

Capítulo 1 : Apresentação da API

Descrição

Esta API foi desenvolvida com o recurso à OpenAPI, usando uma camada de serviços REST.

A API tem o objetivo de ver todos os detalhes de uma Equipa, Pokemon e seus treinadores.

Para esta API foram usados 6 controladores:

- PokemonController
- TreinadoresController
- EquipasController
- TiposController
- TeamController
- TreinadoresByEquipaController

- AuthController
- swagger-controller

> PokemonController

É composto por 6 funcionalidades:

- **GET(Count)** -> Retorna o numero de Pokemon presente na base de dados.
- **GET** -> Retorna um array com todos os Pokemon presentes na base de dados.
- **POST** -> Adiciona um Pokemon à base de dados.
- **GET{id}** -> Dado um ID, retorna o respetivo Pokemon.
- **PUT{id}** -> Efetua alterações a um dado Pokemon com um respetivo ID.
- **DELETE{id}** -> Elimina um Pokemon da base de dados com um dado ID.

> TreinadoresController

É composto por 6 funcionalidades:

- **GET(Count)** -> Retorna o numero de treinadores presente na base de dados.
- **GET** -> Retorna um array com todos os treinadores presentes na base de dados.
- **POST** -> Adiciona um treinador à base de dados.
- **GET{id}** -> Dado um ID, retorna o respetivo treinador.
- **PUT{id}** -> Efetua alterações a um dado treinador com um respetivo ID.
- **DELETE{id}** -> Elimina um treinador da base de dados com um dado ID.

> EquipasController

É composto por 6 funcionalidades:

- **GET(Count)** -> Retorna o numero de equipas presente na base de dados.
- **GET** -> Retorna um array com todos os equipas presentes na base de dados.
- **POST** -> Adiciona uma equipa à base de dados.
- **GET{id}** -> Dado um ID, retorna o respetivo equipa.
- **PUT{id}** -> Efetua alterações a um dado equipa com um respetivo ID.
- **DELETE{id}** -> Elimina uma equipa da base de dados com um dado ID.

> TiposController

É composto por 6 funcionalidades:

- **GET(Count)** -> Retorna o numero de tipos presente na base de dados.
- **GET** -> Retorna um array com todos os tipos presentes na base de dados.
- **POST** -> Adiciona um tipo à base de dados.
- **GET{id}** -> Dado um ID, retorna o respetivo tipo.
- **PUT{id}** -> Efetua alterações a um dado tipo com um respetivo ID.
- **DELETE{id}** -> Elimina um tipo da base de dados com um dado ID.

> BattlesController

É composto por 6 funcionalidades:

- **GET(Count)** -> Retorna o numero de batalhas presente na base de dados.
- **GET** -> Retorna um array com todas as batalhas presentes na base de dados.
- **POST** -> Adiciona uma batalha à base de dados.
- **GET{id}** -> Dado um ID, retorna a respetiva batalha.
- **PUT{id}** -> Efetua alterações a uma dada batalha com um respetivo ID.
- **DELETE{id}** -> Elimina uma batalha da base de dados com um dado ID.

> TreinadoresByEquipaController

É composto por 1 funcionalidade:

- **GET{id}** -> Retorna todos os treinadores associados a uma equipa com um ID.

> AuthController

É composto por 7 funcionalidades:

- **GET** (login) -> Retorna uma página pública onde é possível efetuar login ou ir para a documentação da API.

- **GET** (logout) -> Efetua logout e retorna uma página pública onde é possível efetuar login ou ir para a documentação da API.
- **GET** (protected) -> Retorna uma página privada onde é possível aceder a endpoints que retornam informação do utilizador, ir para a documentação da API ou efetuar logout.
- **GET** (auth/github) -> Operação responsável por efetuar a autenticação.
- **GET** (auth/github/callback) -> Operação responsável por comunicar de volta o token.
- **GET** (me) -> Retorna informações do utilizador.
- **GET** (gitHubMe) -> Retorna informações do utilizador com recurso à token de authentication.

> **swagger-controler**

É composto por 1 funcionalidade:

- **GET** (docs/swagger.json) -> Retorna uma página com a documentação da API no formato .json.

Grupo 02

- André Rodrigues [@AndreRodrigues](#)
- Hugo Martins [@HugoMartins](#)

< **Anterior** **^ Main** **Próximo** >
