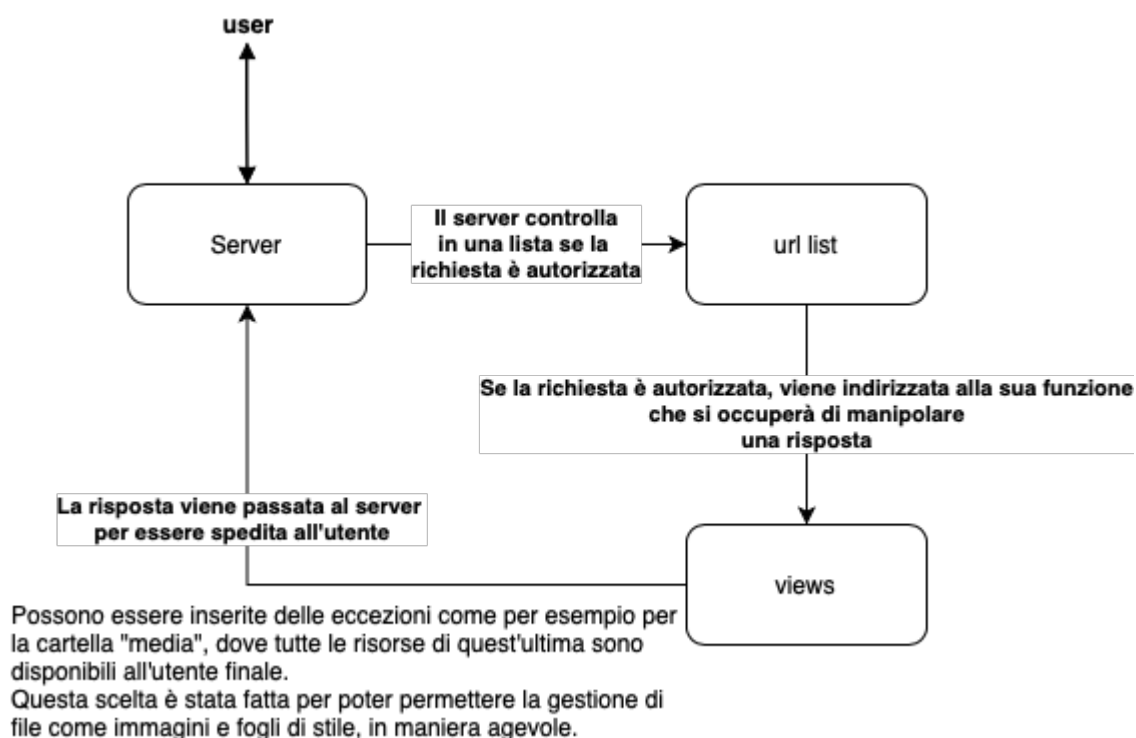


Figura 1: Gestione richieste

## 4 LOGIN

Ho deciso di implementare due tipi di login

- Basic access authentication
- Classico, gestito per mezzo di POST e Cookies

In entrambi i casi ho usato dei Decorator da implementare sulle funzioni in “views.py” per bloccare l’accesso alla risorsa tramite login. Gli username e password per praticità sono nel file “password\_validator.py”.

## 5 Formattazione file

Nei file html, sono presenti delle stringhe `ess:{var}`, che servono per sfruttare la formattazione offerta da python

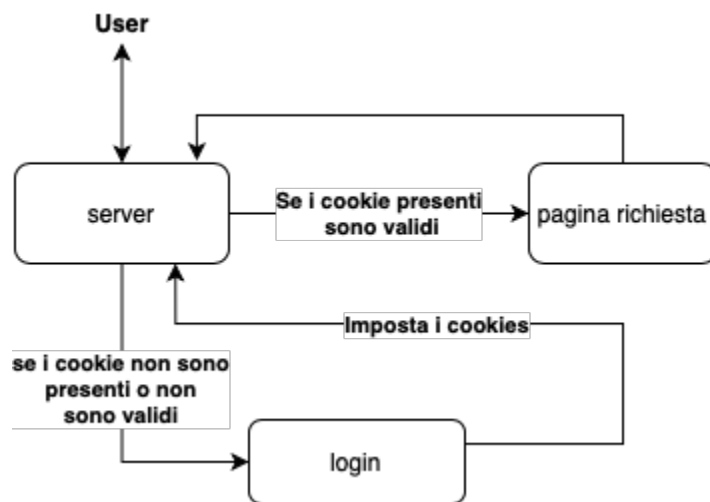


Figura 2: Gestione login

Esempio utilizzo decorator per gestione login

```

@basic_access_authentication
def index(request, page):
    return page.format(username=get_username(request))
  
```

Se non si è loggati non si potrà accedere alla pagina principale, impostando questo decorator su altre funzioni, si limiterà l’accesso anche a queste ultime. utilizzando i cookies, basterà loggarsi una volta per sbloccarle tutte. La basic access authentication è separato dal login normale, quindi il log di uno è indipendente dell’altro

## 6 Librerie utilizzate

- http
- socketserver
- signal
- base64

## 7 Link utili

Codice Sorgente : <https://github.com/Ro0t-set/Progetto-Reti-2021>