

# **ScreenScore Manual Book**

## 1. Pendahuluan

### 1.1 Tujuan Pembuatan Dokumen

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

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

### 1.2 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.

Fitur Utama :

- Manajemen Film: Pengguna dapat melihat daftar film, menambahkan film baru, memperbarui detail film, dan menghapus film yang ada.
- Manajemen Ulasan: Pengguna dapat menambahkan ulasan untuk film tertentu, melihat ulasan yang telah dibuat, dan menghapus ulasan yang sudah tidak diperlukan.
- Manajemen Genre: Pengguna dapat melihat daftar genre, menambahkan genre baru, memperbarui detail genre, dan menghapus genre yang sudah tidak diperlukan.
- Manajemen Pemain: Pengguna dapat melihat daftar pemain, menambahkan pemain baru, memperbarui detail pemain, dan menghapus pemain yang sudah tidak diperlukan.
- Manajemen Karakter: Pengguna dapat melihat daftar karakter dalam film, menambahkan karakter baru, memperbarui detail karakter, dan menghapus karakter yang sudah tidak diperlukan.

### 1.3 Deskripsi Umum Dokumen

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 user manual aplikasi Screenscore

## 2. Cara Penggunaan

### 2.1 Instalasi dan setup

#### 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 0
Receiving objects: 100% (130/130), 3.64 MiB | 2.37 MiB/s, done.
Resolving deltas: 100% (28/28), done.
```

#### 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
```

#### 2.1.3 Setting environment

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

> tests	10
> vendor	11 DB_CONNECTION=mysql
	12 DB_HOST=127.0.0.1
⚙ .editorconfig	13 DB_PORT=3306
⚙ .env	14 DB_DATABASE=screenscore
\$ .env.example	15 DB_USERNAME=root
💎 .gitignore	16 DB_PASSWORD=

#### 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)
```

#### 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.
```

