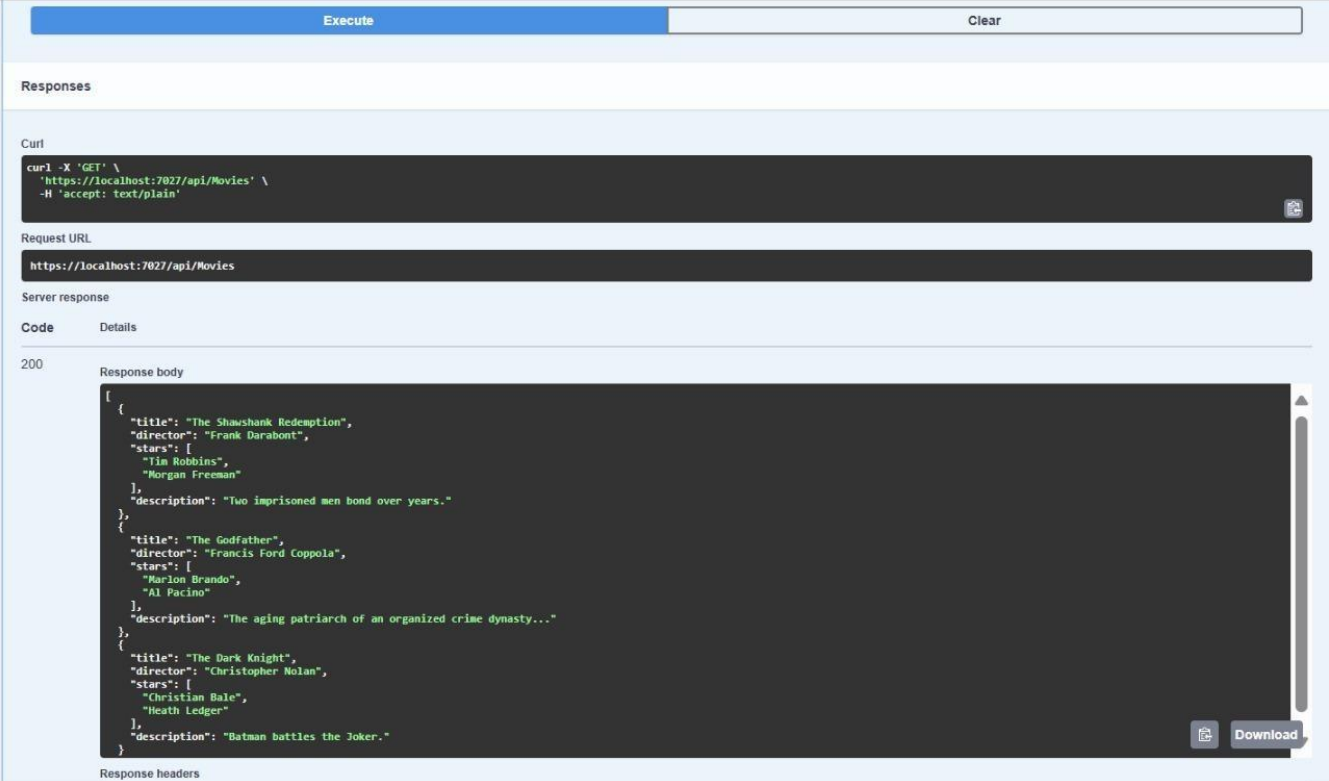


JURNAL MODUL 9

Muhammad Daniel Anugrah Pratama
2311104063
SE07-02

1. 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”



The screenshot displays a REST client interface with the following sections:

- Execute** and **Clear** buttons at the top.
- Responses** section.
- Curl** section showing the command: `curl -X 'GET' \ 'https://localhost:7027/api/Movies' \ -H 'accept: text/plain'`
- Request URL** section showing: `https://localhost:7027/api/Movies`
- Server response** section with a **Code** tab selected, showing a status of **200**.
- Response body** section showing a JSON array of three movie objects:

```
[
  {
    "title": "The Shawshank Redemption",
    "director": "Frank Darabont",
    "stars": [
      "Jim Robbins",
      "Morgan Freeman"
    ],
    "description": "Two imprisoned men bond over years."
  },
  {
    "title": "The Godfather",
    "director": "Francis Ford Coppola",
    "stars": [
      "Marlon Brando",
      "Al Pacino"
    ],
    "description": "The aging patriarch of an organized crime dynasty..."
  },
  {
    "title": "The Dark Knight",
    "director": "Christopher Nolan",
    "stars": [
      "Christian Bale",
      "Heath Ledger"
    ],
    "description": "Batman battles the Joker."
  }
]
```
- Response headers** section at the bottom.

