14 April 2022

Tugas Jurnal modul 8

1. membuat project Web API

Berhubung cara membuat project web api berbeda-beda untuk setiap bahasa pemrograman, langkah-langkah berikut hanya berlaku apabila dilakukan dengan menggunakan .NET dan Visual Studio. Untuk IDE dan bahasa pemrograman lain, yang terpenting adalah nama project yang dibuat yaitu "modul8_NIM".

A. Buka visual studio yang sudah terinstall dengan ASP.NET dan .NET 5.0 SDK atau setelahnya
B. Pilih New Project dan kemudian pilih ASP.NET Core Web API atau API (pastikan opsi 'Enable OpenAPI support' tercentang).
C. Pastikan untuk memilih .NET versi 5.0 atau yang lebih baru.
D. Masukkan nama projek "modul8_NIM".
E. Langkah-langkah yang disertai gambar dapat dilihat pada link berikut ini (cukup dilihat pada bagian "Create a Web API project"):

   https://docs.microsoft.com/en-us/aspnet/core/tutorials/min-web-api?view=aspnetcore-6.0&tabs=visual-studio

F. Setelah project tersebut selesai dibuat, coba run programnya, dan tunggu sampai program selesai di-compile.

2. Melakukan git commit pada project yang dibuat

Task atau langkah-langkah yang perlu dikerjakan adalah sebagai berikut:

A. Buatlah github public repository kosong (pastikan bagian "*Initialize this repository with*" tidak ada yang dicentang pada saat membuat repository baru) melalui https://github.com/
B. Melakukan inisialisasi git repository di folder project yang dibuat.
C. Pastikan untuk menambahkan file ".gitignore" baik manual atau dengan menggunakan visual studio/IDE. Untuk project dengan C# dapat melihat referensi file ".gitignore" pada link berikut ini:

   https://github.com/github/gitignore/blob/main/VisualStudio.gitignore

D. Membuat commit untuk versi pertama dari project yang dibuat dengan pesan commit bebas.
E. Melakukan git push ke github repo.

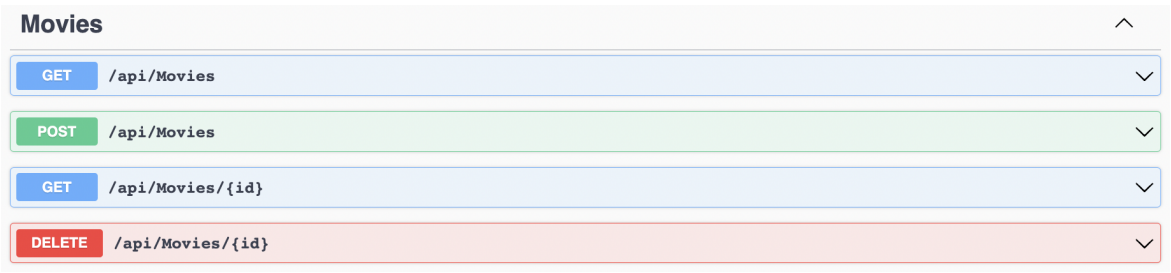
3. Implementasi Web API

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 : string + Director : string + Stars : List<string> + Description: string
+ Movie()

B. API yang dibuat mempunyai lokasi sebagai berikut '**/api/Movies**, URL domain boleh dari port mana saja (port bebas). Dengan menggunakan swagger API tersebut dapat menerima RESTful API dengan metoda sebagai berikut (halaman swagger dapat diakses pada https://localhost:<PORT>/swagger/index.html):



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"

C. Secara default, program yang dibuat memiliki list film yang berasal dari TOP 3 film IMDB dari link: https://www.imdb.com/search/title/?groups=top_100&sort=user_rating,desc