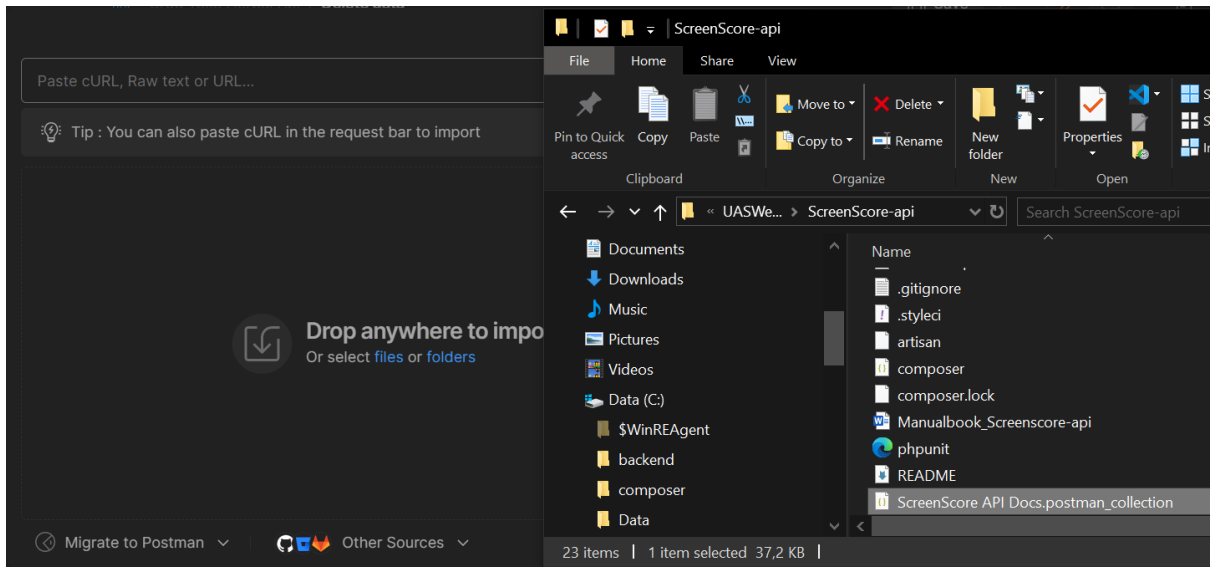
#### 2.1.6 Running Aplikasi

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

## 2.2 Pengujian

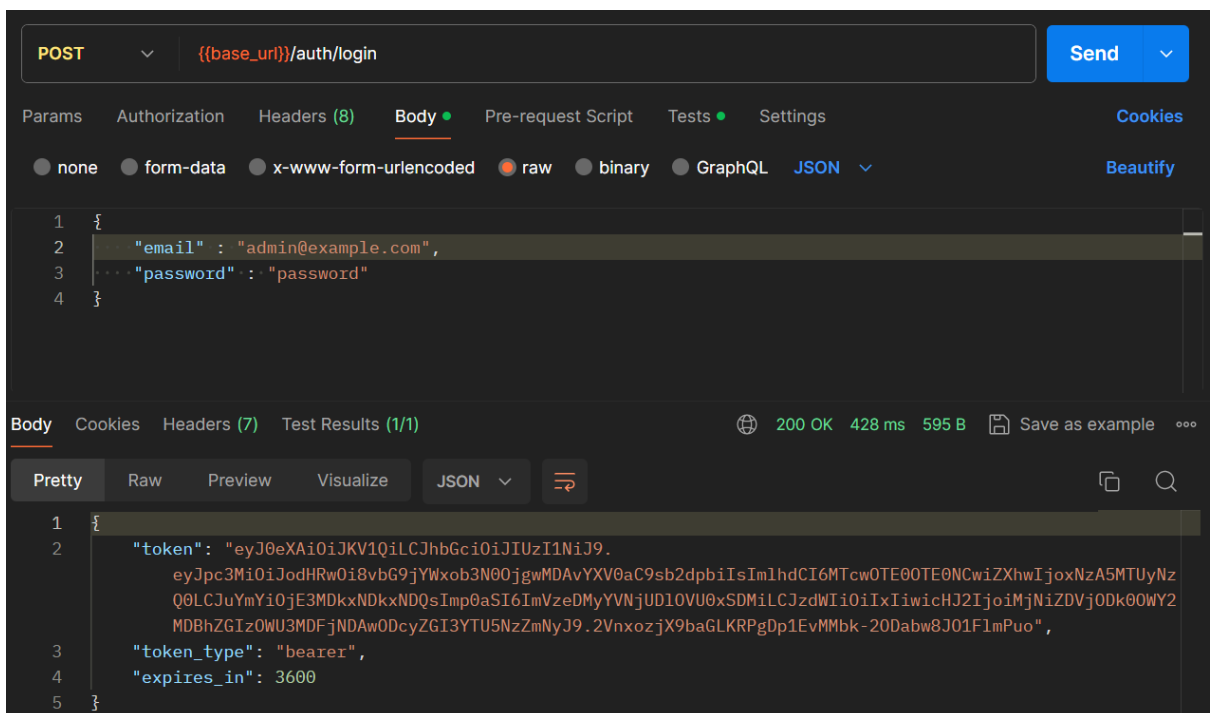
### 2.2.1 Import Collection

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



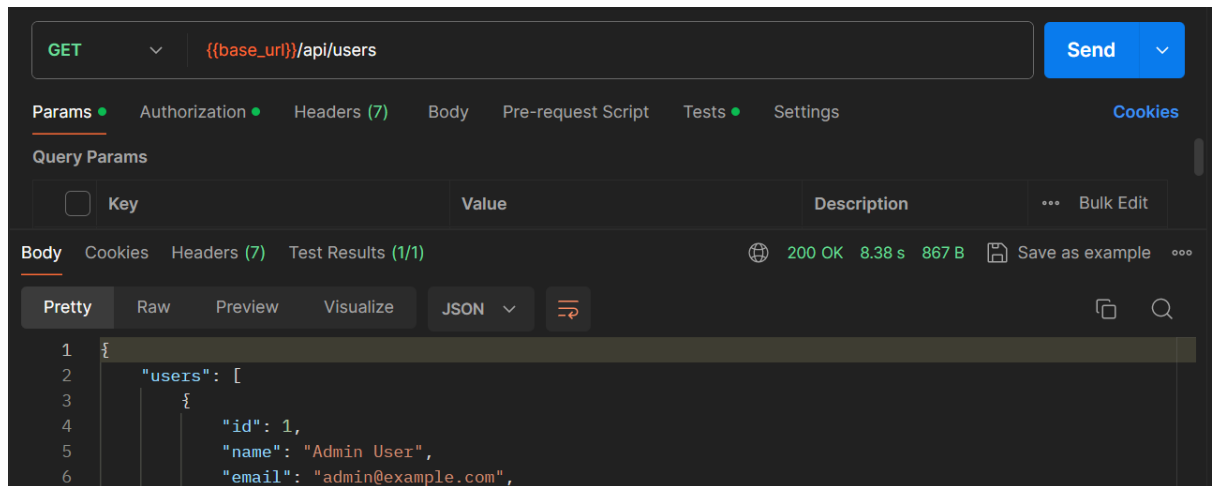
## 2.3 Endpoint Login

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

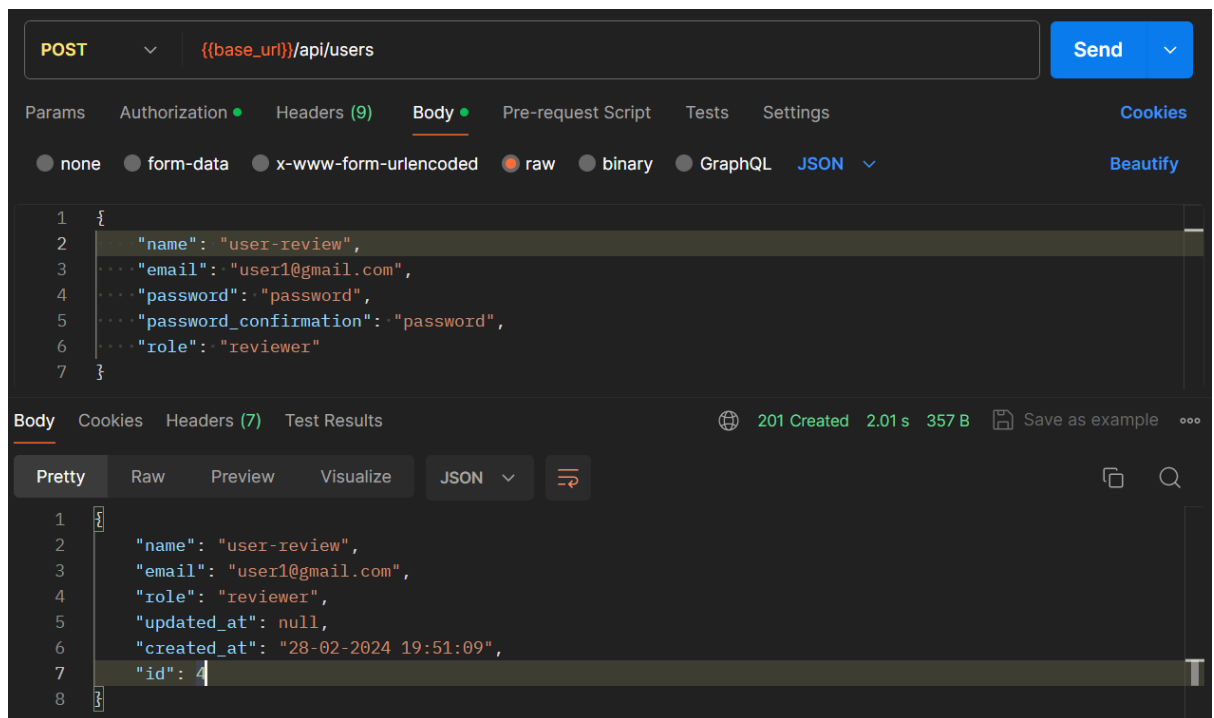


## 2.4 Endpoint User

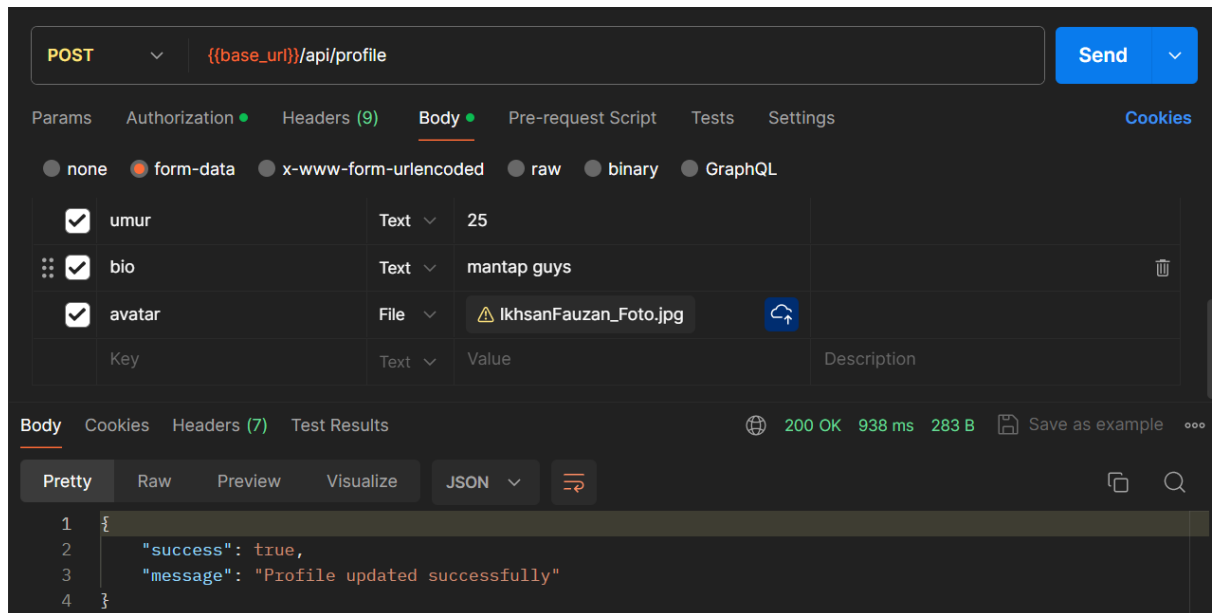
2.4.1 GET /api/user: Mendapatkan daftar semua user.



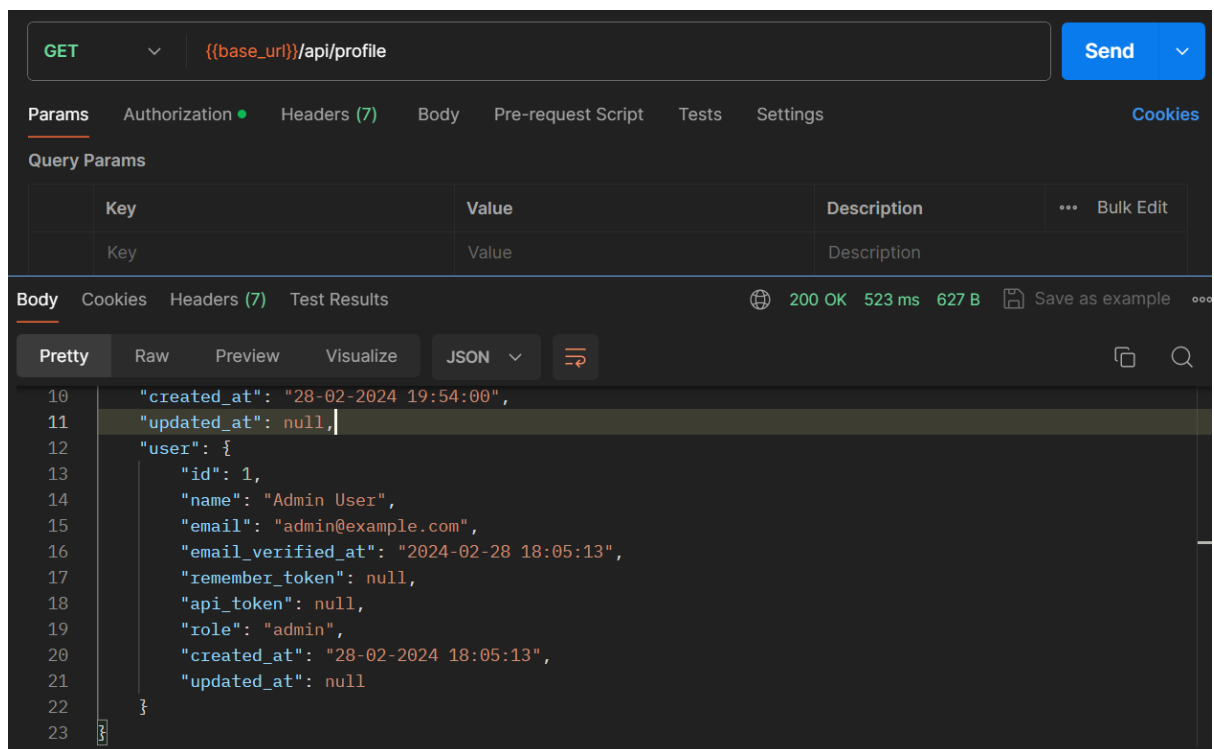
#### 2.4.2 POST /api/user: Menambahkan user baru.



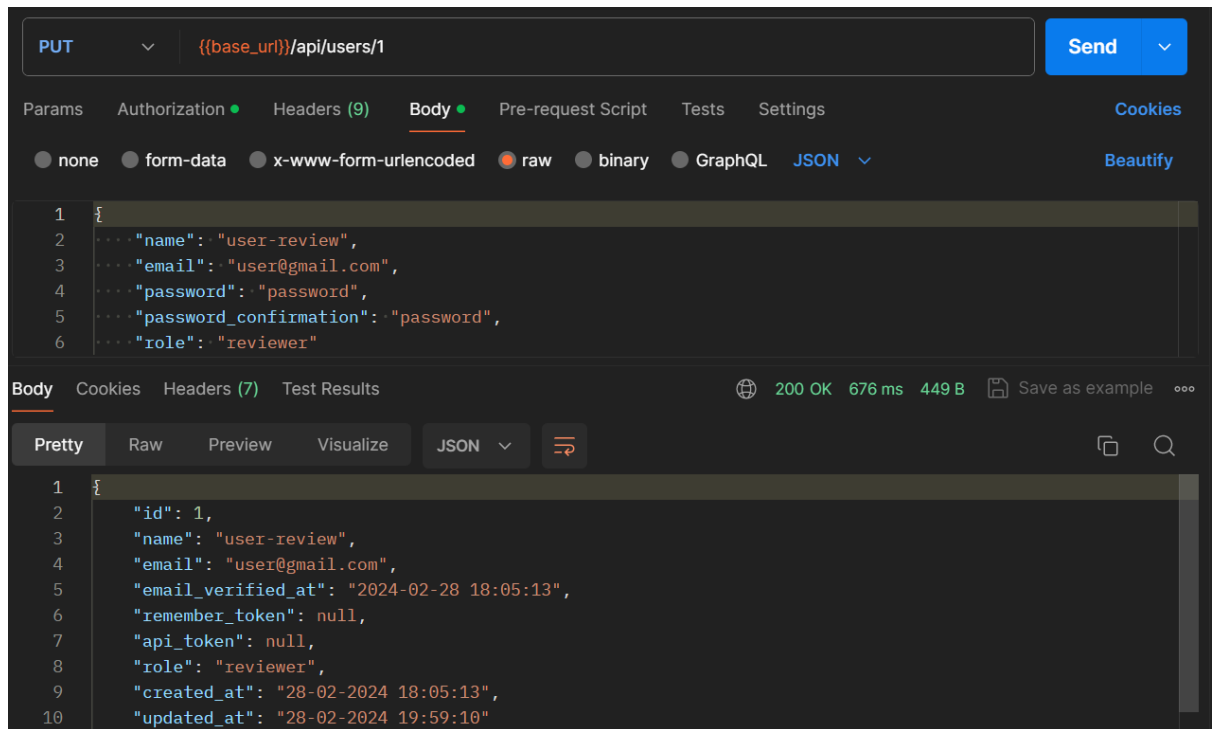
#### 2.4.3 POST /api/profile: Menambahkan user profile baru



