Vidly API Documentation

This project provides an API for managing data related to movies, customers, and genres. You can perform CRUD operations for each entity.

Usage

To interact with the API, you will need a valid JWT token. After logging in or registering, include your token in the headers of your subsequent requests.

Features

- CRUD operations for Genres, Customers, Movies, and Rentals.
- JWT Authentication for secure API access.
- Admin Panel for user and access management.
- Access control via middleware to check for valid tokens.

API Endpoints

Genres

1. **GET /api/genres/**
   - Description: Retrieve a list of all genres.
   - Example URL: `http://localhost:3000/api/genres/`
2. **GET /api/genres/:id**
   - Description: Retrieve details of a genre by its ID.
   - Path Variables: `id` (String) - Genre ID.
   - Example URL: `http://localhost:3000/api/genres/6718d5f442a74b5423296620`
3. **POST /api/genres/**
   - Description: Add a new genre.
   - Headers: `x-auth-token` - JWT authentication token.
   - Body:

   ```
   {
     "name": "action"
   }
   ```
4. **DELETE /api/genres/**
   - Description: Delete a genre by its ID.
   - Headers: `x-auth-token` - JWT authentication token.
   - Path Variables: `id` (String) - Genre ID.

Customers

1. **GET /api/customers/**

   - Descriptoin: Retrieve a list of all customers.
   - Example URL: `http://localhost:3000/api/customers/`

2. **GET /api/customers/**

   - Description: Retrieve details of a customer by their ID.
   - Path Variables: `id` (String) - Customer ID.

3. **POST /api/customers/**

   - Description: Add a new customer.
   - Headers: `x-auth-token` - JWT authentication token.
   - Body:

```
{
  "name": "name",
  "phone": "12345678900"
}
```

4. **PUT /api/customers/**

   - Description: Update a customer by thier ID.
   - Headers: `x-auth-token` - JWT authentication token.
   - Path Variables: `id` (String) - Customer ID.
   - Body:

```
{
  "name": "new name",
  "phone": "new phone"
}
```

5. **DELETE /api/customers/**

   - Description: Delete a customer by thie ID.
   - Headers: `x-auth-token` - JWT authentication token.
   - Path Variables: `id` (String) - customer ID.

Movies

1. **GET /api/movies/**

   - Description: Retrieve a list of all movies.

2. **GET /api/movies/**

- Description: Retrieve details of a movie by its ID.
- Path Variables: `id` (String) - Movie ID.

3. **POST /api/movies/**

   - Description: Add a new movie.
   - Headers: `x-auth-token` - JWT authentication tiken.
   - Body:

```json
{
  "title": "title",
  "genreId": "671fc4488a87f71534bd30a1",
  "numberInStock": 2,
  "dailyRentalRate": 2
}
```

4. **PUT /api/movies/**

   - Description: Update a movie by its ID.
   - Headers: `x-auth-token` - JWT authentication token.
   - Path Variables: `id` (String) - Movie ID.
   - Body:

```json
{
  "title": "new title",
  "genreId": "671fc4488a87f71534bd30a1",
  "numberInStock": 1,
  "dailyRentalRate": 1
}
```

5. **DELETE /api/movies/**

   - Description: Delete a movie by its ID.
   - Headers: `x-auth-token` - JWT authentication token.
   - Path Variables: `id` (Stirng) - Movie ID.

Rentals

1. **GET /api/rentals/**

   - Description: Retrieve a list of all rentals.

2. **POST /api/rentals/**

   - Description: Create a new rental record.
   - Body:

```
{
  "customerId": "671fc8ecb1382d0ad4e66357",
  "movieId": "671fca90f5c0f20e08125bbc"
}
```

Authentication & Users

1. **POST /api/users/**
   - Description: Register a new user.
   - Body:

```
{
  "name": "name",
  "email": "example@gmail.com",
  "password": "your_password"
}
```

2. **POST /api/auth/**
   - Description: Login and receive a JWT authentication token.
   - Body:

```
{
  "email": "example@gmail.com",
  "password": "your_password"
}
```

3. **GET /api/users/me/**
   - Description: Retrieve the current user's details.
   - Headers: `x-auth-token` - JWT authentication token.