

W2Bit - Backend Challenge

Se você chegou até aqui, parabéns! Agora vamos avaliar suas habilidades técnicas com um desafio prático.

Você será avaliado por sua capacidade de escrever códigos legíveis, simples e de fácil manutenção.

› Instruções

- Nome do Projeto: W2Bit Bus Station
- Objetivo: Criar uma API RESTful capaz de gerenciar os ônibus de sua frota e adicionar as viagens realizadas.
- Tecnologia: Node.js com banco de dados relacional.
- Entrega: Crie um repositório pessoal para esse projeto, siga as instruções abaixo, sinalize o término e envie o link do repositório, o link do deploy e a documentação no insomnia para testar sua api.

› Recomendações

- Documente seu projeto em arquivos markdown explicando a estrutura, processo de setup e requisitos.
- Tenha sempre um mindset de usabilidade, escalabilidade e colaboração.
- A organização das branches e os commits no repositório falam muito sobre como você organiza seu trabalho.
- O design/estrutura do código da aplicação deve ser production ready.
- Tenha em mente os conceitos de SOLID, KISS, YAGNI e DRY.
- Use boas práticas de programação.
- Você pode usar bibliotecas como Express, mas queremos que a solução de arquitetura seja sua, portanto não use frameworks que impõem uma arquitetura específica.
- Utilize PostgreSQL como banco de dados.
- Faça deploy no Heroku.
- Inclua no seu readme os desafios/problemas com os quais você se deparou durante a execução do projeto.

› Challenge!

Você foi contratado para fazer um sistema para donos de frotas de ônibus, onde os usuários poderão criar uma conta, se autenticar utilizando um token JWT e poderão adicionar seus ônibus. Após cadastrar o ônibus, ele poderá adicionar passageiros ao ônibus.

- A Sua API deverá ser capaz de:
 - Cadastrar um novo usuário
 - Se autenticar utilizando JWT
 - Cadastrar um novo ônibus
 - Listar todos os ônibus cadastrados
 - Listar os dados de um ônibus
 - Alterar dados de um ônibus
 - Excluir um ônibus

- Adicionar passageiro ao ônibus (mas não ultrapassar o limite de assentos)
 - Alterar algum dado do passageiro
 - Excluir passageiro
- O cadastro do usuário deve ter os seguintes campos:
 - Nome
 - Foto
 - Estado e cidade
 - Senha
 - Confirmação de senha
- O cadastro do ônibus deve ter os seguintes campos:
 - Placa
 - Ano
 - Modelo
 - Numero de assentos (utilizar para limitar o número de passageiros)
- O cadastro do passageiro deve ter os seguintes campos:
 - Nome
 - Idade
 - CPF

’ O que avaliaremos

- Você será avaliado pela qualidade do código, legibilidade e pelo modo que implementou as funcionalidades.
- Você é livre para tomar as decisões técnicas com as quais você se sente mais confortável.

’ Happy coding!