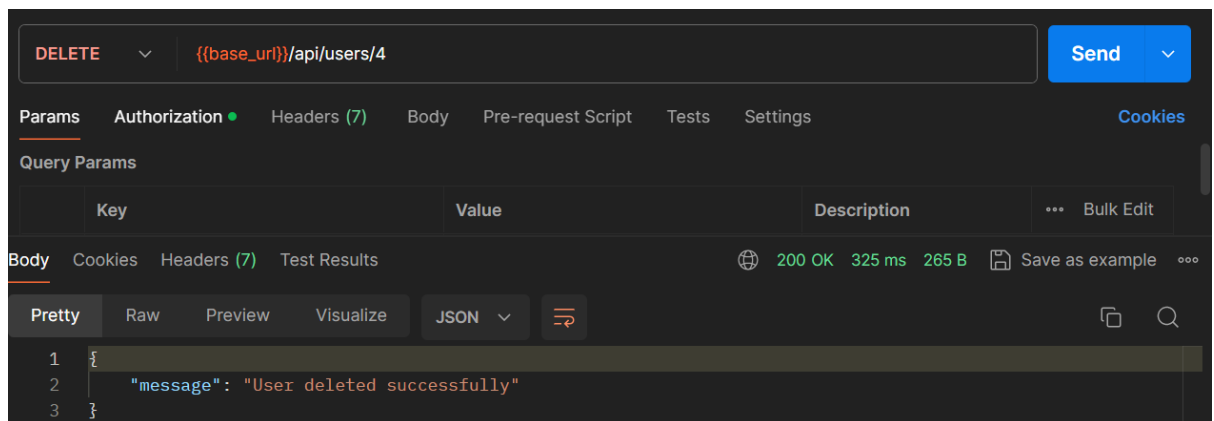
#### 2.4.4 GET /api/user: Mendapatkan daftar user profile.



#### 2.4.5 POST /api/user/{id}: Mengupdate user berdasarkan ID.

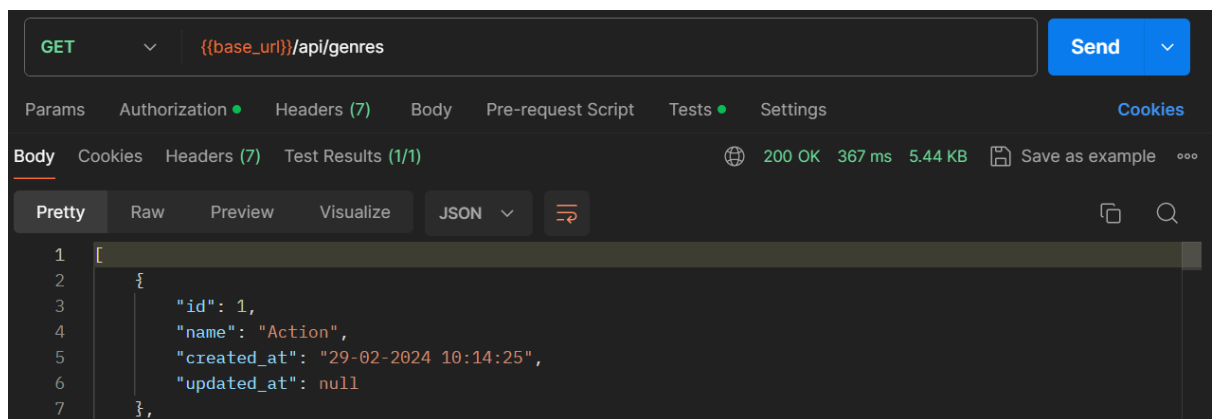


#### 2.4.6 DELETE /api/user/{id}: Menghapus user berdasarkan ID.



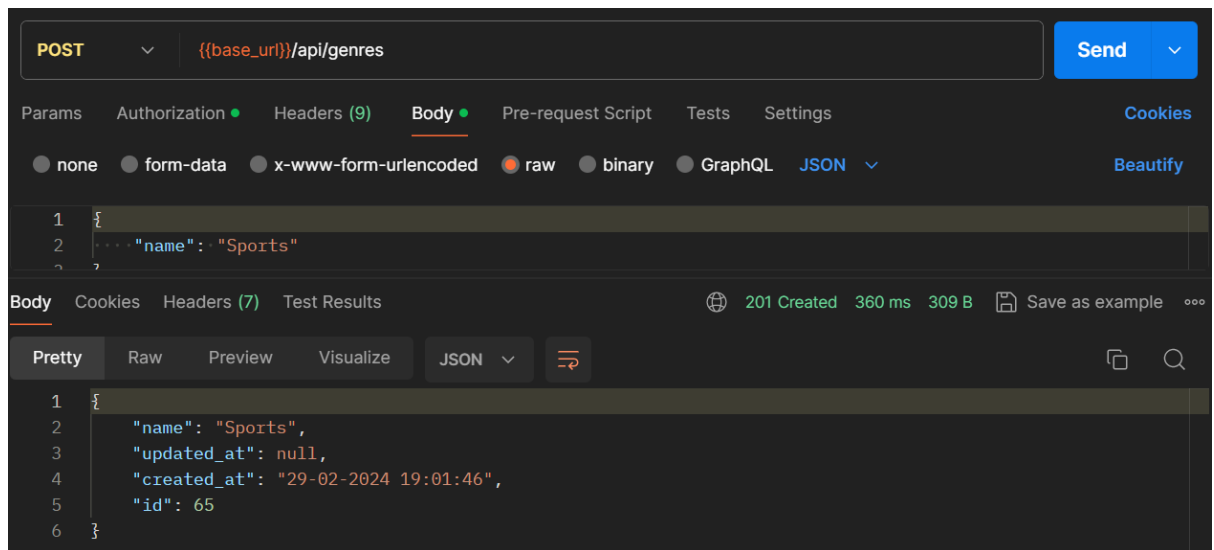
## 2.5 Endpoint Genre

#### 2.5.1 GET /api/genres: Mendapatkan daftar semua genre.

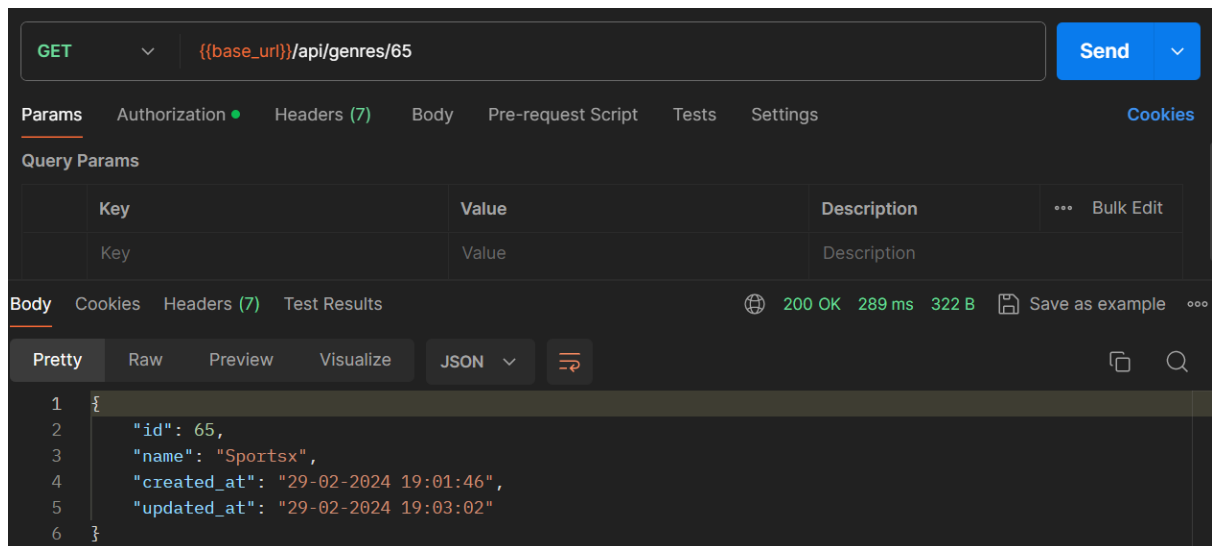




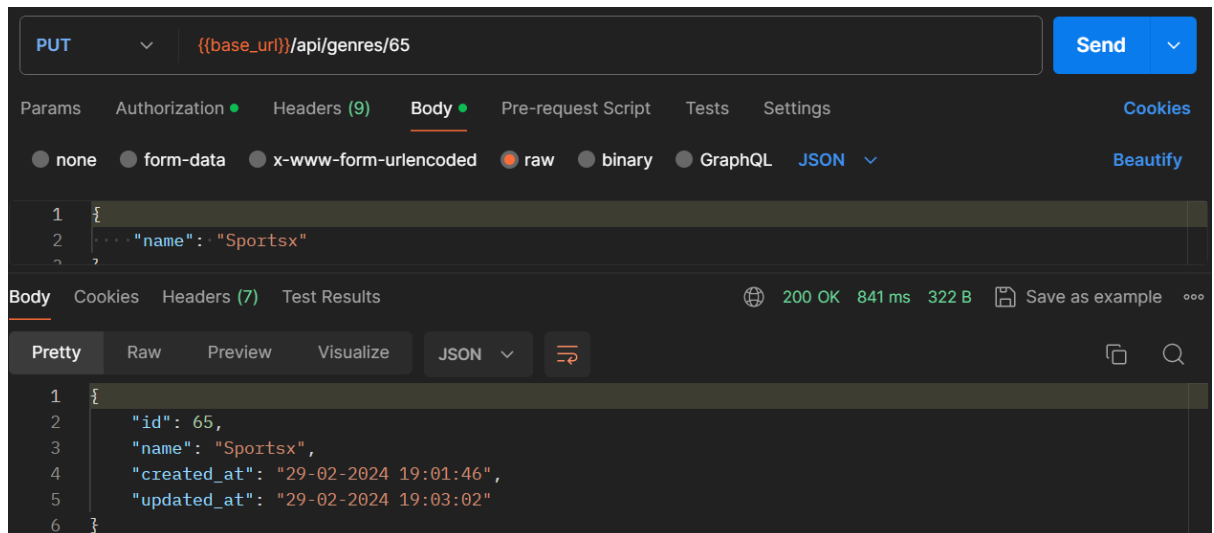
### 2.5.2 POST /api/genres: Menambahkan genre baru.



