

Лабораторна робота №4

Розробка вебсерверів засобами фреймворку ASP.NET.

Зубченко А.С 301-пТК

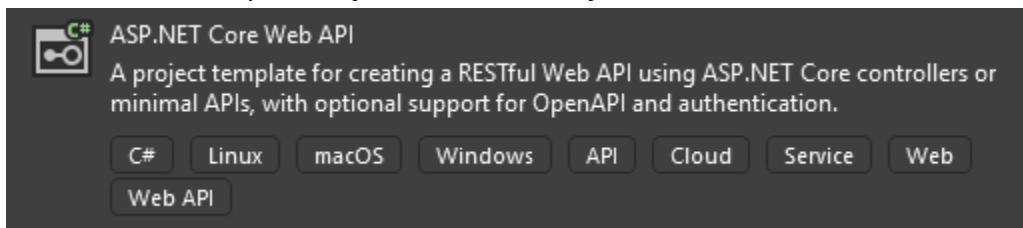
Передумова

Виконайте лабораторну роботу №3. Створіть нову гілку у раніше створеному гіт репозиторії *NUPP_NET_2025_{Номер Групи}_TK_{Прізвище}_Lab* із назвою **lab4**, яка міститиме код із третьої лабораторної роботи.

Для здачі лабораторної роботи, необхідно буде створити пул реквест із гілки **lab4** у **master**, або якщо на момент здачі пул реквест із гілки **lab3** у **master** не закритий - із гілки **lab4** у **lab3**.

Завдання

- Створити проект за шаблоном [ASP.NET](#) Core Web API {Назва тематики}.REST використовуючи Visual Studio або .NET SDK:
`dotnet new webapi -n {Назва тематики}.REST`



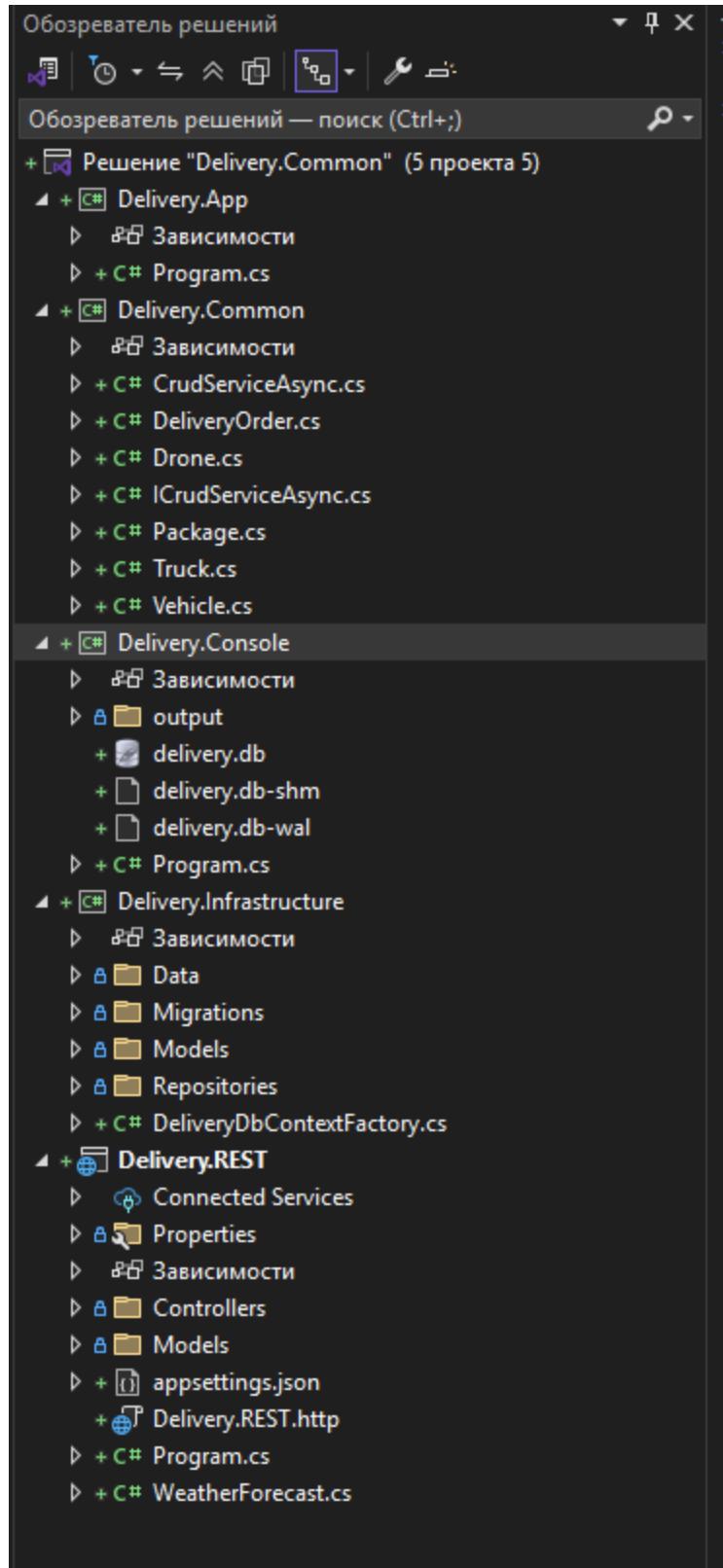
- Створити папку **Models**, у якій будуть зберігатися моделі, які будуть використовуватися контролерами у проекті {Назва тематики}.REST.
- Створіть контролери у папці **Controllers**, які будуть описувати REST API для виконання CRUD операцій (створення, читання, оновлення, видалення) над моделями у вашій тематиці, які ви зберігали у базі даних у третій лабораторній, при цьому моделі, що використовуються у контролерах, повинні бути описані у проекті {Назва тематики}.REST, щоб відповідати [3-рівневій архітектурі](#). Наприклад, якщо у вас є сутність **Bus**, що зберігається у БД, ви створите новий клас **BusModel** у папці **Models** нового проекту, який використовуватиметься у контролерах. Також зверніть увагу, щоб моделі у контролерах, мали лише ненобхідні властивості, та якщо якісь властивості зайді, для якихось із методів - створіть нову модель без цих властивостей.
- При проєктуванні Web API зверніть увагу, щоб створений API відповідав 4-ї умові побудови REST-додатку по Філдингу - однорідності інтерфейсу ([див. 31 слайд](#)), тобто використовувалися унікальні назви сутностей, правильні HTTP методи та поверталися правильні HTTP коди у HTTP відповідях, наприклад 404 якщо сутність не знайдена, або 201 якщо нова сутність створена.

