Documentação da API

Índice

Introdução

Este é o guia oficial da API do nosso aplicativo. Aqui você encontrará todas as informações necessárias para utilizar nossa API e desenvolver sua própria integração.

Para usar a API, é necessário ter uma chave de acesso, que pode ser obtida através de um cadastro em no site OMDB.

Autenticação

Descrição

O sistema de autenticação utiliza tokens JWT (JSON Web Tokens) para autenticar usuários em uma aplicação web. A funcionalidade é composta por duas partes: a função signup, responsável por criar um novo usuário na base de dados, criptografar a senha e gerar um token JWT para autenticar o usuário na aplicação; e a função AuthMiddleware, que é um middleware para proteger as rotas que requerem autenticação.

A função signup é responsável por receber as informações do novo usuário (username, email e password), e verifica se todas as informações foram fornecidas.

Se alguma informação faltar, a função retorna uma resposta HTTP com status 400 e uma mensagem de erro. Em seguida, a função verifica se já existe um usuário com o mesmo e-mail na base de dados. Caso já exista, a função retorna uma resposta HTTP com status 400 e uma mensagem de erro informando que o e-mail já está sendo utilizado. Caso contrário, a função criptografa a senha do usuário e insere um novo registro na base de dados, retornando uma resposta HTTP com status 201 contendo o novo usuário e o token JWT gerado.

Já a função AuthMiddleware é responsável por verificar se o token JWT presente na requisição é válido e corresponde a um usuário existente na base de dados. Primeiramente, a função extrai o token da requisição HTTP, verificando se ele está no formato esperado e não está expirado. Em seguida, a função decodifica o token, obtendo o ID do usuário. Com o ID do usuário, a função faz uma consulta na base de dados, buscando o registro correspondente. Se o registro for encontrado, o middleware adiciona o usuário à requisição HTTP, e a função next() é chamada, permitindo que a requisição prossiga para a rota protegida. Caso contrário, a função retorna uma resposta HTTP com status 401 (Unauthorized) ou 500 (Internal Server Error), informando que o token é inválido ou houve um erro interno no servidor.

O uso do JWT traz algumas vantagens para o sistema de autenticação, como o fato de ser uma solução sem estado (stateless), ou seja, a sessão do usuário não é armazenada no servidor, o que permite uma escalabilidade mais fácil da aplicação. Além disso, o JWT permite que o servidor possa validar rapidamente se o token é válido, evitando consultas desnecessárias na base de dados a cada requisição. No entanto, é importante tomar algumas precauções ao usar tokens JWT, como definir um tempo de expiração adequado, usar algoritmos de criptografia seguros e não armazenar informações sensíveis no payload do token.

POST /auth/register

Essa função é responsável por criar um novo usuário no banco de dados, a partir dos dados fornecidos pelo corpo da requisição (req.body). Além disso, a função também verifica se o email informado já foi cadastrado anteriormente e, em caso afirmativo, retorna uma mensagem de erro.

Parâmetros Os seguintes parâmetros devem ser enviados no corpo da requisição:

- username (obrigatório): nome de usuário do novo usuário.
- email (obrigatório): email do novo usuário.
- password (obrigatório): senha do novo usuário.

Resposta Sucesso - Código: 201 - Conteúdo: objeto JSON com os seguintes campos: - user: objeto JSON contendo os dados do novo usuário cadastrado no banco de dados, incluindo o id gerado pelo banco de dados. - token: token de autenticação do novo usuário, gerado pela biblioteca jsonwebtoken.

Exemplo de resposta:


```
{
    "user": {
        "id": 9,
        "username": "test",
        "email": "test33@test.com"
    },
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6OSwidXNlcm5hbWUiOiJ0ZXN0IiwiaWF0
}
```


Erros

Código: 400 - Conteúdo: objeto JSON com a mensagem de erro correspondente a um dos seguintes casos: - Email is required: email não foi fornecido no corpo da requisição. - Username is required: nome de usuário não foi fornecido no corpo da requisição. - Password is required: senha não foi fornecida no corpo da requisição. - Email already taken: o email fornecido já foi cadastrado anteriormente. Código: 500 - Conteúdo: objeto JSON com a mensagem de erro Internal server error. Isso pode ocorrer em caso de falha ao executar alguma operação no banco de dados ou ao gerar o token de autenticação.

Reviews

Descrição

O CRUD implementado é um sistema básico que permite que os usuários gerenciem avaliações de filmes. As funções incluem criar uma nova avaliação, selecionar todas as avaliações existentes, selecionar uma avaliação específica por seu ID, apagar uma avaliação, atualizar uma avaliação por completo e atualizar parcialmente uma avaliação.

