

TUGAS JURNAL 09
KONSTRUKSI PERANGKAT LUNAK

API DESIGN DAN CONSTRUCTION USING SWAGGER



DISUSUN OLEH:
KHOLIFAH DINA
2211104004

SE 06 01

S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY
2025

1. MEMBUAT PROJECT WEB API

Buka IDE misalnya dengan Visual Studio

- Pilih New Project dan kemudian pilih ASP.NET Core Web API atau API (pastikan opsi 'Enable OpenAPI support' tercentang).
- Masukkan project baru dengan nama modul9_NIM.

The screenshot shows the 'Configure your new project' dialog in Visual Studio. The project type is 'ASP.NET Core Web API'. The project name is 'modul9_2211104004'. The location is 'D:\Kuliah\KPL_KHOLIFAH DINA_2211104004_SE0601\09_API_Design_dan_Construction_Using_Swag'. The solution is 'Create new solution'. The solution name is 'modul9_2211104004'. The checkbox 'Place solution and project in the same directory' is checked. The project will be created in the path 'D:\Kuliah\KPL_KHOLIFAH DINA_2211104004_SE0601\09_API_Design_dan_Construction_Using_Swagger\modul9_2211104004\'.

Configure your new project

ASP.NET Core Web API C# Linux macOS Windows API Cloud Service Web Web API

Project name

modul9_2211104004

Location

D:\Kuliah\KPL_KHOLIFAH DINA_2211104004_SE0601\09_API_Design_dan_Construction_Using_Swag ...

Solution

Create new solution

Solution name ⓘ

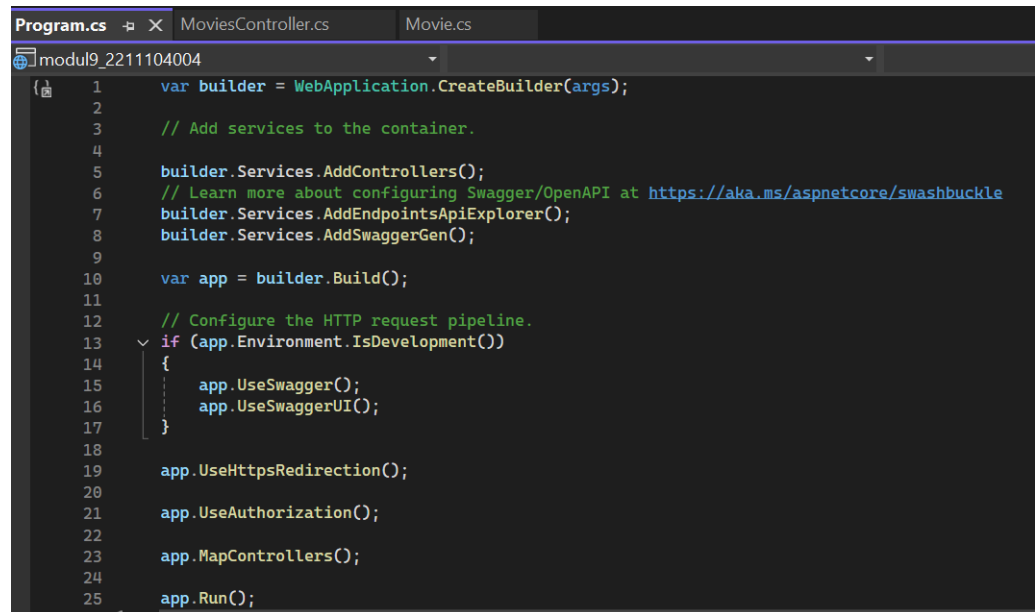
modul9_2211104004

☒ Place solution and project in the same directory

Project will be created in "D:\Kuliah\KPL_KHOLIFAH DINA_2211104004_SE0601\09_API_Design_dan_Construction_Using_Swagger\modul9_2211104004"