5. Використовуйте асинхронну версію дженерік CRUD сервісу, що використовує репозиторій для доступу до даних та який був реалізований у Зій лабораторній роботі, у створених контролерах.:

```
public interface ICrudServiceAsync<T>
{
    public Task<bool> CreateAsync(T element);
    public Task<T> ReadAsync(Guid id);
    public Task<IEnumerable<T>> ReadAllAsync();
    public Task<IEnumerable<T>> ReadAllAsync(int page, int amount);
    public Task<bool> UpdateAsync(T element);
    public Task<bool> RemoveAsync(T element);
    public Task<bool> SaveAsync();}
```

Імплементацію CRUD сервісу, репозиторіїв та контексту, додавайте у контролери використовуючи вбудовану у [ASP.NET](#) Core підтриму впровадження залежностей(Dependency Injection).

До PR готової лабораторної роботи додайте PDF файл у якому будуть результати виконання запитів, використовуючи створений REST API. Можете скористатися сторінкою Swagger, яка генерується [ASP.NET](#) застосунком, або застосунком [Postman](#).



Структура проекту

```
C:\Users\Артем\Desktop\Net > + <
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5153
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\Users\Артем\Desktop\Net\NUPP_NET_2025_301_pTK_Zubchenko_Lab\Lab4\Delivery.REST
```

Результат в консолі

The screenshot shows the Swagger UI for the **Delivery.REST** API, version 1.0. The main navigation bar includes links for **Delivery.REST v1**, **Delivery.REST v2**, and **Delivery.REST v3**. The current view is for **Delivery.REST v1**.

DeliveryOrder

- GET** /api/DeliveryOrder
- POST** /api/DeliveryOrder
- GET** /api/DeliveryOrder/{id}
- DELETE** /api/DeliveryOrder/{id}
- PUT** /api/DeliveryOrder/{id}/status

Package

- GET** /api/Package
- POST** /api/Package
- GET** /api/Package/{id}
- DELETE** /api/Package/{id}

Truck

- GET** /api/Truck

Результат запуску програми

Контрольні запитання

1. WIP

Система оцінювання (ТК)

Усього за лабораторну роботу можна отримати **20 балів**, із яких 14 йдуть у семестровий контроль та 6 у заліковий. Роботи подані через дистанційну систему навчання **до 14 листопада** оцінюються без усного захисту із відповідями на контрольні запитання, при цьому додаткове завдання оцінюється у 4 бали, роботи здані після дедлайну повинні бути захищені згідно контрольних запитань.

Система оцінювання (ТН)

Усього за лабораторну роботу можна отримати **10 балів**. Роботи подані через дистанційну систему навчання **до 15 листопада** оцінюються без усного захисту із відповідями на контрольні запитання, при цьому додаткове завдання оцінюється у 2 бали, роботи здані після дедлайну повинні бути захищені згідно контрольних запитань.