TEKNIK INFORMATIKA
IF - 5K

Manual Book Screenscore API

IKHSAN FAUZAN // D112121054
NURHIDAYAT // D112121057
RIYAN // D11212062

KATA PENGANTAR

Dalam era digital yang semakin berkembang, penilaian dan ulasan pengguna memiliki peran yang sangat penting dalam mempengaruhi keputusan orang-orang dalam memilih film untuk ditonton. Screenscore API hadir sebagai solusi untuk memberikan pengalaman yang lebih baik dalam hal ini.

Screenscore API adalah sebuah API CRUD (Create, Read, Update, Delete) yang dirancang khusus untuk memudahkan pengelolaan data rating dan ulasan film. Dengan Screenscore API, Anda dapat dengan mudah membuat, membaca, memperbarui, dan menghapus data rating dan ulasan film

Ebook ini akan membimbing Anda melalui langkah-langkah penggunaan Screenscore API, mulai dari instalasi dan konfigurasi hingga implementasi pada aplikasi Anda. Dengan menggunakan Screenscore API, Anda dapat mengembangkan aplikasi rating dan review film yang responsif dan efisien, meningkatkan pengalaman pengguna, dan mengoptimalkan pengelolaan data.

Cimahi, 29 Januari 2024

Penulis

DAFTAR ISI

BAB I
PEMBUKAAN

1. Pendahuluan

1.1 Latar Belakang

Dalam era digital yang semakin berkembang, industri hiburan, khususnya film, mengalami transformasi signifikan dalam cara film diproduksi, dikonsumsi, dan dinilai oleh penonton. Di tengah lautan konten yang terus berkembang, penilaian dan ulasan pengguna telah menjadi faktor kunci dalam membentuk preferensi dan keputusan penonton saat memilih film untuk ditonton.

Screenscore API muncul sebagai solusi inovatif untuk memenuhi kebutuhan akan manajemen data rating dan ulasan film yang efektif. Dalam dunia di mana preferensi penonton bisa sangat bervariasi, Screenscore API menawarkan pendekatan yang terintegrasi dan responsif untuk memastikan bahwa pengelolaan data rating dan ulasan film dapat dilakukan dengan efisien.

Screenscore API adalah sebuah antarmuka pemrograman aplikasi (API) CRUD yang dirancang khusus untuk menyederhanakan proses penciptaan (Create), pembacaan (Read), pembaruan (Update), dan penghapusan (Delete) data rating dan ulasan film. Dengan menggunakan teknologi ini, pengembang dan pengelola platform hiburan dapat dengan mudah mengintegrasikan fungsi rating dan review film

1.2 Tujuan Pembuatan Dokumen atau Ebook

Dokumen user manual Screenscore API ini dibuat untuk tujuan sebagai berikut :

a. Menggambarkan dan menjelaskan penggunaan Screenscore API
b. Sebagai panduan penggunaan aplikasi backend

1.3 Deskripsi Umum Sistem

"ScreenScore" adalah sebuah aplikasi backend yang menyediakan layanan untuk menangani informasi terkait film, ulasan, genre, pemain, dan karakter. Aplikasi ini memungkinkan pengguna untuk melihat daftar film, menambahkan ulasan, melihat ulasan film, dan melakukan berbagai operasi CRUD terkait data film, genre, pemain, dan karakter.

1.4 Deskripsi Umum Dokumen atau Ebook

Dokumen ini dibuat untuk memberikan panduan penggunaan aplikasi Screenscore. Dokumen ini berisikan informasi sebagai berikut :

a. BAB I

Berisi informasi umum yang merupakan bagian pendahuluan yang meliputi tujuan pembuatan dokumen, deskripsi umum system serta dokumen

b. BAB II

Berisi informasi umum yang membahas secara mendetail fitur-fitur yang dimiliki oleh Screenscore API. Mulai dari fitur utama seperti membuat, membaca, memperbarui, dan menghapus data rating dan ulasan film, hingga fitur tambahan yang dapat meningkatkan fungsionalitas AP

c. BAB III

Berisi informasi mengenai rancangan fitur-fitur yang akan diimplementasikan dalam Screenscore API. Ini mencakup spesifikasi teknis setiap fitur, alur kerja yang diperlukan untuk mengimplementasikannya

d. BAB IV

Berisi informasi yang membahas rancangan basis data (Entity-Relationship Diagram/ERD) untuk Screenscore API. Bab ini berisi bagaimana entitas seperti film, pengguna, rating, dan ulasan saling terhubung dalam basis data, serta bagaimana struktur ini mendukung operasi CRUD dalam API. Dengan pemahaman yang baik tentang ERD, Anda akan dapat merancang basis data yang efisien dan mudah diakses oleh Screenscore API.

e. BAB V

Berisi rangkuman dari seluruh isi buku panduan atau ebook ini

BAB II
FITUR

2. Fitur Utama

2.1 Manajemen Film

Pengguna dapat melihat daftar film, menambahkan film baru, memperbarui detail film, dan menghapus film yang ada.

2.2 Manajemen Ulasan

Pengguna dapat menambahkan ulasan untuk film tertentu, melihat ulasan yang telah dibuat, dan menghapus ulasan yang sudah tidak diperlukan.

2.3 Manajemen Genre

Pengguna dapat melihat daftar genre, menambahkan genre baru, memperbarui detail genre, dan menghapus genre yang sudah tidak diperlukan.

2.4 Manajemen Pemain

Pengguna dapat melihat daftar pemain, menambahkan pemain baru, memperbarui detail pemain, dan menghapus pemain yang sudah tidak diperlukan.

2.5 Manajemen Karakter

Pengguna dapat melihat daftar karakter dalam film, menambahkan karakter baru, memperbarui detail karakter, dan menghapus karakter yang sudah tidak diperlukan.

2.6 Manajemen Users

Pengguna dapat melihat daftar user, menambahkan user baru, memperbarui detail user, dan menghapus user yang sudah tidak diperlukan

BAB III
RANCANGAN END POINT

3. Rancangan End Point

3.1 End Point API

3.1.1 Films

- GET /api/films: Mendapatkan daftar semua film.
- POST /api/films: Menambahkan film baru.
- GET /api/films/{id}: Mendapatkan detail film berdasarkan ID.
- PUT /api/films/{id}: Memperbarui detail film berdasarkan ID.
- DELETE /api/films/{id}: Menghapus film berdasarkan ID.

3.1.2 Reviews

- GET /api/reviews: Mendapatkan daftar semua ulasan.
- POST /api/reviews/films/{idFilms}: Menambahkan ulasan baru untuk film tertentu.
- GET /api/reviews/{id}: Mendapatkan detail ulasan berdasarkan ID.
- PUT /api/reviews/films/{idFilms}: Memperbarui ulasan untuk film tertentu.
- DELETE /api/reviews/{id}: Menghapus ulasan berdasarkan ID.

3.1.3 Genres

- GET /api/genres: Mendapatkan daftar semua genre.
- POST /api/genres: Menambahkan genre baru.
- GET /api/genres/{id}: Mendapatkan detail genre berdasarkan ID.
- PUT /api/genres/{id}: Memperbarui detail genre berdasarkan ID.
- DELETE /api/genres/{id}: Menghapus genre berdasarkan ID.

3.1.4 Casts

- GET /api/casts: Mendapatkan daftar semua pemain.
- POST /api/casts: Menambahkan pemain baru.
- GET /api/casts/{id}: Mendapatkan detail pemain berdasarkan ID.
- PUT /api/casts/{id}: Memperbarui detail pemain berdasarkan ID.

- DELETE /api/casts/{id}: Menghapus pemain berdasarkan ID.

3.1.5 Characters

- GET /api/characters: Mendapatkan daftar semua karakter dalam film.
- POST /api/characters: Menambahkan karakter baru.
- GET /api/characters/{id}: Mendapatkan detail karakter berdasarkan ID.
- PUT /api/characters/{id}: Memperbarui detail karakter berdasarkan ID.
- DELETE /api/characters/{id}: Menghapus karakter berdasarkan ID.

3.1.6 Users

- GET /api/characters: Mendapatkan daftar semua karakter dalam film.
- POST /api/characters: Menambahkan karakter baru.
- GET /api/characters/{id}: Mendapatkan detail karakter berdasarkan ID.
- PUT /api/characters/{id}: Memperbarui detail karakter berdasarkan ID.
- DELETE /api/characters/{id}: Menghapus karakter berdasarkan ID.

3.2 Cara Penggunaan

3.2.1 Instalasi dan Set Up

3.2.1.1 Unduh atau clone repository

Source code bisa diunduh atau di clone melalui *https://github.com/riyanada/ScreenScore-api.git*

```
Ikhsanfzn@Ikhsanfzn MINGW64 /c/laragon/UASWebService
$ git clone https://github.com/riyanada/ScreenScore-api.git
Cloning into 'ScreenScore-api'...
remote: Enumerating objects: 130, done.
remote: Counting objects: 100% (130/130), done.
remote: Compressing objects: 100% (86/86), done.
remote: Total 130 (delta 28), reused 128 (delta 26), pack-reused 0Receiving obje
Receiving objects: 100% (130/130), 3.64 MiB | 2.37 MiB/s, done.

Resolving deltas: 100% (28/28), done.
```

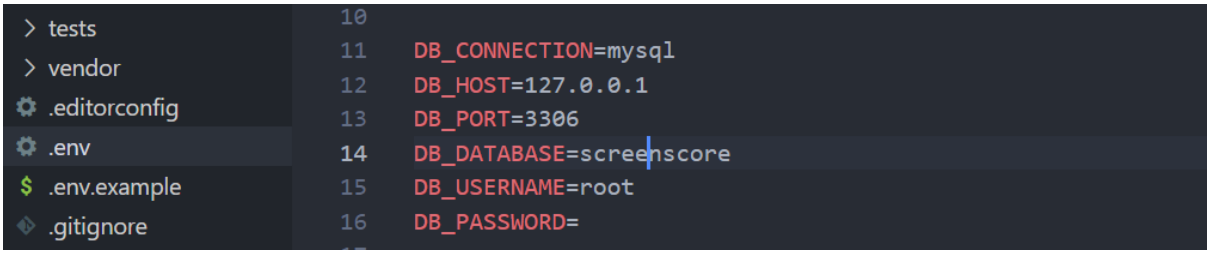
3.2.1.2 Instal composer

Instal semua dependency dengan menjalankan *composer install*

```
Ikhsanfzn@Ikhsanfzn MINGW64 /c/laragon/UASWebService/ScreenScore-api (master)
$ composer install
Installing dependencies from lock file (including require-dev)
Verifying lock file contents can be installed on current platform.
Package operations: 112 installs, 0 updates, 0 removals
  - Installing doctrine/inflector (2.0.8): Extracting archive
  - Installing symfony/polyfill-php72 (v1.28.0): Extracting archive
  - Installing symfony/polyfill-intl-normalizer (v1.28.0): Extracting archive
  - Installing symfony/polyfill-intl-idn (v1.28.0): Extracting archive
  - Installing doctrine/lexer (1.2.3): Extracting archive
  - Installing egulias/email-validator (2.1.25): Extracting archive
  - Installing symfony/deprecation-contracts (v2.5.2): Extracting archive
```

3.2.1.3 Setting environment

Salin file .env.example menjadi .env dan sesuaikan pengaturan database



3.2.1.4 Migrasi database

Jalankan migrasi database dengan menjalankan *php artisan migrate*

```
Ikhsanfzn@Ikhsanfzn MINGW64 /c/laragon/UASWebService/ScreenScore-api (master)
$ php artisan migrate
Migration table created successfully.
Migrating: 2021_10_18_161042_create_users_table
Migrated:  2021_10_18_161042_create_users_table (23.15ms)
Migrating: 2023_10_18_161358_create_profile_table
Migrated:  2023_10_18_161358_create_profile_table (48.95ms)
Migrating: 2024_02_09_161934_create_films_table
Migrated:  2024_02_09_161934_create_films_table (75.94ms)
Migrating: 2024_02_09_162543_create_reviews_table
Migrated:  2024_02_09_162543_create_reviews_table (93.37ms)
Migrating: 2024_02_09_163025_create_genres_table
Migrated:  2024_02_09_163025_create_genres_table (7.33ms)
Migrating: 2024_02_09_164047_create_film_genres_table
Migrated:  2024_02_09_164047_create_film_genres_table (86.39ms)
Migrating: 2024_02_09_171424_create_casts_table
Migrated:  2024_02_09_171424_create_casts_table (7.52ms)
Migrating: 2024_02_09_171440_create_characters_table
Migrated:  2024_02_09_171440_create_characters_table (106.95ms)
```