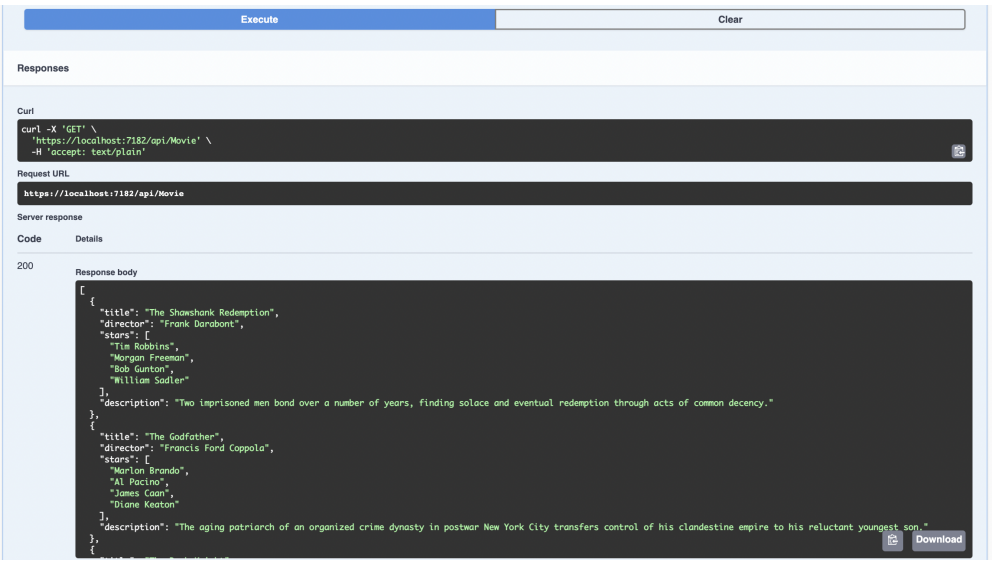
D. Impementasi yang dibuat tidak menggunakan database, cukup disimpan sebagai suatu variable, dan gunakan "static" di variable tersebut yang menyimpan list/array dari objek-objek Movie.

E. Dalam pembuatan program/aplikasi ini, anda dapat mengasumsikan bahwa input dari user selalu benar dan sesuai dengan tipe data yang diharapkan.

