TECHNOLOGIE KOMPONENTOWE GRUPA 2 - 20.10.2017

Sprawdzenie środowiska pracy:

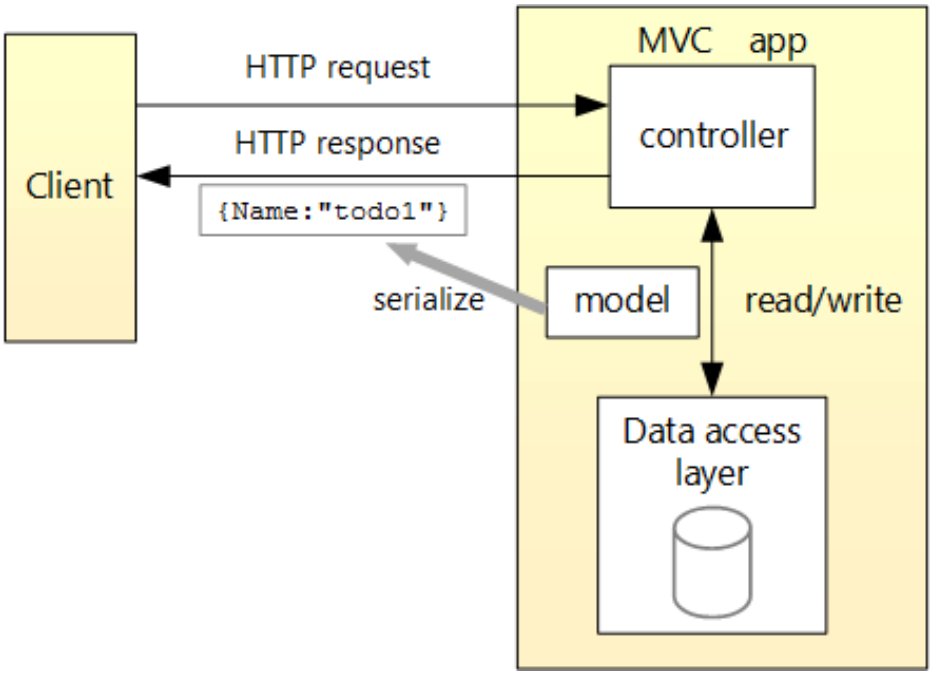
a. git --version
b. dotnet --version
c. Visual Studio Code
d. Postman - doinstalowanie (do testów API)

0. Fork projektu:
https://gitlab.com/lukasikpiotr/WebAPI2
Po ukończeniu zadania proszę o wykonanie Merge Requesta z opisem stworzonej aplikacji. (Opis ma zawierać sposób korzystania z aplikacji oraz przygotowane testy)

I. Zadanie:

Stwórz aplikacje opartą o interfejs komunikacyjny Web API (format JSON).

- Temat aplikacji dowolny z wyłączeniem aplikacji realizującej listę zadań tzw. ToDo.
- Model powinien posiadać przynajmniej 3 własności (property).
- Jako warstwy „Data access layer" proszę użyć "Entity Framework Core InMemory". Microsoft.EntityFrameworkCore.InMemory
  (Paczki nugetowe w VS Code można dodawać poprzez wpis do pliku: nazwa_projektu.csproj
  jako: PackageReference)




Wytyczne:

1. Stworzenie projektu Web API (korzystając z .NET Core): (0.5 pkt)
   a. Użycie odpowiedniego polecenia (dotnet new …)

2. Stworzenie odpowiedniej struktury projektu: (0.5 pkt)
    a. Stworzenie solucji
    b. Rozdzielenie głównego projektu od projektu testów
    c. Rozdzielenie kontrolerów od modeli

3. Aplikacja ma obsługiwać następujące akcje: (5pkt)

| API | Opis | Request body | Response body (odpowiedź serwera) |
|---|---|---|---|
| 1. GET /api/*model*/{id} | Pobierz wskazany model według ID | Brak | Obiekt w formacie JSON |
| 2. POST /api/*model* | Dodaj nowy obiekt | Obiekt w formacie JSON | Obiekt w formacie JSON |
| 3. DELETE /api/*model*/{id} | Usuń wskazany obiekt | Brak | Brak |

*model -> zastąp nazwą swojego modelu
(implementacja z wykorzystaniem async / await => 2 pkt)

4. Akcje powinny zwracać odpowiednie kody odpowiedzi HTTP (według przyjętych standardów https://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html): (1,5pkt)

    a. GET /api/*model*/{id} - kod znalezienia lub nieznalezienia obiektu
    b. POST /api/*model* - kod utworzenia obiekty lub błędu tworzenia
    c. DELETE /api/*model*/{id} - kod usunięcia obiektu

5. Stwórz testy integracyjne (XUnit): (2 pkt)
    a. sporządzenie dla każdej akcji przynajmniej jednego testu integracyjnego sprawdzającego:
        i. kod odpowiedzi
        ii. zawartość odpowiedzi

6. Sporządzenie opisu wykonanego zadania. (0.5 pkt)
    a. W opisie Merge Requesta proszę o przygotowanie opisu stworzonej przez Państwa aplikacji. Opis powinien zawierać przedstawienie wybranej dziedziny, możliwość API oraz przygotowane testy.

II. Ważne pojęcia z ćwiczeń
    b. CRUD
    c. Kody zwracane przez Web API
    d. Integration testing vs Unit testing
    e. Async / await
    f. Task
    g. IActionResult
    h. In memory database
    i. Continuous Integration