3.2.1.5 Seeder

Jalankan seeder dengan menjalankan *php artisan db:seed*

```
Ikhsanfzn@Ikhsanfzn MINGW64 /c/laragon/UASWebService/ScreenScore-api (master)
$ php artisan db:seed
Seeding: Database\Seeders\UsersSeeder
Seeded:  Database\Seeders\UsersSeeder (225.04ms)
Seeding: Database\Seeders\FilmsSeeder
Seeded:  Database\Seeders\FilmsSeeder (15.31ms)
Seeding: Database\Seeders\GenresSeeder
Seeded:  Database\Seeders\GenresSeeder (75.90ms)
Seeding: Database\Seeders\FilmGenresSeeder
Seeded:  Database\Seeders\FilmGenresSeeder (16.89ms)
Seeding: Database\Seeders\CastsSeeder
Seeded:  Database\Seeders\CastsSeeder (37.76ms)
Seeding: Database\Seeders\CharactersSeeder
Seeded:  Database\Seeders\CharactersSeeder (28.17ms)
Database seeding completed successfully.
```

3.2.1.6 Running aplikasi

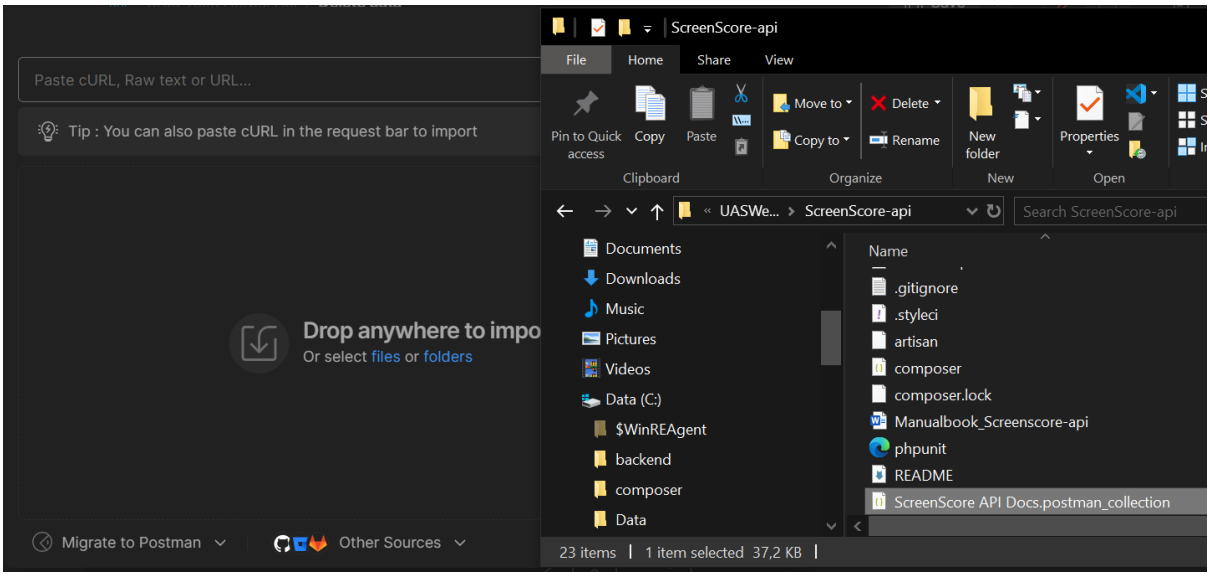
Jalankan aplikasi dengan menjalankan *php –S localhost:800 –t public* atau *php artisan serve*

```
Ikhsanfzn@Ikhsanfzn MINGW64 /c/laragon/UASWebService/ScreenScore-api (master)
$ php -S localhost:8000 -t public
```

3.2.2 Pengujian

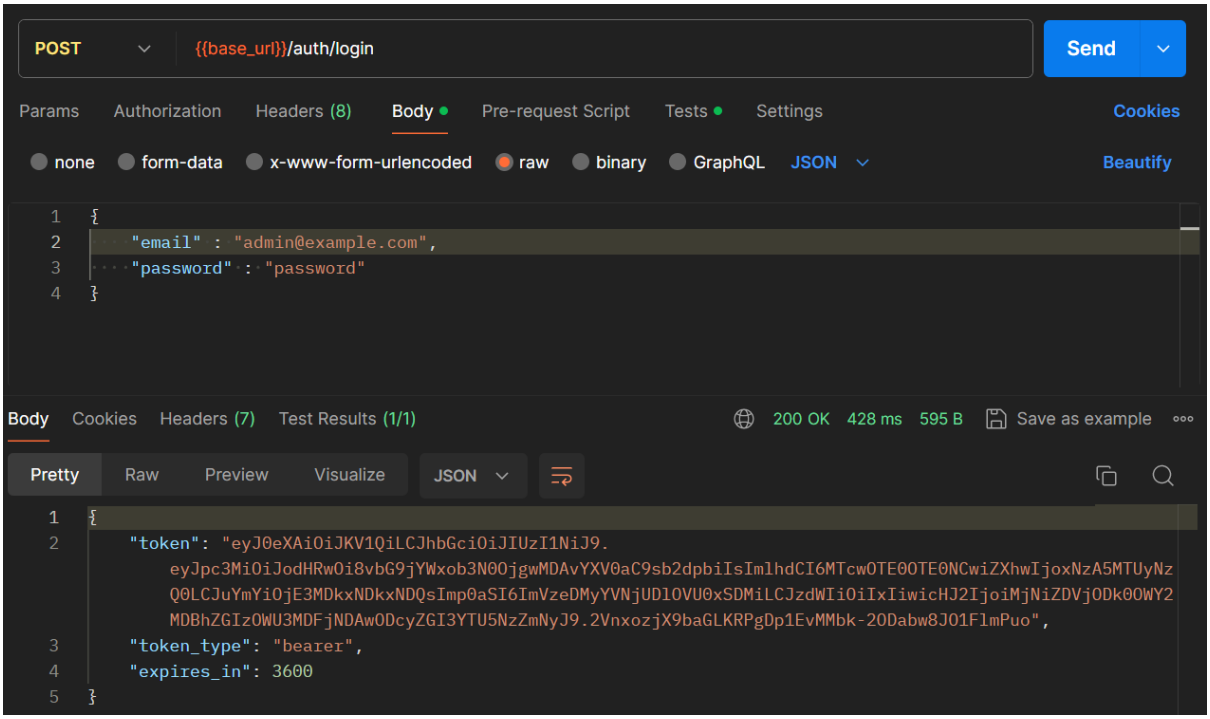
3.2.2.1 Import Collection

Buka aplikasi postman lalu import collection yang sudah tersedia di dalam folder



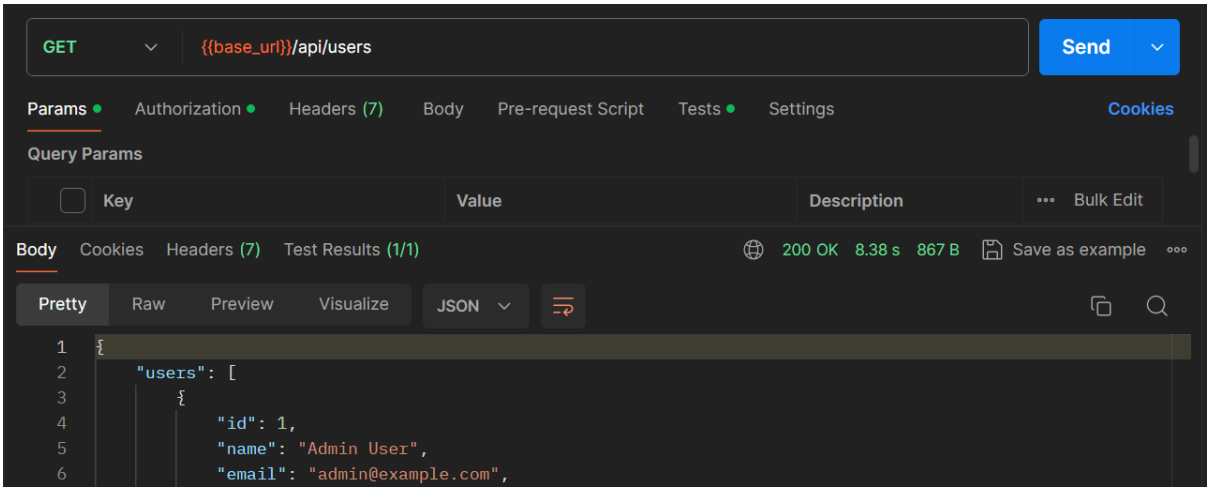
3.2.2.2 Login

Pada endpoint ini menggunakan token dengan type bearer dan expire bernilai 3600 detik atau 1 jam, token ini digunakan untuk berbagai api yang telah dibuat

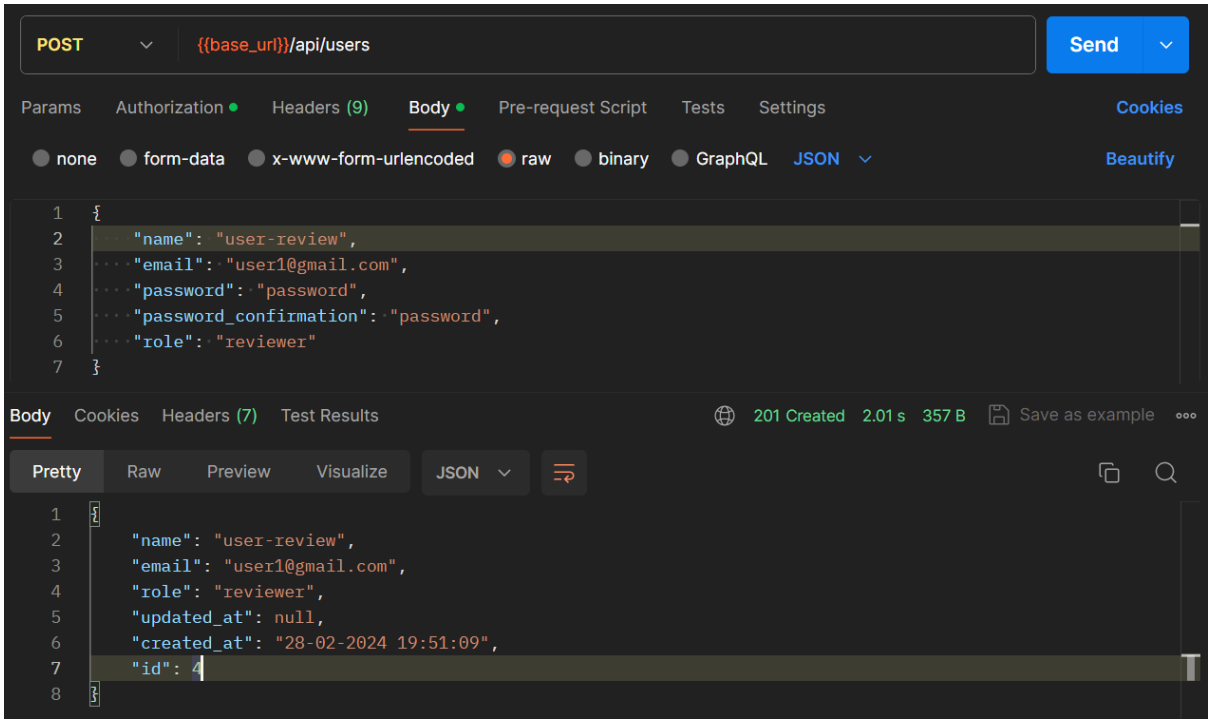


3.2.2.3 End Point User

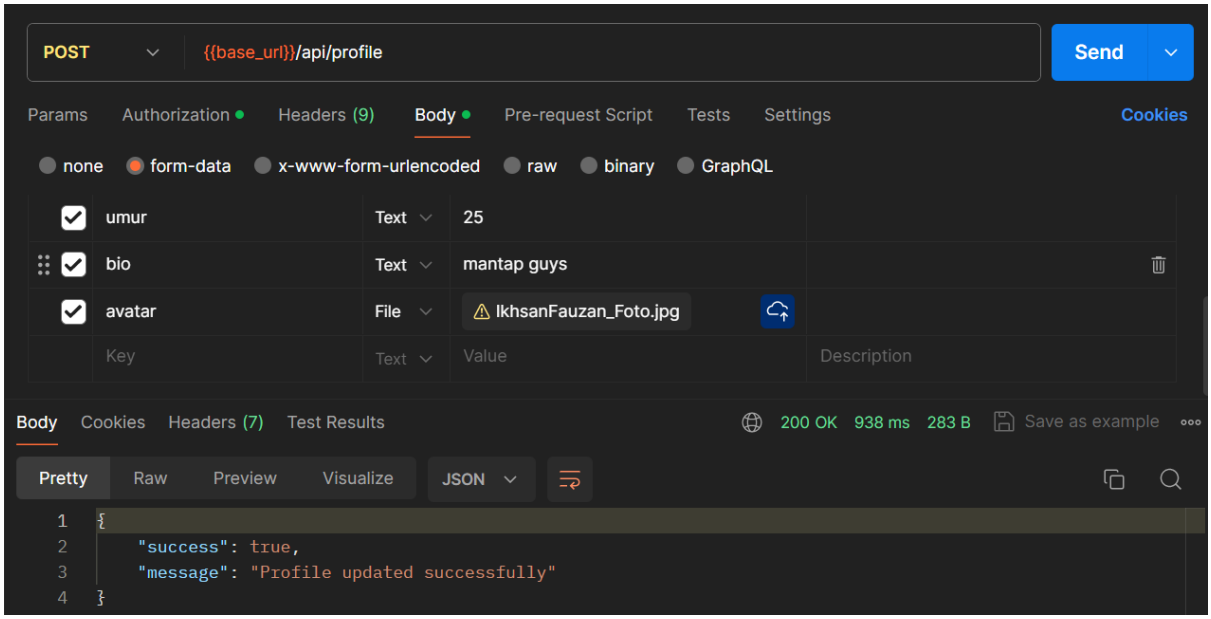
- GET /api/users: Mendapatkan daftar semua user



