Nathaly de Jeus Brito / 20222TADSSAJ0025 / ADS - IFBA

Documentação da API microservice_users

Plano de Implementação:

1. **Serviço de Banco de Dados (Simulado)**:
   -> O script gerado define a classe DatabaseService, que implementa um serviço para gerenciar um "banco de dados" simples simulado com um arquivo JSON.
   a. Gerenciar um arquivo JSON para armazenar e recuperar dados.
   b. Fornecer métodos para ler e gravar no arquivo JSON.

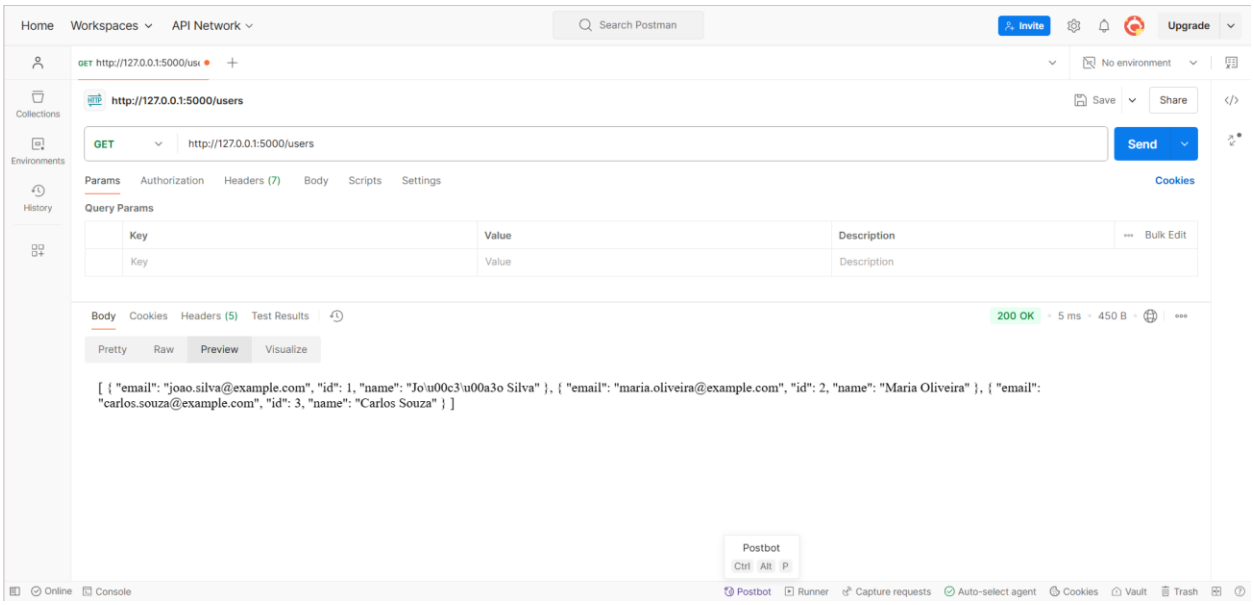
2. **Serviço de API (Gerenciador de Usuários)**:
   -> O script gerado implementa uma API RESTful usando Flask para gerenciar usuários. Ela se integrs com a classe DatabaseService, que fornece funcionalidades para persitência de dados em um arquivo JSON.
- Um endpoint GET /users para listar usuários.
- Um endpoint POST /users para adicionar novos usuários.

3. **Integração**:
   O serviço de API usará o serviço de banco de dados para persistir e recuperar os dados.

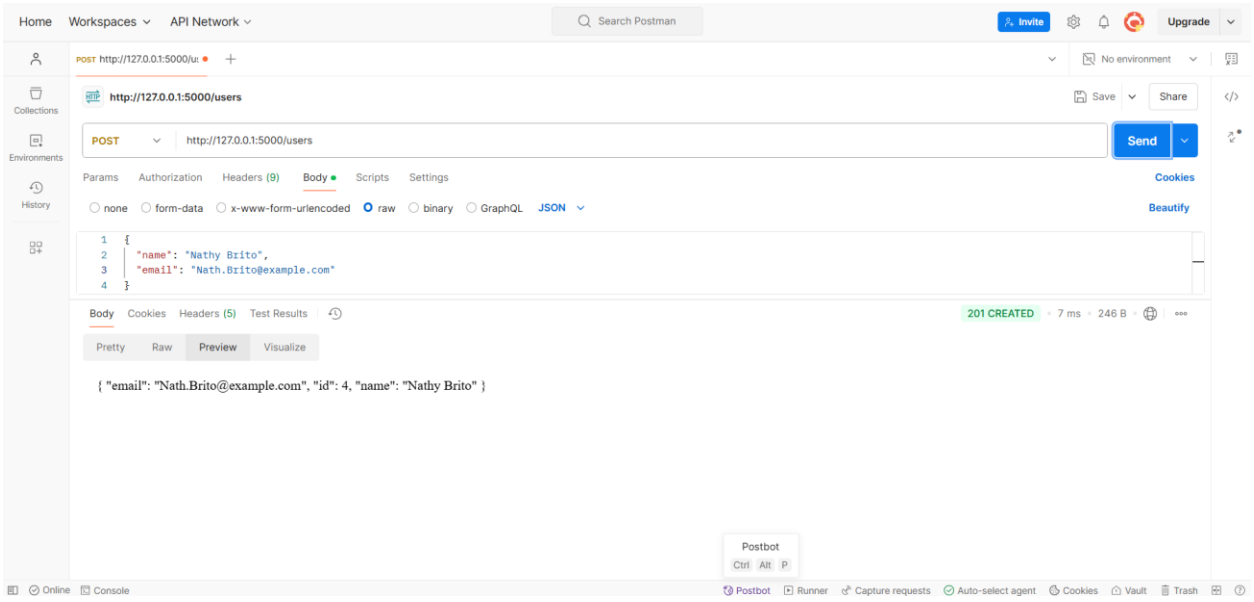
Testes

- **Teste Local com Postman**:
- Comando usado localmente flask run.
- Teste dos endpoints.
- Para rodar uma aplicação Flask em produção, é necessário usar um servidor **WSGI (Web Server Gateway Interface)**, como **Gunicorn**, **uWSGI**, ou configurá-lo com servidores como **Nginx** ou **Apache**. Servidor utilizado: Waitress-> pip install waitress.

Teste do endpoint GET/users no Postman:



Teste do endpoint POST/users:



Deploy

-> Fazer o deploy da API envolve hospedar o aplicativo em um servidor para que ele possa ser acessado via a web.

- Deploy usando **Vercel.**