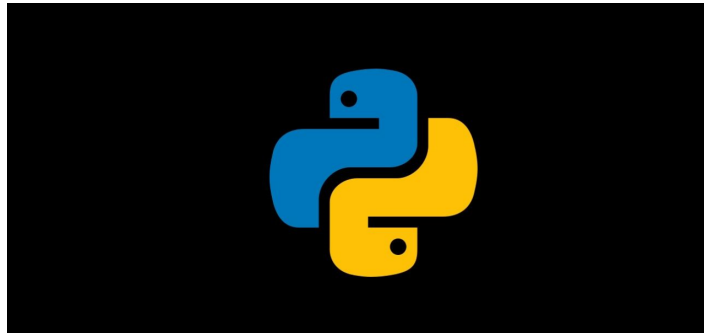


[✎ Edit article](#)[📊 View stats](#)[👁 View post](#)

Desafios ao Desenvolver uma REST API usando Flask e FastAPI: Perspectivas de uma Desenvolvedora Django!



Paula Zaidel

Software Engineer no Banco Itaú

3 articles

April 14, 2024

[📖 Open Immersive Reader](#)

Como desenvolvedora Python e com experiência em Django e Django REST Framework (DRF), aventurar-se em outras tecnologias como Flask e FastAPI foi uma experiência reveladora, com alguns desafios que gostaria de falar um pouco neste artigo.

Flask: Simples, mas exige mais

Flask é conhecido por sua simplicidade e flexibilidade. Ao começar a desenvolver com ele, a primeira coisa que notei é a liberdade que ele oferece. Porém, essa liberdade vem com seus próprios desafios.

1. Estrutura do Projeto: Muitas decisões ficam nas mãos do desenvolvedor, desde a estrutura do projeto até a gestão de dependências, podendo ser um desafio para quem já está acostumado com a estrutura mais opinativa do Django.

2. Extensões de Terceiros: Para funcionalidades como ORM, autenticação e autorização, dependemos de extensões de terceiros.

3. Maior Carga de Trabalho: Muitas vezes é necessário escrever mais código para realizar tarefas que são mais diretas em Django, como serialização e tratamento de requisições e respostas.

FastAPI: Moderno e rápido, mas com curva de aprendizado

FastAPI é um framework moderno e rápido para construir APIs com Python 3.6+ baseado em padrões modernos como async e OpenAPI. Embora traga muitas melhorias em termos de velocidade e padrões de API, também apresenta seus próprios desafios.

1. Assincronicidade: Django suporta assincronicidade através de canais e views assíncronas, mas não é uma funcionalidade nativa como no FastAPI. Aprender a utilizar async e await de forma eficiente requer um entendimento aprofundado dos conceitos de assincronia, o que pode ser um desafio inicial.

2. OpenAPI: FastAPI gera automaticamente documentação usando o OpenAPI, o que é fantástico! Mas entender e ajustar seus mecanismos para documentar corretamente a API pode exigir um esforço adicional.

3. Maturidade e Ecossistema: Sendo relativamente novo comparado ao Django, FastAPI ainda está desenvolvendo seu ecossistema e base de usuários, o que pode levar a uma menor quantidade de recursos de aprendizado e suporte da comunidade.

A Descoberta do Django Ninja

Ao retornar ao conforto do ecossistema Django enquanto procurava por uma experiência de desenvolvimento mais rápida e moderna como a oferecida pelo FastAPI, descobri

o Django Ninja. Django Ninja é um framework de API web rápido para Django, inspirado pelo FastAPI.

1. Velocidade e Simplicidade: Django Ninja utiliza a mesma abordagem de tipo de anotações que o FastAPI, oferecendo um desempenho impressionante e facilitando o processo de desenvolvimento com a geração automática de esquemas OpenAPI.

2. Integração com o Django: O uso de modelos e ORM do Django com o Ninja é praticamente sem costura, permitindo aproveitar toda a robustez e eficiência do Django com a rapidez do FastAPI.

3. Produtividade Aumentada: Com Django Ninja, consegui reduzir significativamente o tempo de desenvolvimento de uma API, mantendo todas as vantagens do ecossistema Django, desde a autenticação até a administração.

Conclusão

Embora Flask e FastAPI ofereçam abordagens válidas e poderosas para a construção de REST APIs, para um desenvolvedora Django, enfrentar suas curvas de aprendizado e peculiaridades pode ser um pouco desafiador. Django Ninja emerge como uma solução que combina o melhor de dois mundos: a simplicidade e rapidez do FastAPI com a robustez e familiaridade do Django, resultando em um desenvolvimento mais eficiente e agradável.

Published by



Paula Zaidel
Software Developer
Published • 3w

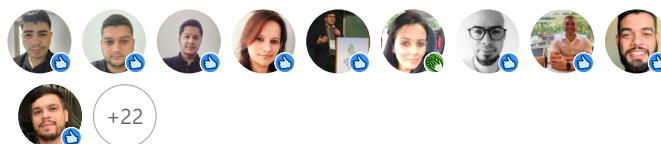
[3 articles](#)

Como desenvolvedora Python, dos projetos que já trabalhei sempre utilizei Django e DRF. Nesses últimos dias enquanto estudava, decidi desenvolver um simples CRUD utilizando Flask e FastAPI e aqui conto um pouco dos desafios que senti.

Like Comment Share

Bianca Nunes and 33 others · 1 comment

Reactions



1 Comment