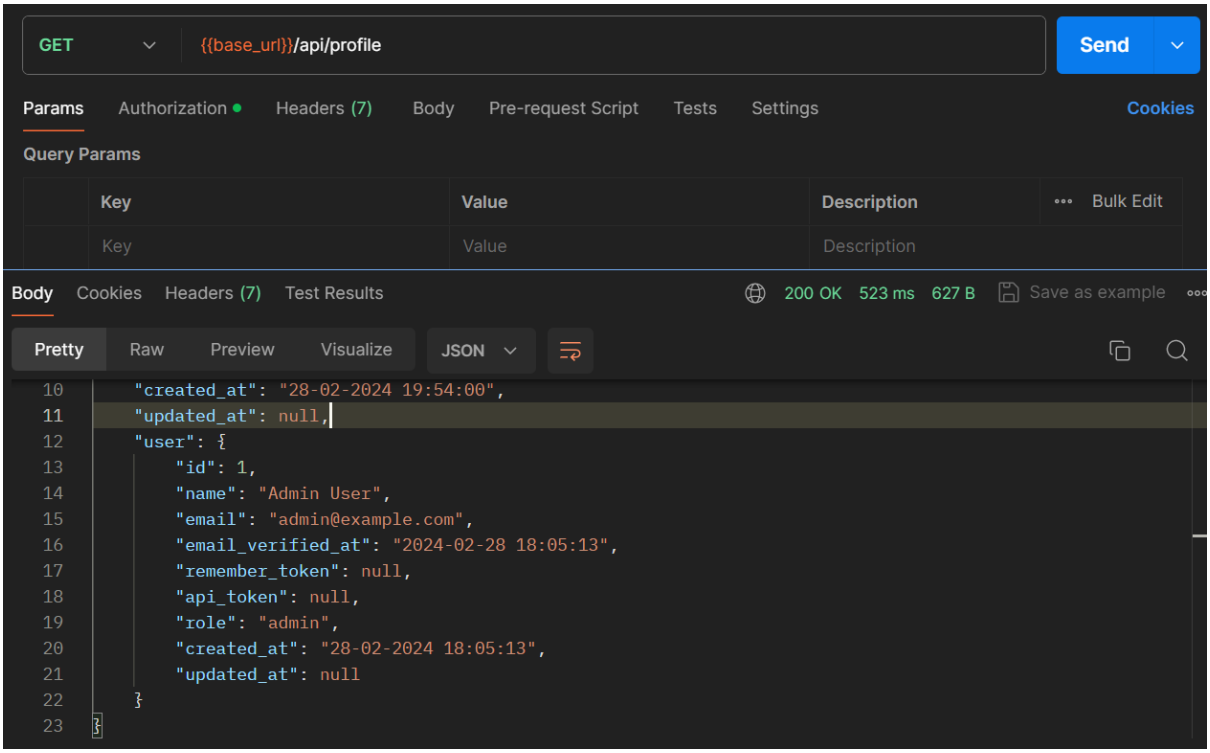
- POST /api/users: Menambahkan user baru



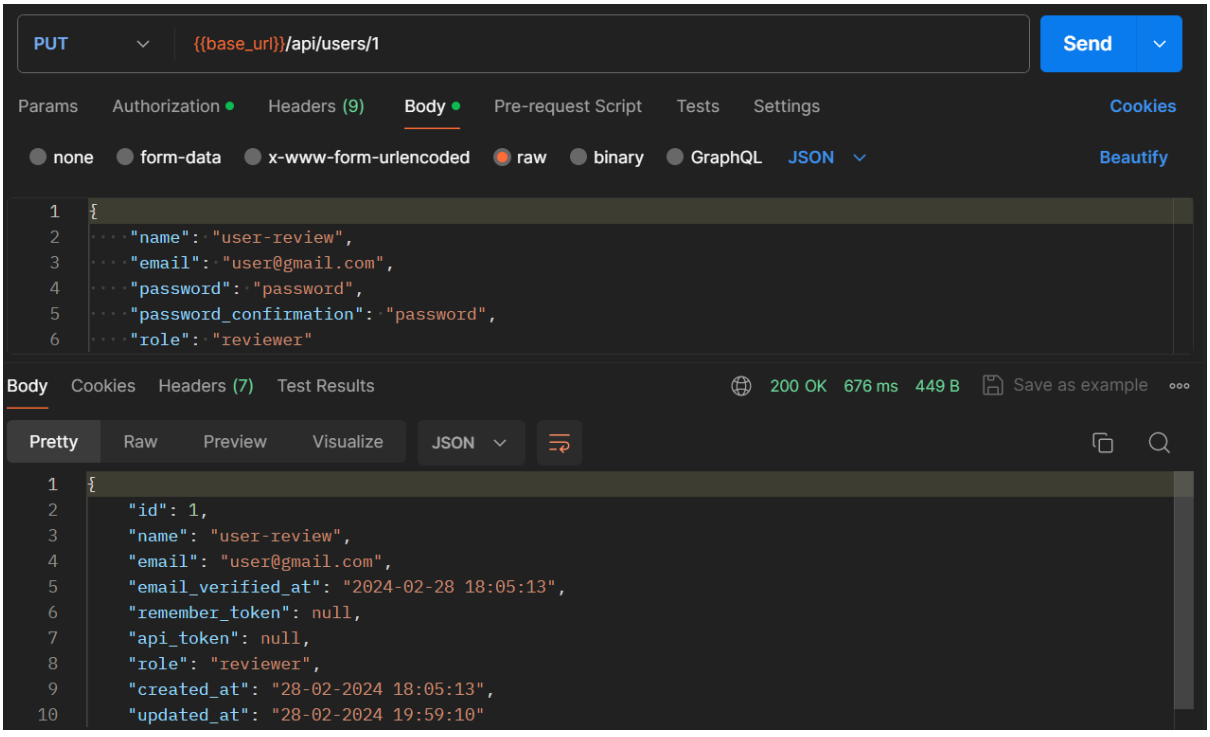
- POST /api/profile: Menambahkan user profile baru



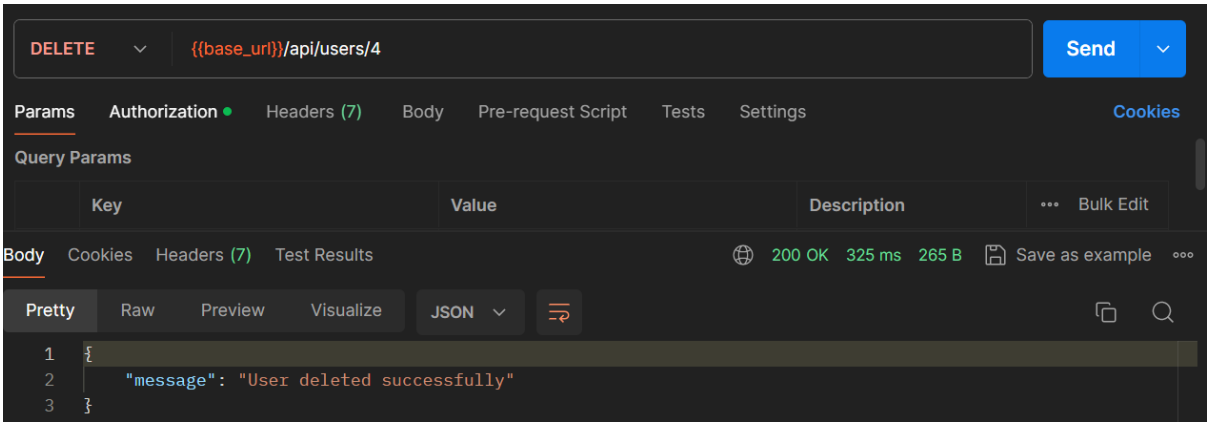
- GET /api/profile: Mendapatkan user profile



- POST /api/user/{id}: Mengupdate user berdasarkan ID

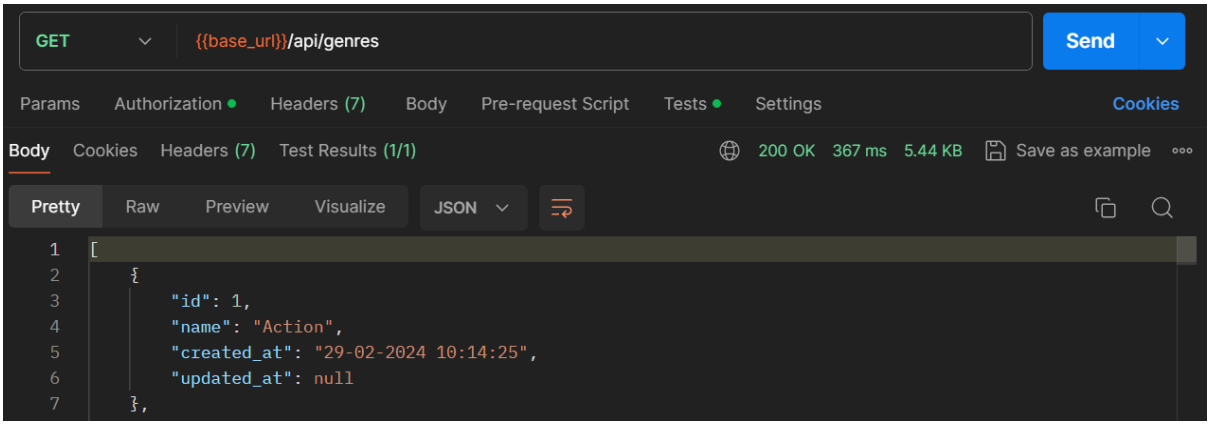


- DELETE /api/user/{id}: Menghapus user berdasarkan ID

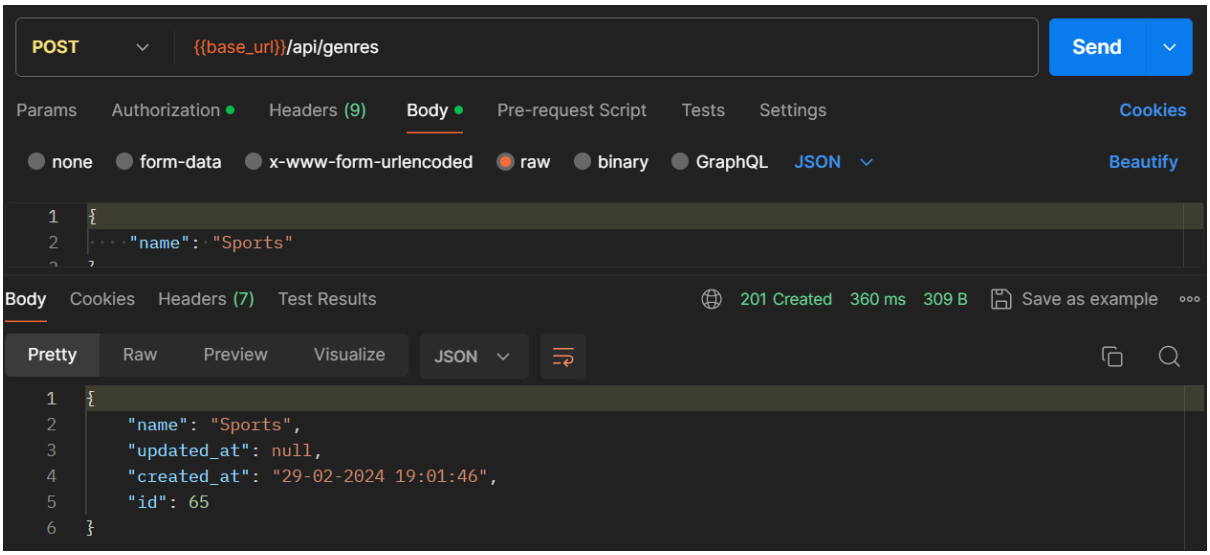


3.2.2.4 End Point Genre

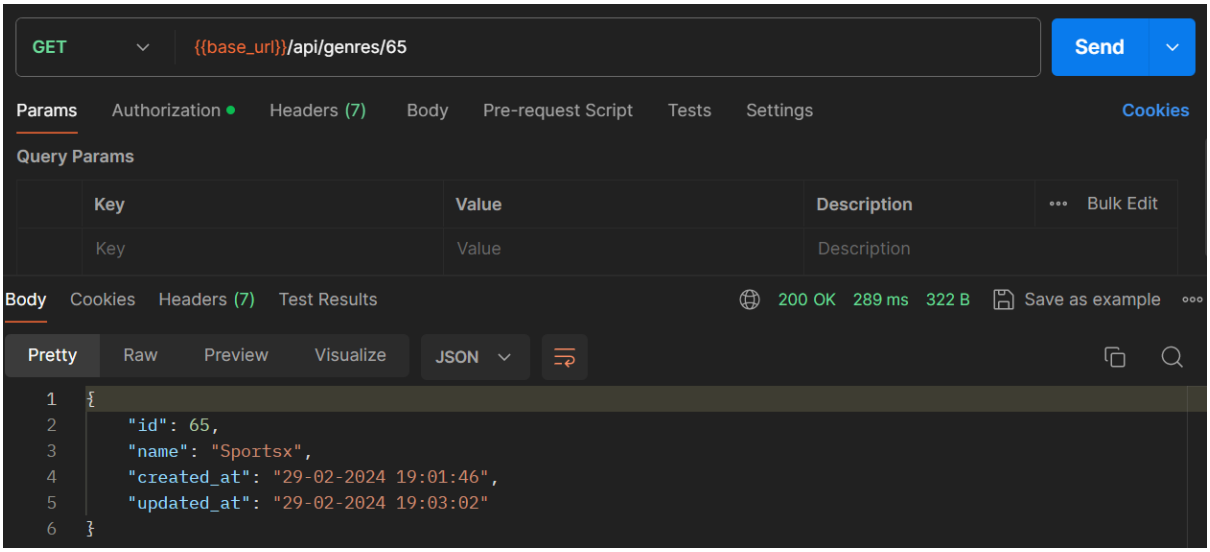
- GET /api/genres: Mendapatkan daftar semua genre.



