

Лабораторна робота №5

Реалізація спроможностей реєстрації, аутентифікації та авторизації засобами фреймворку [ASP.NET Core Identity](#).

Зубченко А.С 301-пТК

Передумова

Виконайте лабораторну роботу №4. Створіть нову гілку у раніше створеному гіт репозиторії *NUPP_NET_2025_{Номер Групи}_TK_{Прізвище}_Lab* із назвою **lab5**, яка міститиме код із четвертої лабораторної роботи.

Для здачі лабораторної роботи, необхідно буде створити пул реквест із гілки **lab5** у **master**, або якщо на момент здачі пул реквест із гілки lab5 у master не закритий - із гілки lab5 у lab4.

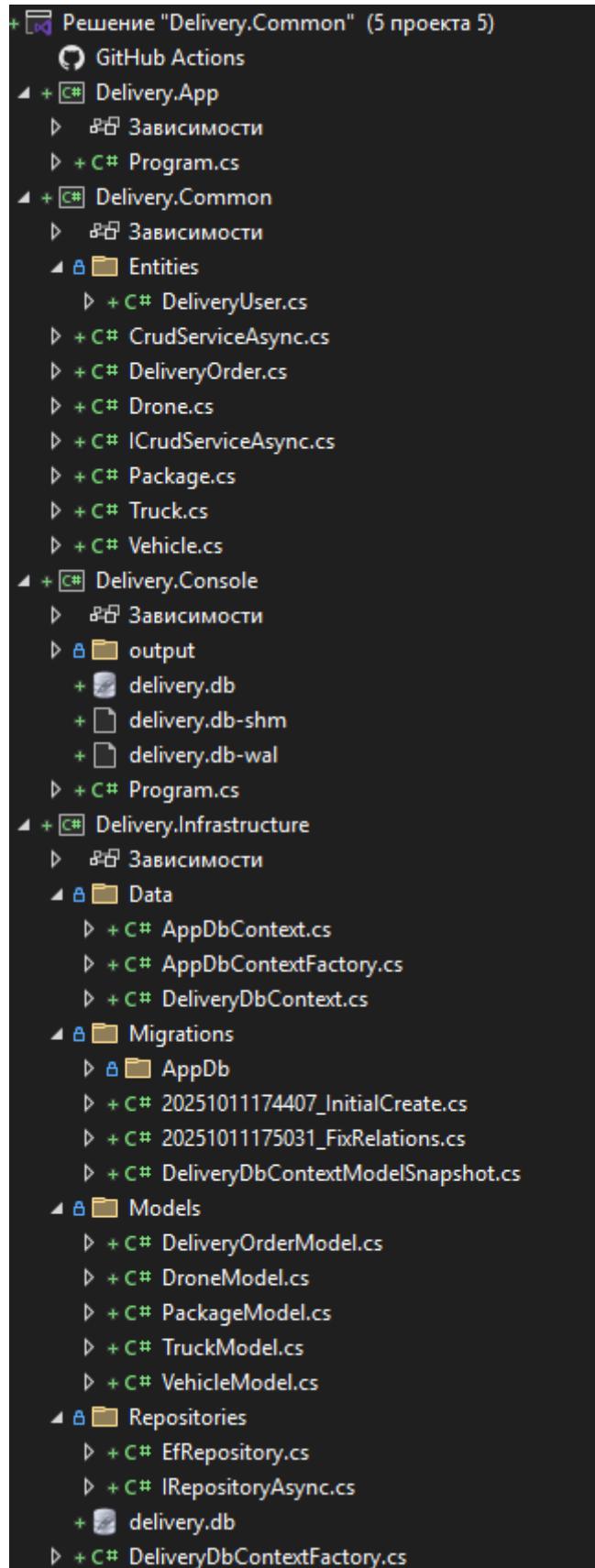
Завдання

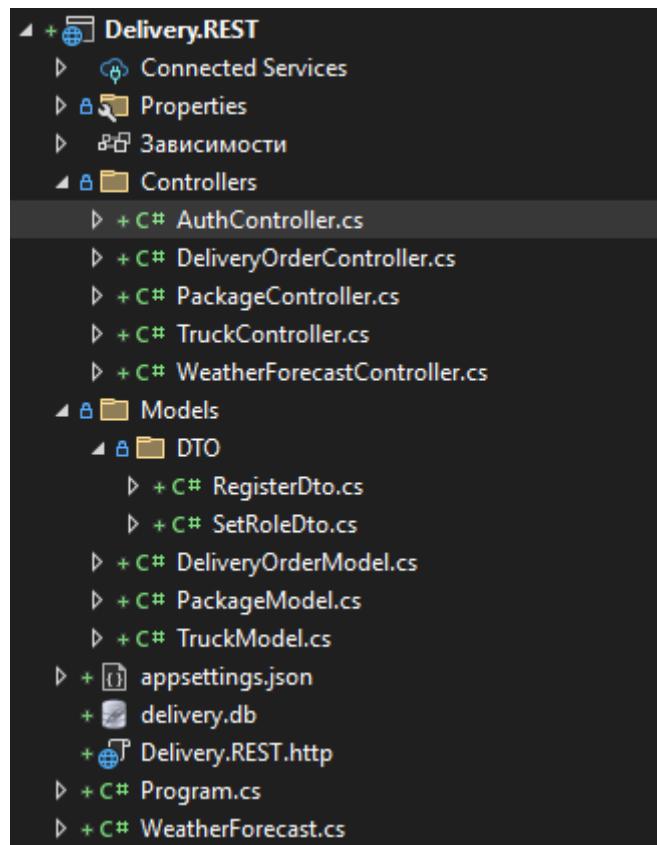
1. Додати в проект {Назва тематики}.Infrastructure нову залежність на NuGET пакет [Microsoft.AspNetCore.Identity.EntityFrameworkCore](#).
2. Модифікуйте створену вами у попередніх лабораторних роботах модель (набір класів), додавши до неї сутність користувача, яка буде наслідуватися від класу `IdentityUser` із фреймворку [ASP.NET Core Identity](#). Модифікуйте, створений у третій лабораторній, `DbContext` таким чином, щоб він наслідувався від класу `IdentityDbContext<T>`, де узагальнюючим типом буде створена вами сутність користувача.
3. Модифікуйте проект {Назва тематики}.REST, щоб він виставляв на зовні вбудовані у [ASP.NET Core Identity](#) кінцеві точки для реєстрації та аутентифікації, при цьому аутентифікація повинна створювати Bearer або JWT токени. Можете спиратися на [офіційний туторіал від Microsoft](#).
4. Модифікуйте створений у четвертій лабораторній REST API, щоб кінцеві точки використовували авторизацію, та надавали доступ відповідно до токену із заголовку HTTP-запиту `Authorization`. Зверніть увагу, щоб авторизація використовувалася лише в необхідних кінцевих точках, наприклад у REST API бібліотеки, перегляд книг не потребує авторизації, але бронювання книги потребує.
