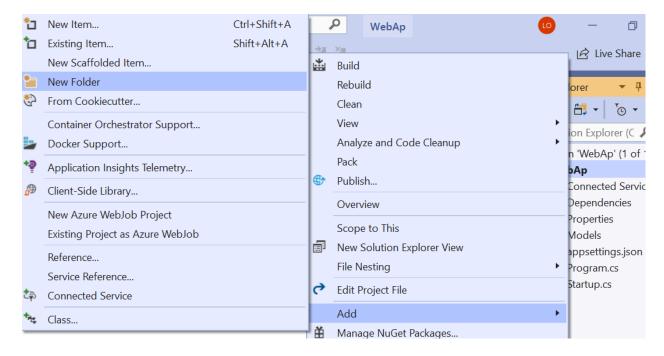
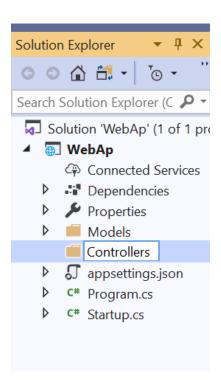
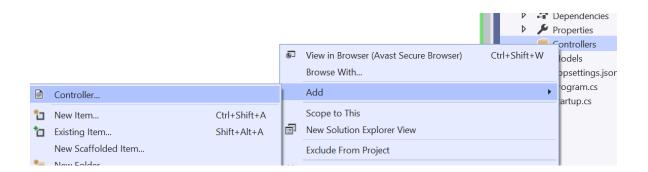
## 1. Створення контролерів

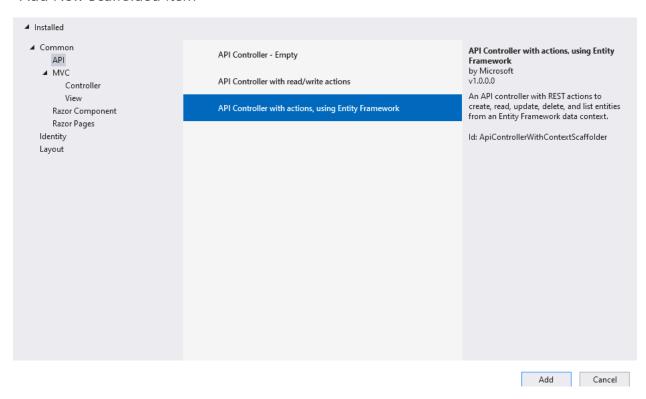
Далі додамо в проєкт нову папку Controllers, а в ній створимо новий арі-контролер для кожного класу моделі. Для цього при додаванні нового елемента в проект можна використовувати шаблон API Controller Class:







## Add New Scaffolded Item



# Add API Controller with actions, using Entity Framework

Model class	Category (LibraryAPIWebApp.Models)		•
Data context class	LibrariAPIContext (LibraryAPIWebApp.Models)		+
Controller name	CategoriesController		
		Add	Cancel

```
CategoriesController.cs → X Program.cs
                                                                     NuGet - Solution
                                                                                     Author.cs
                                  appsettings.json
                                                  LibrariAPIContext.cs
■ LibraryAPIWebApp
                                ⊕ ⊕ ♠ ₽ ७ - ₽ 目 ₽
            #nullable disable
           □using System;
                                                                                                          Solution 'LibraryAPIWebApp' (1 of
            using System.Collections.Generic;

▲ ■ LibraryAPIWebApp

            using System.Linq;
                                                                                                           Connected Services
            using System.Threading.Tasks;
                                                                                                           ▶ ₽₽ Dependencies
            using Microsoft.AspNetCore.Http
                                                                                                             Properties
            using Microsoft.AspNetCore.Mvc;
                                                                                                             mwwroot)
            using Microsoft.EntityFrameworkCore;
                                                                                                             css 🗀 css
           using LibraryAPIWebApp.Models;
                                                                                                                🛅 js
     10
                                                                                                                ₽⊓ lib
          namespace LibraryAPIWebApp.Controllers
     11
                                                                                                                favicon.ico
     12
                                                                                                             Controllers
                [Route("api/[controller]")]
     13
                [ApiController]
     14
                                                                                                             Models 🗎
     15
                public class CategoriesController: ControllerBase
                                                                                                              C# Author.cs
     16
                                                                                                              C# AuthorBook.cs
                    private readonly LibrariAPIContext _context;
     17
                                                                                                              D C# Book.cs
                                                                                                             D C# Category.cs
                                                                                                             D C# LibrariAPIContext.cs
                    public CategoriesController(LibrariAPIContext context)
     19
                                                                                                            Pages
     20
                                                                                                             ▶ ☐ Shared
     21
                        _context = context;
                                                                                                                _ViewImports.cshtml
     22
                                                                                                                _ViewStart.cshtml
     23
                    // GET: api/Categories
     2/1
                                                                                                             25
                    [HttpGet]
                                                                                                               Privacy.cshtml
     26
                    public async Task<ActionResult<IEnumerable<Category>>> GetCategories()
                                                                                                             appsettings.json
                                                                                                           D C# Program.cs
     28
                        return await _context.Categories.ToListAsync();
     29
     30
                                                                                                         Solution Explorer Git Changes
                    // GET: api/Categories/5
     31
                    [HttpGet("{id}")]
                                                                                                        Properties
     32
                                                                                                         CategoriesController.cs File Properties
                    O references
```