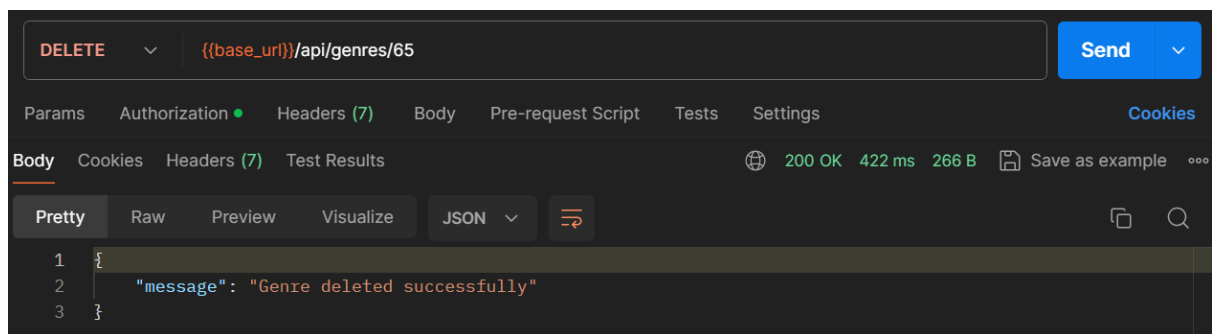
### 2.5.3 GET /api/genres/{id}: Mendapatkan detail genre berdasarkan ID.



### 2.5.4 PUT /api/genres/{id}: Memperbarui detail genre berdasarkan ID.

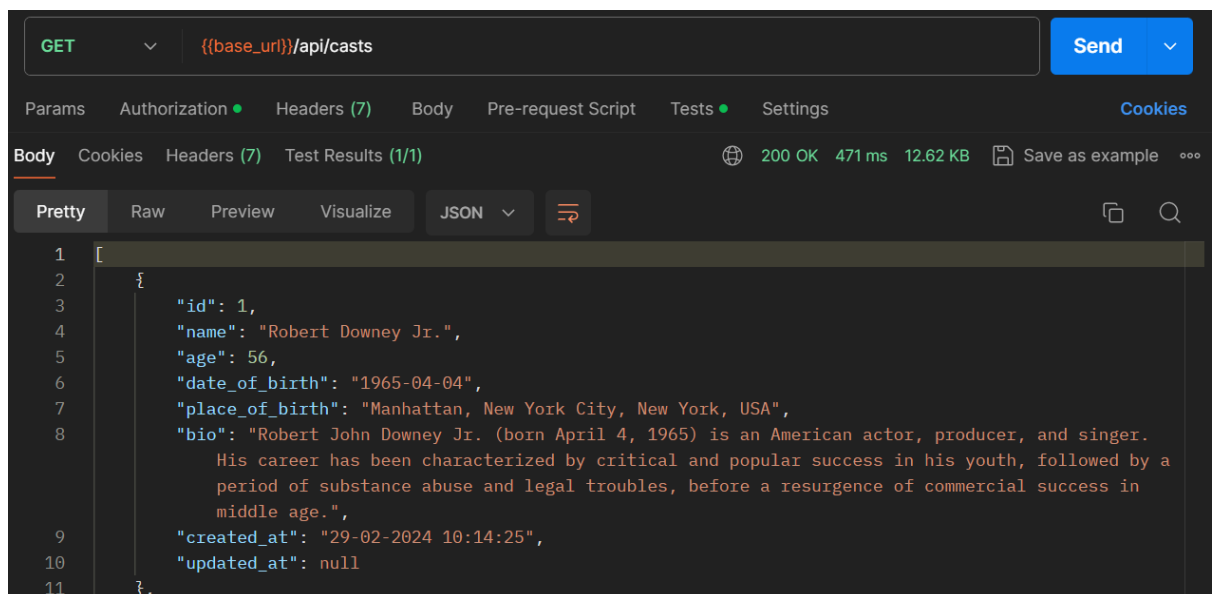


### 2.5.5 DELETE /api/genres/{id}: Menghapus genre berdasarkan ID.

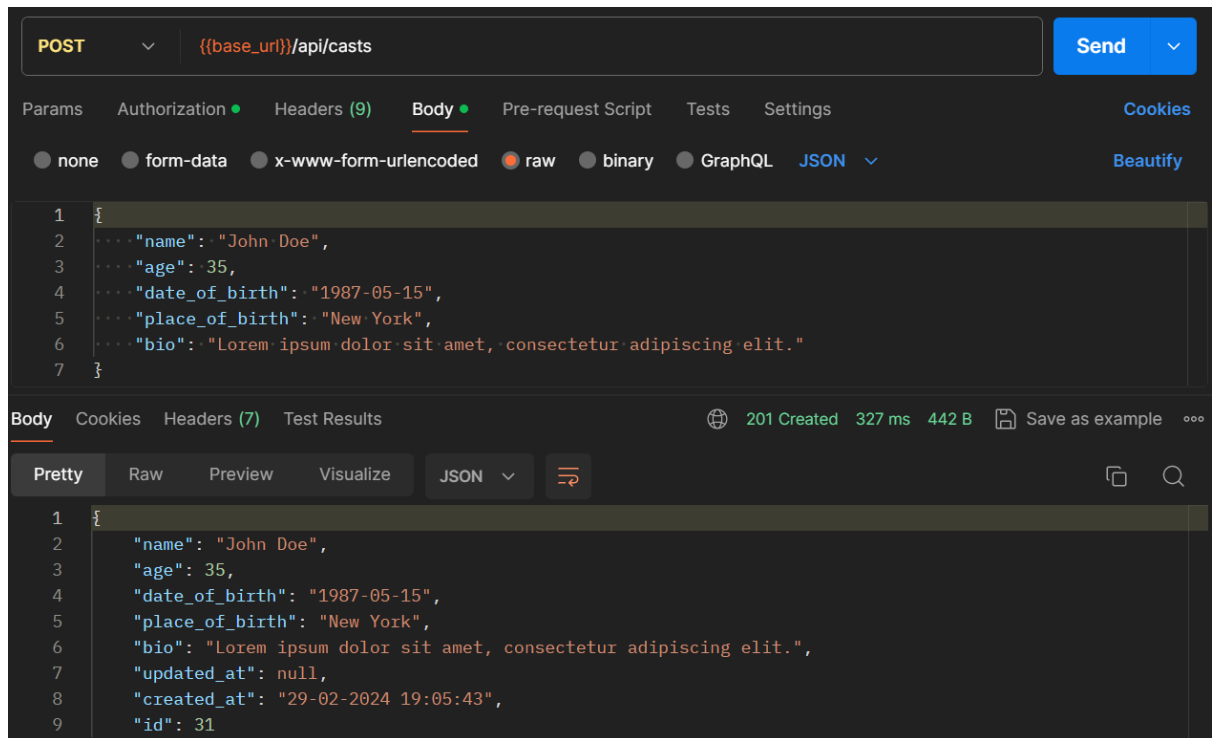


## 2.6 Endpoint Cast

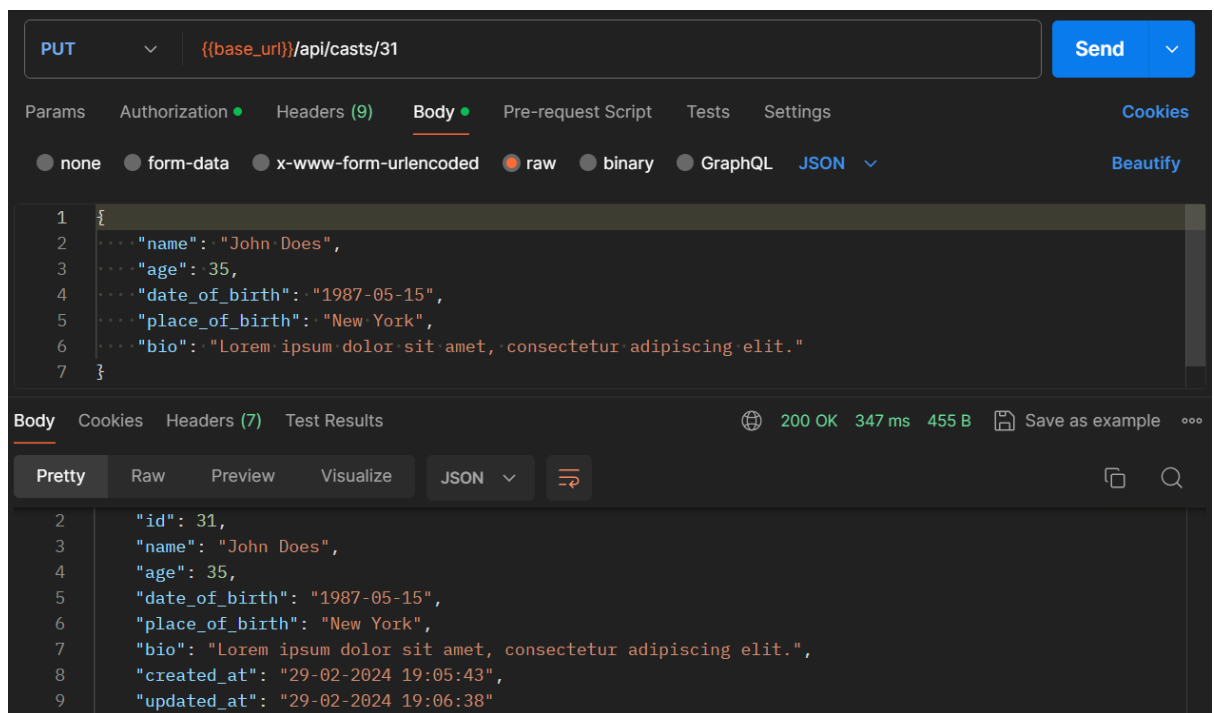
### 2.6.1 GET /api/casts: Mendapatkan daftar semua pemain.



