

Movie Trailer Rating API

Description

You are tasked with building an API to power a website that enables users to discover and watch movie trailers. The website aims to offer a user-friendly interface for users to browse various movie trailers and view movie details. To access to see the movie trailers, users must register and log in.

Objective

The goal is to develop a secure and efficient API using TypeScript, Express.js, MongoDB for data storage, and JWT for user authentication. This API should support user registration, login, viewing of movie trailers. Additionally, implement role-based access control for administrators to manage movies and user accounts.

Features to Implement

User Registration and Login

- Allow users to register with their name, email, and password.
- Enable users to log in with their registered credentials.

JWT Authentication

- Implement user authentication using JWT (JSON Web Tokens).
- Restrict access to certain API endpoints to users with a valid JWT.

View Movie Trailers

- Provide users with a list of movie trailers available on the platform.
- Include details like title, release date, genres, and a trailer link for each movie.

Role-Based Access Control

- Define two roles within the API: "User" and "Admin".
- Restrict CRUD operations on movies and user management to "Admin" users.

Movie Management (Admin Only)

- Enable admin users to add new movies to the platform.
- Allow admin users to update existing movie details.
- Permit admin users to delete movies from the platform.

Static Folder for Poster Images

- Store movie posters in a static folder on the server.
- Include the movie poster image name in movie details.

Technologies to Use

- **TypeScript:** Enhances JavaScript with static typing and other features to improve code quality and maintainability.
- **Express.js:** A minimalistic web framework for building APIs with Node.js.
- **MongoDB:** A NoSQL database for storing data about movies and users.
- **Mongoose:** A MongoDB object modeling tool for Node.js, facilitating interaction with the database.
- **JWT:** Utilizes JSON Web Tokens for user authentication and secure communication between client and server.
- **Bcrypt:** A library for hashing and salting passwords to securely store them.
- **CORS:** Middleware to enable cross-origin resource sharing, allowing the frontend to access the API from a different domain.

API Endpoints

- **POST /auth/register** - User registration.
- **POST /auth/login** - User login to obtain a JWT.
- **GET /auth/users** - (Admin only) Fetch a list of users.
- **PUT /auth/users/:id** - (Admin only) Update user details.
- **DELETE /auth/user/:id** - (Admin only) Delete a user.
- **GET /api/movies** - (Only logged users) Fetch a list of movie trailers with details.
- **GET /api/movies/search** - (Only logged users) Fetch a list of movie with pagination and sort and filters (sortBy realiseDate, search by realise year, name or genre).
- **POST /api/movies** - (Admin only) Add a new movie.
- **PUT /api/movies/:id** - (Admin only) Update movie details.
- **DELETE /api/movies/:id** - (Admin only) Delete a movie.

Project Structure

Adopt a structured MVC (Model-View-Controller) pattern with separate folders for controllers, models, services, middleware, and routes. Implement authentication middleware to protect endpoints requiring authentication.

Models

- **User Model:** Represents a registered website user with fields for username, email, password, and role.
- **Movie Model:** Describes a movie with fields for title, release date, trailer link, poster URL, and genres.

Movie objects examples

```
[
  {
    "title": "Harry Potter and the Sorcerer's Stone",
    "releaseDate": "2001-11-16",
    "trailerLink": "https://www.youtube.com/watch?v=eKSB0gXl9dw",
    "poster": "harry_potter_poster.jpg",
    "genres": ["Adventure", "Family", "Fantasy"]
  },
  {
    "title": "The Lord of the Rings: The Fellowship of the Ring",
    "releaseDate": "2001-12-19",
```

```
    "trailerLink": "https://www.youtube.com/watch?v=V75dMMIW2B4",
    "poster": "lord_of_the_rings_poster.jpg",
    "genres": ["Adventure", "Drama", "Fantasy"]
  },
  {
    "title": "The Matrix",
    "releaseDate": "1999-03-31",
    "trailerLink": "https://www.youtube.com/watch?v=vKQi3bBA1y8",
    "poster": "matrix_poster.jpg",
    "genres": ["Action", "Sci-Fi"]
  }
]
```

Bonus

- All endpoints should be documented using Swagger
- Users can leave a comment or rate the movie trailer (e.g. 1 - 5 stars)