До контролера застосовується атрибут [ApiController], який дозволяє використовувати низку додаткових можливостей, зокрема, в плані прив'язки моделі. Також до контролера застосовується атрибут маршрутизації, який вказує, як контролер буде зіставлятися з запитами.

У конструкторі контролера отримуємо контекст даних і використовуємо його для операцій з даними. Також в конструкторі контролера додаємо початкові дані.

Контролер API призначений переважно для обробки запитів протоколу HTTP: Get, Post, Put, Delete, Patch, Head, Options. В даному випадку для кожного типу запитів в контролері визначено свої методи. Так, метод Get () обробляє запити типу GET і повертає колекцію об'єктів з БД.

Якщо запит Get містить параметр іd (ідентифікатор об'єкта), то він обробляється іншим методом - Get (int id), який повертає об'єкт по переданому іd.

Запити типу Post обробляються методом Post, який отримує з тіла запиту відправлені дані і додає їх в базу даних.

Метод Put обробляє запити типу Put - отримує дані з запиту і змінює ними об'єкт в базі даних.

I метод Delete (int id) обробляє запити типу Delete, тобто запити на видалення - отримує із запиту параметр id і по цим ідентифікатором видаляє об'єкт з БД.

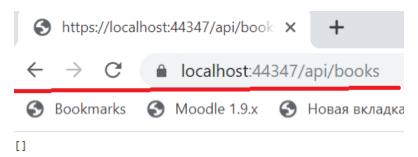
Аналогічно створюємо контролери для усіх класів моделі (за необхідності додаємо методи та/або змінюємо існуючі — це є обов'язковим для здачі етапу на максимальний бал):

```
ntroller.cs + X AuthorsController.cs
                                       BooksController.cs
                                                         CategoriesController.cs
                                                                                                      🗝 😽 Library APIWeb App. Controllers. Author Books C 🕶 🗞 _context
                                                                                                            #nullable disable
                                                                                                           Search Solution Explorer (Ctrl+;)
      using System;
                                                                                                            Solution 'LibraryAPIWebApp' (1 of 1 project)
       using System.Collections.Generic;

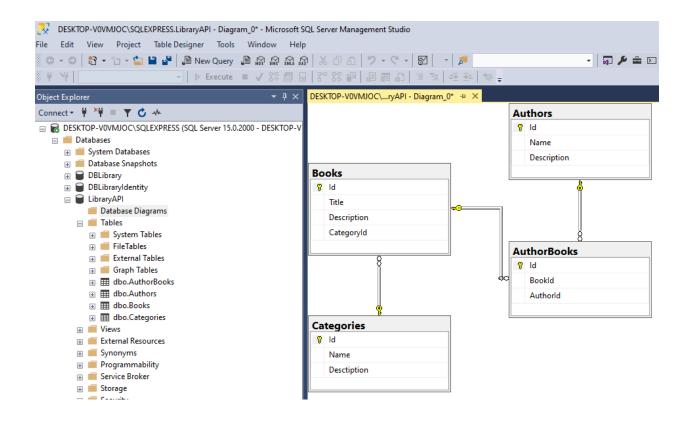
▲ LibraryAPIWebApp

        using System.Linq;
                                                                                                                 (A) Connected Services
        using System.Threading.Tasks;
                                                                                                                 ₽₽ Dependencies
        using Microsoft.AspNetCore.Http;
                                                                                                                 Properties
       using Microsoft.AspNetCore.Mvc;
       using Microsoft.EntityFrameworkCore;
                                                                                                                 D 🗀 css
       using LibraryAPIWebApp.Models;
                                                                                                                   🛅 js
10
                                                                                                                    lib
11
      □namespace LibraryAPIWebApp.Controllers
                                                                                                                    favicon.ico
12
13
            [Route("api/[controller]")]
                                                                                                                    C# AuthorBooksController.cs
14
            [ApiController]
                                                                                                                    C# AuthorsController.cs
15
            public class AuthorBooksController: ControllerBase
                                                                                                                    C# BooksController.cs
16
                                                                                                                   C# CategoriesController.cs
                private readonly LibrariAPIContext _context;
17
                                                                                                                 Models
18
                                                                                                                 D C# Author.cs
                                                                                                                 D C# AuthorBook.cs
                public AuthorBooksController(LibrariAPIContext context)
19
                                                                                                                   C# Book.cs
```