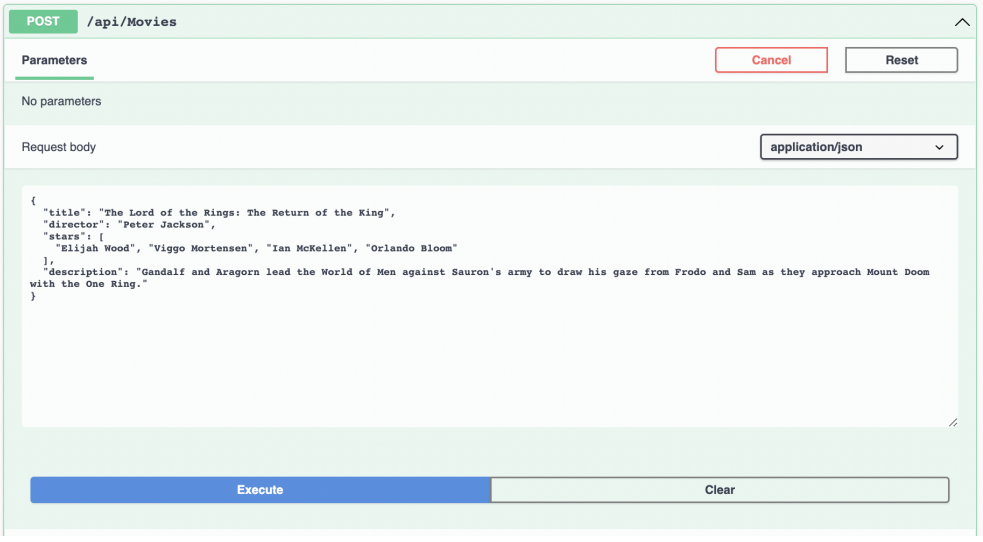
4. Mendemonstrasi Web API

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

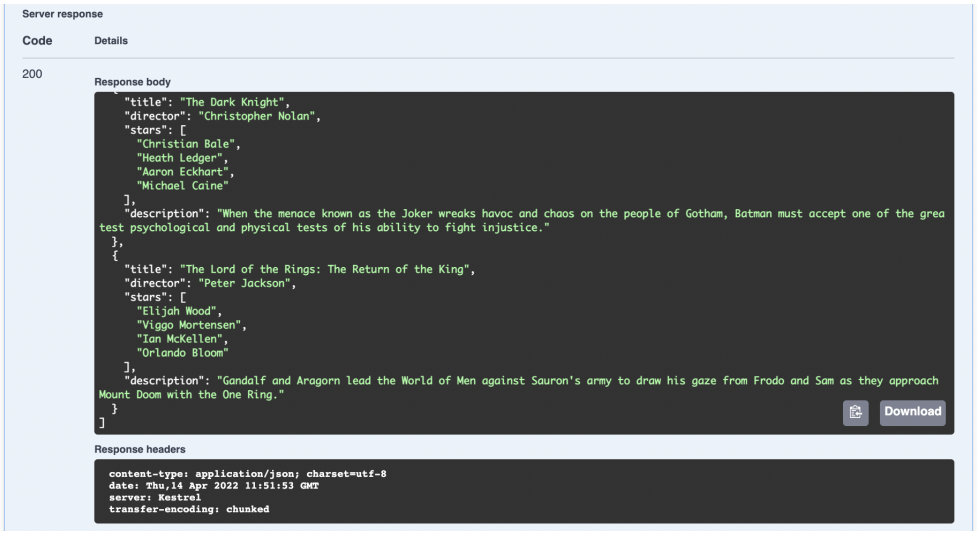
A. Mencoba "GET /api/Movies" saat baru dijalankan yang mengeluarkan list film dari TOP 3 IMDB seperti pada tampilan berikut pada saat dicoba dengan menekan tombol "Try it out" dan tombol "Execute"



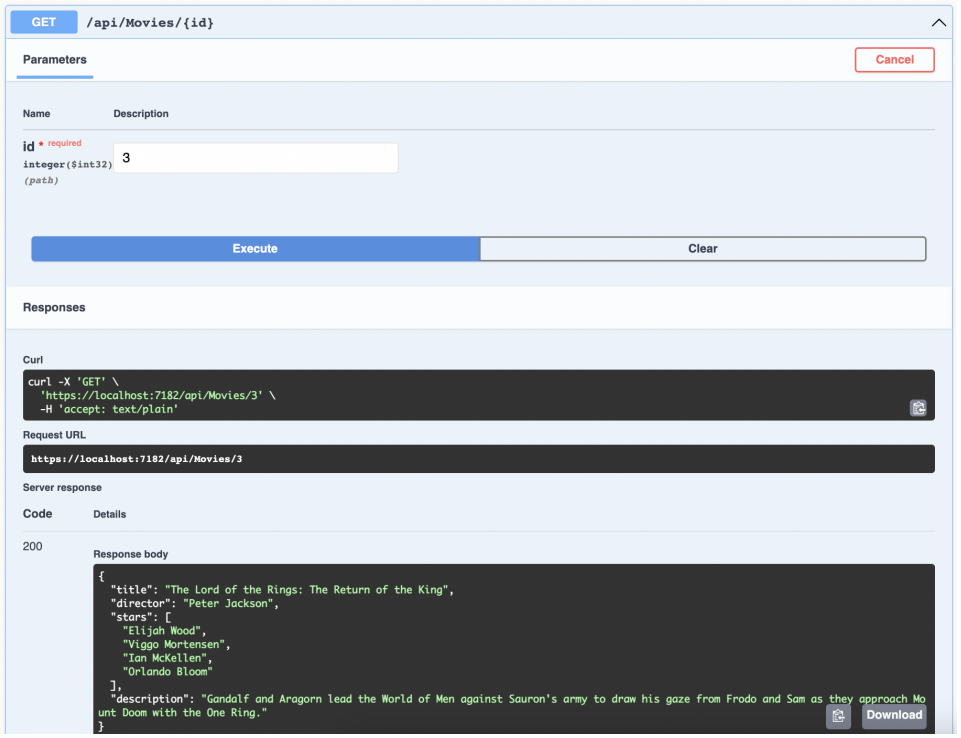
B. Menambahkan Movie baru yaitu urutan ke-4 pada TOP IMDB list dengan memanggil API pada bagian “POST /api/Movies”