5. Додайте для створеного вебдодатку ролі, усього ролей повинно бути не менше трьох. Оновіть кінцеві точки REST API, що використовують авторизацію, щоб вони виконувалися залежно від ролі клієнта, наприклад у REST API бібліотеки, забронювати книгу може звичайний користувач, але змінити інформацію про книгу у БД лише бібліотекар.

До готової лабораторної роботи додайте PDF файл у якому будуть результати виконання запитів, використовуючи модифікований REST API, який буде повертати результати реєстрації та аутентифікації, виконання запитів для

авторизованих, неавторизованих та клієнтів, із недостатнім рівнем доступа.
Можете скористатися сторінкою [Swagger UI](#) або застосунком [Postman](#).

Структура проекту





Перша реєстрація та логін

Auth

POST /api/Auth/register

Parameters

No parameters

Request body

application/json

```
{
  "email": "user@example.com",
  "password": "Abc123!@#",
  "fullName": "Андрій"
}
```

Execute Clear

Responses

This screenshot shows the Postman application interface. A new request is being prepared with the following details:

- Method:** POST
- URL:** /api/Auth/register
- Parameters:** None
- Request body:** application/json
- Body Content:**

```
{
  "email": "user@example.com",
  "password": "Abc123!@#",
  "fullName": "Андрій"
}
```
- Buttons:** Execute (blue button), Clear (white button)
- Responses:** (Empty section)

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:5153/api/Auth/register' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "email": "user@example.com",
    "password": "Abc123!@#",
    "fullname": "Андрей"
  }'
```

Request URL

