

DOCUMENTAÇÃO APP TAKA v.1

Este documento tem o objetivo de fornecer uma base de conhecimento do funcionamento geral do app taka. No swagger tem algumas informações extras sobre características de cada endpoint.

Autenticação

Métodos de Autenticação Suportados

O sistema utiliza **JSON Web Tokens (JWT)** para autenticação. Além disso, há suporte para o recurso de **Refresh Token** para renovar os tokens de acesso.

Integração com Sistemas Externos

Não há integração com sistemas externos para autenticação. O sistema gerencia a autenticação internamente usando JWT.

Fluxos de Autenticação e Autorização

O sistema permite apenas dois tipos de usuários: **admin** e **entregador**. Aqui estão os principais fluxos:

1.Admin:

- O usuário admin é único e é cadastrado com um email e senha padrão definidos durante a inicialização do sistema.
- O admin pode alterar sua senha futuramente.
- Para autenticar no sistema, o admin deve informar o email e a senha já cadastrados.
- Após a validação, o sistema emite um token JWT com um tempo de validade estipulado no arquivo `appsettings` do microserviço.
- Esse token é usado para autenticação nos demais microserviços. O perfil entregador não tem acesso aos microserviços de motos.

2.Entregador:

- Todo novo usuário cadastrado terá o perfil entregador.
- O entregador também se autentica usando JWT, mas seu acesso é restrito aos microserviços específicos.

Microserviço de Motos

Informações Gerenciadas

Este microserviço gerencia informações exclusivas para o perfil **admin**. As informações incluem detalhes da moto, status e localização.

Operações de CRUD para Motos

- Uma moto é única, identificada pela **PLACA**.
- O cadastro de uma nova moto não é síncrono. O microserviço utiliza mensageria para registrar o evento de solicitação de cadastro (produtor).
- Um background service (consumidor) lê essa fila e tenta realizar o cadastro.
- Uma moto só é efetivada no sistema quando seu ano for **2024**.
- Foi implementada uma nova fila para gerar mensagens representando o sucesso ou falha do cadastro.

Integrações Específicas

Não há integrações específicas com outros microserviços ou APIs externas.

Microserviço de Entregadores

Dados dos Entregadores

Os dados dos entregadores armazenados e gerenciados incluem:

- **Nome**
- **CNPJ**
- **Data de Nascimento**
- **CNH**

Cadastro e Gestão dos Entregadores

- O critério de unicidade é CNPJ único e CNH única.
- A autenticação é realizada usando o token JWT gerado pelo microserviço de autenticação.

Locação de Motos

Processo de Locação

- Para realizar a locação, o entregador precisa estar cadastrado e possuir carta **A** ou **AB**.
- A moto selecionada não deve estar locada.

Integração com Outros Microserviços

- O microserviço de locação integra-se com os microserviços de motos e entregadores usando chamadas REST e Refit para verificar se as motos e entregadores são cadastros válidos.

Arquitetura Geral e Integrações

Comunicação entre Microserviços

- Os microserviços se comunicam entre si por meio de **chamadas REST**.

Padrões de Design Utilizados

Além de Clean Code, SOLID, os principais padrões de design incluem:

- Mediator
- Command
- Repository
- Factory
- Singleton

Resiliência e Fallback

•Foi implementado um mecanismo de **retry** na parte de mensageria para garantir resiliência, utilizando o Polly.

Infraestrutura e Deploy

Deployment e Ambientes

•O deployment e a gestão dos microserviços são feitos de acordo com ambientes separados para **desenvolvimento**, **teste** e **produção**.

Monitoramento e Logging

Foi utilizado o Serilog e Elasticsearch para essas atividades.

Testes Unitários e Integrados

Testes Unitários

Implementados para Autenticação, Motos e Entregadores.

Integrados

Implementados apenas para uso no Postman. Apenas microserviço de autenticação.