

Documentação API RestFull Controle de Motos

Autora e Dev: Lara Kássia Gomes de Andrade.

Visão Geral do Projeto:

Este Projeto Consiste em uma API para gerenciamento de motos desenvolvida com .Net e C#. A aplicação permite ao administrador cadastrar, consultar, alterar e deletar motos do sistema.

Funcionalidades:

- Cadastrar motos com identificador, ano, modelo e placa.
- Garantir que a placa seja única.
- Consultar motos existentes e filtrar a procura por meio da placa.
- Modificar a placa de uma moto.
- Remover motos, desde que não possuam registro de locação

Requisitos:

Tecnologias: .Net Core 8.0, C#, Entity Framework, Docker e Postgres

Pré-requisitos:

Antes de rodar a aplicação certifique-se de ter os seguintes softwares instalados:

- .NET SDK (versão 8.0)
- Docker
- PostgreSQL (pacotes)
- Insonmia ou postman para testar as rotas da API (também é possível utilizar swagger)

Configuração do Banco de dados:

Rodar o comando:

```
docker run --name my-container -e POSTGRES_PASSWORD=12345 -p 5432:5432 -d postgres
```

Ou

Docker-compose up

Como rodar a Aplicação:

1. Clone o repositório do projeto para sua máquina:
git clone <https://github.com/httpGaspar/Gerenciador-de-Motos.git>
2. Navegue até a pasta do projeto:
cd MotoApi
3. Instale as bibliotecas:
dotnet add package Npgsql.EntityFrameworkCore.PostgreSQL
dotnet add package Microsoft.EntityFrameworkCore.Design
4. Restaure os pacotes e compile a solução:
dotnet restore
dotnet build
5. Rode a aplicação:
dotnet run

A API estará disponível para o insomnia ou postman no endereço:
<https://localhost:5193>

Endpoints e Casos de Uso:

1. **Cadastrar uma nova moto:**
Rota: POST **/api/motos**
Requisição:
{
 "Id": "11111",
 "ano": 2020,
 "modelo": "Yamaha MT-07",
 "placa": "ABC-1234"
}
2. Consultar motos existentes:
Rota: GET **/motos**
Exemplo de requisição (com filtro): GET
<https://localhost:5192/api/motos?placa=ABC-1234>

3. **Modificar** a placa de uma moto:
Rota: PUT /motos/{id}
4. **Remover** uma moto:
Rota: DELETE /motos/{id}

Problemas Conhecidos

1. **Erro 500 ao tentar realizar uma Operação CRUD:** Pode ocorrer indicando que houve uma falha na conexão com o banco de dados, verifique as configurações de conexão no appsettings.json.

Considerações finais

Esse projeto está parcialmente funcional com o projeto sendo compilado sem erros e atende aos principais requisitos do desafio.

Embora as funcionalidades de CRUD estejam implementadas, ainda existem algumas melhorias e correções necessárias para que funcionem corretamente no momento de utilizar o código no Insonmia (postman ou swagger).