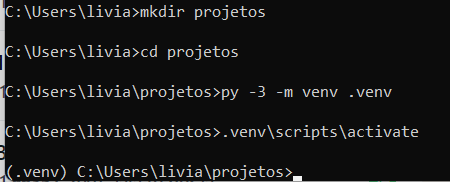
Tutorial marotinho (exemplos estão em windows ok?)   
  
Pré requisitos:  
- Python 3 <https://www.python.org/downloads/>  
Usando o prompt de comando do windows no modo administrador:

Vamos usar o mkdir para criar uma pasta para o projeto e em seguida vamos executar 2 comandos para criar um ambiente virtual

py -3 -m venv .venv   
.venv\scripts\activate



( mais infos aqui)  
<https://code.visualstudio.com/docs/python/tutorial-flask#_create-a-project-environment-for-the-flask-tutorial>

Agora tudo será executado dentro desse venv ok ?  
python -m pip install --upgrade pip  
python -m pip install flask

pip install -U Flask-SQLAlchemy

pip install pymssql

Beleza, agora vamos a criação da app em si!  
Deixem o prompt com o venv aberto, e abram o editor de código da sua escolha para criar os arquivos a seguir ( na msm pasta ok?) :

Vamos criar um arquivo app.py com o seguinte conteúdo:

from flask import Flask

from datetime import datetime

import re

app = Flask(\_\_name\_\_)

@app.route("/")

def home():

return "Hello, Flask!"

@app.route("/hello/<name>")

def hello\_there(name):

now = datetime.now()

formatted\_now = now.strftime("%A, %d %B, %Y at %X")

# Filter the name argument to letters only using regular expressions. URL arguments

# can contain arbitrary text, so we restrict to safe characters only.

match\_object = re.match("[a-zA-Z]+", name)

if match\_object:

clean\_name = match\_object.group(0)

else:

clean\_name = "Friend"

content = "Hello there, " + clean\_name + "! It's " + formatted\_now

return content

O importante nesta app é apenas observar como o flask funciona. Nesse caso estamos criando duas rotas para a aplicação, uma diretamente na chamada do ip, e a outra no ip + /hello/ + <nome>

Para visualizar o funcionamento, volte ao prompt do venv e execute:

python -m flask run

Esse comando irá colocar sua aplicação python flask em execução e irá te retornar um ip que você pode acessar no seu browser para visualizar a aplicação.

Agora vamos alterar esse app.py para incluir o sqlalchemy e fazer uma query no bd!

novo app.py:

from flask import Flask

from sqlalchemy import create\_engine, text

from datetime import datetime

import re

app = Flask(\_\_name\_\_)

db\_url = "mssql+pymssql://sa:Senh4Forte@localhost:52733/sonserina"

# A cadeia de conexão é formada por dialect[+driver]://user:password@host:port/dbname

engine = create\_engine(db\_url, pool\_size=5, pool\_recycle=3600)

conn = engine.connect()

sql\_text = text("SELECT \* FROM dbo.usuarios")

result = conn.execute(sql\_text)

print(f"Number of rows = {result.rowcount}.")

for row in result:

print(row)

conn.close()

@app.route("/")

def home():

return "Hello, Flask!"

ALTER LOGIN sa ENABLE ;

GO

ALTER LOGIN sa WITH PASSWORD = '[enterStrongPasswordHere]' ;

GO

Doc usada como referencia:<https://betterprogramming.pub/how-to-execute-plain-sql-queries-with-sqlalchemy-627a3741fdb1>

Vamos então mostrar no browser o resultado de uma query?   
Altere o app.py e teste novamente:

from flask import Flask

from sqlalchemy import create\_engine, text

from datetime import datetime

import re

app = Flask(\_\_name\_\_)

db\_url = "mssql+pymssql://sa:Senh4Forte@localhost:52733/sonserina"

# A cadeia de conexao é formada por dialect[+driver]://user:password@host:port/dbname

engine = create\_engine(db\_url, pool\_size=5, pool\_recycle=3600)

conn = engine.connect()

sql\_text = text("SELECT \* FROM dbo.usuarios")

result = conn.execute(sql\_text)

print(f"Number of rows = {result.rowcount}.")

for row in result:

print(row)

conn.close()

@app.route("/")

def home():

return "Hello, Flask!"

@app.route("/hello/<name>")

def hello\_there(name):

now = datetime.now()

formatted\_now = now.strftime("%A, %d %B, %Y at %X")

conn = engine.connect()

sql\_text = text("SELECT \* FROM dbo.usuarios")

result = conn.execute(sql\_text)

content = ''

for row in result:

content = content + str(row)

conn.close()

return content

Agora pra deixar bonito:

<https://learningactors.com/how-to-use-bootstrap-with-flask/>

É possível criar diferentes páginas utilizando o html com bootstrap para montar a parte estática e popular os campos com os dados vivos e manipulados pelo python.

Quer testar um exemplo? <https://github.com/bradtraversy/myflaskapp>