Endpoints

POST /api/review

Essa função é responsável por criar uma nova avaliação no banco de dados, a partir dos dados fornecidos pelo corpo da requisição (req.body). O usuário deve fornecer um titulo, um comentario e uma nota. Além disso, a função pode realizar algumas verificações adicionais, como verificar se o usuário que está criando a avaliação tem permissão para avaliar o produto ou serviço em questão. Caso alguma verificação falhe, a função retorna uma mensagem de erro. Se a avaliação for criada com sucesso, a função retorna a nova avaliação criada. #### Parâmetros Os seguintes parâmetros devem ser enviados no corpo da requisição:

- title (obrigatório): titulo do filme.
- rating (obrigatório): nota do filme (0 a 5).

- comment (obrigatório): review do filme.
- ispublic: booleano para definir se outros usuarios podem ver a sua avaliação (Default=false).
- token: autenticação do usuário.

Resposta Sucesso - Código: 201 - Conteúdo: objeto JSON com os seguintes campos: - message: Review criada com sucesso! - review: objeto json contendo todos dados da nova review.

Exemplo de resposta:


```
{
    "message": "Review criada com sucesso!",
    "body": {
        "review": {
            "id": 32,
            "userid": 1,
            "movieid": 1,
            "rating": 1,
            "review": "Não gostei",
            "ispublic": false,
            "created_at": "2023-05-15T23:45:52.763Z"
        }
    }
}
```


Erros

Código: 400 - Conteúdo: objeto JSON com a mensagem de erro correspondente a um dos seguintes casos: - Title is required: titulo não foi fornecido no corpo da requisição. - Rating is required: nota não foi fornecido no corpo da requisição. - Review is required: avaliação não foi fornecida no corpo da requisição. - O campo 'rating' deve ser um número entre 0 e 5: usuario digitou um rating invalido. Código: 500 - objeto JSON com a mensagem de erro Internal server error. Isso pode ocorrer em caso de falha ao executar alguma operação no banco de dados ou ao gerar o token de autenticação.

GET /api/review

Essa função é responsável por retornar todas as avaliações existentes no banco de dados. Ela não requer nenhum parâmetro no corpo da requisição. No entanto, o usuário pode realizar algumas verificações adicionais, como verificar se ele tem permissão para visualizar todas as avaliações existentes. Caso alguma verificação falhe, a função retorna uma mensagem de erro.

Parâmetros Os seguintes parâmetros devem ser enviados no corpo da requisição:

- token (obrigatório): autenticação do usuário

Resposta Sucesso - Código: 200 - Conteúdo: objeto JSON com os seguintes campos: - reviews: array contendo todos objetos JSON com as informações das avaliações. - message: Review(s) encontrada(s) com sucesso!

Exemplo de resposta:


```
{
    "reviews": [
        {
            "id": 32,
            "userid": 1,
            "movieid": 1,
            "rating": 1,
            "review": "Não gostei",
            "ispublic": false,
            "created_at": "2023-05-15T23:45:52.763Z"
        },
        {
            "id": 33,
            "userid": 2,
            "movieid": 1,
            "rating": 4,
            "review": "Gostei muito",
            "ispublic": true,
            "created_at": "2023-05-16T00:22:10.523Z"
        }
    ],
    "message": "Review(s) encontrada(s) com sucesso!"
}
```


Erros

Código: 400 - O usuario não possui reviews

Código: 500 - objeto JSON com a mensagem de erro Internal server error. Isso pode ocorrer em caso de falha ao executar alguma operação no banco de dados.

GET /api/review/:id

Essa função é responsável por retornar uma avaliação específica do banco de dados, com base no id fornecido na URL da requisição (req.params.id). Ela pode realizar algumas verificações adicionais, como verificar se o usuário tem permissão para visualizar a avaliação em questão. Caso alguma verificação falhe, a função retorna uma mensagem de erro.

Parâmetros

O seguinte parâmetro deve ser enviado na URL da requisição:

id (obrigatório): id da avaliação que se deseja buscar. token: autenticação do usuario.

Resposta Sucesso

Código: 200 Conteúdo: objeto JSON com os seguintes campos: review: objeto json contendo todas informações da avaliação. message: Review encontrada com sucesso! Exemplo de resposta:

```
{
"review": {
"id": 32,
"userid": 1,
"movieid": 1,
"rating": 1,
"review": "Não gostei",
"ispublic": false,
"created_at": "2023-05-15T23:45:52.763Z"
},
"message": "Review encontrada com sucesso!"
}
```

Erros

Código: 400 - objeto JSON com a mensagem de erro correspondente a um dos seguintes casos: - Id is required: o id não foi fornecido na URL da requisição. - Invalid id: o id fornecido na URL da requisição não é válido.

Código: 404 - objeto JSON com a mensagem de erro Review not found. Isso ocorre quando a avaliação com o id fornecido na URL não foi encontrada no banco de dados.