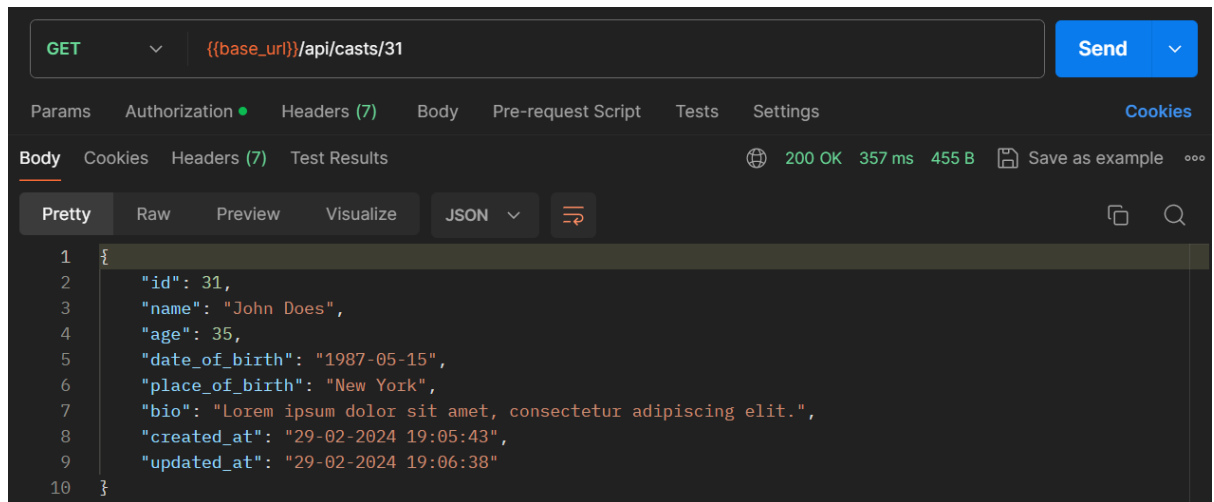
### 2.6.2 POST /api/casts: Menambahkan pemain baru.



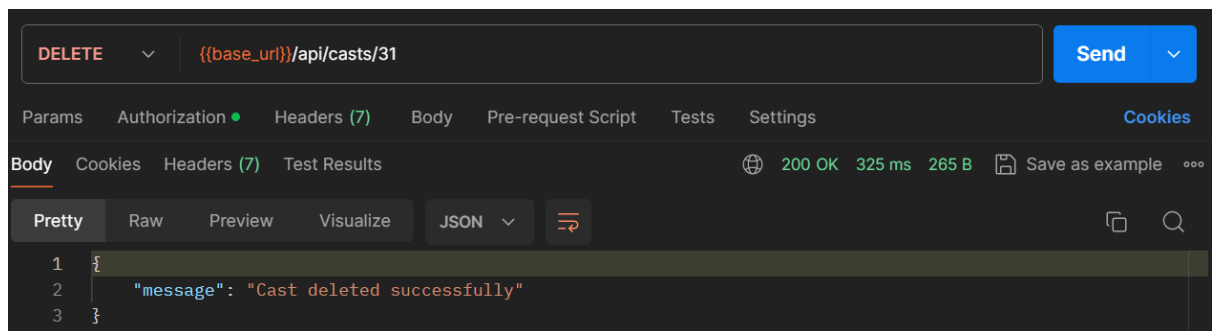
### 2.6.3 GET /api/casts/{id}: Mendapatkan detail pemain berdasarkan ID.