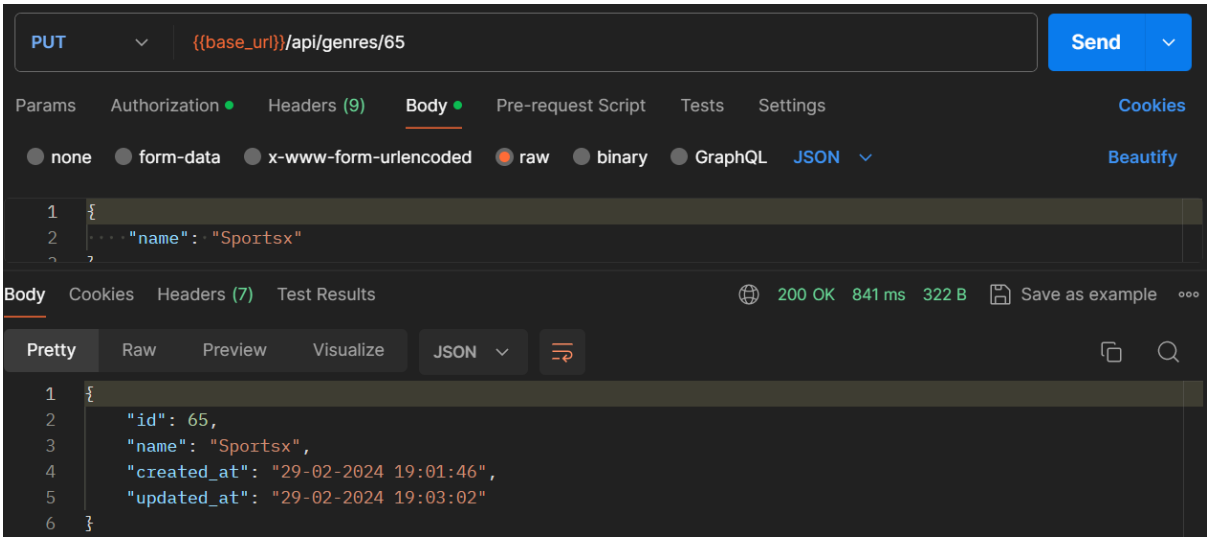
- POST /api/genres: Menambahkan genre baru



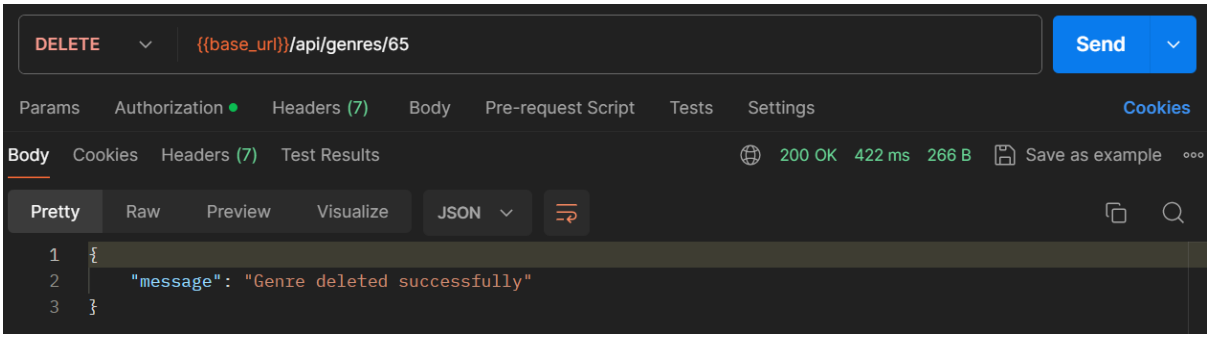
- GET /api/genres/{id}: Mendapatkan detail genre berdasarkan ID



- PUT /api/genres/{id}: Memperbarui detail genre berdasarkan ID

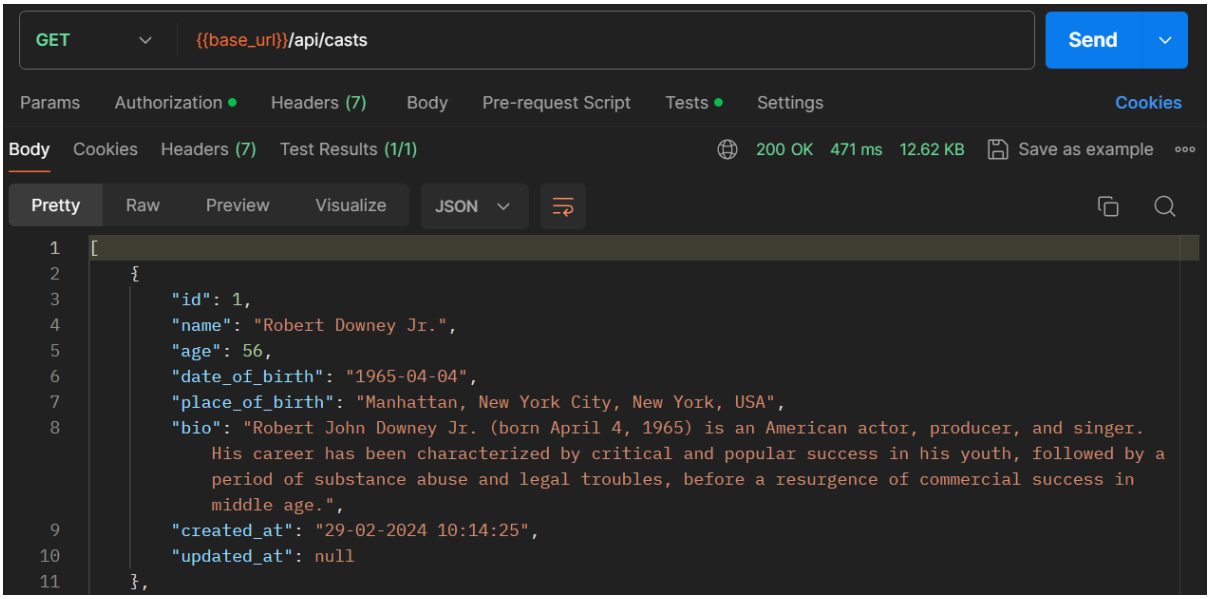


- DELETE /api/genres/{id}: Menghapus genre berdasarkan ID

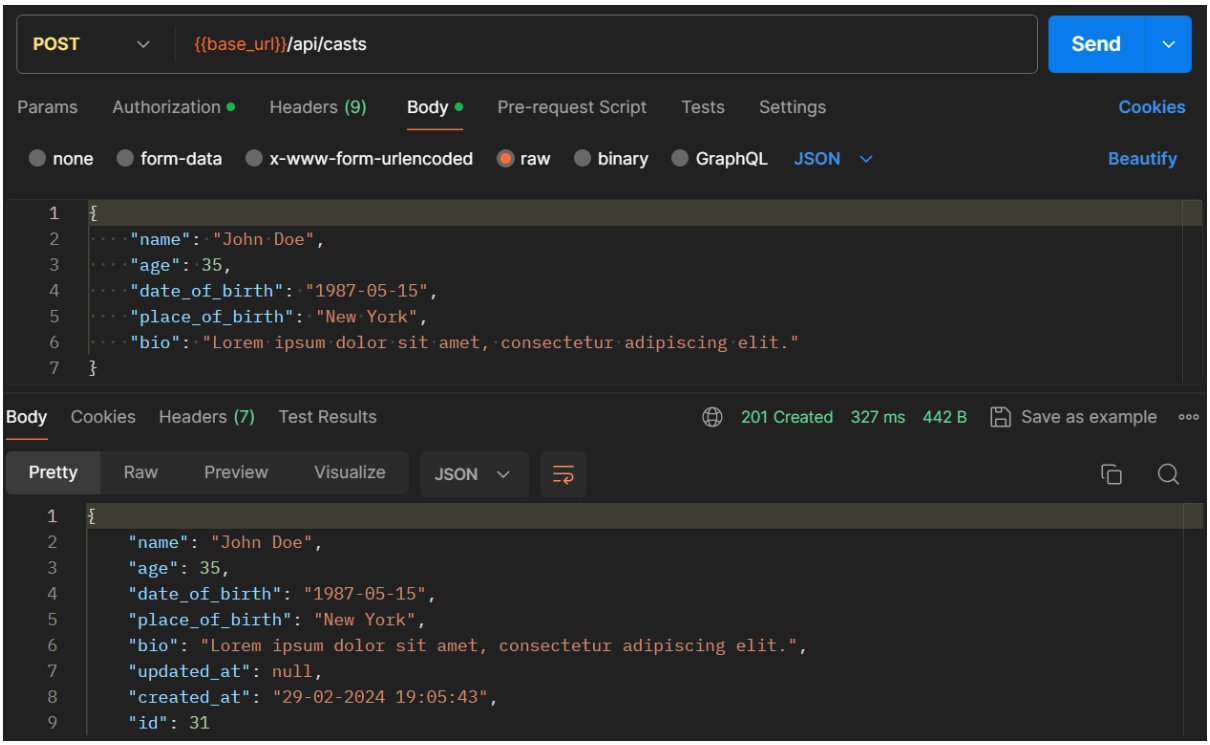


3.2.2.5 Endpoint Cast

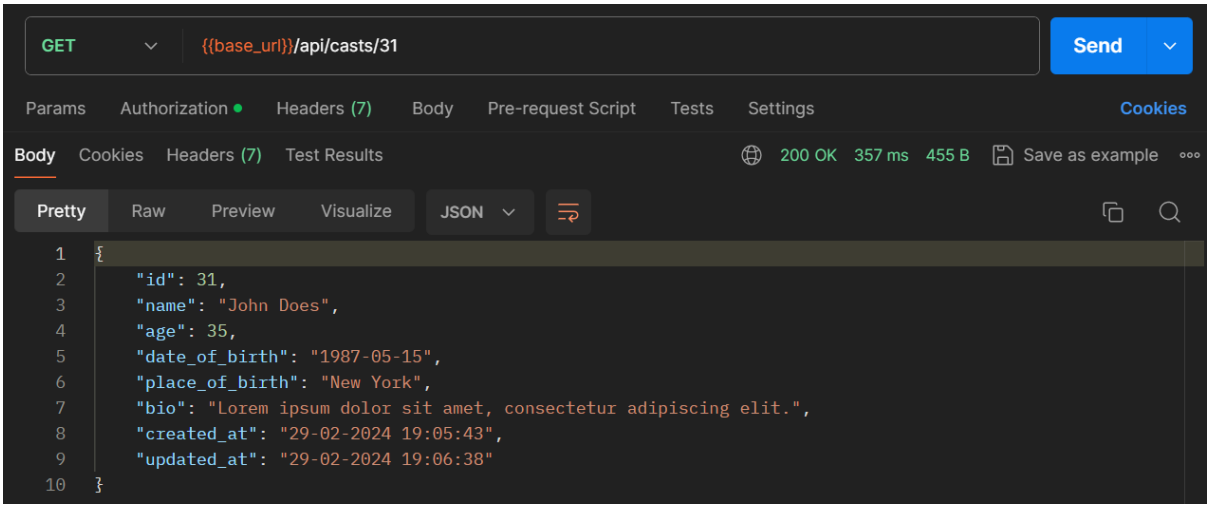
- GET /api/casts: Mendapatkan daftar semua pemain



