# **Documentação da API**

## **Visão Geral**

Esta API é uma aplicação de gerenciamento de veículos com autenticação baseada em tokens JWT. Os usuários podem se autenticar, listar, criar, atualizar e excluir veículos. As operações são protegidas por autenticação e todas as requisições devem incluir um token JWT válido no cabeçalho.

## **Endpoints**

### **1. Obter Token**

#### **Endpoint:**

**POST /token**

#### **Descrição:**

Obtém um token de autenticação JWT para um usuário com base em nome de usuário e senha.

#### **Parâmetros:**

**Corpo da Requisição ():**  
  
{

"username": "string",

"password": "string"

}

#### **Respostas:**

**Sucesso (200 OK):**  
  
{

"access\_token": "string",

"token\_type": "bearer"

}

**Erro (401 Unauthorized):**  
  
{

"detail": "Usuário ou senha incorretos."

}

#### **Exemplo de Uso no Postman:**

1. **Método:** POST
2. **URL:** http://localhost:8000/token
3. **Cabeçalhos:**
   * Content-Type: application/json
4. **Corpo:**{

"username": "user1",

"password": "password1"

}

### **2. Listar Nomes dos Veículos**

#### **Endpoint:**

**GET /veiculos**

#### **Descrição:**

Obtém uma lista dos nomes de todos os veículos disponíveis.

#### **Parâmetros:**

* **Cabeçalhos:**
  + Authorization: Bearer {token}

#### 

#### 

#### 

#### 

#### **Respostas:**

**Sucesso (200 OK):**  
  
[

"Fusca",

"Civic",

"Corsa",

"Onix",

"Focus",

"Golf"

]

**Erro (401 Unauthorized):**  
  
{

"detail": credenciais inválidas."

}

#### **Exemplo de Uso no Postman:**

1. **Método:** GET
2. **URL:** http://localhost:8000/veiculos
3. **Cabeçalhos:**
   * Authorization: Bearer {token}

### **3. Criar Veículo**

#### **Endpoint:**

**POST /veiculos**

#### **Descrição:**

Cria um novo veículo.

#### **Parâmetros:**

* **Cabeçalhos:**
  + Authorization: Bearer {token}

**Corpo da Requisição ():**  
  
{

"nome": "string",

"modelo": "string",

"status": "CONECTADO" | "DESCONECTADO"

}

#### **Respostas:**

**Sucesso (201 Created):**  
  
{

"id": 1,

"nome": "string",

"modelo": "string",

"status": "CONECTADO" | "DESCONECTADO"

}

**Erro (401 Unauthorized):**  
  
{

"detail": credenciais inválidas."

}

**Erro (400 Bad Request):**  
  
{

"detail": "Status inválido"

}

#### **Exemplo de Uso no Postman:**

1. **Método:** POST
2. **URL:** http://localhost:8000/veiculos
3. **Cabeçalhos:**
   * Authorization: Bearer {token}
   * Content-Type: application/json
4. **Corpo:**  
     
   {

"nome": "Civic",

"modelo": "Honda Civic",

"status": "CONECTADO"

}

### **4. Obter Detalhes do Veículo por Nome**

#### **Endpoint:**

**GET /veiculos/{nome\_veiculo}**

#### **Descrição:**

Obtém os detalhes de um veículo com base no nome do veículo.

#### **Parâmetros:**

* **Cabeçalhos:**
  + Authorization: Bearer {token}
* **URL Path Parameters:**
  + nome\_veiculo (string): Nome do veículo.

#### **Respostas:**

**Sucesso (200 OK):**  
  
{

"id": 1,

"nome": "string",

"modelo": "string",

"status": "CONECTADO" | "DESCONECTADO"

}

**Erro (404 Not Found):**  
  
{

"detail": "Veículo não encontrado"

}

**Erro (401 Unauthorized):**  
  
{

"detail": "credenciais inválidas."

