

C1 : Apresentação do Projeto

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 diretores.

Para esta API foram usados 11 controladores:

- ActorController
- ActorMoviesController
- DirectorController
- DirectorMoviesController
- GenreController
- GenreMoviesController
- MoviesController
- MoviesActorController
- MoviesDirectorController
- MoviesGenreController
- ThroughController

> ActorController

É composto por 8 funcionalidades:

- **GET (count)** - Retorna o nº total de atores presentes na base de dados.
- **PUT{id}** - Efetua alterações a um ator com um respetivo ID.
- **PATCH{id}** - Requisições sucessivas idênticas de atores através do ID podem obter efeitos distintos.
- **GET{id}** - Dado um ID, retorna o respetivo ator.
- **DELETE{id}** - Remove da base de dados um ator com um respetivo ID.
- **POST** - Adiciona um ator à base de dados.
- **PATCH** - Requisições sucessivas idênticas de atores podem obter efeitos distintos.
- **GET** - Retorna todos os atores presentes na base de dados.

> ActorMoviesController

É composto por 4 funcionalidades:

- **POST{id}** - Adiciona um filme através do ID do ator à base de dados.
- **PATCH{id}** - Requisições sucessivas idênticas de atores através do ID podem obter efeitos distintos.
- **GET{id}** - Dado um ID de um ator, retorna o(s) respetivo(s) filme(s).
- **DELETE{id}** - Remove da base de dados um/os filme/s com um respetivo ID do ator.

> DirectorController

É composto por 8 funcionalidades:

- **GET (count)** - Retorna o nº total de diretores presentes na base de dados.
- **PUT{id}** - Efetua alterações a um diretor com um respetivo ID.
- **PATCH{id}** - Requisições sucessivas idênticas de atores através do ID podem obter efeitos distintos.
- **GET{id}** - Dado um ID, retorna o respetivo diretor.
- **DELETE{id}** - Remove da base de dados um diretor com um respetivo ID.
- **POST** - Adiciona um diretor à base de dados.
- **PATCH** - Requisições sucessivas idênticas de diretores podem obter efeitos distintos.
- **GET** - Retorna todos os diretores presentes na base de dados.

> DirectorMoviesController

É composto por 4 funcionalidades:

- **POST{id}** - Adiciona um filme através do ID do diretor à base de dados.
- **PATCH{id}** - Requisições sucessivas idênticas de diretor através do ID podem obter efeitos distintos.
- **GET{id}** - Dado um ID de um diretor, retorna o(s) respetivo(s) filme(s).
- **DELETE{id}** - Remove da base de dados um/os filme/s com um respetivo ID do diretor.

> GenreController

É composto por 8 funcionalidades:

- **GET (count)** - Retorna o nº total de géneros presentes na base de dados.
- **PUT{id}** - Efetua alterações a um género com um respetivo ID.
- **PATCH{id}** - Requisições sucessivas idênticas de género através do ID podem obter efeitos distintos.
- **GET{id}** - Dado um ID, retorna o respetivo género.
- **DELETE{id}** - Remove da base de dados um género com um respetivo ID.
- **POST** - Adiciona um género à base de dados.
- **PATCH** - Requisições sucessivas idênticas de género podem obter efeitos distintos.
- **GET** - Retorna todos os géneros presentes na base de dados.

> GenreMoviesController

É composto por 4 funcionalidades:

- **POST{id}** - Adiciona um filme através do ID do género à base de dados.
- **PATCH{id}** - Requisições sucessivas idênticas de género através do ID podem obter efeitos distintos.
- **GET{id}** - Dado um ID de um género, retorna o(s) respetivo(s) filme(s).
- **DELETE{id}** - Remove da base de dados um/os filme/s com um respetivo ID do género.

> MoviesController

É composto por 8 funcionalidades:

- **GET (count)** - Retorna o nº total de filmes presentes na base de dados.
- **PUT{id}** - Efetua alterações a um filme com um respetivo ID.
- **PATCH{id}** - Requisições sucessivas idênticas de filmes através do ID podem obter efeitos distintos.
- **GET{id}** - Dado um ID, retorna o respetivo filme.
- **DELETE{id}** - Remove da base de dados um filme com um respetivo ID.
- **POST** - Adiciona um filme à base de dados.
- **PATCH** - Requisições sucessivas idênticas de filme podem obter efeitos distintos.
- **GET** - Retorna todos os filmes presentes na base de dados.

> MoviesActorController

É composto por 1 funcionalidade:

- **GET{id}** - Dado um ID de um ator, retorna o(s) respetivo(s) filme(s).

> MoviesDirectorController

É composto por 1 funcionalidade

- **GET{id}** - Dado um ID de um ator, retorna o(s) respetivo(s) filme(s).

> MoviesGenreController

É composto por 4 funcionalidades:

- **POST{id}** - Adiciona um filme através do ID do género à base de dados.
- **PATCH{id}** - Requisições sucessivas idênticas de género através do ID podem obter efeitos distintos.
- **GET{id}** - Dado um ID de um género, retorna o(s) respetivo(s) filme(s).
- **DELETE{id}** - Remove da base de dados um/os filme/s com um respetivo ID do género.

> TroughController

É a tabela de relação entre 'Movies' e 'Genre', composto por 8 funcionalidades:

- **GET (count)** - Retorna o nº total de filmes e géneros presentes na base de dados.
- **PUT{id}** - Efetua alterações a um filme e género com um respetivo ID.
- **PATCH{id}** - Requisições sucessivas idênticas de filmes e géneros através do ID podem obter efeitos distintos.

- **GET{id}** - Dado um ID, retorna o respetivo filme e género.
- **DELETE{id}** - Remove da base de dados um filme e género com um respetivo ID.
- **POST** - Adiciona um filme e género à base de dados.
- **PATCH** - Requisições sucessivas idênticas de filme e género podem obter efeitos distintos.
- **GET** - Retorna todos os filmes e géneros presentes na base de dados.

Group 01

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

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