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

The screenshot shows a REST client interface with a green header bar. The method is set to POST and the URL is /api/Movies. The 'Parameters' tab is selected, showing 'No parameters'. The 'Request body' tab is also selected, showing a JSON object for a movie titled '12 Angry Men'. The 'Execute' button is highlighted in blue. Below the interface, the 'Responses' section shows the raw curl command used for the request.

```
curl -X 'POST' \
  'https://localhost:7027/api/Movies' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "title": "12 Angry Men",
    "director": "Sidney Lumet",
    "stars": [
      "Henry Fonda",
      "Lee J. Cobb"
    ],
    "description": "A jury holdout attempts to prevent a miscarriage of justice."
  }'
```

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

The screenshot shows the same REST client interface, but now with a GET request to /api/Movies. The 'Execute' button is highlighted in blue. Below the interface, the 'Responses' section shows the raw curl command and the server response. The response is a JSON array containing three movie objects, including the one added in the previous step.

```
curl -X 'GET' \
  'https://localhost:7027/api/Movies' \
  -H 'accept: text/plain'
```

Request URL: https://localhost:7027/api/Movies

Server response: 200

```
{
  "title": "The Godfather",
  "director": "Francis Ford Coppola",
  "stars": [
    "Marlon Brando",
    "Al Pacino"
  ],
  "description": "The aging patriarch of an organized crime dynasty..."
},
{
  "title": "The Dark Knight",
  "director": "Christopher Nolan",
  "stars": [
    "Christian Bale",
    "Heath Ledger"
  ],
  "description": "Batman battles the Joker."
},
{
  "title": "12 Angry Men",
  "director": "Sidney Lumet",
  "stars": [
    "Henry Fonda",
    "Lee J. Cobb"
  ],
  "description": "A jury holdout attempts to prevent a miscarriage of justice."
}
]
```

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

The screenshot shows a REST client interface with a GET request to `/api/Movies/{id}`. The parameter `id` is set to 3. The response is a 200 status code with a JSON body containing movie details.

Parameters

Name	Description
<code>id</code> * required <code>integer(\$int32)</code> (path)	3

Execute **Clear**

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7027/api/Movies/3' \
  -H 'accept: text/plain'
```

Request URL

```
https://localhost:7027/api/Movies/3
```

Server response

Code	Details
200	<p>Response body</p> <pre>{ "title": "12 Angry Men", "director": "Sidney Lumet", "stars": ["Henry Fonda", "Lee J. Cobb"], "description": "A jury holdout attempts to prevent a miscarriage of justice." }</pre>

Response headers

Download

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

The screenshot shows a REST client interface with a DELETE request to `/api/Movies/{id}`. The parameter `id` is set to 1. The response is a 200 status code with empty headers.

Parameters

Name	Description
<code>id</code> * required <code>integer(\$int32)</code> (path)	1

Execute **Clear**

Responses

Curl

```
curl -X 'DELETE' \
  'https://localhost:7027/api/Movies/1' \
  -H 'accept: */*'
```

Request URL

```
https://localhost:7027/api/Movies/1
```

Server response

Code	Details
200	<p>Response headers</p> <pre>content-length: 0 date: Wed, 30 Apr 2025 13:33:20 GMT server: Kestrel</pre>

Responses

Code	Description	Links
200	OK	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:

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7027/api/Movies' \
  -H 'accept: text/plain'
```

Request URL

```
https://localhost:7027/api/Movies
```

Server response

Code

Details

200

Response body

```
{
  "title": "The Shawshank Redemption",
  "director": "Frank Darabont",
  "stars": [
    "Tim Robbins",
    "Morgan Freeman"
  ],
  "description": "Two imprisoned men bond over years."
},
{
  "title": "The Dark Knight",
  "director": "Christopher Nolan",
  "stars": [
    "Christian Bale",
    "Heath Ledger"
  ],
  "description": "Batman battles the Joker."
},
{
  "title": "12 Angry Men",
  "director": "Sidney Lumet",
  "stars": [
    "Henry Fonda",
    "Lee J. Cobb"
  ],
  "description": "A jury holdout attempts to prevent a miscarriage of justice."
}
```

Download

Response headers