```
http://localhost:5153/api/Auth/register
```

Server response

Code	Details
200	Response body Registered <div style="text-align: right; margin-top: -10px;"> Copy Download </div>
	Response headers access-control-allow-origin: * content-type: text/plain; charset=utf-8 date: Fri, 21 Nov 2025 21:20:24 GMT server: Kestrel transfer-encoding: chunked

Responses

Code	Description	Links
200	OK	No links

POST /api/Auth/login

Parameters

No parameters

Request body

```
{
  "email": "user@example.com",
  "password": "Abc123!@#"
}
```

Execute **Clear**

Responses

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:5153/api/Auth/login' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "email": "user@example.com",
    "password": "Abc123!@#"
}'
```

Request URL

```
http://localhost:5153/api/Auth/login
```

Server response

Code	Details
200	Response body <pre>{ "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwaIi8vc2NoZWdhcy54bbcb2Fwdm9yZy93cy8yMDA1LzA1L21kZV50aXR512NsYWlItcy9uYw11joiNdEx1ck81eGFTcGx1LnWvbSIsImh0dHA6Ly92Y2h1MFpZLahtbHNvYXhUb37nE3dzLzIwMDUvaIR1bnRpdtIkvY2xhaW1zL25hbWpZGVudmlmaWly1jo10dg1Ye5zDctMDQ405000m4LTkzOTktYz81NtI4Yzg5OTIx1iw1ZXhwIjoxNzY0MDE5NjU1fQ.533yMw8RCGPjN10Su0hY0lnIp-{f000xM6QLzt1V/</pre> <div style="text-align: right; margin-top: -10px;"> Copy Download </div>
	Response headers access-control-allow-origin: * content-type: application/json; charset=utf-8 date: Fri, 21 Nov 2025 21:27:35 GMT server: Kestrel transfer-encoding: chunked

Responses

Code	Description	Links
200	OK	No links

Available authorizations

X

Bearer (http, Bearer)

Вставте JWT токен у форматі: Bearer {token}

Value:

Bearer eyJhbGciOiJIUzI1Nils

Authorize

Close

Available authorizations

X

Bearer (http, Bearer)

Authorized

Вставте JWT токен у форматі: Bearer {token}

Value: *****

Logout

Close

Створюєм кілька користувачів та проповісння ролей

1) User

email: user@test.com

password: Abc123!@

fullName: User

```
{  
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1hcyc54bwzb2FwLm9yZy93cy8yM  
  DA1LzA1L21kZW50aXR5L2NsYWltcy9uYW11IjoidXN1ckB0ZXN0LmNvbSIsImh0dHA6Ly9zY2h1bWFzLnhtbHNvYXAub3J  
  nL3dzLzIwMDUvMDUvaW1bnRpdHkvY2xhaw1zL25hbWVpZGVudGlmaWVyIjoiZWNmNjYwZWItZTY2YS00MmE1LWFhNzQtY  
  TQ0M2EzOGIwYTdjIiwiHR0cDovL3NjaGVtYXMubWljcm9zb2Z0LmNvbS93cy8yMDA4LzA2L21kZW50aXR5L2NsYWltcy9  
  yb2x1IjoiVXNlciIsImV4cCI6MTc2NDAyMzEyOH0.J_haXUNFjoP0ivGAMNjIePGngUhgA1dj1qkRZtiRQfc"  
}
```

2) Manager

email: manager@test.com

password: Abc123!@

fullName: Manager

```
{ "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1hcy54bWxzb2FwLm9yZy93cy8yM DA1LzA1L2lzkW50aXR5L2NsYWh1tcy9uYW1lIjoibWFuYWh1lckB0ZXN0LmNvbSISImh0dHA6Ly9zY2h1bwFzLnhtbHNvYXA ub3JnL3dzLzIwMDUvMDUvaWR1bnRpdkHkvY2xhaW1zL25hbWpZGVudGlmaWVijoiYTdmZmNiYTgtMWUzYy00OWY4LWEz0 DIYTJmNDg2MjYwOTBmIiwiHaHR0cDovL3NjaGVtYXMubWljcm9zb2Z0LmNvbS93cy8yMDA4LzA2L21kZW50aXR5L2NsYWh1 tcy9yb2x1IjoiVXNlcjIiMv4cCI6MTc2NDAyMzA5Nn0.ZGOMZx4zhzGH-Ua9SN1w6WyPsZgsZlRE47FmpHUHBqA" }
```