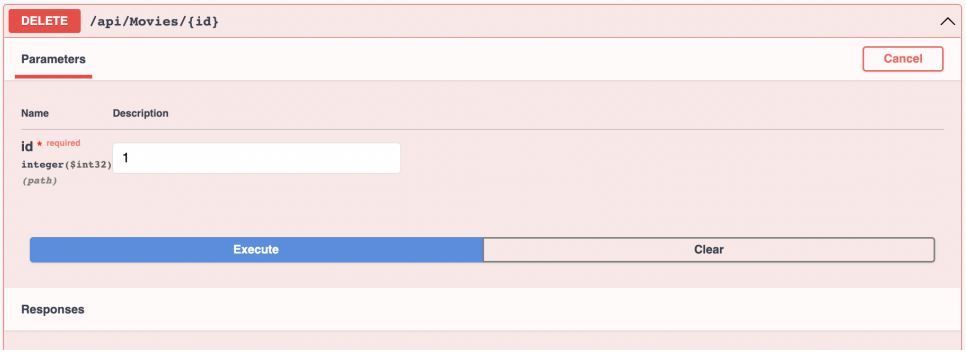
C. Cek list/array dari semua Movie lagi dengan “GET /api/Movies”, pastikan Movie yang baru ditambahkan sebelumnya sudah ada:



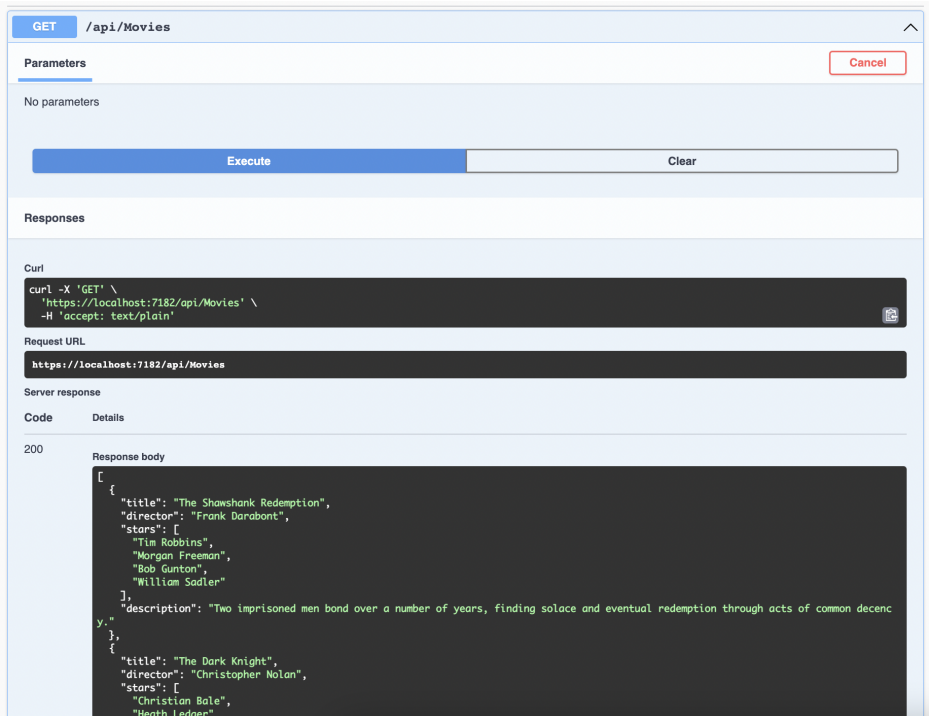
D. Mencoba meminta Movie dengan index 3, “GET /api/Movies/3” yang seharusnya mengembalikan Movie yang baru saja ditambah:



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



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



5. Melakukan Commit

Pada branch master/main:

A. Lakukan commit dengan pesan “menambahkan API Movies”.

B. Lakukan push ke github ke branch yang dibuat di bagian sebelumnya.

6. Pengumpulan file/tugas JUrnal

Sebelum pengumpulan, praktikan wajib menunjukkan hasil run via share screen ke asprak. Kumpulkan semua file berikut dalam bentuk file zip/rar/7zip:

A. Source code dari project yang dibuat

B. File docx/pdf yang berisi:

i. Link github repository kelompok

ii. Screenshot hasil run (hasil console output dari proses run sesuai bagian 4)

iii. Penjelasan singkat dari kode implementasi yang dibuat (beserta screenshot dari potongan source code yang dijelaskan).

Catatan: Tidak ada file docx/pdf (screenshot dan penjelasan) yang dikumpulkan DAN juga tidak melakukan demo/menunjukkan hasil run ke asprak maka nilai jurnal akan 0

KOMponen penilaian

A. Penggunaan Git dan Github **[10 pts]**

B. Implementasi Web API **[45 pts]**

C. Demo program/aplikasi **[30 pts]**

D. Laporan jurnal **[15 pts]**