

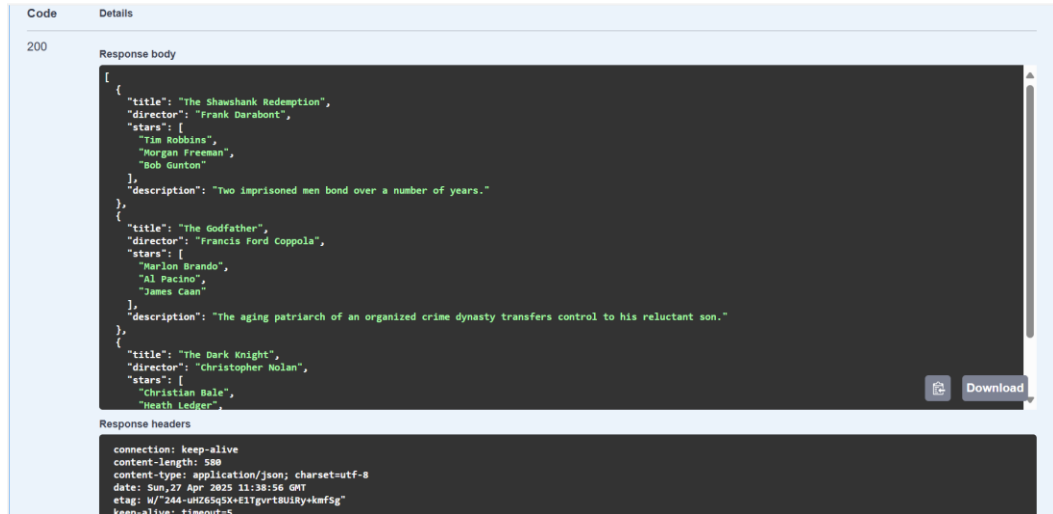
JM9_2311104002_Shilfi Habibah_SE0701

I. Link github

https://github.com/shilfihabibah/KPL_Shilfi-Habibah_2311104002_SE07-01/tree/master/09_API%20Design%20Costruction%20using%20swagger/JM9_2311104002

II. Screenshot hasil running

- a. Mencoba “GET /api/Movies” saat baru dijalankan yang mengeluarkan list film dari TOP 3 IMDB seperti pada tampilan berikut pada saat dicoba dengan menekan tombol “Try it out” dan tombol “Execute”

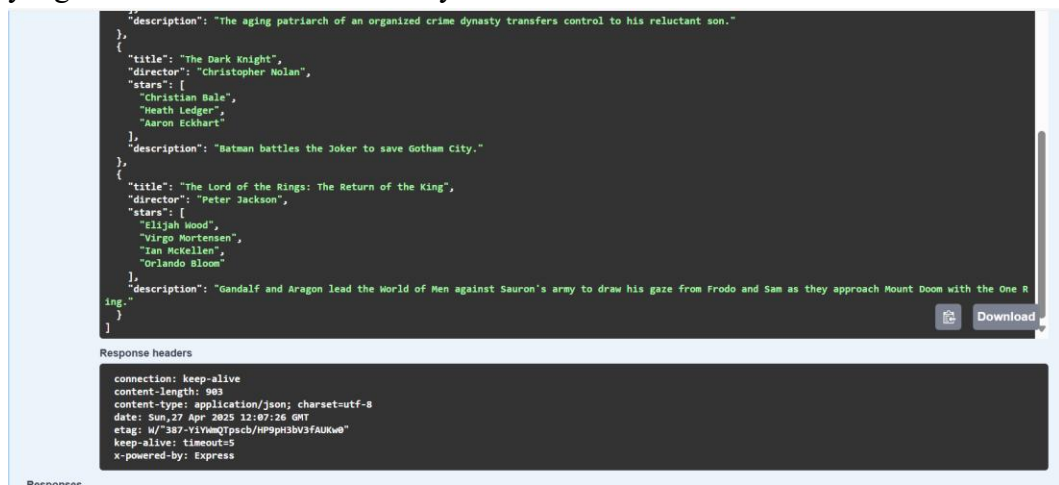


- b. Menambahkan Movie baru yaitu urutan ke-4 pada TOP IMDB list dengan memanggil API pada bagian “POST /api/Movies”