2. SOURCE CODE

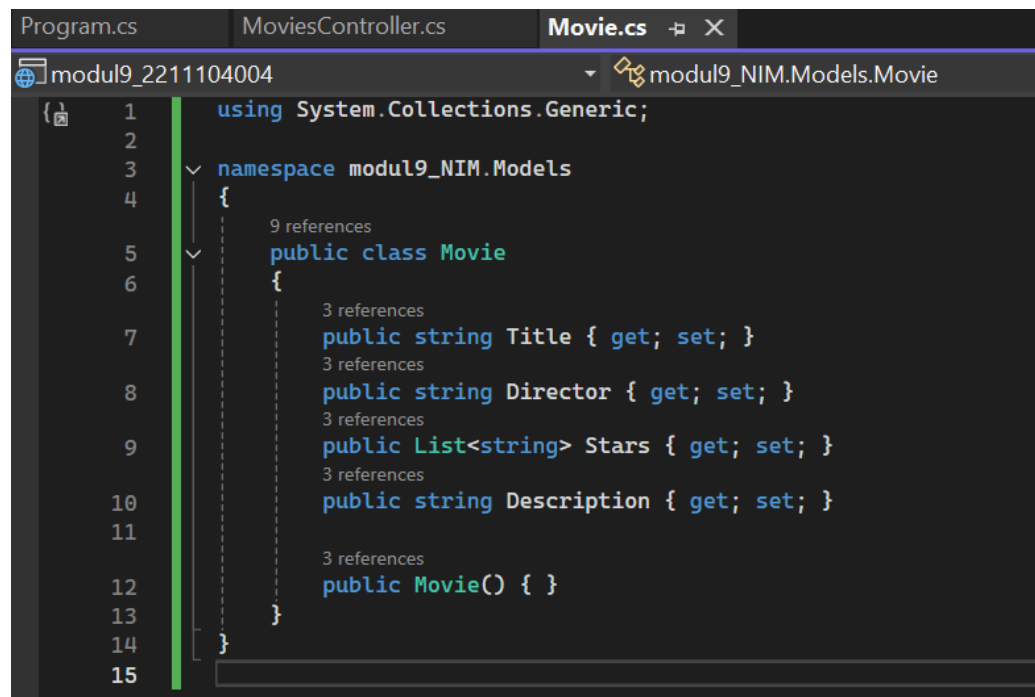
A. Program.cs



The screenshot shows the `Program.cs` file in a Visual Studio editor. The file is part of a project named `modul9_2211104004`. The code defines the `main` method for the application, which creates a `WebApplication` builder, adds services, builds the application, and runs it. The code is as follows:

```
1 var builder = WebApplication.CreateBuilder(args);
2
3 // Add services to the container.
4
5 builder.Services.AddControllers();
6 // Learn more about configuring Swagger/OpenAPI at https://aka.ms/aspnetcore/swashbuckle
7 builder.Services.AddEndpointsApiExplorer();
8 builder.Services.AddSwaggerGen();
9
10 var app = builder.Build();
11
12 // Configure the HTTP request pipeline.
13 if (app.Environment.IsDevelopment())
14 {
15     app.UseSwagger();
16     app.UseSwaggerUI();
17 }
18
19 app.UseHttpsRedirection();
20
21 app.UseAuthorization();
22
23 app.MapControllers();
24
25 app.Run();
```

B. Movie.cs



The screenshot shows the `Movie.cs` file in a Visual Studio editor. The file is part of a project named `modul9_2211104004`. The code defines a `Movie` class within the `modul9_NIM.Models` namespace. The code is as follows:

```
1 using System.Collections.Generic;
2
3 namespace modul9_NIM.Models
4 {
5     public class Movie
6     {
7         public string Title { get; set; }
8         public string Director { get; set; }
9         public List<string> Stars { get; set; }
10        public string Description { get; set; }
11
12        public Movie() { }
13    }
14 }
15
```

