

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 ser possível a consulta de filmes. Consegue ver todos os detalhes de um filme, dos generos, atores e realizadores.

Para esta API foram usados 6 controladores:

- ActorsController
 - DirectorsController
 - GenreByMoviesController
 - GenresController
 - MoviesByGenreController
 - MoviesController
-
- AuthController
 - swagger-controller

> ActorsController

É composto por 6 funcionalidades:

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

> DirectorsController

É composto por 6 funcionalidades:

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

> GenresController

É composto por 6 funcionalidades:

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

> MoviesController

É composto por 6 funcionalidades:

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

> GenreByMoviesController

É composto por 1 funcionalidade:

- **GET{id}** -> Retorna todos os generos associados a um filme com um ID.

> MoviesByGenreController

É composto por 1 funcionalidade:

- **GET{id}** -> Retorna todos os filems associados a um genero 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 01

- Ana Filipa Azevedo [@AnaAzevedo](#)
- Mariana Lopes [@MarianaLopes](#)
- Mário Rodrigues [@MarioRodrigues](#)
- Rodrigo Moreira [@Rodrigo Moreira](#)

< **Anterior** [^ Main](#) **Próximo** >
