

Roteiro Flask

Sumário

- [Roteiro Flask](#)
 - [Sumário](#)
 - [Pré-requisitos](#)
 - [1.1. Python e pip](#)
 - [1.2. Apache2](#)
 - [1.3. Mod-WSGI](#)
 - [Fase de Teste](#)
 - [Passo 1: Configuração Inicial](#)
 - [Passo 2: Criar o Aplicativo Flask](#)
 - [Passo 3: Configuração do WSGI](#)
 - [Passo 4: Permissões](#)
 - [Passo 5: Configuração do Apache](#)
 - [Fase de Projeto](#)
 - [Passo 1: Estrutura do Projeto](#)
 - [Passo 2: Mover Projeto para o Servidor](#)
 - [Passo 3: Permissões](#)
 - [Extra: Configuração SSL](#)
-

Pré-requisitos

Certifique-se de ter os seguintes componentes instalados na sua máquina:

1.1. Python e pip

```
sudo apt install python3  
sudo apt install python3-pip
```

1.2. Apache2

```
sudo apt install apache2 -y
```

1.3. Mod-WSGI

```
sudo apt install libapache2-mod-wsgi-py3
```

Atualização do sistema:

```
sudo apt update && sudo apt upgrade -y
```

Fase de Teste

Passo 1: Configuração Inicial

Crie um diretório em `/home/user/` para os testes:

```
mkdir ~/teste
```

No diretório criado, configure um ambiente virtual (venv):

```
cd ~/teste  
sudo virtualenv env  
source env/bin/activate
```

Com a venv ativa, o prompt deve ser semelhante a:

```
(env) user@maquina:~/teste$
```

Instale o Flask:

```
pip3 install flask
```

Passo 2: Criar o Aplicativo Flask

Crie o arquivo `app.py` com o seguinte conteúdo:

```
from flask import Flask  
app = Flask(__name__)  
  
@app.route('/')  
def index():  
    return 'Hello World'
```

Teste o aplicativo com:

```
flask run
```

Você verá algo como:

```
* Debug mode: off
WARNING: This is a development server. Do not use it in a production
deployment.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

Passo 3: Configuração do WSGI

Crie o arquivo `app.wsgi` com o seguinte conteúdo:

```
import sys
sys.path.insert(0, '/home/user/teste')
from app import app as application
```

A estrutura do diretório deve ser:

```
/teste
  /env
  app.py
  app.wsgi
```

Passo 4: Permissões

Dê permissão para o Apache acessar o diretório de teste:

```
sudo chown :www-data /home/user/teste
```

Passo 5: Configuração do Apache

Navegue para o diretório de configuração do Apache:

```
cd /etc/apache2/sites-available/
```

Crie o arquivo de configuração do site (`teste.conf`):

```
sudo nano teste.conf
```

Adicione o seguinte conteúdo:

```
<VirtualHost *:80>
    ServerAdmin admin@example.com
    ServerName example.com
    ServerAlias www.example.com
    DocumentRoot /home/user/teste

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    Alias /static /home/user/teste/static

    <Directory /home/user/teste/static>
        Require all granted
    </Directory>

    Alias /media /home/user/teste/media

    <Directory /home/user/teste/media>
        Require all granted
    </Directory>

    <Directory /home/user/teste/>
        <Files app.wsgi>
            Require all granted
        </Files>
    </Directory>

    WSGIDaemonProcess teste python-path=/home/user/teste python-
home=/home/user/teste/env
    WSGIProcessGroup teste
    WSGIScriptAlias / /home/user/teste/app.wsgi
</VirtualHost>
```

Ative o site e reinicie o Apache:

```
sudo a2ensite teste.conf
sudo systemctl restart apache2
```

Para verificar erros de sintaxe:

```
apachectl -t
```

Após essas configurações, o site deve estar disponível no IP da máquina. Caso esteja usando outra porta que não 80, especifique-a no final do IP.

Fase de Projeto

Passo 1: Estrutura do Projeto

Crie um diretório para o projeto:

```
mkdir ~/projeto_flask
```

Clone o repositório do projeto desejado:

```
cd ~/projeto_flask  
git clone <url.do.projeto>
```

Crie uma venv e instale os requisitos:

```
virtualenv env  
source env/bin/activate  
pip3 install -r requirements.txt
```

A estrutura deve ser:

```
/projeto_flask  
  /env  
  /projeto  
    app.py  
  /database  
    database.db  
  /static  
  /templates
```

Teste o aplicativo localmente:

```
flask run
```

Passo 2: Mover Projeto para o Servidor

Mova o diretório para o Apache:

```
sudo mv projeto_flask /var/www/html/
```

Atualize o arquivo `app.wsgi`:

```
import sys
sys.path.insert(0, '/var/www/html/projeto_flask')
from app import app as application
```

A estrutura deve ser:

```
/var/www/html/projeto_flask
    /env
    /projeto
        app.py
    /database
        database.db
    /static
    /templates
    app.wsgi
```

Passo 3: Permissões

Dê permissões necessárias:

```
sudo chmod 664 /var/www/html/projeto_flask/database/database.db
sudo chown :www-data /var/www/html/projeto_flask/database/database.db
sudo chown :www-data /var/www/html/projeto_flask/
```

Atualize ou crie o arquivo `.conf` no Apache:

```
sudo nano /etc/apache2/sites-available/projeto_flask.conf
```

Ative o site e reinicie o Apache:

```
sudo a2ensite projeto_flask.conf
sudo systemctl restart apache2
```

Para verificar logs de erro:

```
tail -f /var/log/apache2/error.log
```

Extra: Configuração SSL

Crie um diretório para armazenar os certificados SSL:

```
sudo mkdir -p /etc/apache2/ssl/projeto_flask
```

Gere os certificados:

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout  
/etc/apache2/ssl/projeto_flask/projeto.key -out  
/etc/apache2/ssl/projeto_flask/projeto.crt
```

Habilite o módulo SSL:

```
sudo a2enmod ssl
```

Atualize o arquivo `.conf` com as configurações SSL:

```
<VirtualHost *:443>  
    ServerAdmin admin@example.com  
    ServerName example.com  
    ServerAlias www.example.com  
    DocumentRoot /var/www/html/projeto_flask  
  
    SSLEngine on  
    SSLCertificateFile /etc/apache2/ssl/projeto_flask/projeto.crt  
    SSLCertificateKeyFile /etc/apache2/ssl/projeto_flask/projeto.key  
  
    ErrorLog ${APACHE_LOG_DIR}/error.log  
    CustomLog ${APACHE_LOG_DIR}/access.log combined  
  
    Alias /static /home/user/teste/static  
  
    <Directory /home/user/teste/static>  
        Require all granted  
    </Directory>  
  
    Alias /media /home/user/teste/media  
  
    <Directory /home/user/teste/media>  
        Require all granted  
    </Directory>  
  
    <Directory /home/user/teste/>  
        <Files app.wsgi>  
            Require all granted
```

```
</Files>
</Directory>

WSGIDaemonProcess projeto_flask python-path=/var/www/html/projeto_flask
python-home=/var/www/html/projeto_flask/env
WSGIProcessGroup projeto_flask
WSGIScriptAlias / /var/www/html/projeto_flask/app.wsgi
</VirtualHost>
```

Reinicie o Apache:

```
sudo systemctl restart apache2
```

Agora, seu site estará acessível em HTTPS:

```
https://<IP>
```