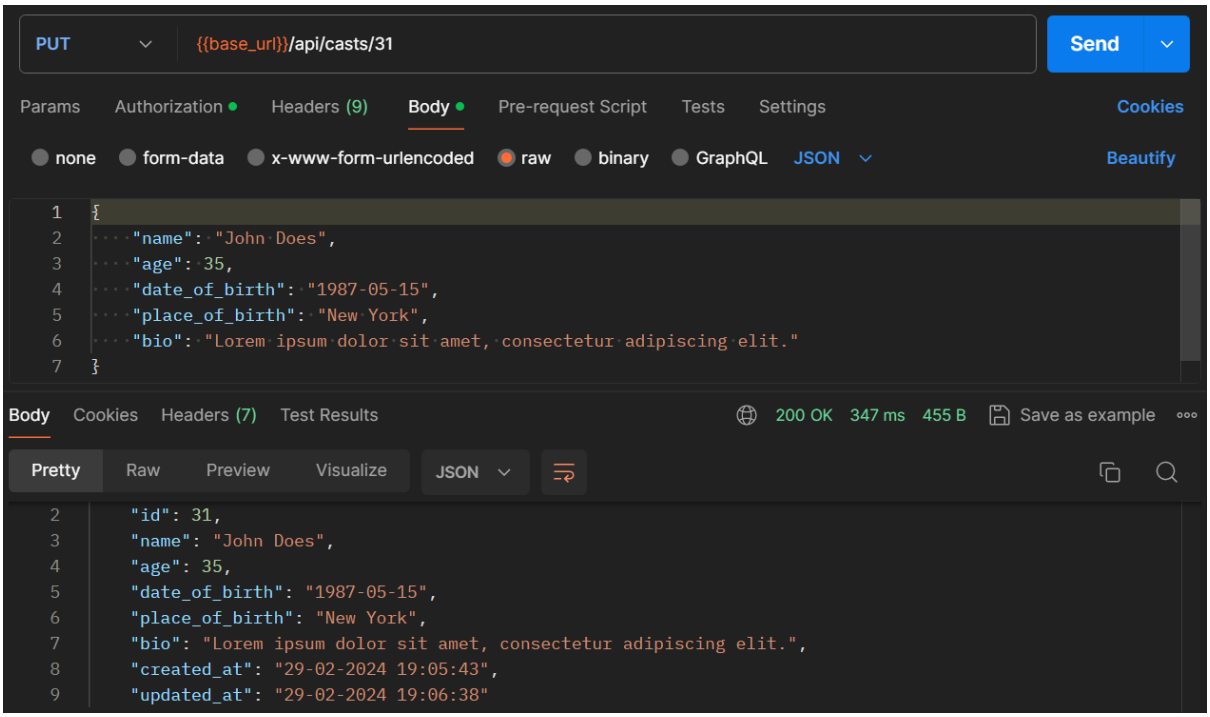
- POST /api/casts: Menambahkan pemain baru



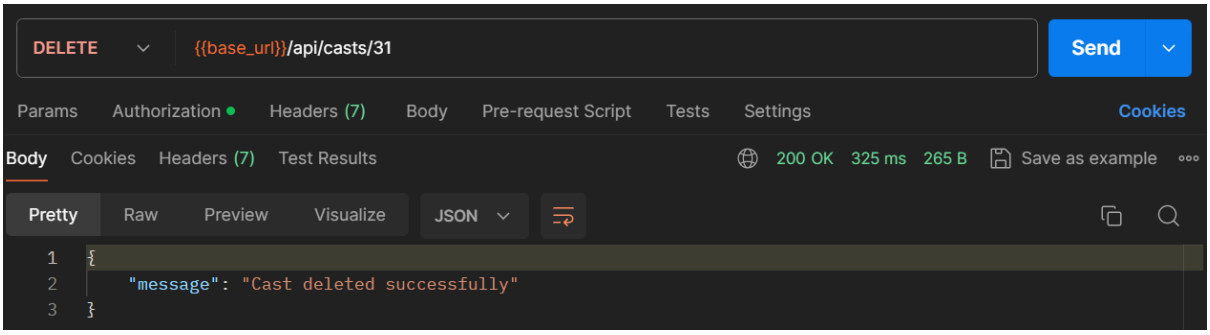
- GET /api/casts/{id}: Mendapatkan detail pemain berdasarkan ID



- PUT /api/casts/{id}: Memperbarui detail pemain berdasarkan ID



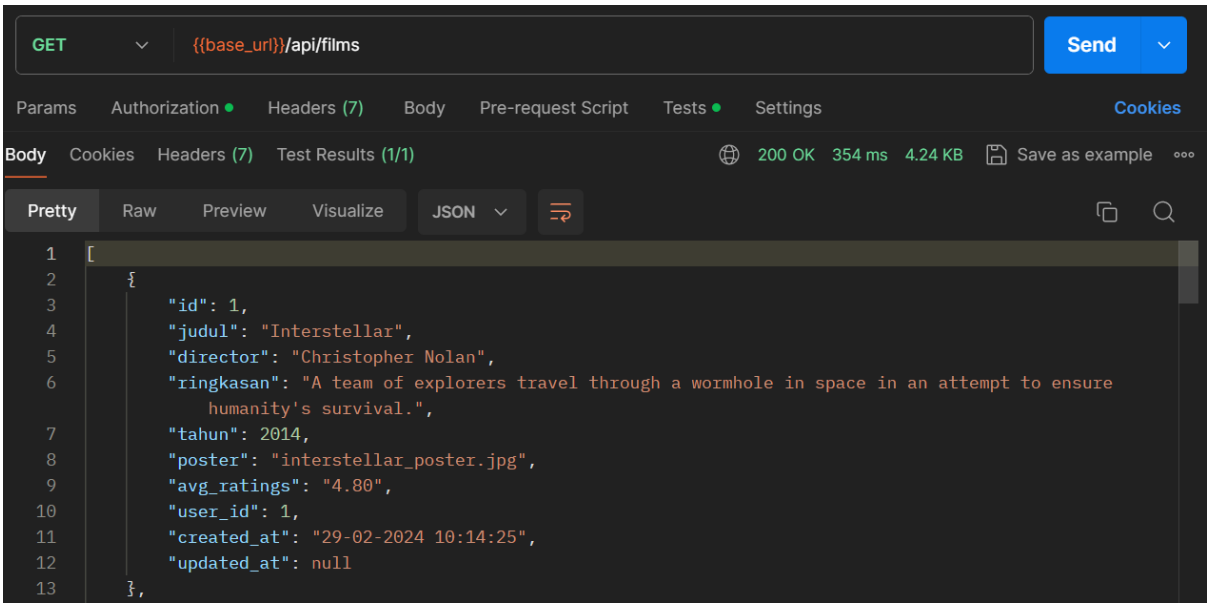
- DELETE /api/casts/{id}: Menghapus pemain berdasarkan ID



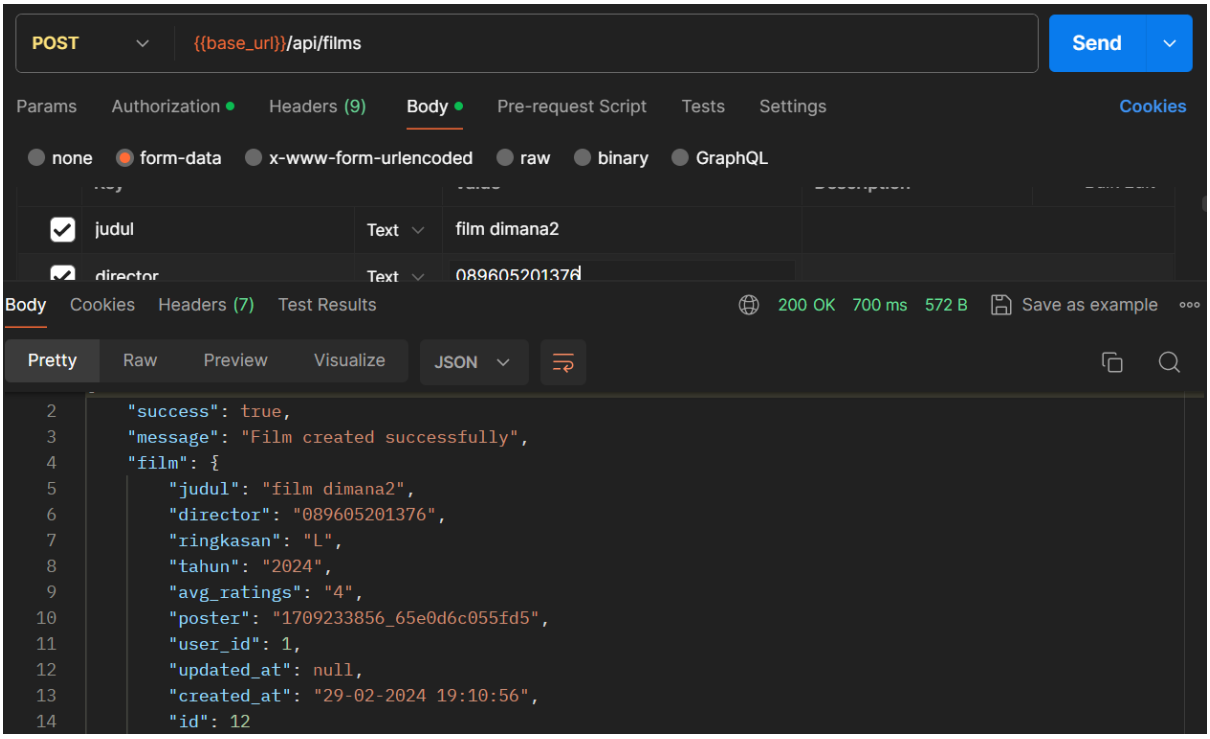
.

3.2.2.6 Endpoint Film

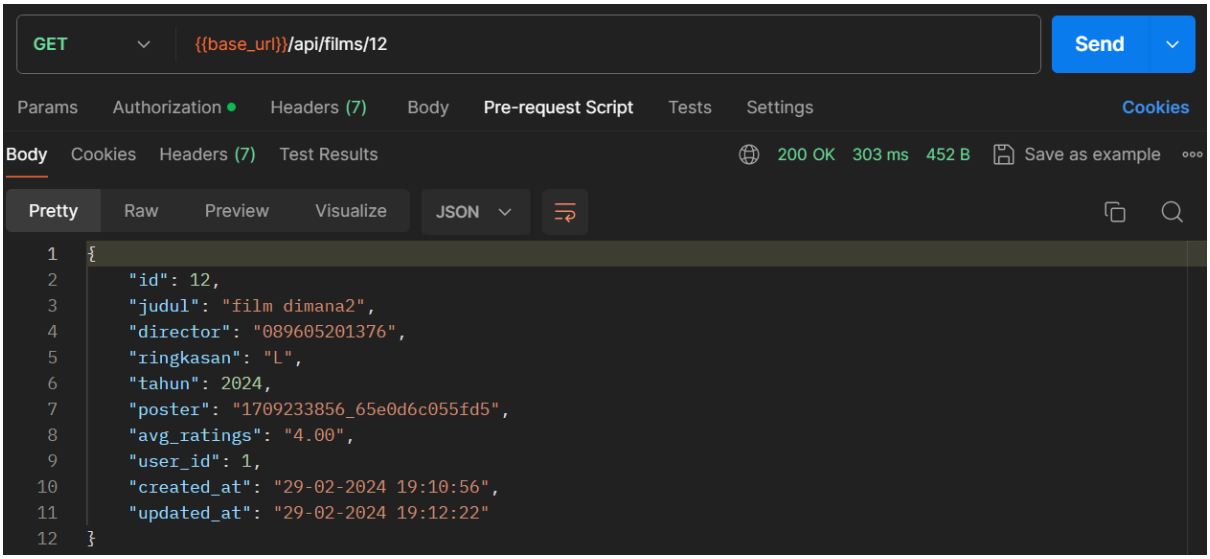
- GET /api/films: Mendapatkan daftar semua film.