# Запускаємо:



Ми здійснили звернення до БД, перевіримо її:

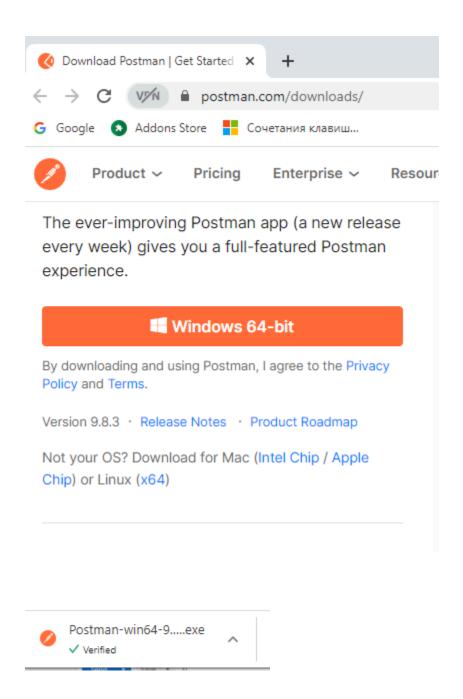


#### 2. Тестування контролерів

Ми створили контролери Web API, і протестовали роботу методу GET контролера Books. Однак безпосередньо з рядка браузера крім запитів GET інші типи запитів ми протестувати не можемо. Звичайно, ми можемо створити клієнт у вигляді веб-сторінки, мобільного додатка під якусь платформу або навіть графічного або консольного десктопного додатка, але створення клієнта може зайняти досить багато часу, тоді як нам просто треба протестувати обробку запитів.

Для тестування контролера Web API можна застосовувати спеціальні інструменти, які встановлюються у вигляді окремих додатків, або у вигляді розширень для браузерів, наприклад, Fiddler або Postman.

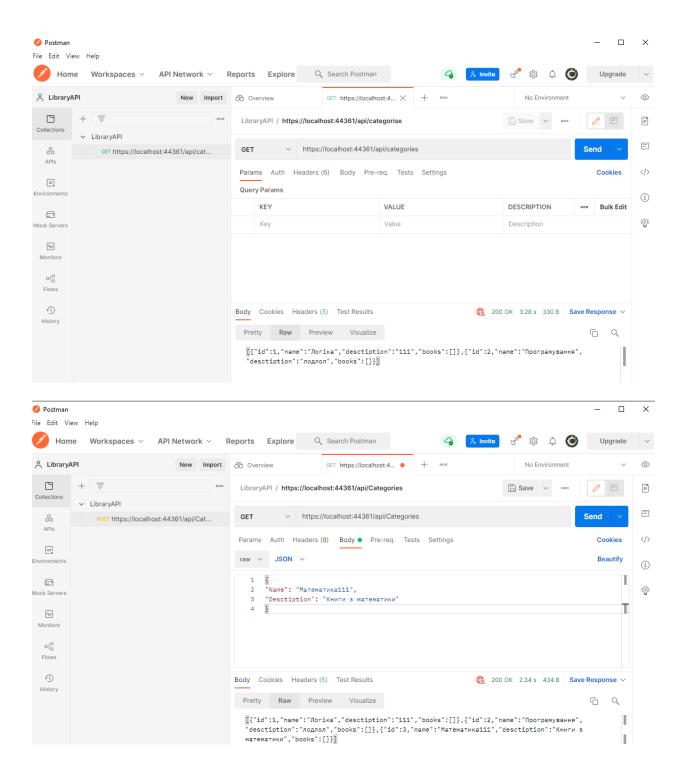
Встановіть Postman (<a href="https://www.getpostman.com/downloads/">https://www.getpostman.com/downloads/</a>):

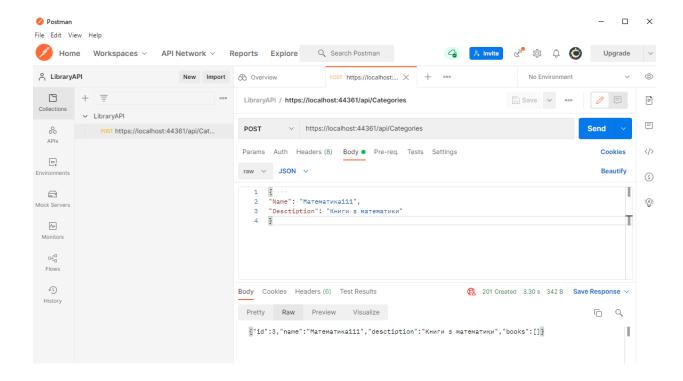


Запустіть веб-додаток.

Запустіть Postman (В File-> Settings відключіть параметр SSL certificate verification – після завершення роботи увімкніть знову).

Створіть новий запит.





Для здачі цього етапу підготуйте тестові приклади для усіх методів контролерів.