

Лабораторна робота №5

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

Виконав студент 301-пТК Тараненко Олександр

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 тутторіал від Microsoft](#).

POST /api/Auth/login

Parameters

No parameters

Request body

application/json

```
{
  "email": "s.taranenko2012@gmail.com",
  "password": "Dimatar21!"
}
```

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:5155/api/Auth/login' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "email": "s.taranenko2012@gmail.com",
    "password": "Dimatar21!"
}'
```

Request URL

<http://localhost:5155/api/Auth/login>

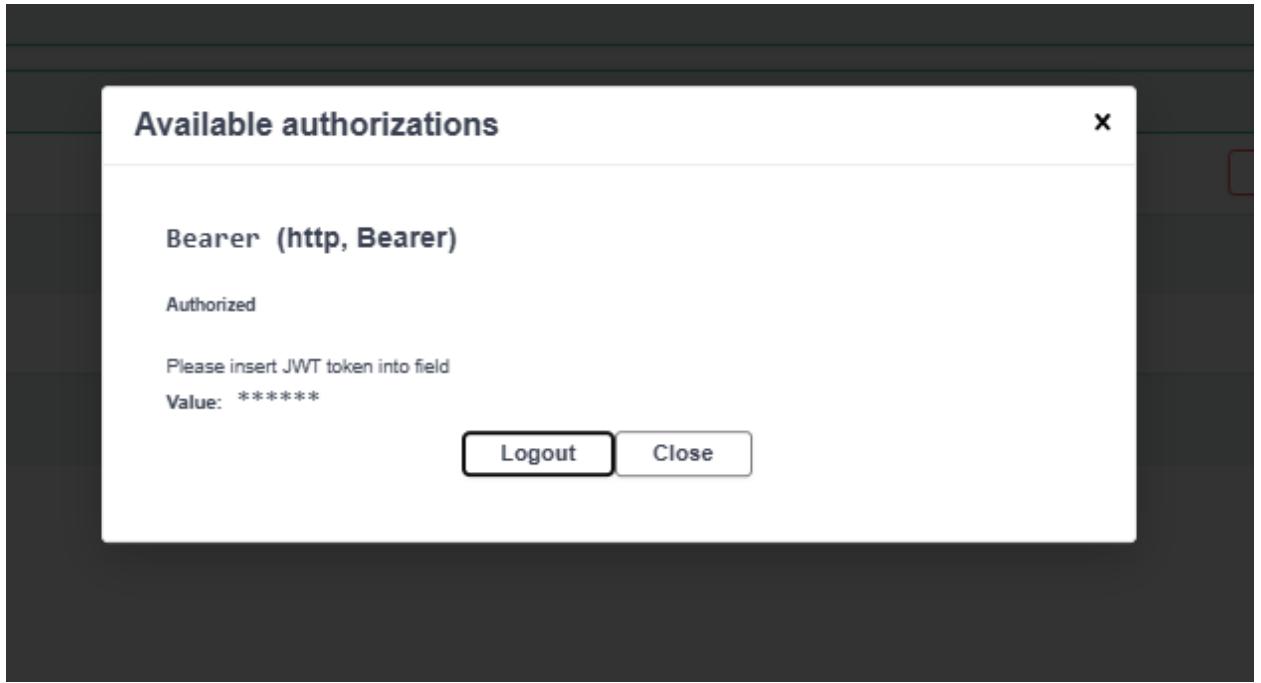
Server response

Code	Details	Links
200	Response body <pre>{ "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWJlOiI4MDk1YmNkNCB0MGFkLTQ4YmUyNzE4ZGE1LC31bWFpbCTGjzb20ILC3kaXNubGF5TmFtZST6InRlc2ciLC1leHA0jE3NjIxMDcwNTUsImIzcyI6IkdhbwzQXVbaFN1cn2lc1ci1m51zC16IkdhbwzQ2xpZw58Im8.Y1_XehDd-9-_mESSheIH7xk75ftc4-dUEhMhVWBo-N" }</pre> Response headers <pre>content-type: application/json; charset=utf-8 date: Sun, 02 Nov 2025 15:10:54 GMT server: Kestrel transfer-encoding: chunked</pre>	Download

Responses

Code	Description	Links
200	OK	No links

- 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.

Game

GET /api/game

Parameters

No parameters

Cancel

Execute Clear

Responses

Call

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 123
{
    "id": "5e0f2a00-0000-4000-a000-000000000000",
    "name": "Football"
}
```

Request (R):

```
http://localhost:3001/api/game
```

Server response

Code detail

Failed to fetch.
Possible Reasons:

- CORS
- Network Failure
- URL scheme must be "http" or "https" for CORS request.

Responses

Code	Description	Links
200	OK	No links

Header

Content-Type: application/json

Example Value: Schema

```
{
    "id": "5e0f2a00-0000-4000-a000-000000000000",
    "name": "Football"
}
```

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](#).