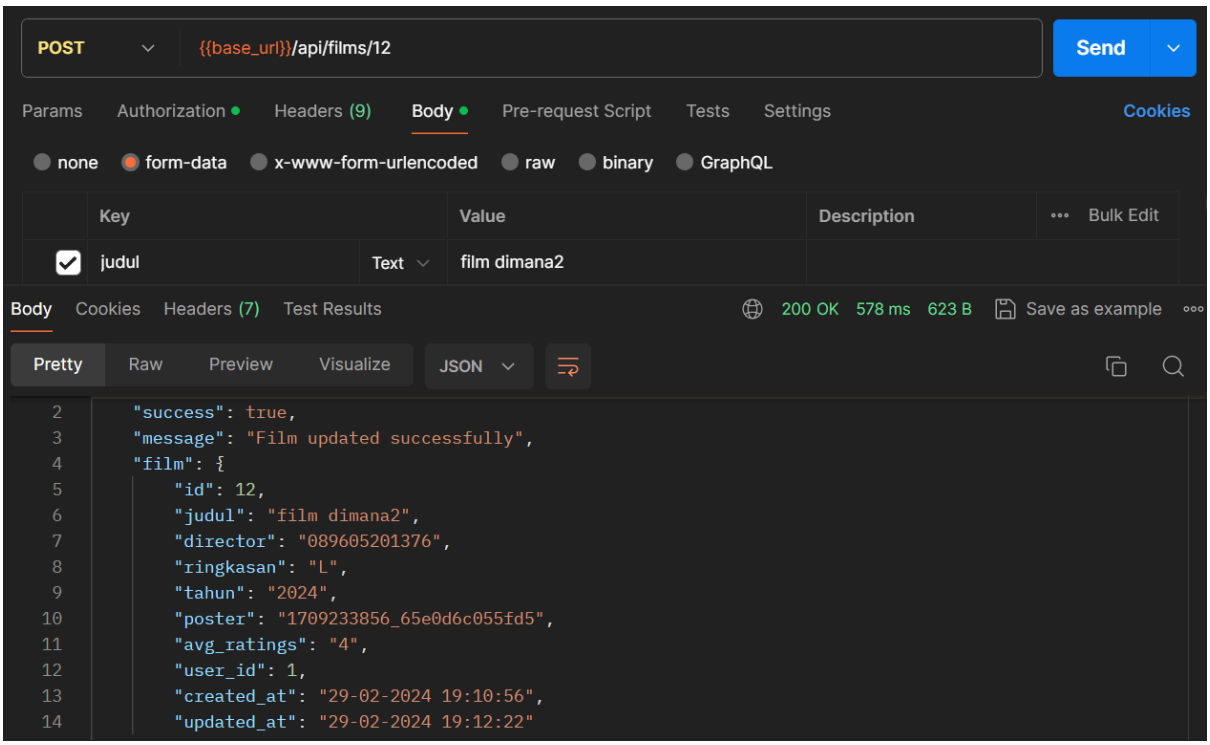
- POST /api/films: Menambahkan film baru



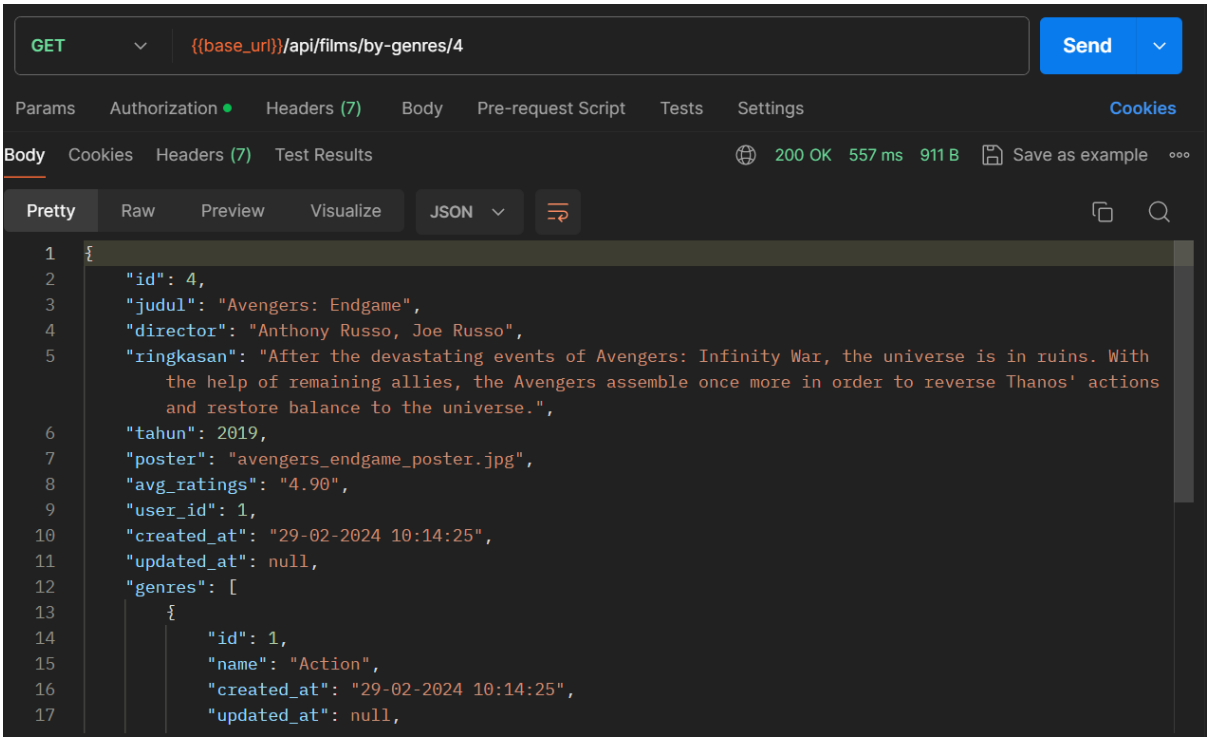
- GET /api/films/{id}: Mendapatkan detail film berdasarkan ID.



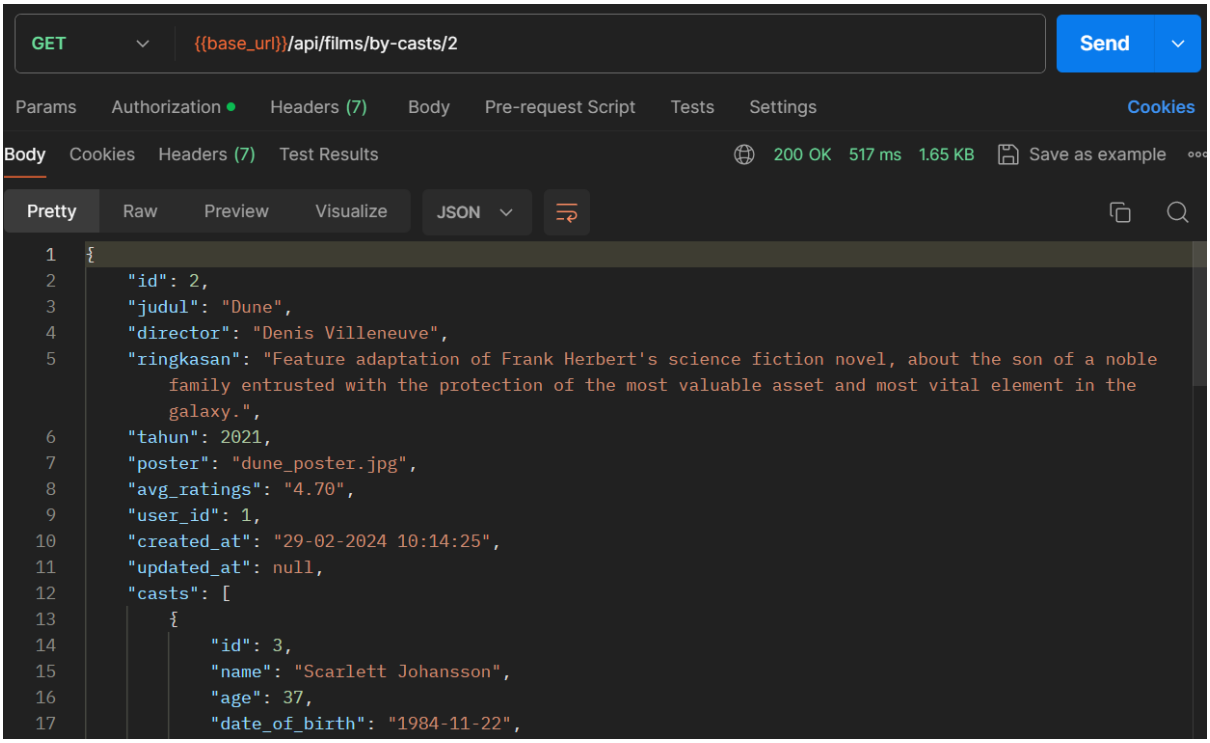
- PUT /api/films/{id}: Memperbarui detail film berdasarkan ID



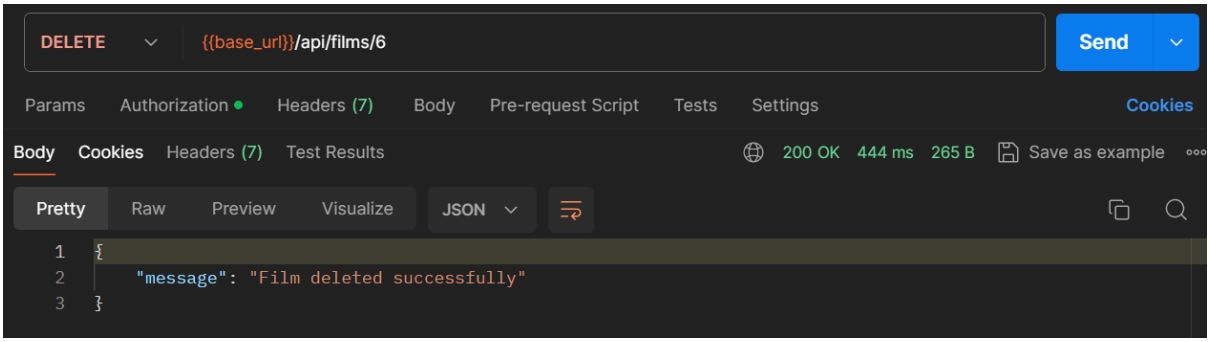
- GET /api/films/by-genres/{id}: Mendapatkan detail film berdasarkan ID genre