3) Admin

email: admin@test.com

password: Abc123!@

fullName: Admin

```
{  
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCIKpXVCJ9.eyJodHRwOi8vc2NoZW1hcj54bWxzb2FwLm9yZy93cy8yM  
DA1LzA1L2lkZW50aXR5L2NsYWhlcty9uYW1lIjoiYWRTaW5AdGVzdC5jb20iLCJodHRwOi8vc2NoZW1hcj54bWxzb2FwLm9  
yZy93cy8yMDA1LzA1L2lkZW50aXR5L2NsYWhlcty9uYW1laWRlbnRpZmllcjI6ImExOWM00Tc2LTk4NmYtNDM5NS1i0GJ1L  
TIwNGFjYmE3MjIyZSI5Imh0dHA6Ly9zY2h1bWFzLm1pY3Jvc29mdC5jb20vd3MvMjAwOC8wNi9pZGVudG10eS9jbGFpbXM  
vcm9sZSI6I1VzZXIiLCJleHAiOjE3NjQwMjMwNjZ9.9WZyIuT6Wgf5bWYwvxpI80hxgwd0nkaml1r240Rp0tA"  
}
```

Вивід інфо в консолі під час запуску

```
C:\Users\Artem\Desktop\Net X + ▾

", "a"."LockoutEnabled", "a"."LockoutEnd", "a"."NormalizedEmail", "a"."NormalizedUserName", "a"."PasswordHash", "a"."PhoneNumber", "a"."PhoneNumberConfirmed", "a"."SecurityStamp", "a"."TwoFactorEnabled", "a"."UserName"
    FROM "AspNetUsers" AS "a"
    WHERE "a"."NormalizedUserName" = @_normalizedUserName_0
    LIMIT 1
[info: Microsoft.EntityFrameworkCore.Database.Command[20101]
Executed DbCommand (1ms) [Parameters=@p0=? (Size = 36), @p1=? (Size = 36)], CommandType='Text', CommandTimeout='30']
DELETE FROM "AspNetUserRoles"
WHERE "RoleId" = @p0 AND "UserId" = @p1
RETURNING 1;
[info: Microsoft.EntityFrameworkCore.Database.Command[20101]
Executed DbCommand (1ms) [Parameters=@p0=? (Size = 36), @p1=? (DbType = Int32), @p2=? (Size = 36), @p3=? (DbType = Boolean), @p4=? (Size = 7), @p5=? (DbType = Boolean), @p6=? (DbType = DateTimeOffset), @p7=? (Size = 16), @p8=? (Size = 16), @p9=? (Size = 84), @p10=? (DbType = Boolean), @p11=? (DbType = Boolean), @p12=? (Size = 32), @p13=? (DbType = Boolean), @p14=? (Size = 16)], CommandType='Text', CommandTimeout='30']
UPDATE "AspNetUsers" SET "AccessFailedCount" = @p0, "ConcurrencyStamp" = @p1, "Email" = @p2, "EmailConfirmed" = @p3, "FullName" = @p4, "LockoutEnabled" = @p5, "LockoutEnd" = @p6, "NormalizedEmail" = @p7, "NormalizedUserName" = @p8, "PasswordHash" = @p9, "PhoneNumber" = @p10, "PhoneNumberConfirmed" = @p11, "SecurityStamp" = @p12, "TwoFactorEnabled" = @p13, "UserName" = @p14
WHERE "Id" = @p15 AND "ConcurrencyStamp" = @p16
RETURNING 1;
[info: Microsoft.EntityFrameworkCore.Database.Command[20101]
Executed DbCommand (0ms) [Parameters=@_normalizedName_0=? (Size = 7)], CommandType='Text', CommandTimeout='30']
SELECT "a"."Id", "a"."ConcurrencyStamp", "a"."Name", "a"."Normalized Name"
FROM "AspNetRoles" AS "a"
WHERE "a"."Normalized Name" = @_normalizedRoleName_0
LIMIT 1
[info: Microsoft.EntityFrameworkCore.Database.Command[20101]
Executed DbCommand (0ms) [Parameters=@_normalizedRoleName_0=? (Size = 7)], CommandType='Text', CommandTimeout='30']
```