

Tecnologias e Programação Integrada



Adicionar extensão sqlite para visualizar banco de dados mais facilmente.

Imports necessários

Para criação de uma API em Python, podemos usar o **Flask**, que é um micro framework para criar aplicações web.

1. Instalar o Python

<https://www.python.org/downloads/>

ou

Abrir o terminal > Python > Instalar pela loja do Windows

2. Instalar a biblioteca Flask

- Terminal > New terminal > pip install flask

3. Instalar biblioteca flask_sqlalchemy

- pip install flask_sqlalchemy

4. Instalar biblioteca pytest

- pip3 install pytest

```
//Importação das bibliotecas
from flask import Flask, request, jsonify

app = Flask(__name__)
```

- Flask: Biblioteca que cria a aplicação web.
- request: Usada para obter dados que vêm do cliente (como parâmetros da URL).

- jsonify: Facilita a conversão de resultados em formato JSON, que é o padrão para APIs.
- Aqui estamos criando uma instância da aplicação Flask. `__name__` é um parâmetro que indica o nome do módulo onde a aplicação está sendo executada.

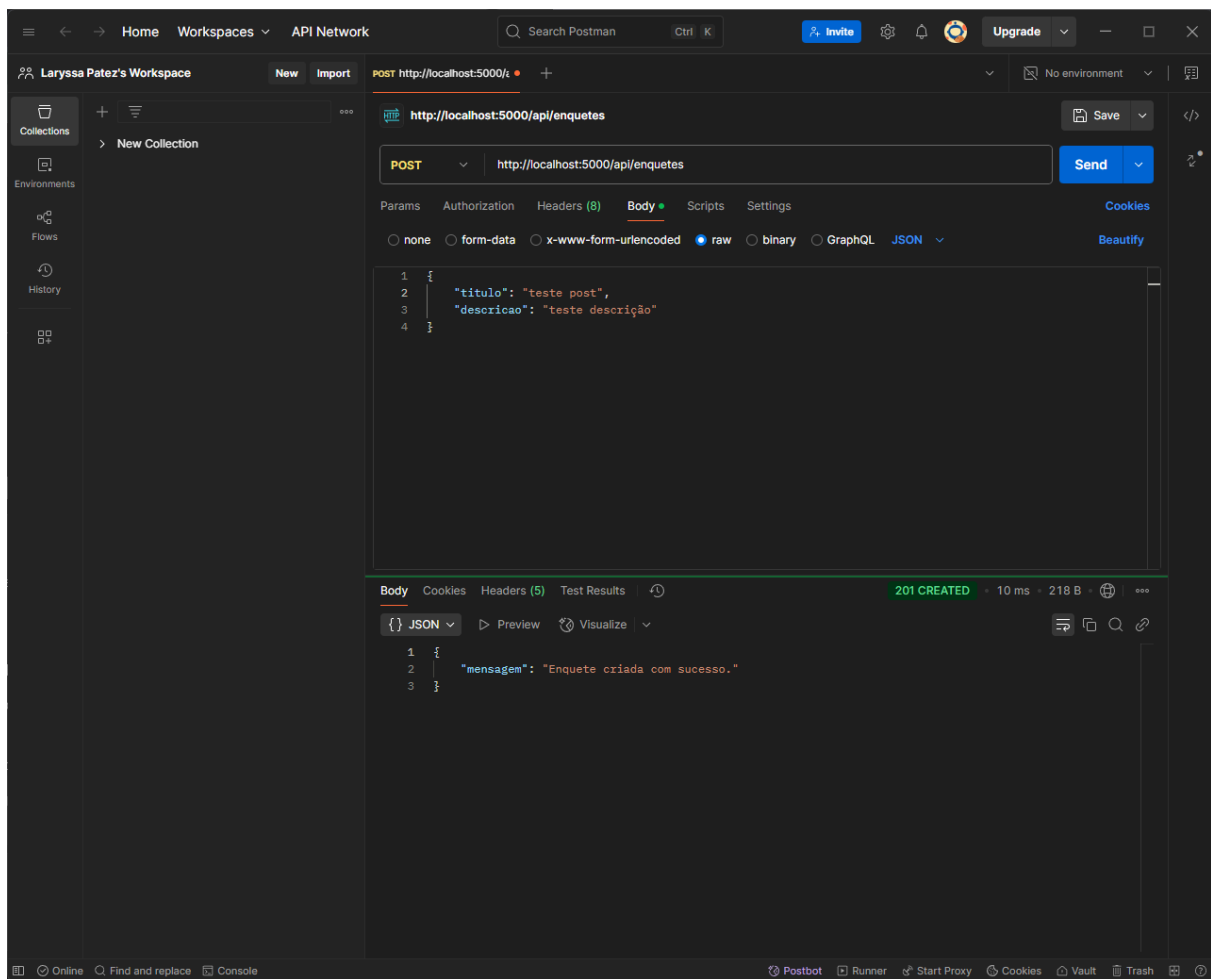
```
@app.route('/add', methods=['GET'])
def add():
    # Obtém os parâmetros 'num1' e 'num2' da URL
    num1 = float(request.args.get('num1'))
    num2 = float(request.args.get('num2'))

    # Realiza a adição e retorna o resultado em formato JSON
    return jsonify(result=num1 + num2)

# Inicia o servidor na porta 5000
app.run(port=5000,host='localhost',debug=True)
```

- A notação `@app.route()` define a rota que pode ser acessada via HTTP. O parâmetro `methods=['GET']` indica que a rota será acessível apenas para requisições GET.
- `add` define o nome do método.
- Dentro de cada função, usamos `request.args.get()` para obter os números passados como parâmetros na URL. O método `get()` acessa os valores de query strings (como `?num1=10&num2=5`).
- `app.run(port=5000,host='localhost',debug=True)` inicia o servidor na porta 5000 e host 'localhost'
 - `http://localhost/add?num1=1&num2=1`

Exemplo postman



Testes unitários

Anotações