

Captalys — BackEnd Developer

Sobre nós

Na Captalys, acreditamos no crédito como um bem comum, e assim, uma força para o desenvolvimento social e econômico.

Por esse motivo, investimos nosso capital financeiro, humano e intelectual para desenvolver novas tecnologias que tornem esta visão uma realidade. Gerimos quase R\$ 3 bilhões em ativos de crédito privado

Desafio técnico

O objeto deste desafio é criar uma API REST que se comunica com a API REST oficial do Github.

<https://api.github.com/>

A aplicação deverá ter:

- Um endpoint para listar os repositórios públicos de um usuário;
- Um endpoint para mostrar os detalhes de um repositório específico;

Ou seja, através dos endpoints que forem desenvolvidos, sua API deverá ser capaz de se comunicar com a API do Github.

Tecnologias

- Python ou Clojure como linguagem de desenvolvimento;
- Flask(Python) e FlaskRestplus(Python) ou Reitit (Clojure) para montar a sua API;
- Requests (Python) ou clj-http(Clojure) ou HttpKit (Clojure) para fazer a comunicação com a API do GitHub

Recomendamos que, caso escolha Python, utilize o UWSGI para subir sua API. Mas, fique a vontade para escolher outro serviço WSGI, como o Gunicorn, para subir a API.

Restrições

- Utilizar apenas as bibliotecas Requests (Python) ou clj-http(Clojure) ou HttpKit (Clojure) para fazer a comunicação com a API do GitHub.
- Escrever no arquivo README.md as instruções para ter subir a API em outro computador.
- Criar um repositório público no Github ou Gitlab para armazenar o projeto.

Bonus Points

Não são necessários para o teste, mas os seguintes pontos seriam "nice to have"

- Rodar a aplicação dentro do Docker;
- Inserir os dados obtidos do Github em uma base de dados;

Como banco de dados, pode ser utilizado o banco SQLite. E, para isso, são necessários os seguintes pontos;

- Criar um tabela para armazenar os dados de um usuário;
- Criar uma tabela para armazenar os dados de um repositório;
- Criar um endpoint que liste os usuários salvos na sua tabela;
- Criar um endpoint na sua API que receba um id de um usuário e retorne os dados dele e os detalhes de todos seus repositórios. Dica: Utilize [Join Statement](#) para isso.

Documentação relevante

- <https://developer.github.com/v3/>
- <https://requests.readthedocs.io/en/master/>
- <https://flask.palletsprojects.com/en/1.1.x/>
- <https://flask-restplus.readthedocs.io/en/stable/>
- <https://uwsgi-docs.readthedocs.io/en/latest/>
- <https://gunicorn.org/>
- <https://github.com/dakrone/clj-http>
- <https://github.com/http-kit/http-kit>
- <https://github.com/metosin/reitit>
- <https://www.docker.com/>
- <https://www.sqlite.org/index.html>