C. MovieController.cs

```
Program.cs  MoviesController.cs  Movie.cs
modul9_2211104004  modul9_NIM.Cor

1  using Microsoft.AspNetCore.Mvc;
2  using modul9_NIM.Models;
3  using System.Collections.Generic;
4
5  namespace modul9_NIM.Controllers
6  {
7      [Route("api/[controller]")]
8      [ApiController]
9      public class MoviesController : ControllerBase
10     {
11         private static List<Movie> Movies = new List<Movie>
12         {
13             new Movie
14             {
15                 Title = "The Shawshank Redemption",
16                 Director = "Frank Darabont",
17                 Stars = new List<string> { "Tim Robbins", "Morgan Freeman", "Bob Gunton" },
18                 Description = "Two imprisoned men bond over a number of years..."
19             },
20             new Movie
21             {
22                 Title = "The Godfather",
23                 Director = "Francis Ford Coppola",
24                 Stars = new List<string> { "Marlon Brando", "Al Pacino", "James Caan" },
25                 Description = "The aging patriarch of an organized crime dynasty..."
26             },
27             new Movie
28             {
29                 Title = "The Dark Knight",
30                 Director = "Christopher Nolan",
31                 Stars = new List<string> { "Christian Bale", "Heath Ledger", "Aaron Eckhart" },
32                 Description = "When the menace known as the Joker wreaks havoc..."
33             }
34         };
35
36         [HttpGet]
37         public ActionResult<List<Movie>> GetAll() => Movies;
38
39         [HttpGet("{id}")]
40         public ActionResult<Movie> Get(int id)
41         {
42             if (id < 0 || id >= Movies.Count) return NotFound();
43             return Movies[id];
44         }
45
46         [HttpPost]
47         public ActionResult Add(Movie movie)
48         {
49             Movies.Add(movie);
50             return Ok();
51         }
52
53         [HttpDelete("{id}")]
54         public ActionResult Delete(int id)
55         {
56             if (id < 0 || id >= Movies.Count) return NotFound();
57             Movies.RemoveAt(id);
58             return Ok();
59         }
60     }
61 }
62
```

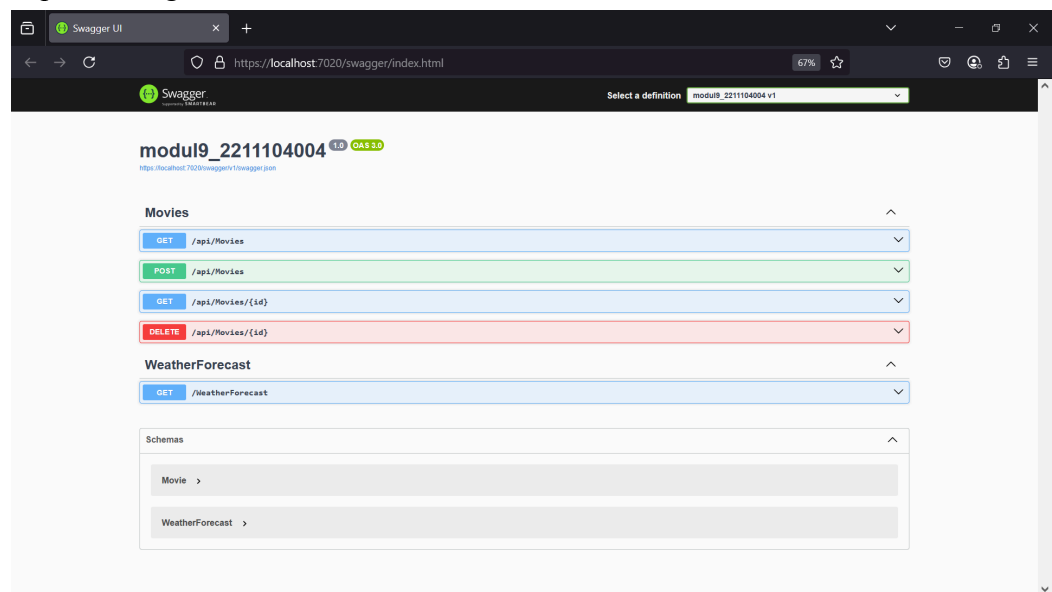
3. IMPLEMENTASI WEB API

Dari master/main branch dan class utama, buatlah program/aplikasi web API dari spesifikasi sebagai berikut ini:

A. API yang dibuat menggunakan data dari kelas Movie.

```
Movie {
  title
  director
  stars
  description
}
```

B. Gunakan teknologi API sehingga program tersebut dapat menerima HTTP request sebagai berikut:

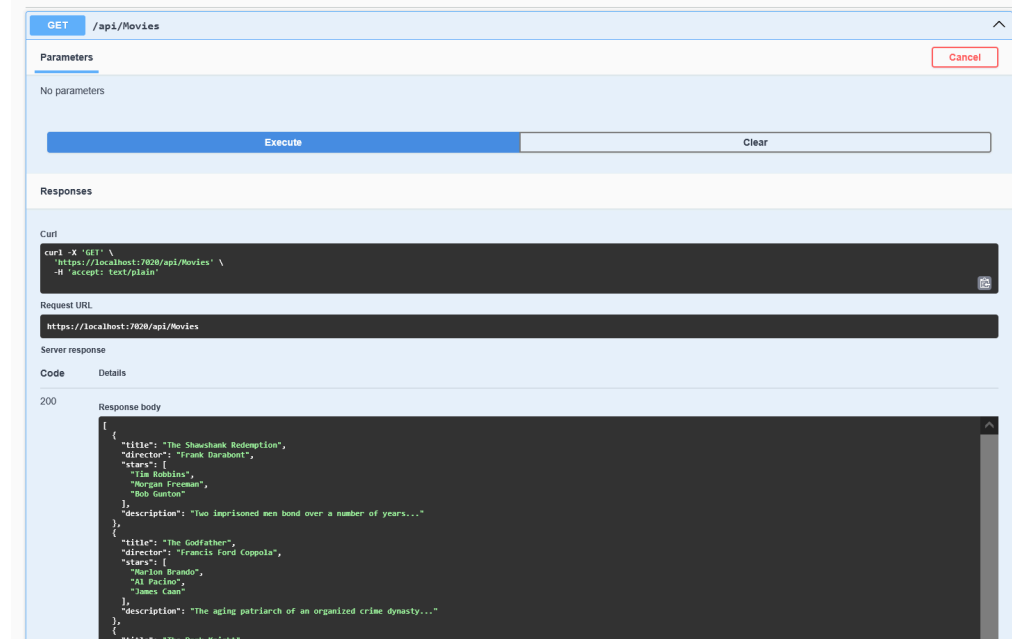


- i. GET /api/Movies: mengembalikan output berupa list/array dari semua objek Movies
- ii. GET /api/Movies/{id}: mengembalikan output berupa objek Movie untuk index “id”
- iii. POST /api/Movies: menambahkan objek Movie baru
- iv. DELETE /api/Movies/{id}: menghapus objek Movie pada index “id”

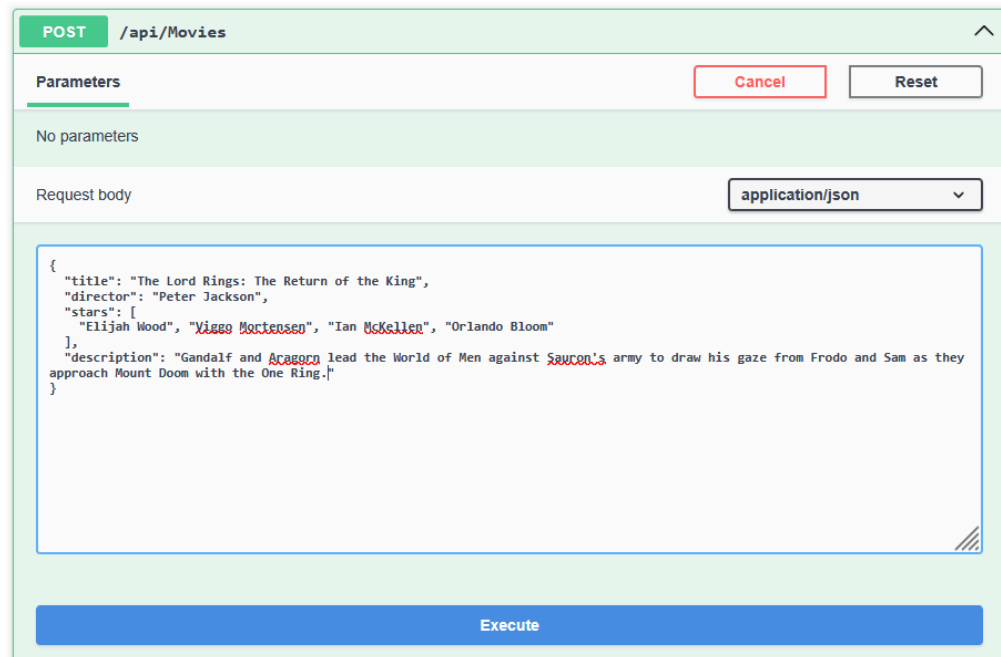
4. MENDEMONSTRASI WEB API

Beberapa skenario yang harus dicoba untuk memastikan jika program telah berjalan dengan baik. Buatlah dokumen yang berisi semua screenshot dari hasil uji coba skenario yang disebutkan pada list berikut ini:

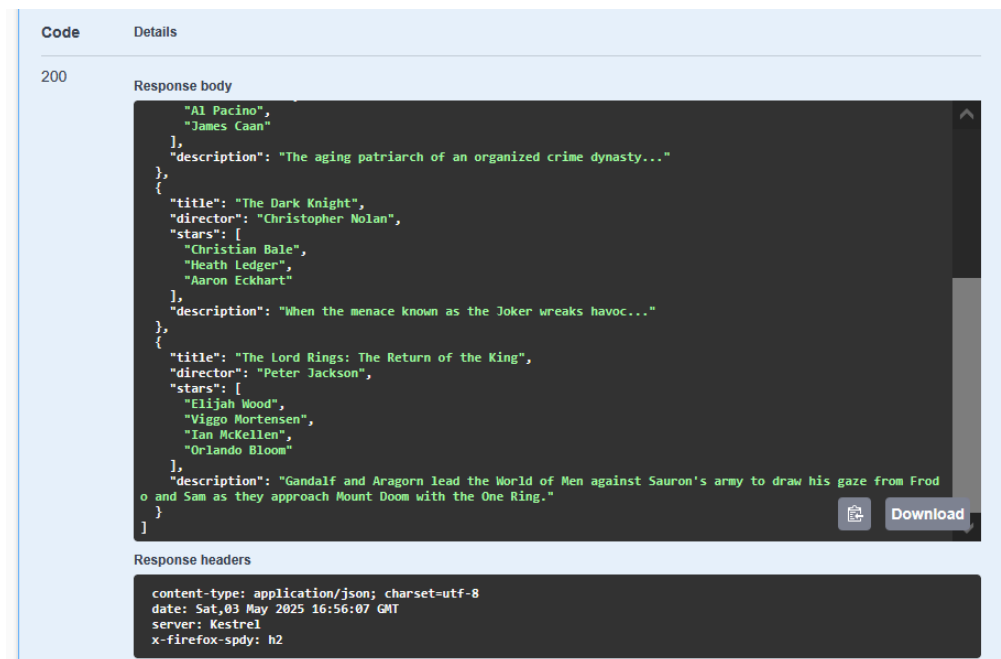
- A. Mencoba “GET /api/Movies” saat baru dijalankan yang mengeluarkan list film dari TOP 3 IMDB seperti pada tampilan berikut pada saat dicoba dengan menekan tombol “Try it out” dan tombol “Execute”



- 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



Code Details

200

Response body

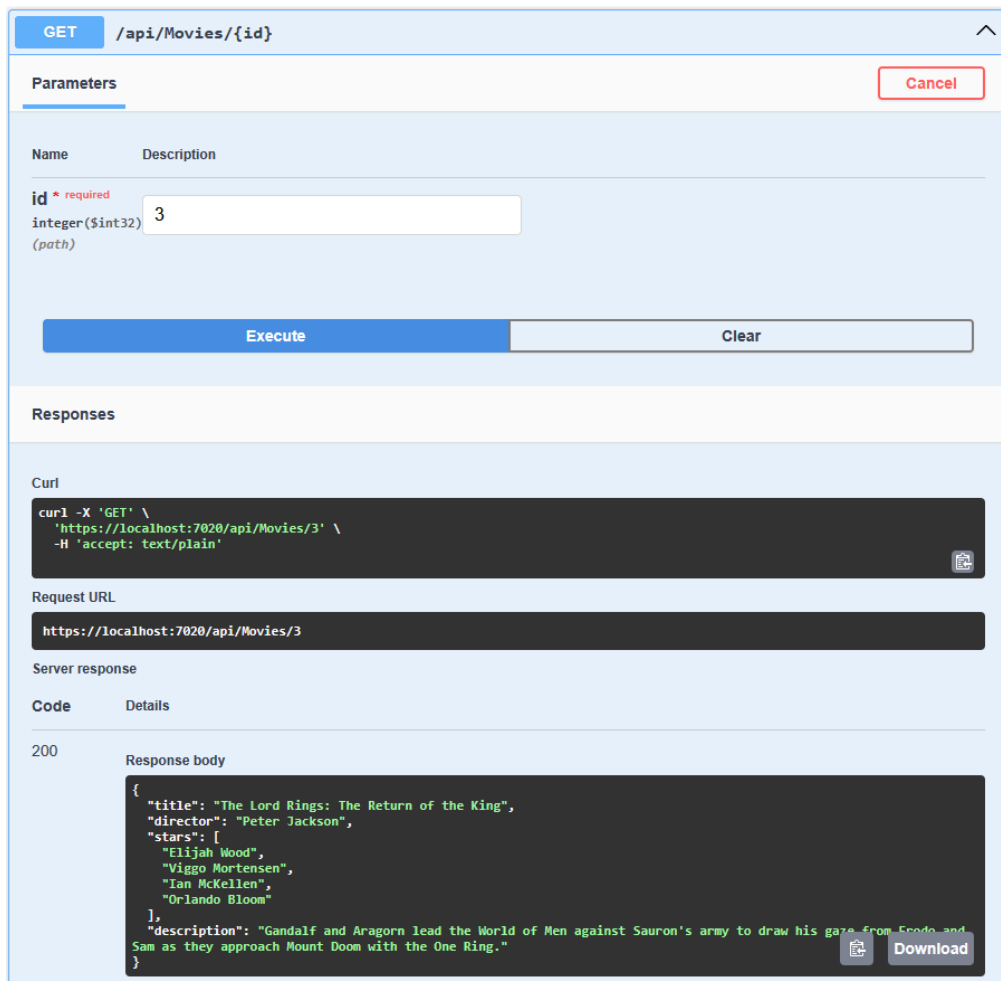
```
{
  "title": "The Dark Knight",
  "director": "Christopher Nolan",
  "stars": [
    "Christian Bale",
    "Heath Ledger",
    "Aaron Eckhart"
  ],
  "description": "When the menace known as the Joker wreaks havoc..."
},
{
  "title": "The Lord Rings: The Return of the King",
  "director": "Peter Jackson",
  "stars": [
    "Elijah Wood",
    "Viggo Mortensen",
    "Ian McKellen",
    "Orlando Bloom"
  ],
  "description": "Gandalf and Aragorn 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."
}
]
```