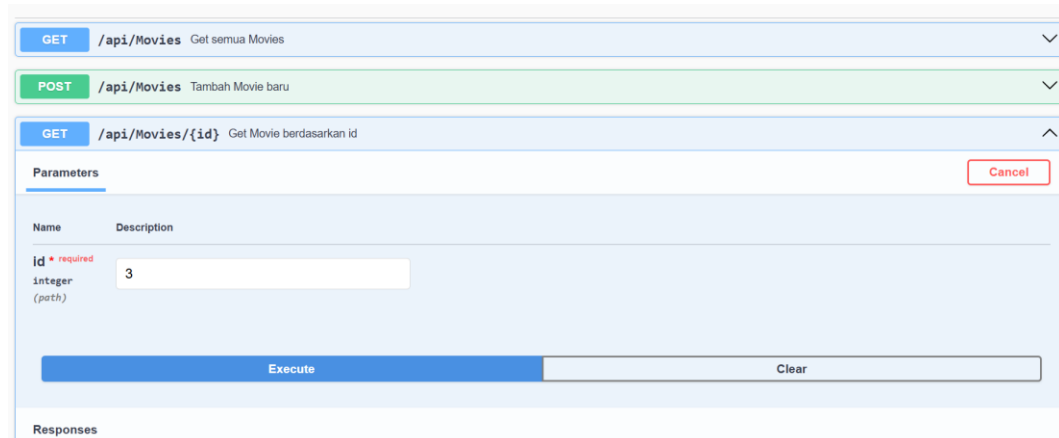




- c. Cek list/array dari semua Movie lagi dengan “GET /api/Movies”, pastikan Movie yang baru ditambahkan sebelumnya sudah ada:



- d. Mencoba meminta Movie dengan index 3, “GET /api/Movies/3” yang seharusnya mengembalikan Movie yang baru saja ditambah:



Curl

```
curl -X 'GET' \
  'http://localhost:5000/api/Movies/3' \
  -H 'accept: */*'
```

Request URL

http://localhost:5000/api/Movies/3

Server response

Code	Details
200	<p>Response body</p> <pre>{ "title": "The Lord of the Rings: The Return of the King", "director": "Peter Jackson", "stars": ["Elijah Wood", "Viggo Mortensen", "Ian McKellen", "Orlando Bloom"], "description": "Gandalf and Aragon lead the World of Men against Sauron's army to draw his gaze from Frodo and Sam as they approach Mount Doom with the One Ring." }</pre> <p>Response headers</p> <pre>connection: keep-alive content-length: 322 content-type: application/json; charset=utf-8 date: Sun, 27 Apr 2025 12:09:16 GMT etag: W/"142-d92634nQ6ey8CL9gy6jY6Alr90I" keep-alive: timeout=5 x-powered-by: Express</pre>

e. Menghapus objek Movie dengan index ke-1 dengan “DELETE /api/Movies/1”

Parameters Cancel

Name	Description
id required	1
integer	
(path)	

Execute Clear

Responses

Curl

```
curl -X 'DELETE' \
  'http://localhost:5000/api/Movies/1' \
  -H 'accept: */*'
```

