Exercicio 4all

Especificação Técnica

Sistema de Locadora – *Rent Store*

Desenvolvido em NodeJs

Banco de Dados

Mysql

- A tabela de usuários foi criada com intuito de gravar > nome, e-mail e senha.
- A tabela de filmes foi criada para gravar discos, para que possa obter várias cópias do mesmo título
- A tabela de aluguéis, irá gravar a chave de identificação do filme alugado.

Instalação banco de dados (opcional)

- Instalação do banco de dados postgres, se tiver o docker instalado na máquina é necessário que a porta 5432 esteja liberada e que o comando "docker run --name postgresDB -p 5432:5432 -d -t kartoza/postgis" seja rodado e finalizado corretamente. (Opcional)

Os dados da API foram trabalhados encima do Sequelize.
Portanto, é necessário informar os dados do banco no arquivo de configuração em:

- src/Config/database.js.

Com as informações do banco de dados já fornecidas, execute a seguinte instrução dentro do diretório da API através do Client do Sequelize:

Para rodar o back-end;

Dependências

- Entrar na pasta raiz e rodar o comando yarn ou npm install, para instalar as dependências.

Migração DB

- Após a configuração do arquivo (/Config/database.js) na raiz do projeto (src), rodar o comando

"npx sequelize db:migrate ou
yarn sequelize db:migrate”

Start

- Na raiz (src) rodar o comando yarn dev ou npm run start.

Também é possível criar a base de dados manualmente através de um script de criação de banco de dados. Nesta base de dados, também são realizados registros de filmes e discos. [O schema de banco de dados é disponibilizado em (schema.sql).

Aplicação

A API foi desenvolvida utilizando o Sucrase.

Através do Sucrase, a sintaxe Import/Export é utilizada. Para executar a API através do script definido junto ao package.json:

Criação de Usuários

Para iniciar no sistema, será necessário informar:

- **Nome**
- **E-mail**
- **Senha**

URL (POST):

```
http://127.0.0.1:3000/users
```

Body da Requisição:

```json
{
  "name": "Andrey",
  "email": "a.elyan.s@gmail.com",
  "password": "123456"
}
```

Update de Usuários

O usuário pode realizar alteração de seus dados, por exemplo:

URL (PUT):

```
http://127.0.0.1:3000/users
```

Body da Requisição:

```json
{
  "name": "Andrey",
  "email": a.elyan.s@gmail.com
}
```

Update da Senha

Para realizar a atualização da senha será necessário informar:

- **Nome**
- **E-mail**
- **Senha Antiga**
- **Nova Senha**
- **Confirmação da Nova senha**

URL (PUT):

Body da Requisição:

```
{
  "name":"Andrey",
  "email":"a.elyan.s@gmail.com",
  "oldPassword": "123456",
  "password":"10203040",
  "password":"10203040"
}
```

Sessões de acesso

Um token será gerado para cada sessão de acesso no servidor

URL (PUT):

```
http://127.0.0.1:3000/sessions
```

Body da Requisição:

```
{
  "email":"a.elyan.s@gmail.com",
  "password":"123456789"
}
```

Resultado Esperado:

```
{
  "user": {
    "id": 1,
    "name": "Andrey",
    "email": " a.elyan.s@gmail.com "
  },
  "token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwiaWF0IjoxNTYyMjU3ODE2LCJleHAi
fgfgytjhtyjtyjyukuikrtyh.ertrtertrtertertrt2345546756778SCSDF"
}
```

Aluguel de Filmes

Baseando-se em grandes empresas como **Netflix**, vamos considerar que, para o Usuário locar os filmes, seja necessário possuir um token de autenticação no corpo da requisição. No *body* da requisição, enviamos o ID do filme que o usuário quer locar. Será gerado ID automaticamente através do token enviado.

URL (POST):

```
http://127.0.0.1:3000/rent
```

Body da Requisição:

```
{
 "disk_id": 1
}
```

Devolução de Filmes

A devolução do filme é realizada através do método PUT.

URL (PUT):

```
http://127.0.0.1:3000/rent
```

Body da Requisição:

```
{
 "disk id": 1
}
```

Lista de Filmes

URL (GET):

```
http://127.0.0.1:3000/movies
```

Resposta Esperada:

http://localhost:3000/movies/search?title=EndGame


```
[
  {
    "id": 2,
    "title": " End Game ",
    "director": "Anthony Russo, Joe Russo"
  },
  {
    "id": 3,
    "title": "DoctorStrange
    ", "director": "Jon
  }
]
```


Busca de Filmes por Nome

URL (GET):

Resposta Esperada:

```
http://127.0.0.1:3000/users
```


```
[
  {
    "id": 2,
    "title": "End Game",
    "director": "Anthony Russo, Joe Russo"
  }
]
```