- GET /api/films/by-casts/{id}: Mendapatkan detail film berdasarkan ID pemain

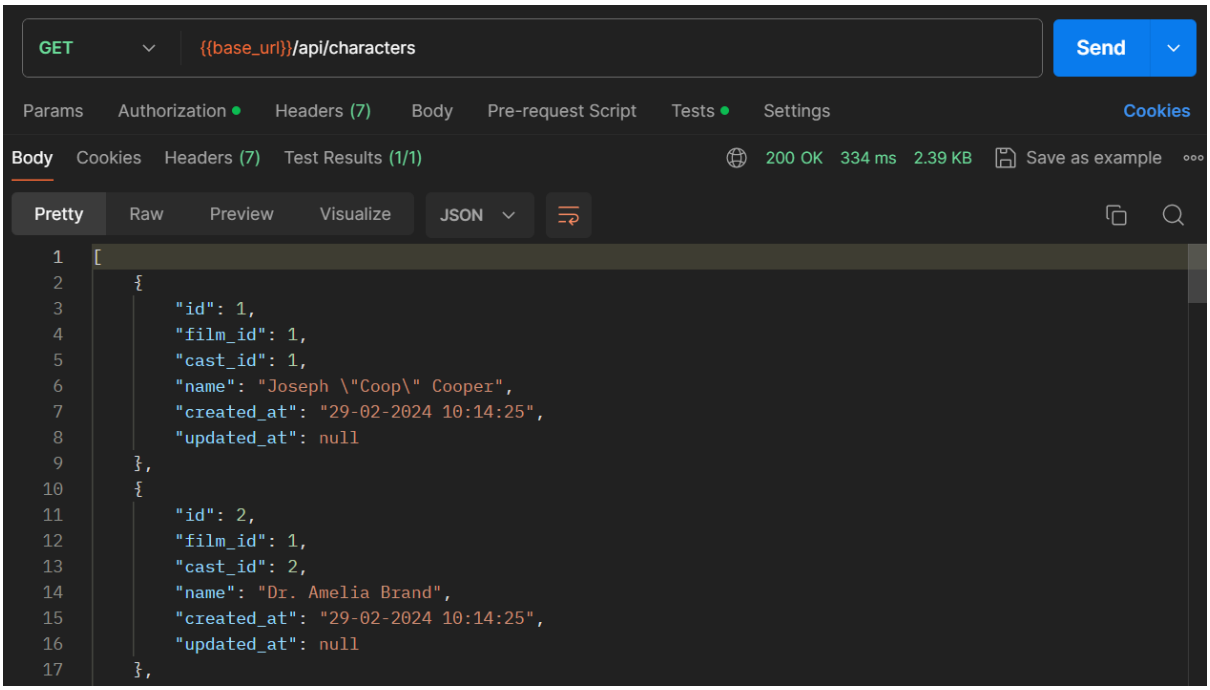


- DELETE /api/films/{id}: Menghapus film berdasarkan ID

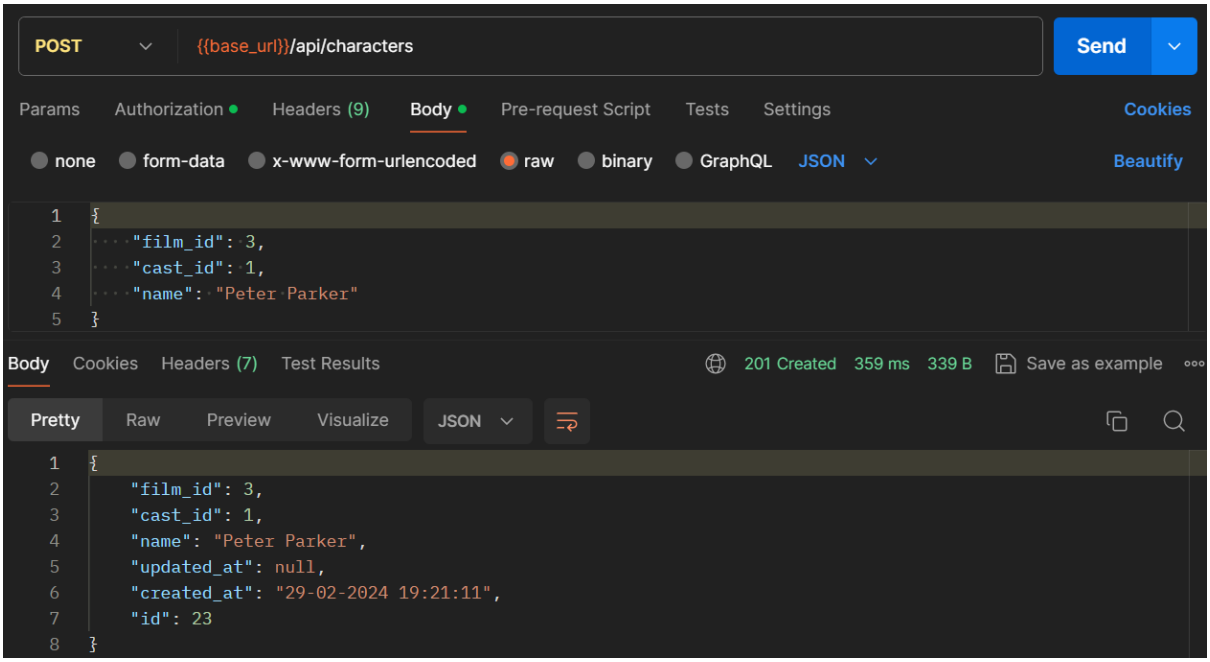


3.2.2.7 Endpoint Character

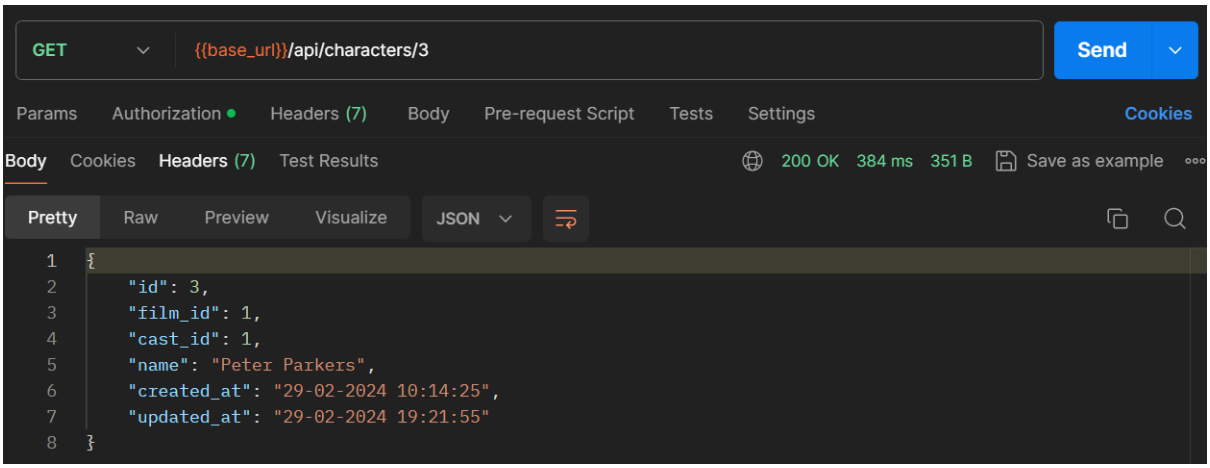
- GET /api/characters: Mendapatkan daftar semua karakter dalam film



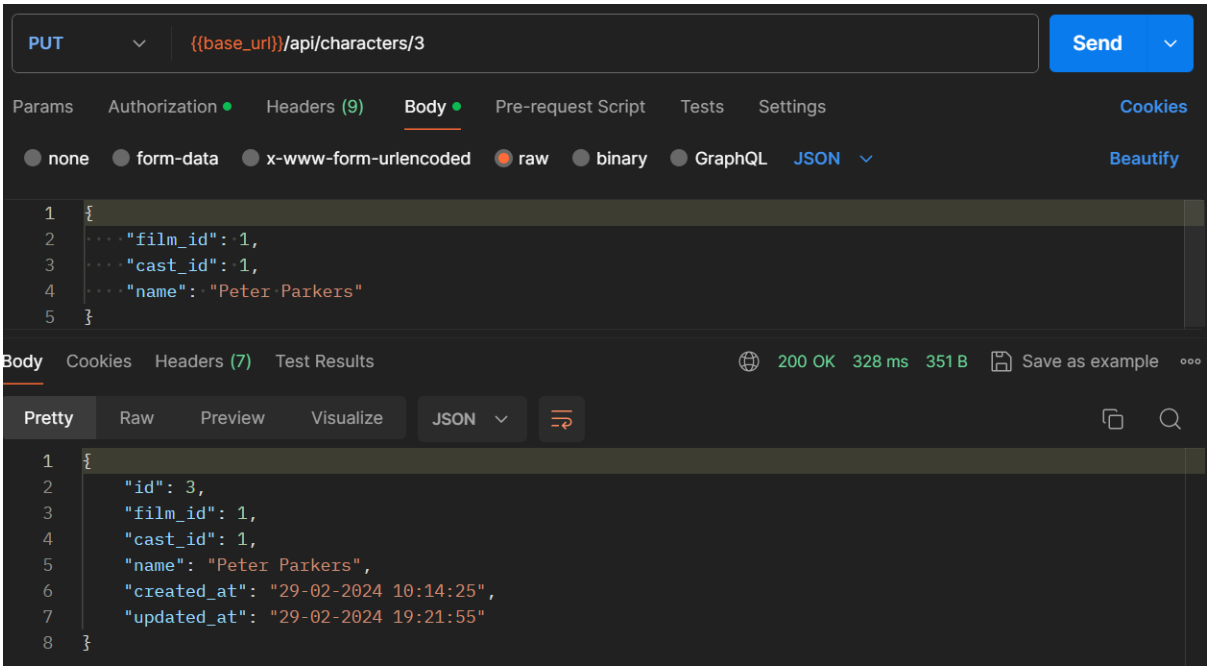
- POST /api/characters: Menambahkan karakter baru



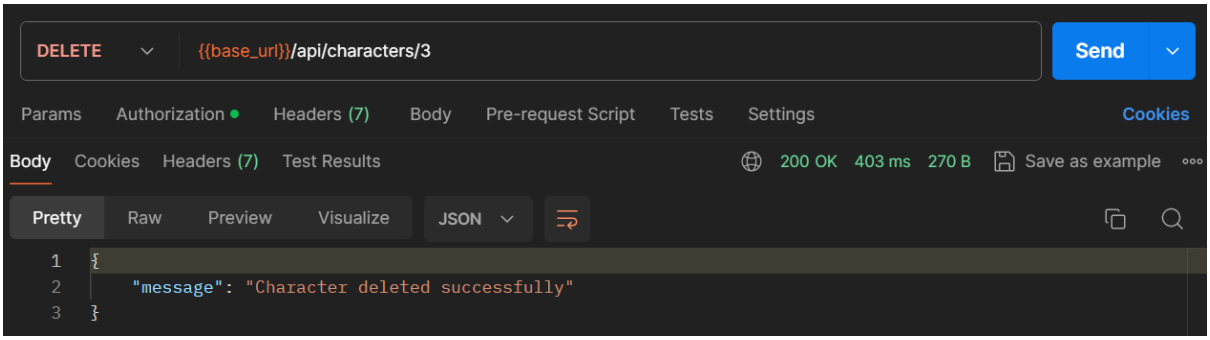
- GET /api/characters/{id}: Mendapatkan detail karakter berdasarkan ID



