

DOCUMENTAÇÃO MovieStore API REST

Este documento explicita com exemplos, como utilizar os recursos disponíveis no REST API de Cadastro e Pesquisa por Filmes. Assim como, as formas de se realizar uma requisição e suas possíveis respostas.

1. Consultar Filmes

Requisição

Requisição para listar todos os filmes do sistema, podendo opcionalmente receber filtros personalizados via path, de forma que se o cliente não definir nenhum parâmetro de consulta (nenhum filtro), os parâmetros receberão os valores padrão.

- Possíveis parâmetros de consulta
 - 🕒 `faixa_etaria_min` ⇒ Filtra filmes que possuam a faixa etária determinada em diante. Padrão: 0
 - 🕒 `faixa_etaria_max` ⇒ Filtra filmes que possuam a faixa etária de até no máximo aquilo que foi determinado. Padrão: 100
 - 🕒 `ano_min` ⇒ Filtra filmes pelo ano no qual foram lançados em diante. Padrão: 0
 - 🕒 `ano_max` ⇒ Filtra todos os filmes que foram lançados até o `ano_max` determinado. Padrão: 10000
 - 🕒 `genero` ⇒ Filtra os filmes pelo seu gênero tais como: *ação*, *comédia*, *terror*, *suspense*, *romance* e *drama*. Padrão: Nulo
 - 🕒 `limit` ⇒ Quantidade máxima de elementos exibidos por página. Padrão: 10
 - 🕒 `offset` ⇒ Quantidade de elementos pular (geralmente múltiplo de `limit`). Padrão: 0

Method	URL
GET	/filmes?faixa_etaria_min=14&ano_min=2016&genero=ação

Resposta

Como resposta, obtém-se uma lista de hotéis que se enquadram nos filtros da requisição acima:

Status	Response Body
200 OK	<pre> { "filmes": [{ "nome": "https://jovemnerd.com.br/wp-content/uploads/2019/04/hqx3vmrkk0u21-810x1200.jpg", "faixa_etaria": "Projeto Gemini", "ano": 14, "genero": 2019, "diretor": "ação", "sinopse": "Ang Lee" }, { "nome": "https://br.web.img3.acsta.net/pictures/17/06/08/16/58/110585.jpg", "faixa_etaria": "Dunkirk", "ano": 14, "genero": 2017, "diretor": "ação", "sinopse": "Cristopher Nolan" }] } </pre>

Requisição

Requisição para visualizar os dados de um filme específico. Faz-se um GET de `/filmes/{filme_id}`

Method	URL
GET	/filmes/1

Resposta

Como resposta, obtém-se um JSON com os dados do filme requisitado.

Status	Response Body
200 OK	<pre>"filme_id": 1, "imagem": "https://s1.static.brasile Escola.uol.com.br/be/conteudo/images "nome": "Eu Sou a Lenda", "faixa_etaria": 14, "ano": 2017, "genero": "terror", "diretor": "Francis Lawrence", "sinopse": "Robert Neville é um brilhante cientista e o único sobrevive "ator_id": 1</pre>

Requisição

Requisição exemplo de quando o usuário pesquisar por um filme que não existe.

Method	URL
GET	/filmes/1902381841

Resposta

Como resposta, obtém-se uma mensagem de erro, dizendo que o filme não foi encontrado.

Status	Response Body
404 Not Found	<pre>"message": "Movie not found."</pre>

2. Cadastro de Usuário

Requisição

Exemplo de Requisição cadastrar um novo usuário.

Method	URL
POST	/user/cadastro

Header	
Content-Type	application/json

Request Body
<pre>{ "login": "Arthur01", "password": "1234" }</pre>

Resposta

Como resposta, obtém-se uma mensagem de sucesso informado que usuário foi criado, e status code 201 Created (Criado).

Status	Response Body
201 Created	<pre>{ "message": "User created successfully!" }</pre>

Requisição

Exemplo de Requisição para tentar cadastrar outro usuário com login “Arthur01”.

Method	URL
POST	user/cadastro

Header	
Content-Type	application/json

Request Body
<pre>{ "login": "Arthur01", "password": "1234" }</pre>

Resposta

Como resposta, obtém-se uma mensagem de erro, informando que um usuário chamado “ana” já existe.

Status	Response Body
400 Bad Request	<pre>{ "message": "The login Arthur01 already exists." }</pre>

3. Login de Usuário

Requisição

Exemplo de Requisição logar com um usuário.

Method	URL
POST	/user/login

Header	
Content-Type	application/json

Request Body
<pre>{ "login": "Arthur01", "password": "1234" }</pre>

Resposta

Como resposta, obtém-se uma mensagem o token de acesso que será necessário para fazer as requisições que só podem ser feitas com login.

Status	Response Body
200 OK	<pre>{ "access token": "eyJ0eXAiOiJKV1QiLCJhbGc:</pre>

Requisição

Exemplo de Requisição para tentar fazer login com um usuário que não existe.

Method	URL
POST	/user/login

Header	
Content-Type	application/json

Request Body
<pre>{ "login": "nao_existo", "password": "1234" }</pre>

Resposta

Como resposta, obtém-se uma mensagem de erro 401 não autorizado, informando que usuário ou senha estão incorretos.