### 2.6.4 PUT /api/casts/{id}: Memperbarui detail pemain berdasarkan ID.

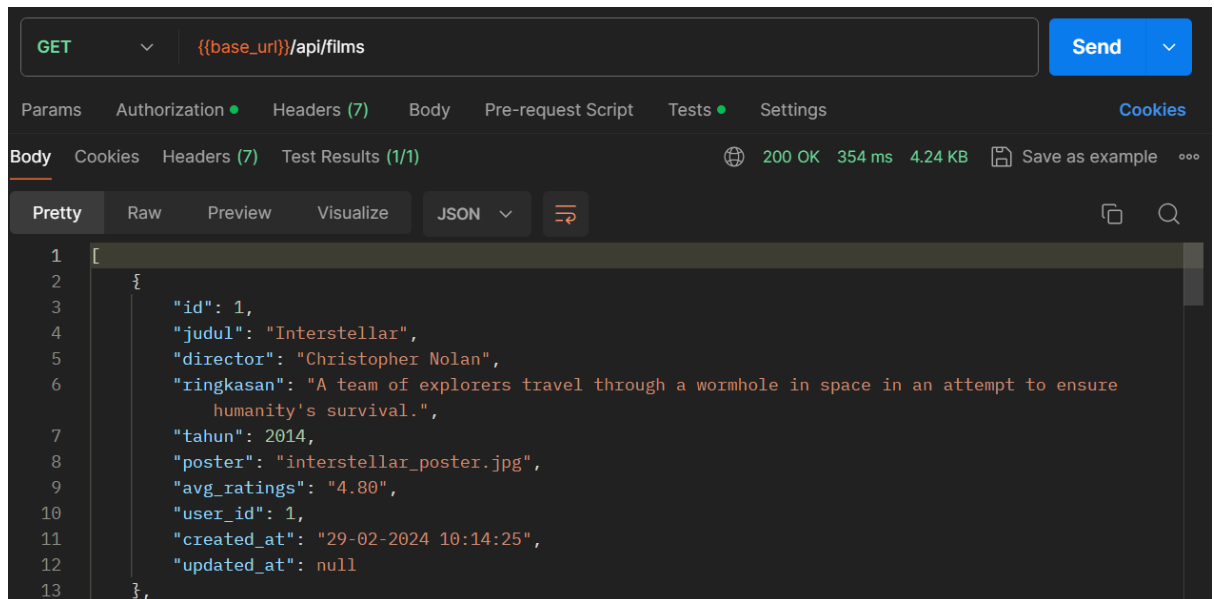


### 2.6.5 DELETE /api/casts/{id}: Menghapus pemain berdasarkan ID.

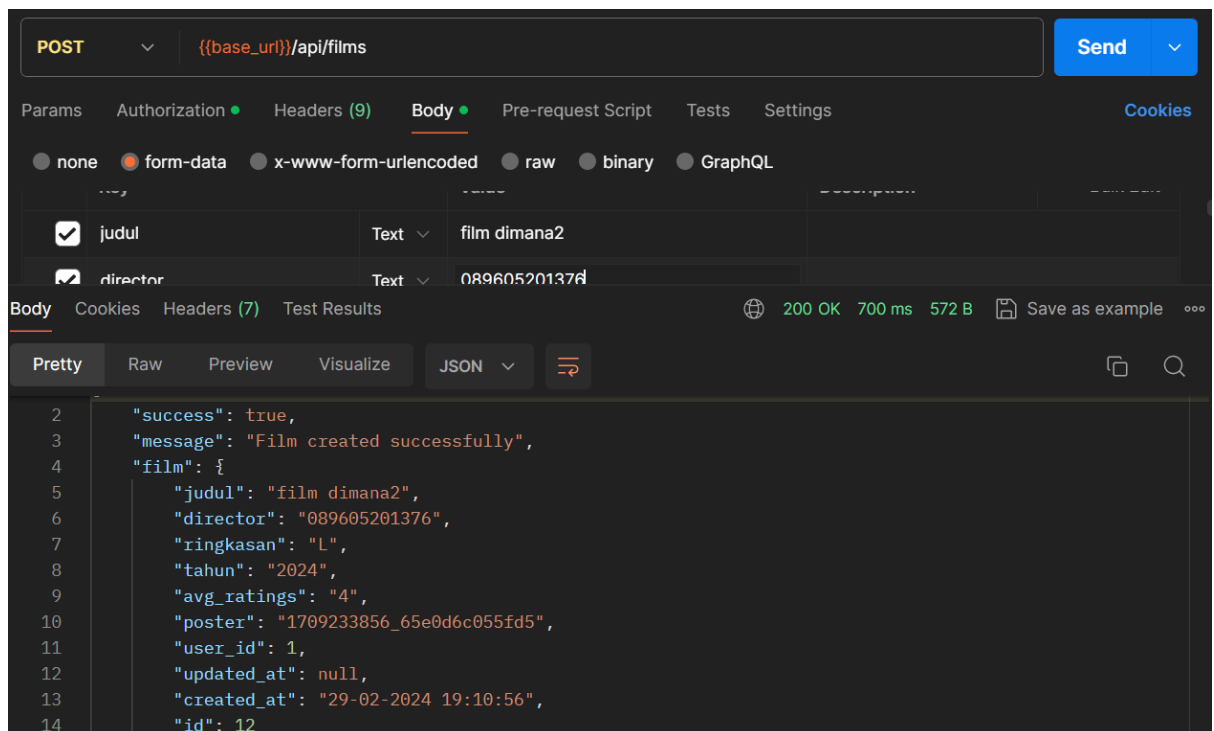


## 2.7 . Endpoint Film

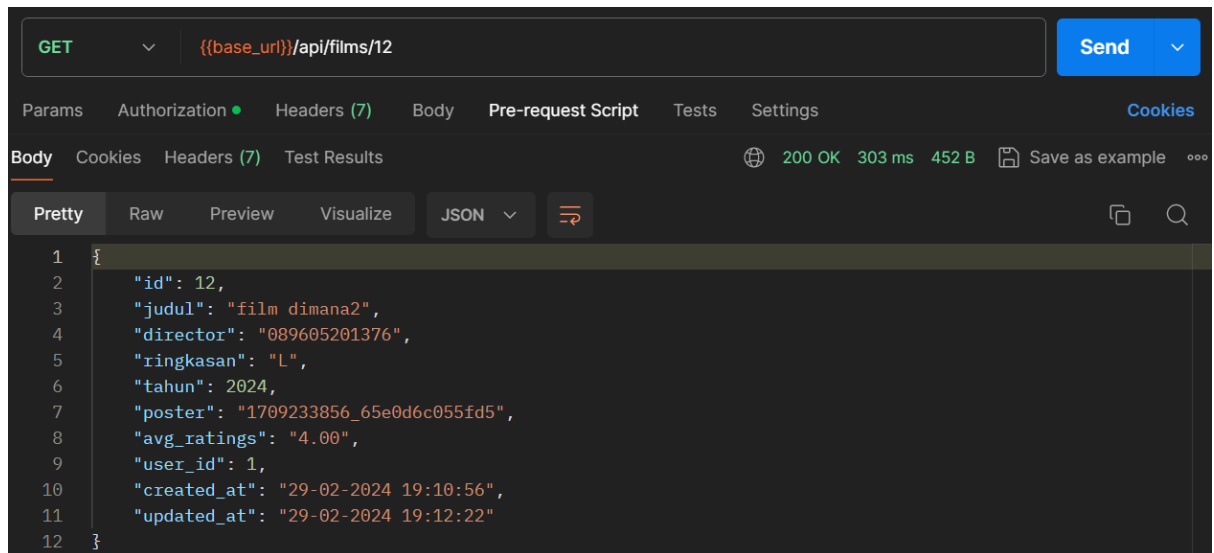
### 2.7.1 GET /api/films: Mendapatkan daftar semua film.



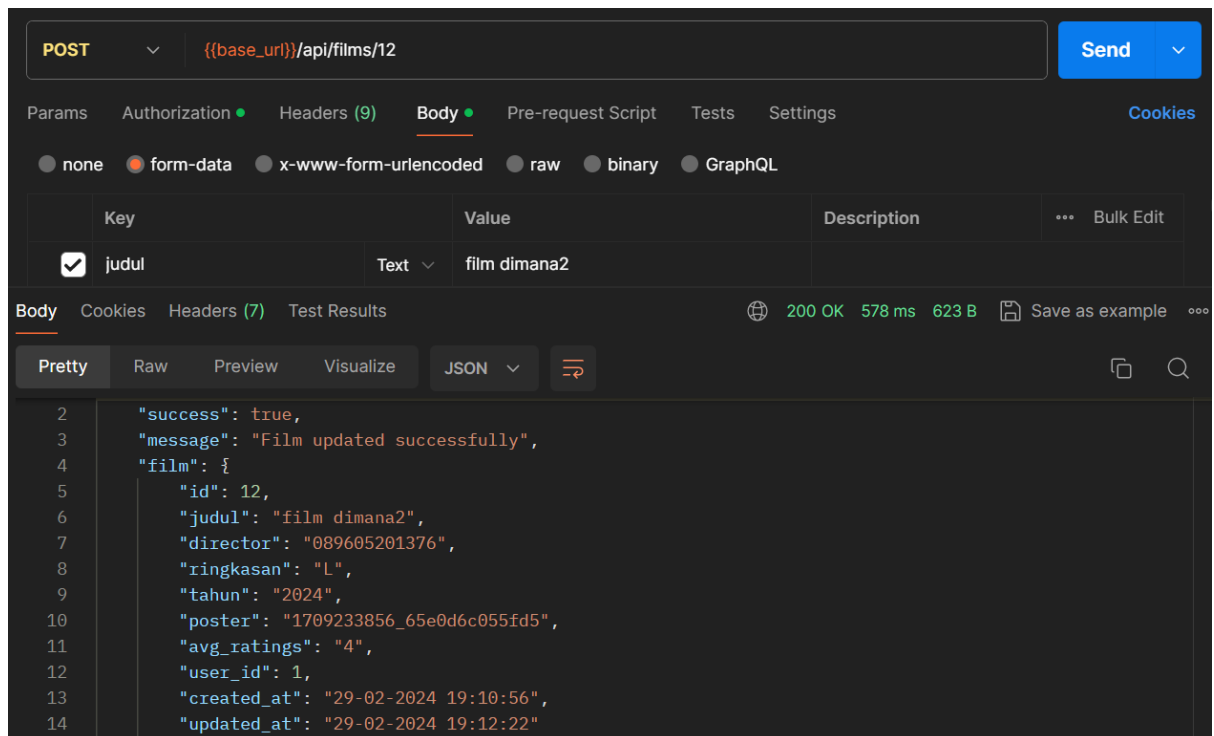
### 2.7.2 POST /api/films: Menambahkan film baru.



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



#### 2.7.4 PUT /api/films/{id}: Memperbarui detail film berdasarkan ID.



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

```
GET {{base_url}}/api/films/by-genres/4 Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (7) Test Results 200 OK 557 ms 911 B Save as example

Pretty Raw Preview Visualize JSON

1 {
2   "id": 4,
3   "judul": "Avengers: Endgame",
4   "director": "Anthony Russo, Joe Russo",
5   "ringkasan": "After the devastating events of Avengers: Infinity War, the universe is in ruins. With the help of remaining allies, the Avengers assemble once more in order to reverse Thanos' actions and restore balance to the universe.",
6   "tahun": 2019,
7   "poster": "avengers_endgame_poster.jpg",
8   "avg_ratings": "4.90",
9   "user_id": 1,
10  "created_at": "29-02-2024 10:14:25",
11  "updated_at": null,
12  "genres": [
13    {
14      "id": 1,
15      "name": "Action",
16      "created_at": "29-02-2024 10:14:25",
17      "updated_at": null,
```

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

```
GET {{base_url}}/api/films/by-casts/2 Send

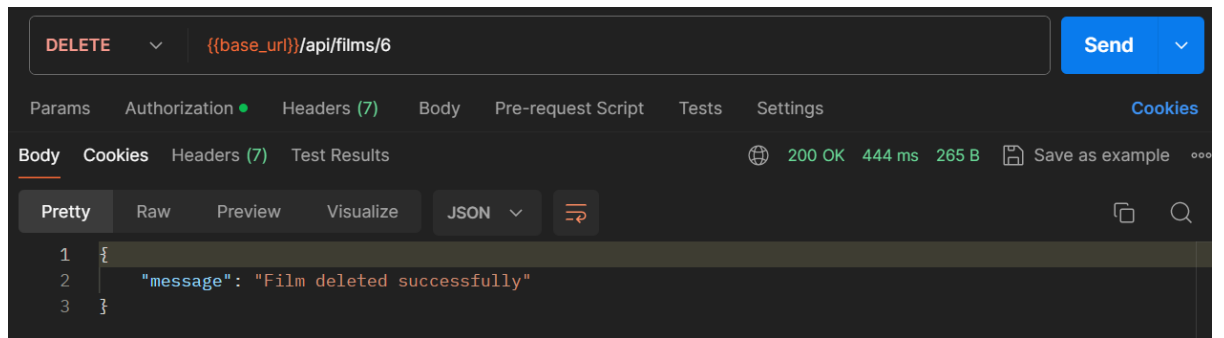
Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (7) Test Results 200 OK 517 ms 1.65 KB Save as example

Pretty Raw Preview Visualize JSON

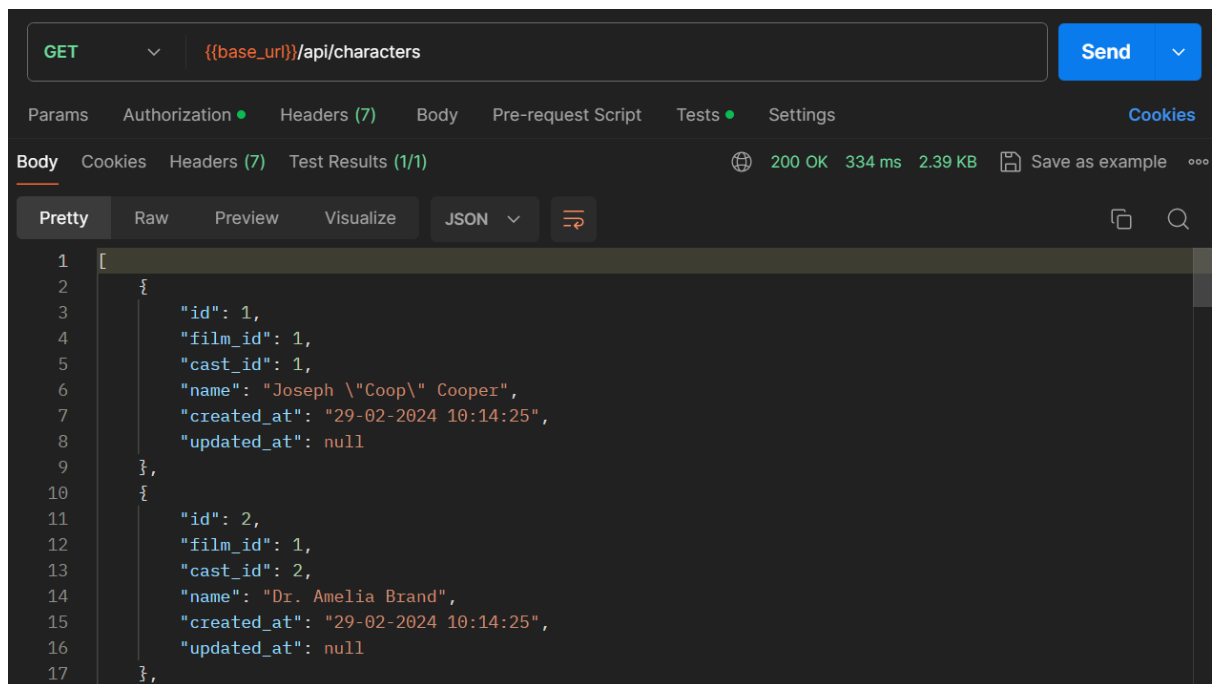
1 {
2   "id": 2,
3   "judul": "Dune",
4   "director": "Denis Villeneuve",
5   "ringkasan": "Feature adaptation of Frank Herbert's science fiction novel, about the son of a noble family entrusted with the protection of the most valuable asset and most vital element in the galaxy.",
6   "tahun": 2021,
7   "poster": "dune_poster.jpg",
8   "avg_ratings": "4.70",
9   "user_id": 1,
10  "created_at": "29-02-2024 10:14:25",
11  "updated_at": null,
12  "casts": [
13    {
14      "id": 3,
15      "name": "Scarlett Johansson",
16      "age": 37,
17      "date_of_birth": "1984-11-22",
```

