

# Documentação da API de Filmes e Usuários

## Integrantes

- Felipe Anselmo - 560661
- João Vinicius Alves - 559369
- Matheus Mariotto - 560276

---

Esta documentação descreve os endpoints da **API** para gerenciar dados de filmes, usuários, avaliações, gêneros e listas de favoritos. A API foi desenvolvida utilizando **Spring Boot** e segue o padrão **OpenAPI 3.0.1**.

---

## Modelos de Dados (Schemas)

### User (Usuário)

Propriedade	Tipo	Formato	Descrição	Obrigatório
id	integer	int64	Identificador único do usuário.	Não
name	string		Nome do usuário.	Sim
email	string		Endereço de e-mail.	Sim
password	string		Senha.	Sim
age	integer	int32	Idade do usuário.	Não
reviews	array	Review	Lista de avaliações feitas pelo usuário.	Não

### Movie (Filme)

Propriedade	Tipo	Formato	Descrição	Obrigatório
-------------	------	---------	-----------	-------------

id	integer	int64	Identificador único do filme.	Não
title	string		Título do filme.	Sim
description	string		Descrição detalhada.	Sim
releaseDate	string	date	Data de lançamento (ex: AAAA-MM-DD).	Sim
averageRating	number	double	Classificação média do filme.	Não
genres	array	Genre	Lista de gêneros associados.	Não
reviews	array	Review	Lista de avaliações do filme.	Não

Review (Avaliação)

Propriedade	Tipo	Formato	Descrição	Obrigatório
id	integer	int64	Identificador único da avaliação.	Não
user	object	User	Usuário que fez a avaliação.	Não
movie	object	Movie	Filme avaliado.	Não
rating	number	double	Nota da avaliação.	Não
comment	string		Comentário da avaliação.	Não

Genre (Gênero)

Propriedade	Tipo	Formato	Descrição	Obrigatório
-------------	------	---------	-----------	-------------

id	integer	int64	Identificador único do gênero.	Não
name	string		Nome do gênero.	Sim

### WatchList (Lista de Favoritos)

Propriedade	Tipo	Formato	Descrição	Obrigatório
id	integer	int64	Identificador único da lista.	Não
name	string		Nome da lista.	Sim
user	object	User	Usuário proprietário da lista.	Não
movies	array	Movie	Lista de filmes na WatchList.	Não

---

## Endpoints (APIs)

### 1. Gerenciamento de Filmes (/movies)

Método	Caminho	Descrição	Parâmetros/Corpo	Resposta de Sucesso
GET	/movies	Retorna todos os filmes com filtragem e paginação.	filters (MovieFilter), pageable (Pageable)	200 OK (PageMovie)
POST	/movies	Cria um novo filme.	Corpo: Movie	201 Created
GET	/movies/{id}	Busca um filme pelo ID.	Parâmetro de Path: id (int64)	200 OK (Movie)

PUT	/movies/{id}	Atualiza um filme existente.	Parâmetro de Path: id (int64), Corpo: Movie	200 OK (Movie)
DELETE	/movies/{id}	Exclui um filme pelo ID.	Parâmetro de Path: id (int64)	204 No Content

## 2. Gerenciamento de Usuários (/users)

Método	Caminho	Descrição	Parâmetros/Corpo	Resposta de Sucesso
GET	/users	Retorna todos os usuários com filtragem e paginação.	filters (UserFilter), pageable (Pageable)	200 OK (PageUser)
POST	/users	Cria um novo usuário.	Corpo: User	201 Created
GET	/users/{id}	Busca um usuário pelo ID.	Parâmetro de Path: id (int64)	200 OK (User)
PUT	/users/{id}	Atualiza um usuário existente.	Parâmetro de Path: id (int64), Corpo: User	200 OK (User)
DELETE	/users/{id}	Exclui um usuário pelo ID.	Parâmetro de Path: id (int64)	204 No Content

## 3. Gerenciamento de Avaliações (/reviews)

Método	Caminho	Descrição	Parâmetros/Corpo	Resposta de Sucesso
GET	/reviews	Retorna todas as avaliações com filtragem e paginação.	filters (ReviewFilter), pageable (Pageable)	200 OK (PageReview)
POST	/reviews	Cria uma nova avaliação.	Corpo: Review	201 Created

GET	/reviews/{id}	Busca uma avaliação pelo ID.	Parâmetro de Path: id (int64)	200 OK (Review)
PUT	/reviews/{id}	Atualiza uma avaliação existente.	Parâmetro de Path: id (int64), Corpo: Review	200 OK (Review)
DELETE	/reviews/{id}	Exclui uma avaliação pelo ID.	Parâmetro de Path: id (int64)	204 No Content

#### 4. Gerenciamento de Gêneros (/genres)

Método	Caminho	Descrição	Parâmetros/Corpo	Resposta de Sucesso
GET	/genres	Retorna todos os gêneros com filtragem e paginação.	filters (GenreFilter), pageable (Pageable)	200 OK (PageGenre)
POST	/genres	Cria um novo gênero.	Corpo: Genre	201 Created
GET	/genres/{id}	Busca um gênero pelo ID.	Parâmetro de Path: id (int64)	200 OK (Genre)
PUT	/genres/{id}	Atualiza um gênero existente.	Parâmetro de Path: id (int64), Corpo: Genre	200 OK (Genre)
DELETE	/genres/{id}	Exclui um gênero pelo ID.	Parâmetro de Path: id (int64)	204 No Content

#### 5. Gerenciamento de Listas de Favoritos (/watch-lists)

Método	Caminho	Descrição	Parâmetros/Corpo	Resposta de Sucesso
GET	/watch-lists	Retorna todas as listas com filtragem e paginação.	filters (WatchListFilter), pageable (Pageable)	200 OK (PageWatchList)

POST	/watch-lists	Cria uma nova lista.	Corpo: WatchList	201 Created
GET	/watch-lists/{id}	Busca uma lista pelo ID.	Parâmetro de Path: id (int64)	200 OK (WatchList)
PUT	/watch-lists/{id}	Atualiza uma lista existente.	Parâmetro de Path: id (int64), Corpo: WatchList	200 OK (WatchList)
DELETE	/watch-lists/{id}	Exclui uma lista pelo ID.	Parâmetro de Path: id (int64)	204 No Content

## Detalhes de Filtragem e Paginação

Os endpoints de listagem (GET em coleções) usam parâmetros de *query* para **filtragem** e **paginação** no padrão do Spring Data JPA:

### Paginação (Pageable)

Propriedade	Tipo	Descrição
page	integer (int32)	Número da página (começa em 0).
size	integer (int32)	Número de elementos por página (mínimo 1).
sort	array of string	Campos para ordenação (ex: title, asc ou name, desc).

### Filtros (Exemplos)

Filtro	Controlador	Propriedades
MovieFilter	/movies	title, description, starDate (data de início), endDate (data de fim), minRating, maxRating, genresIds
UserFilter	/users	name, minAge, maxAge
ReviewFilter	/reviews	userId, movieId, minRating, maxRating

WatchListFilter	/watch-lists	name, moviesIds
GenreFilter	/genres	name