

Desenvolvimento de uma Aplicação de Conversão Monetária com FastAPI e PostgreSQL

Jean Carlo Gengnagel

October 31, 2023

1 Introdução

Este documento descreve o desenvolvimento de uma aplicação de conversão monetária e destaca as escolhas de frameworks e banco de dados, bem como os problemas enfrentados e suas soluções. A aplicação foi construída com o uso do framework FastAPI e utiliza o PostgreSQL como banco de dados para fornecer um serviço de conversão monetária em tempo real.

2 Escolha de Frameworks e Banco de Dados

O FastAPI foi a escolha ideal para a construção desta aplicação de conversão monetária por vários motivos. Primeiramente, o FastAPI é conhecido por sua incrível velocidade e eficiência. Isso é essencial para uma aplicação que lida com consultas de taxas de câmbio em tempo real. O FastAPI utiliza o poder da tipagem estática, o que permite a detecção rápida de erros e oferece uma excelente documentação automática, tornando o desenvolvimento mais rápido e fácil.

Além disso, o FastAPI é construído sobre o Starlette, um framework ASGI, o que o torna ideal para operações I/O intensivas, como as requisições à API de cotações. Isso garante que nossa aplicação seja escalável e responsiva.

Quanto à escolha do PostgreSQL como banco de dados, isso se deve principalmente à capacidade de gerenciar dados estruturados com eficiência e à robustez do PostgreSQL. Ele oferece suporte para dados complexos, consultas avançadas e possui uma comunidade ativa e madura. Isso torna o PostgreSQL uma escolha sólida para uma aplicação que precisa armazenar e recuperar cotações de moedas em tempo real com precisão.

Embora Django e Flask sejam excelentes frameworks em seus próprios direitos, o FastAPI se destaca quando se trata de desenvolvimento de APIs rápidas e escaláveis. A escolha do PostgreSQL como banco de dados é especialmente adequada para aplicativos que envolvem manipulação de dados financeiros e consultas complexas.

Em resumo, o FastAPI e o PostgreSQL foram escolhidos para esta aplicação de conversão monetária devido à sua eficiência, velocidade e capacidade de gerenciar dados financeiros complexos de forma confiável.

3 Problemas e Soluções

Um dos problemas identificados é que as credenciais de acesso ao banco de dados estão expostas dentro do projeto. Para corrigir isso, é necessário armazená-las em um serviço de nuvem. Quando a pipeline de CI/CD é acionada, as variáveis serão substituídas pelos valores correspondentes.

Outro problema potencial é o fato de que a API está construída em cima do motor de busca do Google. Caso a estrutura HTML da página mude, teremos um problema. Por isso, foram construídos testes unitários. Caso isso aconteça, o teste irá quebrar, e poderemos saber o que mudar na estrutura do código.