Response headers

```
content-type: application/json; charset=utf-8
date: Sat, 03 May 2025 16:56:07 GMT
server: Kestrel
x-firefox-spy: h2
```

Download

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



GET /api/Movies/{id}

Parameters

Cancel

Name	Description
id * required	
integer(\$int32)	3
(path)	

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7020/api/Movies/3' \
  -H 'accept: text/plain'
```

Request URL

```
https://localhost:7020/api/Movies/3
```

Server response

Code Details

200

Response body

```
{
  "title": "The Lord Rings: The Return of the King",
  "director": "Peter Jackson",
  "stars": [
    "Elijah Wood",
    "Viggo Mortensen",
    "Ian McKellen",
    "Orlando Bloom"
  ],
  "description": "Gandalf and Aragorn 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."
}
```

Download

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

The screenshot shows a REST client interface with a red header bar. The method is **DELETE** and the URL is `/api/Movies/{id}`. Below the header, there is a **Parameters** section with a table:

Name	Description
id * required	
integer(\$int32)	1
(path)	

At the bottom of the parameters section, there are two buttons: **Execute** (blue) and **Clear** (light blue). Below the parameters section is a **Responses** section, which is currently empty.

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

The screenshot shows a REST client interface with a blue header bar. The method is **GET** and the URL is `/api/Movies`. Below the header, there is a **Parameters** section with the text "No parameters". At the bottom of the parameters section, there are two buttons: **Execute** (blue) and **Clear** (light blue). Below the parameters section is a **Responses** section. It contains a **Curl** section with the following command:

```
curl -X 'GET' \
  'https://localhost:7020/api/Movies' \
  -H 'accept: text/plain'
```

Below the curl section is a **Request URL** section with the following URL:

```
https://localhost:7020/api/Movies
```

Below the request URL section is a **Server response** section. It contains a **Code** section with the value **200** and a **Details** section with the **Response body**:

```
[
  {
    "title": "The Shawshank Redemption",
    "director": "Frank Darabont",
    "stars": [
      "Tim Robbins",
      "Morgan Freeman",
      "Bob Gunton"
    ],
    "description": "Two imprisoned men bond over a number of years..."
  },
  {
    "title": "The Lord Rings: The Return of the King",
    "director": "Peter Jackson",
    "stars": [
      "Elijah Wood",
      "Viggo Mortensen",
      "Ian McKellen",
      "Orlando Bloom"
    ],
    "description": "Gandalf and Aragorn 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."
  }
]
```