

API de Veículos

Descrição

Esta é uma API RESTful desenvolvida em Django usando Django REST Framework. A API permite a criação, leitura, atualização e exclusão de informações sobre veículos, além de implementar autenticação JWT. Este projeto foi desenvolvido como um caso para entrevista de emprego.

Índice

- Tecnologias Utilizadas
- Funcionalidades
- Instalação
- Uso
- Endpoints
- Testes
- Contribuição
- Licença

Tecnologias Utilizadas

- Django: Framework web de alto nível para desenvolvimento rápido.
- Django REST Framework: Ferramenta poderosa para construção de APIs RESTful.
- Django REST Framework Simple JWT: Biblioteca para autenticação baseada em JSON Web Tokens (JWT).

Funcionalidades

- Listar todos os veículos.
- Criar um novo veículo.
- Atualizar o status de um veículo (Conectado/Desconectado).
- Excluir um veículo.
- Autenticação via JWT.

Instalação

Para instalar e configurar o projeto, siga os passos abaixo:

1. Clone este repositório:

```
git clone https://github.com/seu-usuario/seu-repositorio.git
cd seu-repositorio
```

2. Crie um ambiente virtual e ative-o:

```
python -m venv venv
source venv/bin/activate # Para Windows use: venv\Scripts\activate
```

3. Instale as dependências:

```
pip install -r requirements.txt
```

4. Faça as migrações do banco de dados:

```
python manage.py migrate
```

5. Crie um superusuário (opcional, para acessar o painel administrativo):

```
python manage.py createsuperuser
```

6. Execute o servidor:

```
python manage.py runserver
```

Uso

A API estará disponível em <http://localhost:8000/api/>. Para autenticação, utilize os endpoints fornecidos para obter e renovar tokens JWT.

Endpoints

Método	Endpoint	Descrição
GET	/api/veiculos/	Lista todos os veículos.
POST	/api/veiculos/	Cria um novo veículo.
GET	/api/veiculos/{id}/	Retorna os detalhes de um veículo específico.
PUT	/api/veiculos/{id}/	Atualiza o status de um veículo.
DELETE	/api/veiculos/{id}/	Exclui um veículo.
POST	/api/token/	Obtém um novo token JWT.
POST	/api/token/refresh/	Renova o token JWT.
GET	/swagger/	Documentação API

Testes

Para executar os testes, utilize o seguinte comando:

```
python manage.py test
```

Os testes estão implementados na pasta `veiculos/tests.py`.

Contribuição

Contribuições são bem-vindas! Sinta-se à vontade para abrir issues ou pull requests. Para contribuir, siga estas diretrizes:

1. Faça um fork do projeto.
2. Crie uma branch para suas alterações (`git checkout -b feature/nome-da-sua-feature`).
3. Realize suas alterações e faça um commit (`git commit -m 'Adicionando nova funcionalidade'`).

4. Envie sua branch para o repositório remoto (git push origin feature/nome-da-sua-feature).
5. Abra um pull request.

Licença

Este projeto está licenciado sob a MIT License - veja o arquivo LICENSE para mais detalhes.