Status	Response Body
401 unauthorized	<pre>{ "message": "This username or password is incorrect" }</pre>

4. Criar Filme

Requisição

Exemplo de Requisição para criar um novo filme: **/filme**

Method	URL
POST	/filme

Header	
Content-Type	application/json
Authorization	Bearer {token_de_acesso}

Request Body
<pre>{ "imagem": "https://br.web.img2.acsta.net/pictures/20/11/20/11/21/5597135.jpg", "nome": "Vozes que Inspiram", "faixa_etaria": 12, "ano": 2020, "genero": "documentario", "diretor": "Fernando Villena e James D. Stern", "sinopse": "Seis ambiciosos estudantes de teatro encaram os testes para participar da August Wilson Monologue Competition, que culmina em uma final incrível na Broadway.", "ator_id": 10 }</pre>

Resposta

Exemplos de Respostas: primeiro, a mensagem de sucesso ao criar pela primeira vez. Ao tentar criar novamente um filme com nome “Vozes que Inspiram”, o sistema emite uma mensagem de erro, informando que o nome “Vozes que Inspiram” já existe.

Status	Response Body
--------	---------------

201 Created	<pre> { "message": { "filme_id": 50, "imagem": "https://br.web.img2.acsta.net/pictures/20/11/20/11/21/5597135.jpg", "nome": "Vozes que Inspiram", "faixa_etaria": 12, "ano": 2020, "genero": "documentario", "diretor": "Fernando Villena e James D. Stern", "sinopse": "Seis ambiciosos estudantes de teatro encaram os testes para participar da August Wilson Monologue", "ator_id": 10 } } </pre>
400 Bad Request	<pre> { "message": "The movie 'Vozes que Inspiram' already exists." } </pre>

5. Deletar Filme

Requisição

Exemplo de Requisição para deletar um filme: `/filmes/{filme_id}`

Method	URL
DELETE	/filmes/50

Header	
Authorization	Bearer {token_de_acesso}

Resposta

Exemplos de Respostas: primeiro, a mensagem de sucesso ao deletar um filme existente. Depois, ao tentar deletar o mesmo filme, obtém-se o erro 404 not found, informando que o filme não existe. No terceiro exemplo, o cliente esqueceu de enviar o token de autorização.

Status	Response Body
200 OK	<pre> { "message": "Movie Name: Vozes que Inspiram, has deleted" } </pre>

404 Not Found	<pre>"message": "Movie not found."</pre>
401 Unauthorized	<pre>"msg": "Missing Authorization Header"</pre>

7. Logout de Usuário

Requisição

Exemplo de Requisição para fazer logout de usuário. Envia-se o token de acesso, e esse token é invalidado.

Method	URL
POST	user/logout

Header	
Authorization	Bearer {token_de_acesso}

Resposta

Exemplo de Resposta: mensagem de sucesso informando que o usuário foi deslogado. Ao tentar usar esse token de acesso em qualquer requisição, ele não funcionará mais, a não ser que o usuário faça o login.

Status	Response Body
200 OK	<pre>"message": "Logged out successfully"</pre>

8. Consultar dados de Usuário

Requisição

Exemplo de Requisição para ler dados de usuário: `/users/{user_id}`

Method	URL
GET	/users/1

Header	
Content-Type	application/json

Resposta

Como resposta, obtém-se os dados do usuário com id “1” exceto a senha.

Status	Response Body
200 OK	<pre>{ "user_id": 1, "login": "Arthur01" }</pre>

9. Deletar Usuário

Requisição

Exemplo de Requisição para deletar um usuário: `/users/{user_id}`

Method	URL
DELETE	/users/1

Header	
--------	--

Authorization	Bearer {token_de_acesso}
---------------	--------------------------

Resposta

Exemplos de Respostas, primeiro, a mensagem de sucesso ao deletar um usuário existente. Depois, ao tentar deletar o mesmo usuário, obtém-se o erro 404 not found, informando que o usuário não existe ou não foi encontrado. No terceiro exemplo, o cliente enviou um token de autorização expirado.

Status	Response Body
200 OK	<pre>{ "message": "User 'Arthur01' has deleted." }</pre>
404 Not Found	<pre>{ "message": "User not found." }</pre>
401 Unauthorized	<pre>{ "msg": "Token has expired" }</pre>

5. Consultar Atores

Ao consultar os atores, os filmes associados a eles através de chave estrangeira e chave primeira são retornados.

Requisição

Exemplo de Requisição para consultar os atores: **/atores**

Method	URL
GET	/atores

Resposta

Status	Response Body
200 OK	<pre>{ "atores": [{ "ator_id": 1, "foto": "http://t0.gstatic.com/licensed-image?q=tbn:ANd9GcRsnzWW3oUc0FdrEb2y0BZLbetDRgC5zJnm404eG6SH_aR4Yt1U", "nome": "Will Smith", "idade": 53, "altura": 1.88, "filmes": [{ "filme_id": 1, "imagem": "https://s1.static.brasilecola.uol.com.br/be/conteudo/images/o-filme-eu-sou-lenda-pode-se-", "nome": "Eu Sou a Lenda", "faixa_etaria": 14, "ano": 2017, "genero": "terror", "diretor": "Francis Lawrence", "sinopse": "Robert Neville é um brilhante cientista e o único sobrevivente de uma epidemia que trans:", "ator_id": 1 }] }], }</pre>

Também podemos consultar os atores individualmente usando o seu id.

Requisição

Exemplo de Requisição para consultar os atores: `/atores/{id}`

Method	URL
GET	/atores/4

Resposta

Status	Response Body
200 OK	<pre>{ "message": { "ator_id": 4, "foto": "https://media.gettyimages.com/photos/actress-jennifer-aniston-attends-the-21st-annual-screen-actors-guild-picture-id462239", "nome": "Jennifer Aniston", "idade": 53, "altura": 1.64, "filmes": [{ "filme_id": 19, "imagem": "https://br.web.img2.acsta.net/medias/nmedia/18/87/34/28/20028730.jpg", "nome": "Esposa de Mentirinha", </pre>
404 Not Found	<pre>{ "message": "Actor id '12' not found." }</pre>