

Projeto Backend Taxis

Objetivo

Implementar o backend de um aplicativo de smartphone que mostra um mapa com os taxistas ativos da 99Taxis. O sistema deve assumir que tem diversos taxistas cadastrados e permitir atualizar o status desses taxistas, consultar status deles e encontrar taxistas em uma dada área. Toda comunicação deve ser feita com JSON.

Endpoints:

1. Grava a posição atual e estado de um taxista.

1. Parâmetros:

1. latitude

2. longitude

3. driverId (id único de taxista no sistema)

4. driverAvailable: true/false. Representa se o taxista está disponível ou não para corridas.

2. Resposta:

1. Apenas status code HTTP

3. Exemplo:

1. POST /drivers/8475/status '{"latitude":-23.60810717,"

2. Lista os taxistas ativos dentro de um retângulo geográfico.

1. Parâmetros:

1. sw: Ponto extremo sul, extremo oeste do retângulo, no formato "latitude,longitude". Ex: -23.612474,-46.702746

2. ne: Ponto extremo norte, extremo leste do retângulo, no formato "latitude,longitude". Ex: -23.589548,-46.673392

2. Resposta: Um array em formato json, de um objeto com atributos:

1. latitude

2. longitude

3. driverId (id único de taxista no sistema)

4. driverAvailable: true/false. Representa se o taxista está disponível ou não para corridas. Neste endpoint retorna sempre true.

3. Exemplo:

GET /drivers/inArea?sw=-23.612474,

[{"latitude":-23.60810717,"

3.Estado de um taxista. Recebe o id de um taxista e retorna os dados dele:

1.Parâmetros:

1.driverId

2.Resposta:

1.latitude

2.longitude

3.driverId (id único de taxista no sistema)

4.driverAvailable: true/false. Representa se o taxista está disponível ou não para corridas.

3.Exemplo:

1.GET /drivers/73456/status

'{"latitude":-23.60810717,"

Objetivos Bonus:

1.Subir a aplicação "na nuvem" e enviar o link para testarmos os endpoints.

2.Descrever algoritmo ótimo de procura dos taxistas.

3.Autenticação nos endpoints, explicando a solução.

4.Criar outros endpoints que possibilitem a mesma funcionalidade mas com outro fluxo.

5.Endpoint Cria taxista:

1.Parâmetros:

1.name

2.carPlate: placa do carro

2.Resposta:

1.Apenas status code HTTP

3.Exemplos:

1.POST /drivers

'{"name":"Pedro","carPlate":'