#### 2.7.7 DELETE /api/films/{id}: Menghapus film berdasarkan ID.



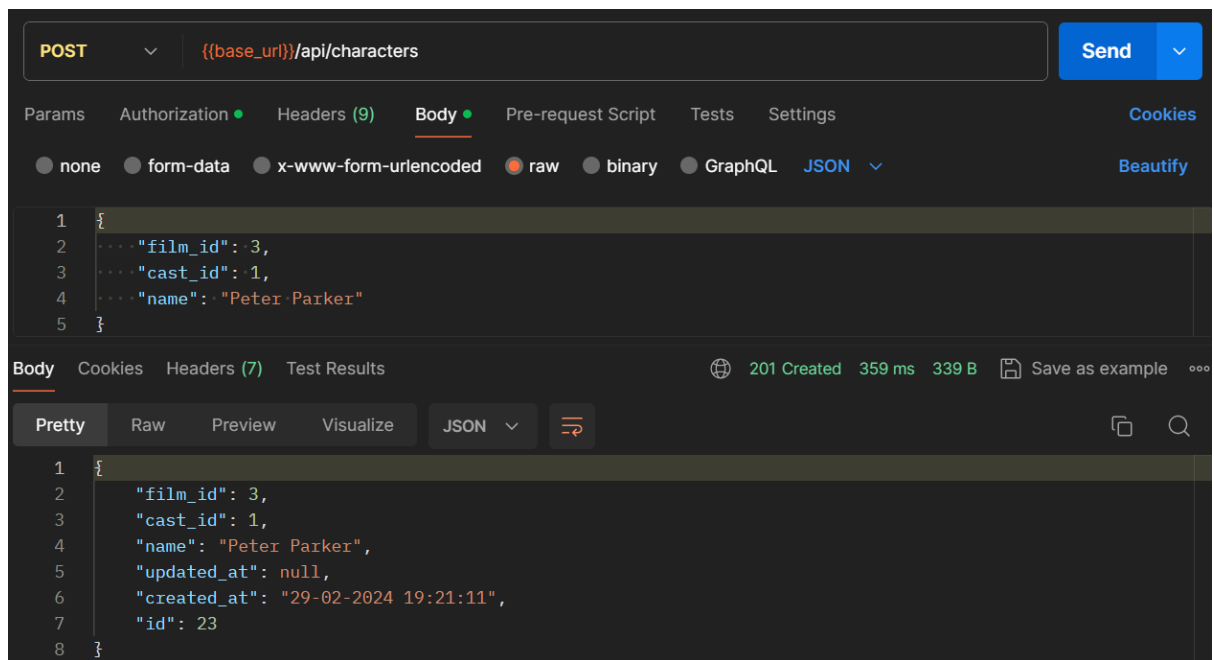
## 2.8 Endpoint Character

### 2.8.1 GET /api/characters: Mendapatkan daftar semua karakter dalam film.

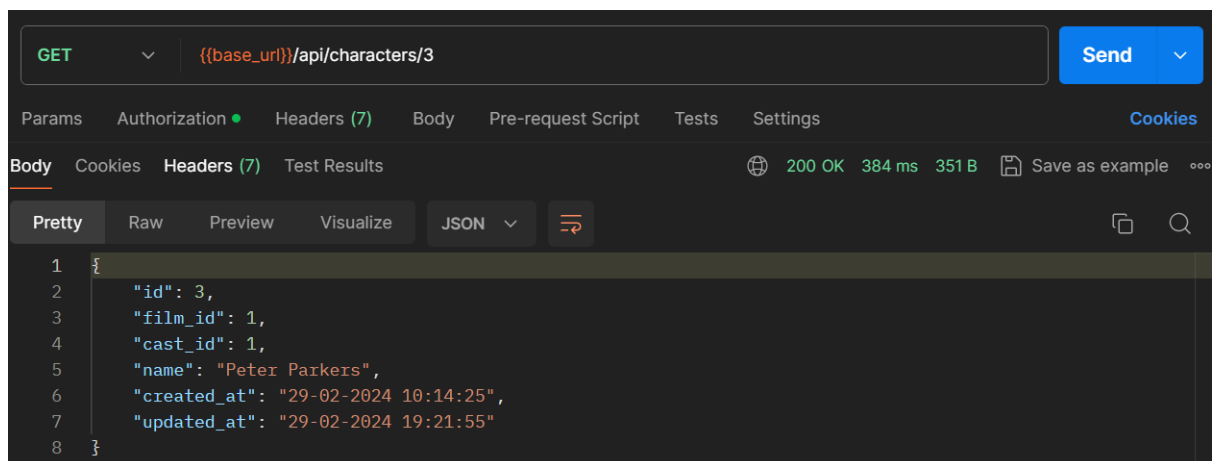


### 2.8.2 POST /api/characters: Menambahkan karakter baru.

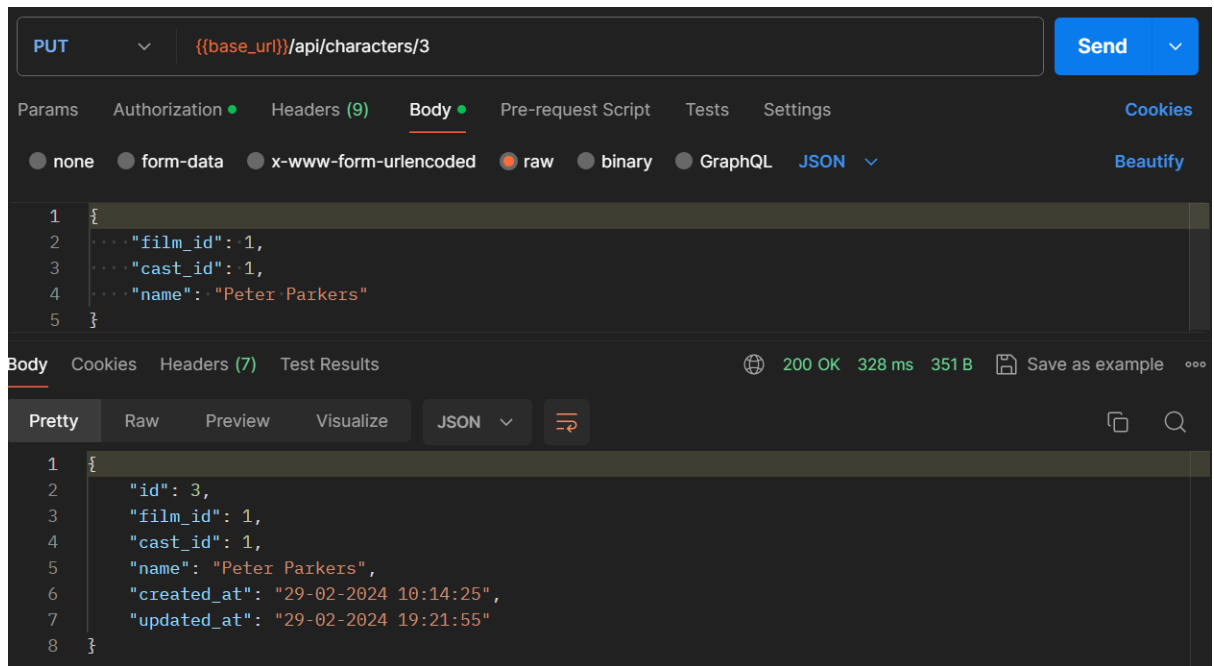




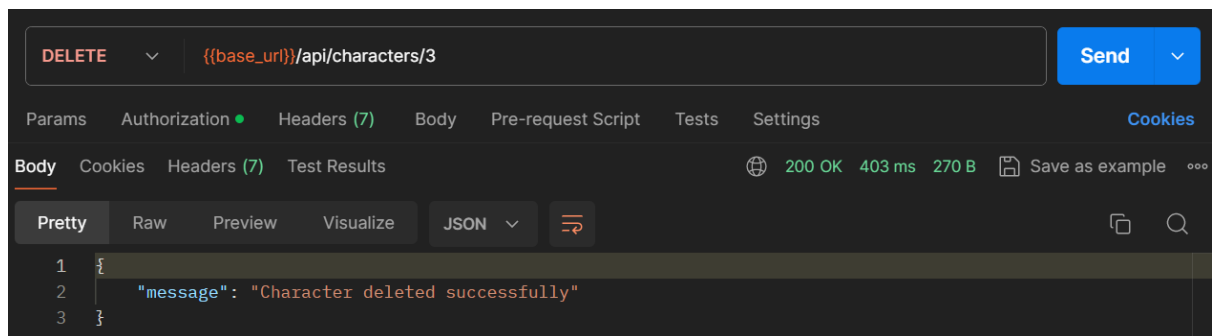
2.8.3 GET /api/characters/{id}: Mendapatkan detail karakter berdasarkan ID.