}

#### **Exemplo de Uso no Postman:**

1. **Método:** GET
2. **URL:** http://localhost:8000/veiculos/Civic
3. **Cabeçalhos:**
   * Authorization: Bearer {token}

### **5. Atualizar Status do Veículo**

#### **Endpoint:**

**PUT /veiculos/{nome\_veiculo}**

#### **Descrição:**

Atualiza o status de um veículo com base no nome do veículo.

#### **Parâmetros:**

* **Cabeçalhos:**
  + Authorization: Bearer {token}
* **URL Path Parameters:**
  + nome\_veiculo (string): Nome do veículo.

**Corpo da Requisição ():**  
  
{

"status": "CONECTADO" | "DESCONECTADO"

}

#### **Respostas:**

**Sucesso (200 OK):**  
  
{

"id": 1,

"nome": "string",

"modelo": "string",

"status": "CONECTADO" | "DESCONECTADO"

}

**Erro (404 Not Found):**  
  
{

"detail": "Veículo não encontrado"

}

**Erro (400 Bad Request):**  
  
{

"detail": "Status inválido"

}

**Erro (401 Unauthorized):**  
  
{

"detail": "credenciais inválidas."

}

#### **Exemplo de Uso no Postman:**

1. **Método:** PUT
2. **URL:** http://localhost:8000/veiculos/Civic
3. **Cabeçalhos:**
   * Authorization: Bearer {token}
   * Content-Type: application/json
4. **Corpo:**  
     
   {

"status": "DESCONECTADO"

}

### **6. Excluir Veículo**

#### **Endpoint:**

**DELETE /veiculos/{nome\_veiculo}**

#### **Descrição:**

Exclui um veículo com base no nome do veículo.

#### **Parâmetros:**

* **Cabeçalhos:**
  + Authorization: Bearer {token}
* **URL Path Parameters:**
  + nome\_veiculo (string): Nome do veículo.

#### **Respostas:**

**Sucesso (200 OK):**  
  
{

"id": 1,

"nome": "string",

"modelo": "string",

"status": "CONECTADO" | "DESCONECTADO"

}

**Erro (404 Not Found):**  
  
{

"detail": "Veículo não encontrado"

}

**Erro (401 Unauthorized):**  
  
{

"detail": "credenciais inválidas."

}

#### **Exemplo de Uso no Postman:**

1. **Método:** DELETE
2. **URL:** http://localhost:8000/veiculos/Civic
3. **Cabeçalhos:**
   * Authorization: Bearer {token}

### **Rodar a Aplicação**

### Instale as dependências: Certifique-se de que todas as dependências estão instaladas. Execute o seguinte comando na raiz do seu projeto: **pip install -r requirements.txt**

### Execute o servidor: Use o **uvicorn** para rodar o servidor. Se a aplicação está no arquivo **app/main.py** e a instância **FastAPI** é chamada app, o comando seria: **uvicorn app.main:app --reload**

### Acesse a aplicação: Depois de iniciar o servidor, você pode acessar a API em <http://localhost:8000>.

### **Rodar os Testes**

### Certifique-se de que a API está em execução: Para alguns testes, especialmente os que envolvem a criação de dados, a API pode precisar estar ativa. No entanto, para testes unitários puros que não dependem da execução real da API, isso não é necessário.

### 

### Execute os testes unitários: Use o **unittest** para executar seus testes. Navegue até o diretório onde seu arquivo de teste está localizado e execute: **python -m unittest discover -s testes**

### **Observações**

### **Pasta app**: Contém a aplicação FastAPI e é onde o código principal reside.

### **Pasta testes**: Contém os testes unitários e deve ser referenciada corretamente nos comandos de teste.

### 

### **Considerações Finais**

* Certifique-se de substituir {token} pelo token JWT real obtido no endpoint /token.
* Para testar as APIs no Postman, adicione o token obtido no cabeçalho Authorization como Bearer {token}.