Desafio Técnico | BTG Pactual Empresas

Descrição:

Sua missão, caso você aceite, é criar uma API REST que gerencia tokens OTP (One-Time Password) e permite a criação e validação desses tokens. Para este desafio, a API deve ser desenvolvida utilizando **Typescript** ou alguma outra linguagem de sua preferência e deve seguir os princípios de **Arquitetura Limpa** ou **Hexagonal**.

Dica: Não existe uma forma certa ou errada de resolver o desafio! Vamos avaliar a qualidade do seu código, a clareza, a organização do projeto, a quantidade e qualidade dos testes, e a preocupação com segurança e escalabilidade.

Restrições:

- O código deve estar no GitHub ou algum repositório de código.
- Deve ter pelo menos 1 commit por cada endpoint.
- Todos os commits devem ser feitos pelo mesmo usuário que criou o projeto.
- Deve seguir exatamente os endpoints descritos a seguir.
- Deve aceitar e responder com objetos JSON.
- A arquitetura deve seguir os princípios de Arquitetura Limpa ou Hexagonal.
- Por motivos de segurança, não podemos aceitar projetos enviados como arquivos. Você deve disponibilizar seu projeto publicamente.
- Não deve fazer fork de nenhum outro projeto.
- Pode utilizar um banco de dados de sua preferência (SQL ou NoSQL).

Requisitos:

A seguir os endpoints que devem estar presentes na sua API e a funcionalidade de cada um deles.

1. Criar token otp
    a. Este endpoint irá gerar um novo token otp.
2. Validar token otp
    a. Este endpoint irá validar um token otp existente.

Extras:

A seguir, temos alguns desafios extras caso você queira testar seus conhecimentos ao máximo! Nenhum desses requisitos é obrigatório, mas são desejados e podem ser um diferencial!

- Testes automatizados: Implemente testes unitários e/ou funcionais para garantir a qualidade do seu código.
- Containerização: Disponibilizar sua aplicação como um container (ex: Docker) para facilitar a implementação e escalabilidade. Se desejar, você pode implementar a infraestrutura em nuvem (ex: AWS) e utilizar serviços serverless, como AWS Lambda para a execução da sua API, ou opções de banco de dados como DynamoDB.
- Logs: Implementar logs informando o que está acontecendo na aplicação. Logs são úteis para auxiliar na solução de problemas e para a observabilidade do sistema.
- Documentação da API: Documentar sua API de forma clara, utilizando ferramentas como Swagger ou OpenAPI para facilitar a compreensão dos endpoints e suas funcionalidades.
- Documentação do Sistema: Criar um Readme.md que explique como compilar e executar a aplicação para alguém que esteja pegando o projeto pela primeira vez.
- Configurações: Permitir que a aplicação seja configurável, como por exemplo, definindo o tempo de expiração do token otp.

Considerações Finais:

Lembre-se de que a qualidade do código, a estrutura do projeto e a clareza da documentação são extremamente importantes. Ao final do desafio, você deverá disponibilizar seu projeto publicamente em uma plataforma de hospedagem de código, como Github ou Gitlab, para que possamos revisar seu trabalho.

Estamos ansiosos para ver sua solução e como você aplicou seus conhecimentos em Typescript, Arquitetura Limpa ou Hexagonal, e práticas de desenvolvimento de software. Boa sorte!

Referências:

https://www.keepersecurity.com/blog/2023/12/20/what-is-a-one-time-password-otp/

https://forense.io/glossario/o-que-e-one-time-password-otp-seguranca-digital/

https://canaltech.com.br/internet/o-que-e-uma-otp-one-time-password/