- PUT /api/characters/{id}: Memperbarui detail karakter berdasarkan ID

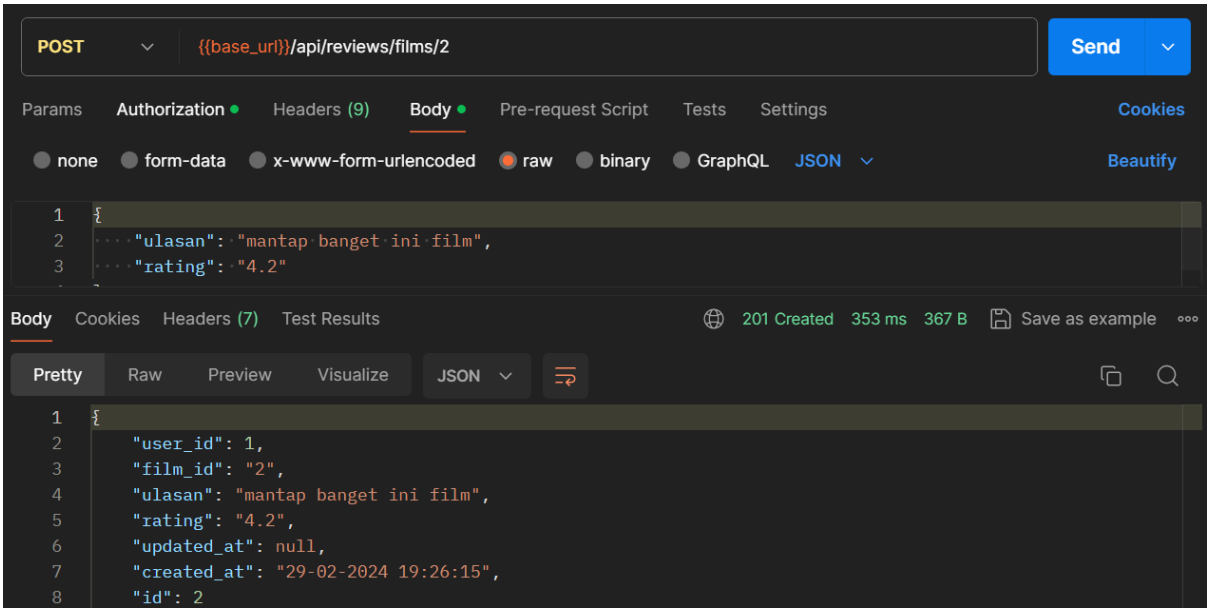


- DELETE /api/characters/{id}: Menghapus karakter berdasarkan ID

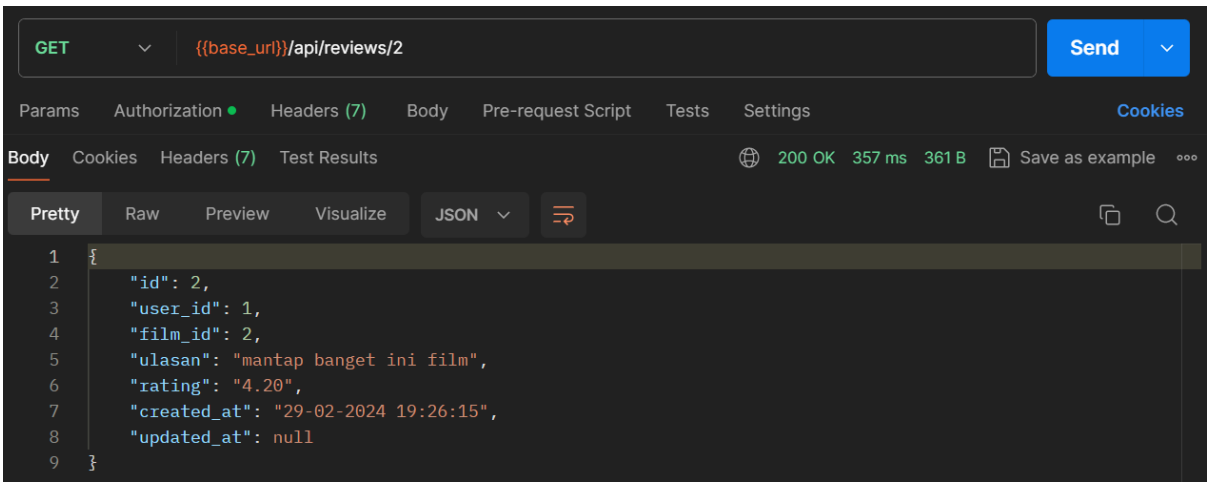


3.2.2.8 Endpoint Review

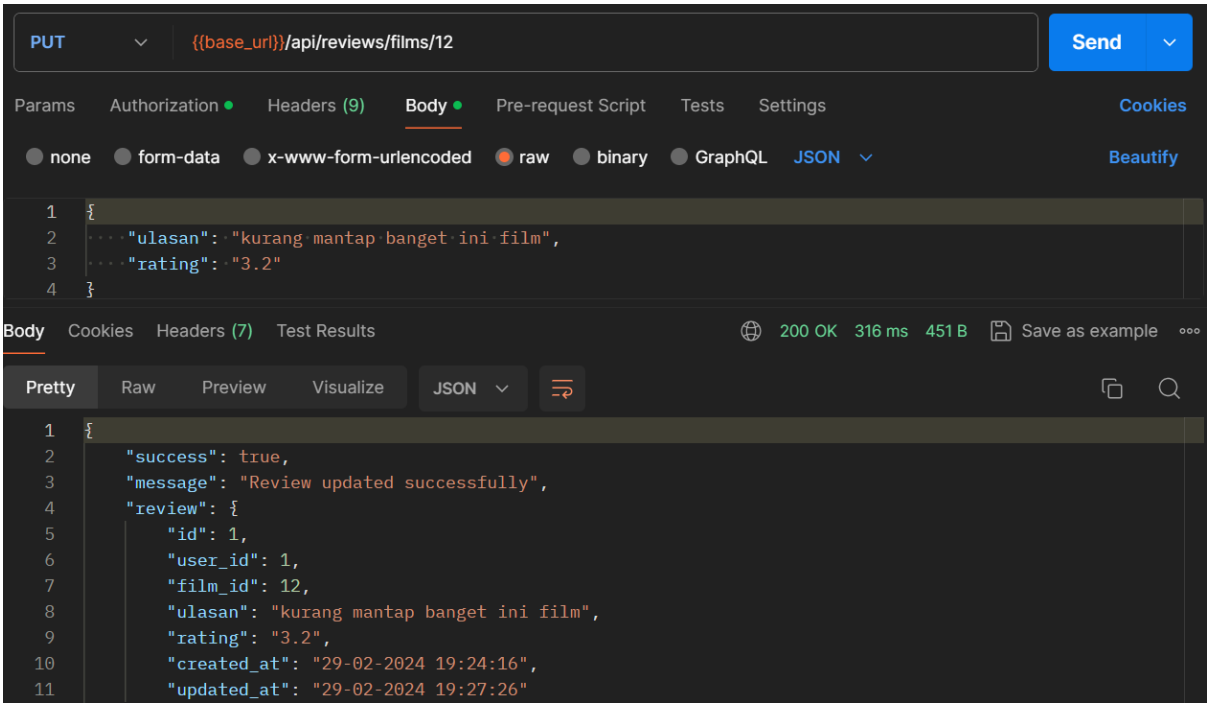
- POST /api/reviews/films/{idFilms}: Menambahkan ulasan baru untuk film tertentu



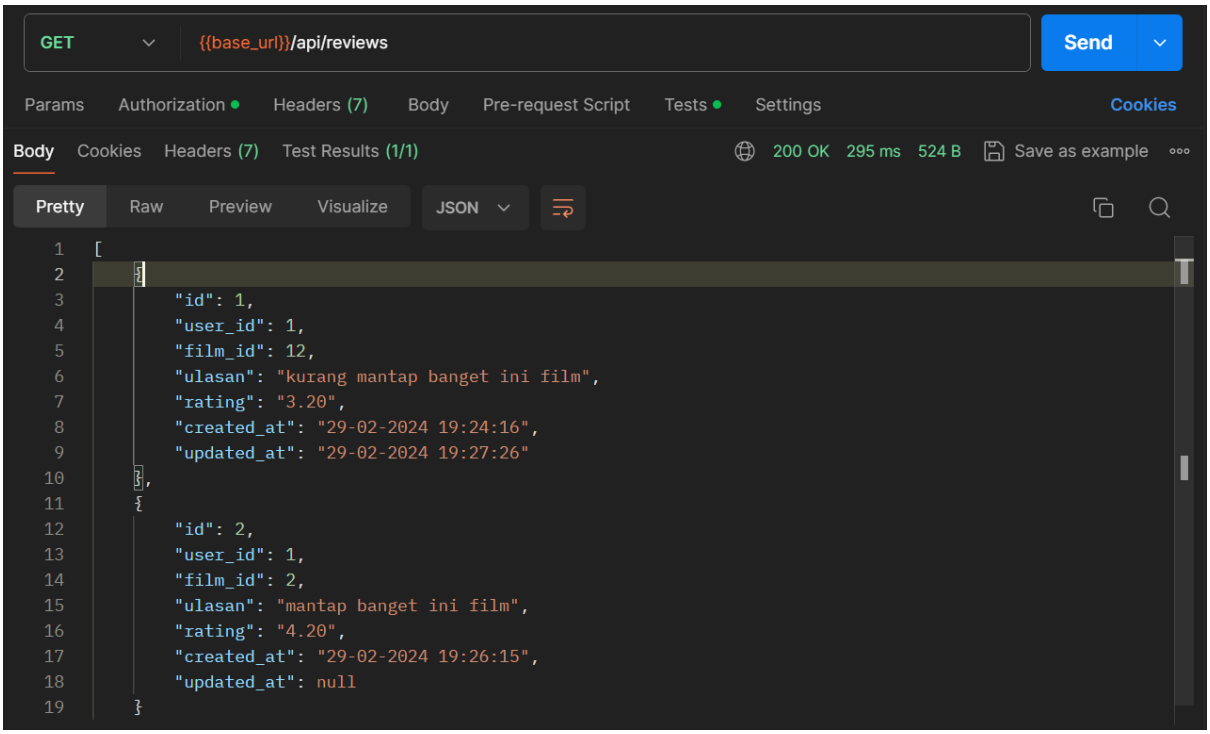
- GET /api/reviews/{id}: Mendapatkan detail ulasan berdasarkan ID



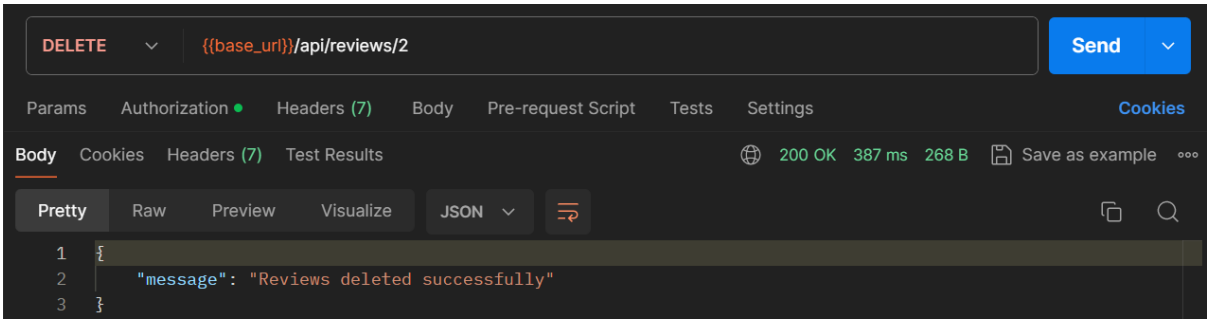
- PUT /api/reviews/films/{idFilms}: Memperbarui ulasan untuk film tertentu



- GET /api/reviews: Mendapatkan semua ulasan



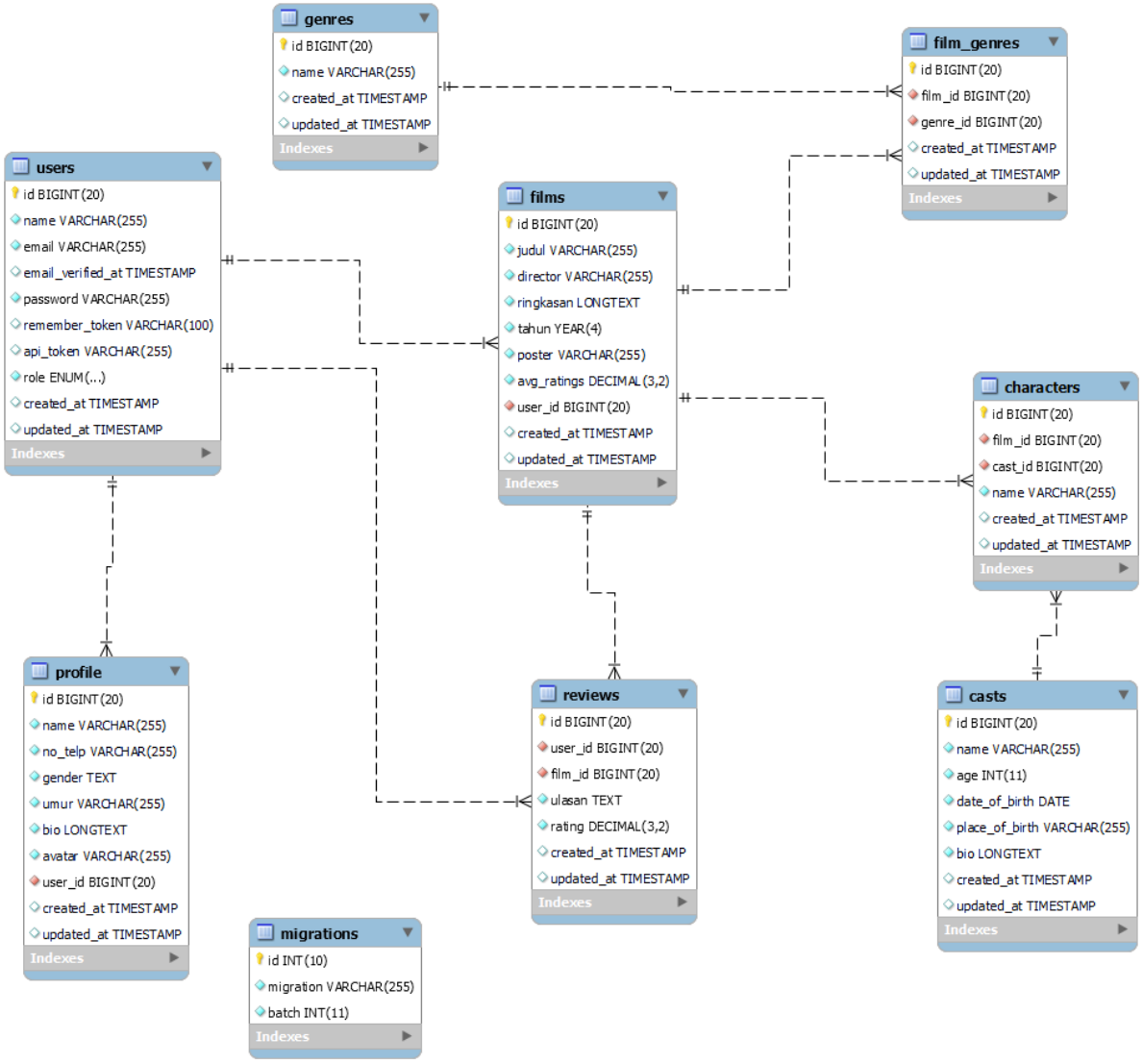
- DELETE /api/reviews/{id}: Menghapus ulasan berdasarkan ID.


DELETE
{{base_url}}/api/reviews/2
Send
Params
Authorization
Headers (7)
Body
Pre-request Script
Tests
Settings
Cookies
Body
Cookies
Headers (7)
Test Results
200 OK
387 ms
268 B
Save as example
Pretty
Raw
Preview
Visualize
JSON
{
"message": "Reviews deleted successfully"
}

BAB IV
RANCANGAN DATABASE (ERD)

4. Desain ERD

Berikut adalah desain atau rancangan database (ERD) yang telah dibuat dan digunakan pada screenscore api.



BAB V
KESIMPULAN

5. Kesimpulan

Dalam rangka menghadapi tuntutan era digital yang terus berkembang, Screenscore API muncul sebagai solusi terdepan untuk memperbaiki dan menyederhanakan pengelolaan data rating dan ulasan film. Dengan kemampuan CRUD yang dirancang khusus, Screenscore API memberikan keleluasaan bagi pengembang dan pengelola platform hiburan untuk membuat, membaca, memperbarui, dan menghapus data dengan mudah.

Ebook ini disusun sebagai panduan lengkap untuk memandu pembaca melalui proses penerapan Screenscore API dalam pengembangan aplikasi rating dan review film. Dari instalasi hingga konfigurasi, ebook ini menyajikan langkah-langkah praktis yang memungkinkan Anda memanfaatkan potensi Screenscore API secara optimal. Dengan implementasi Screenscore API, Anda dapat menghasilkan aplikasi yang responsif dan efisien, tidak hanya meningkatkan interaksi pengguna, tetapi juga memperbaiki pengelolaan data secara menyeluruh.

Melalui pembacaan ebook ini, diharapkan para pembaca dapat menguasai teknologi Screenscore API dan mengaplikasikannya dalam proyek-proyek pengembangan. Dengan demikian, Screenscore API tidak hanya menjadi alat yang mempermudah, tetapi juga menjadi katalisator untuk meningkatkan pengalaman pengguna, serta mengoptimalkan manajemen data dalam ekosistem hiburan digital yang dinamis. Dengan mengakhiri perjalanan ini, kita mengundang Anda untuk menggali lebih dalam potensi Screenscore API dan membuka pintu menuju era baru dalam pengembangan aplikasi rating dan review film.