Código: 500 - objeto JSON com a mensagem de erro Internal server error. Isso pode ocorrer em caso de falha ao executar alguma operação no banco de dados.

DELETE /api/review/:id

Essa função é responsável por deletar uma avaliação específica do banco de dados. O parâmetro ":id" é o identificador único da avaliação a ser deletada e deve ser informado na URL da requisição.

Parâmetros

:id (obrigatório) - Identificador único da avaliação a ser deletada. token: autenticação do usuario.

Resposta

Sucesso

Código: 200 Conteúdo: objeto JSON com a mensagem de sucesso. Exemplo de resposta:

```
{
"message": "Review deletada com sucesso!"
}
```

Erros

Código: 404 - objeto JSON com a mensagem de erro Review não encontrada. Isso pode ocorrer caso o identificador único informado não corresponda a uma avaliação existente no banco de dados.

Código: 500 - objeto JSON com a mensagem de erro Internal server error. Isso pode ocorrer em caso de falha ao executar alguma operação no banco de dados.

PUT /api/review/:id

Essa função é responsável por atualizar completamente uma avaliação específica no banco de dados. O parâmetro ":id" é o identificador único da avaliação a ser atualizada e deve ser informado na URL da requisição.

Parâmetros

:id (obrigatório) - Identificador único da avaliação a ser atualizada. token: autenticação do usuario

Corpo da requisição: objeto JSON contendo os seguintes campos:

- userid (obrigatório) - Identificador único do usuário que realizou a avaliação.
- movieid (obrigatório) - Identificador único do filme que foi avaliado.
- rating (obrigatório) - Nota atribuída à avaliação, deve ser um número inteiro entre 1 e 5.
- review (opcional) - Texto contendo a avaliação em si.
- ispublic (opcional) - Indica se a avaliação pode ser visualizada por outros usuários. Deve ser um valor booleano (true ou false).

Resposta

Sucesso Código: 200

Conteúdo: objeto JSON com a mensagem de sucesso e as informações atualizadas da avaliação.

message: Review atualizada com sucesso!

review: objeto JSON contendo as informações atualizadas da avaliação. Exemplo de resposta:


```
{
"message": "Review atualizada com sucesso!",
"review": {
"id": 32,
"userid": 1,
"movieid": 1,
"rating": 4,
"review": "Gostei muito!",
"ispublic": true,
"created_at": "2023-05-15T23:45:52.763Z",
"updated_at": "2023-05-16T01:22:10.523Z"
}
}
```


Erros

Código: 400 - objeto JSON com a mensagem de erro Dados inválidos. Isso pode ocorrer caso os dados informados no corpo da requisição sejam inválidos ou estejam incompletos.

Código: 404

- objeto JSON com a mensagem de erro Review não encontrada. Isso pode ocorrer caso o identificador único informado não corresponda a uma avaliação existente no banco de dados.

Código: 500

- objeto JSON com a mensagem de erro Internal server error. Isso pode ocorrer em caso de falha ao executar alguma operação no banco de dados.

PATCH /api/review/:id

Essa função é responsável por atualizar parcialmente uma avaliação específica no banco de dados. O parâmetro ":id" é o identificador único da avaliação a ser atualizada e deve ser informado na URL da requisição.

Parâmetros

:id (obrigatório) - Identificador único da avaliação a ser atualizada. token: autenticação de usuario

Corpo da requisição: objeto JSON contendo um ou mais dos seguintes campos: rating - Nota atribuída à avaliação, deve ser um número inteiro entre 1 e 5. review - Texto contendo a avaliação em si. ispublic - Indica se a avaliação pode ser visualizada por outros usuários. Deve ser um valor booleano (true ou false).

Resposta

Sucesso Código: 200

Conteúdo: objeto JSON com a mensagem de sucesso e as informações atualizadas da avaliação.

message: Review atualizada com sucesso!

review: objeto JSON contendo as informações atualizadas da avaliação.

Exemplo de resposta:


```
{
"message": "Review atualizada com sucesso!",
"review": {
"id": 32,
"userid": 1,
"movieid": 1,
"rating": 4,
"review": "Gostei muito!",
"ispublic": true,
"created_at": "2023-05-15T23:45:52.763Z",
"updated_at": "2023-05-16T01:22:10.523Z"
}
}
```


Erros

Código: 400 - objeto JSON com a mensagem de erro Dados inválidos. Isso pode ocorrer caso os dados informados no corpo da requisição sejam inválidos ou estejam incompletos.

Código: 404 - objeto JSON com a mensagem de erro Review não encontrada. Isso pode ocorrer caso o identificador único informado não corresponda a uma avaliação existente no banco de dados.

Código: 500 - objeto JSON com a mensagem de erro Internal server error. Isso pode ocorrer em caso de falha ao executar alguma operação no banco de dados.