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

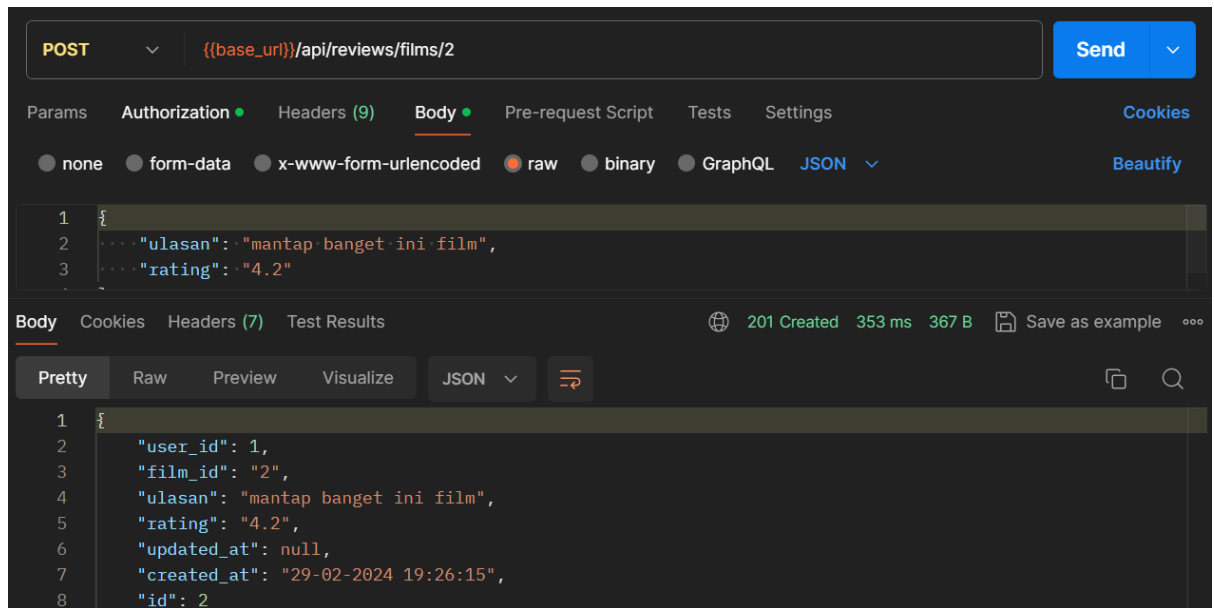


2.8.5 DELETE `/api/characters/{id}`: Menghapus karakter berdasarkan ID.

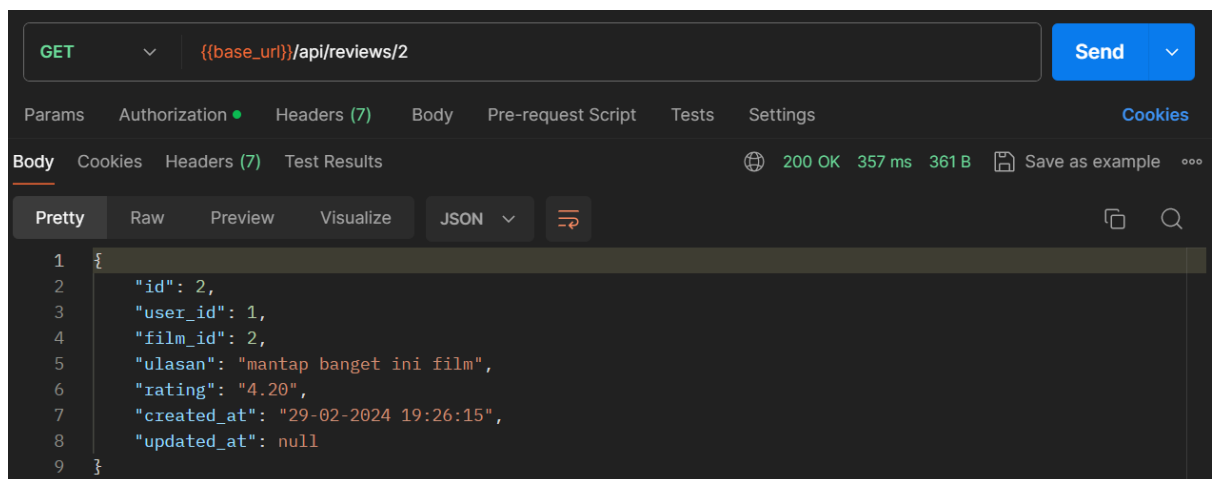


## 2.9 Endpoint Review

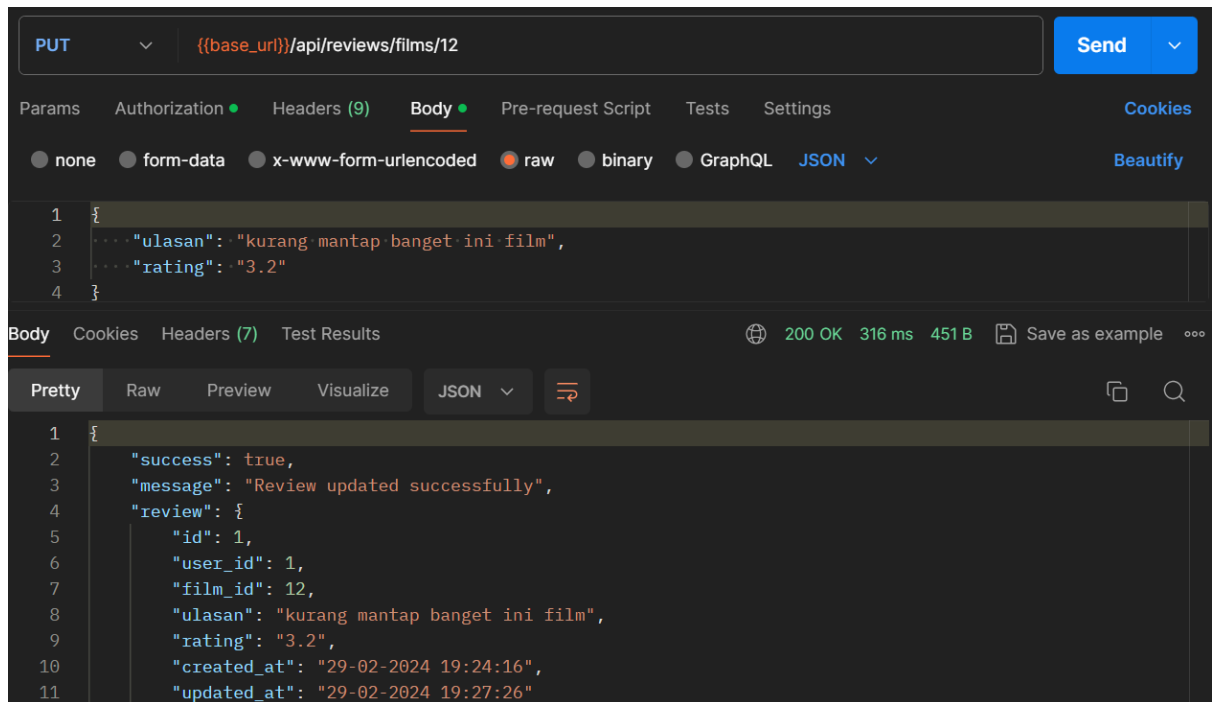
2.9.1 POST `/api/reviews/films/{idFilms}`: Menambahkan ulasan baru untuk film tertentu.



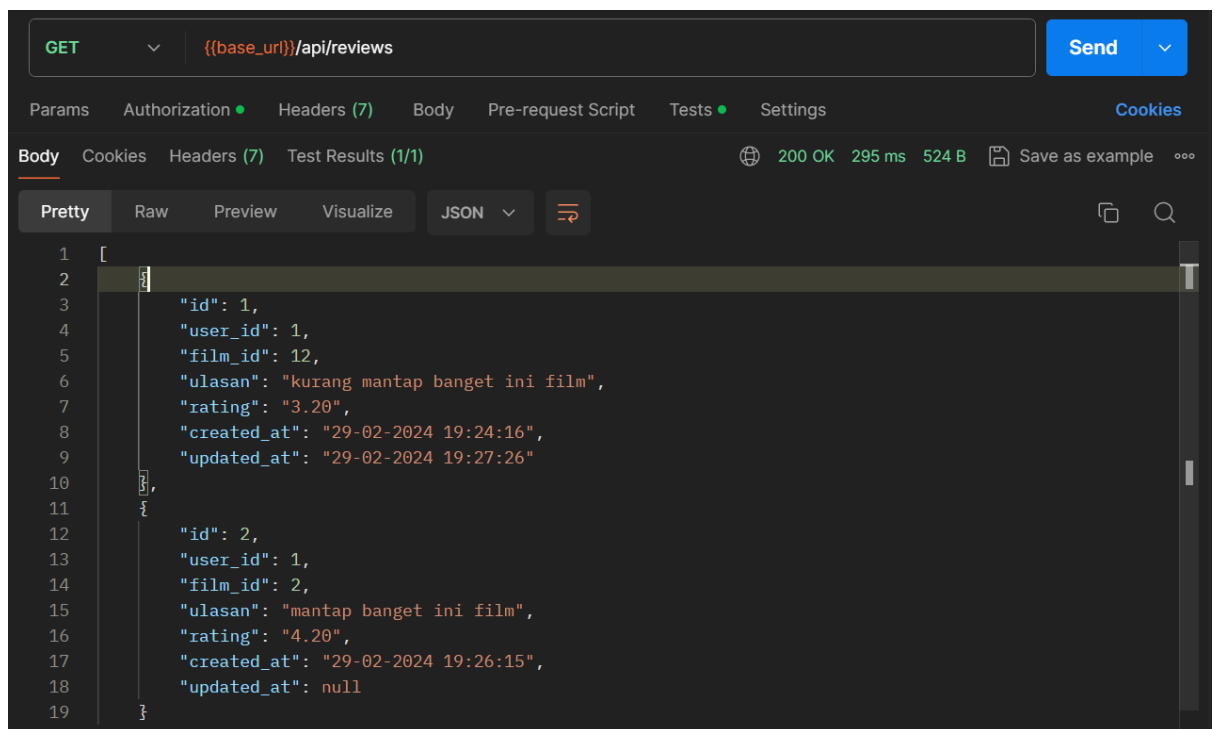
2.9.2 GET `/api/reviews/{id}`: Mendapatkan detail ulasan berdasarkan ID.



2.9.3 PUT `/api/reviews/films/{idFilms}`: Memperbarui ulasan untuk film tertentu.



#### 2.9.4 GET /api/reviews: Mendapatkan semua ulasan



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