Request URL

http://localhost:5000/api/Movies/1

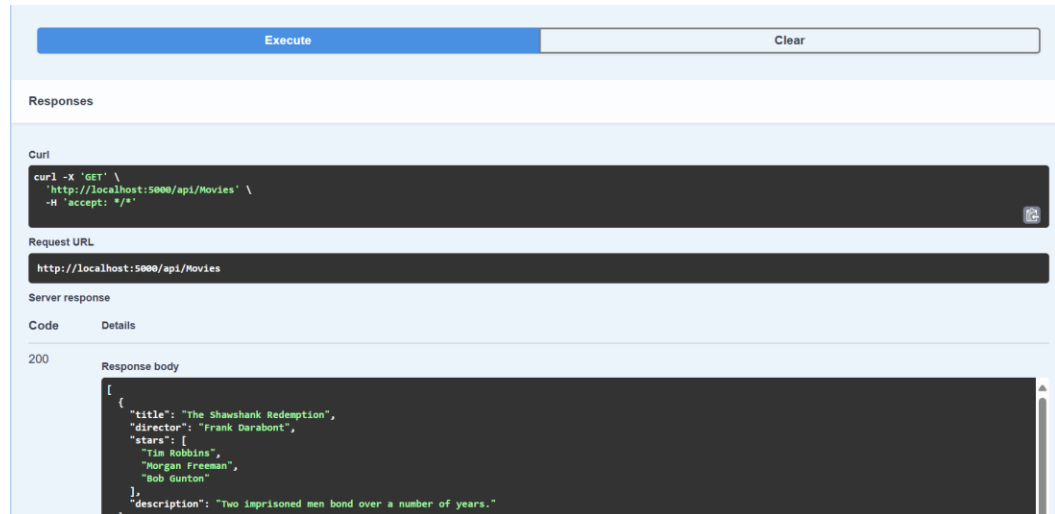
Server response

Code	Details
200	<p>Response body</p> <pre>Movie berhasil dihapus</pre> <p>Response headers</p> <pre>connection: keep-alive content-length: 22 content-type: text/html; charset=utf-8 date: Sun, 27 Apr 2025 12:11:27 GMT etag: W/"16-EzVR2w3Ac1X0x9pk8pi7LkSVmhw" keep-alive: timeout=5 x-powered-by: Express</pre>

Responses

Code	Description	Links
200	Movie berhasil dihapus	No links
404	Movie tidak ditemukan	No links

f. Cek list/array dari semua Movie sekali lagi dengan “GET /api/Movies”, film dengan ranking kedua “Godfather” sudah tidak ada di list:



III. Codingan

Code :

app.js

```
1  const express = require('express');
2  const swaggerUi = require('swagger-ui-express');
3  const movies = require('./movies');
4  const swaggerDocument = require('./swagger.json');
5
6  const app = express();
7  const port = 5000;
8
9  app.use(express.json());
10 app.use('/api-docs', swaggerUi.serve, swaggerUi.setup(swaggerDocument));
11
12 // GET /api/Movies
13 app.get('/api/Movies', (req, res) => {
14   res.json(movies);
15 });
16
17 // GET /api/Movies/:id
18 app.get('/api/Movies/:id', (req, res) => {
19   const id = parseInt(req.params.id);
20   if (id < 0 || id >= movies.length) {
21     res.status(404).send('Movie tidak ditemukan');
22   } else {
23     res.json(movies[id]);
24   }
25 });
```

```

27 // POST /api/Movies
28 app.post('/api/Movies', (req, res) => {
29   const { title, director, stars, description } = req.body;
30   movies.push({ title, director, stars, description });
31   res.status(201).send('Movie berhasil ditambahkan');
32 });
33
34 // DELETE /api/Movies/:id
35 app.delete('/api/Movies/:id', (req, res) => {
36   const id = parseInt(req.params.id);
37   if (id < 0 || id >= movies.length) {
38     res.status(404).send('Movie tidak ditemukan');
39   } else {
40     movies.splice(id, 1);
41     res.send('Movie berhasil dihapus');
42   }
43 });
44
45 app.listen(port, () => {
46   console.log(`Server berjalan di http://localhost:${port}`);
47   console.log(`Swagger UI: http://localhost:${port}/api-docs`);
48 });

```

movies.js

```

1 const movies = [
2   {
3     title: "The Shawshank Redemption",
4     director: "Frank Darabont",
5     stars: ["Tim Robbins", "Morgan Freeman", "Bob Gunton"],
6     description: "Two imprisoned men bond over a number of years."
7   },
8   {
9     title: "The Godfather",
10    director: "Francis Ford Coppola",
11    stars: ["Marlon Brando", "Al Pacino", "James Caan"],
12    description: "The aging patriarch of an organized crime dynasty transfers control to his reluctant son."
13  },
14  {
15    title: "The Dark Knight",
16    director: "Christopher Nolan",
17    stars: ["Christian Bale", "Heath Ledger", "Aaron Eckhart"],
18    description: "Batman battles the Joker to save Gotham City."
19  }
20 ];
21
22 module.exports = movies;

```

Penjelasan:

Kode ini membuat API sederhana menggunakan Node.js dan Express.js untuk mengelola data film.

- Menggunakan Express.js untuk membuat server dan menangani rute.
- Menyediakan endpoint:
 - GET /api/Movies: Mengambil semua data film.
 - GET /api/Movies/:id: Mengambil data film berdasarkan ID.
 - POST /api/Movies: Menambahkan film baru.
 - DELETE /api/Movies/:id: Menghapus film berdasarkan ID.
- Menggunakan middleware `express.json()` untuk memproses data JSON.
- Mengintegrasikan Swagger UI di `/api-docs` untuk dokumentasi API.
- Data film disimpan dalam array `movies` di dalam file terpisah (`movies.js`).
- Server berjalan di port 5000.