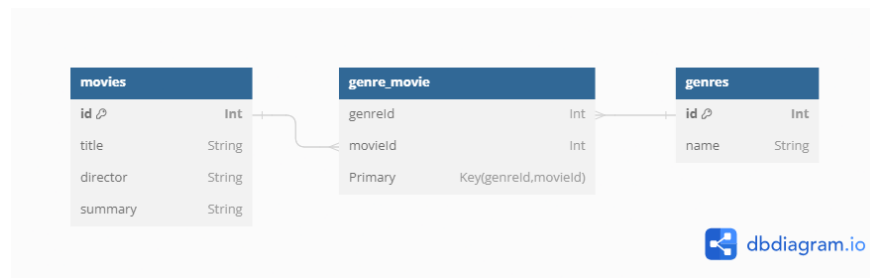


Movies Collection API

Summary

Ini adalah API sederhana untuk manage *movies collection* yang menerapkan prinsip CRUD (Create, Read, Update, dan Delete).

Desain ERD database dari API ini dapat dilihat berikut:



di mana ada 3 table. Tabel utamanya adalah tabel movies yang terdiri dari id, title, director, dan summary. Hubungan movies dan genres adalah many to many, maka diterapkan prinsip Atomic dengan cara menormalisasi genres menjadi tabel sendiri dengan pivot tabelnya genre_movie.

Api ini didesain menggunakan Node.js, Express.js dan Prisma ORM.

Endpoints

1. Endpoint: POST `/api/movies`

- **Deskripsi:** Menambahkan film baru ke database
- **Request Body:**
 - `title` : Judul film.
 - `director` : Nama sutradara film.
 - `summary` : Ringkasan film.
 - `genre` : Array ID genre yang terkait dengan film.
- **Response:** Jika berhasil, mengembalikan status 201 dan objek film yang baru dibuat, termasuk array ID genre. Jika gagal, mengembalikan status

400 dan pesan kesalahan.

2. **Endpoint:** GET `/api/movies`

- **Deskripsi:** Mengambil semua film dan menyertakan nama genre.
- **Response:** Jika berhasil, mengembalikan status 200 dan array film dengan array ID genre. Jika gagal, mengembalikan status 400 dan pesan kesalahan.

3. **Endpoint:** GET `/api/movies/:id`

- **Deskripsi:** Mengambil film berdasarkan ID.
- **Parameter:** `id` - ID film yang ingin diambil.
- **Response:** Jika berhasil, mengembalikan status 200 dan film dengan array ID genre. Jika film tidak ditemukan, mengembalikan status 404. Jika gagal, mengembalikan status 400 dan pesan kesalahan.

4. **Endpoint:** GET `/api/movies/search/:title`

- **Deskripsi:** Mencari film berdasarkan judul.
- **Parameter:** `title` - Judul film yang ingin dicari.
- **Response:** Jika berhasil, mengembalikan status 200 dan array film dengan judul yang mengandung string pencarian, serta array ID genre. Jika tidak ada film yang ditemukan, mengembalikan status 404. Jika gagal, mengembalikan status 400 dan pesan kesalahan.

5. **Endpoint:** PUT `/api/movies/:id`

- **Deskripsi:** Memperbarui film berdasarkan ID.
- **Parameter:** `id` - ID film yang ingin diperbarui.
- **Request Body:**
 - `title` : Judul film baru.
 - `director` : Nama sutradara film baru.
 - `summary` : Ringkasan film baru.
 - `genre` : Array ID genre baru yang terkait dengan film.

- **Response:** Jika berhasil, mengembalikan status 200 dan film yang diperbarui dengan array ID genre. Jika gagal, mengembalikan status 400 dan pesan kesalahan.

6. **Endpoint:** DELETE `/api/movies/:id`

- **Deskripsi:** Menghapus film berdasarkan ID.
- **Parameter:** `id` - ID film yang ingin dihapus.
- **Response:** Jika berhasil, mengembalikan status 200 dan film yang dihapus. Jika gagal, mengembalikan status 400 dan pesan kesalahan.