**So uh idk none abt ts, so I started working with a deadline of 2 weeks I think I can finish it within days so idk**

Ts like a devlog or sum type shi so anyone ever reads this be not surprised

First I gotta set up ide’s and stuff to work on api’s

I alr had vs, vs code, postgres17 with pgadmin, datagrip and later on installed postman to api test

The project is going to be in visual studio, maybe even start a github repo but idk time shows.

Github Repo: <https://github.com/K0d0ku/web-serv_midterm>

**The requirements :**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Минимальные требования к работам 1-ой аттестации** | | | | |  |
|  |  |  |  |  |  |
| **№** | **Наименование** | **Описание** |  | **Балл** |  |
| **1.** | **Реализация CRUD операций с применением архитектурного стиля REST** |  | Обязательно | 8 | Done |
| 1.1. | Реализация валидации (DataAnnotation, FluentAPI) |  |  |  | Done with dataAnnotation |
| **2.** | **Реализация Dependency Injection** |  | Обязательно | 8 | Done |
| 2.1. | Логирование (Seq, Serilog) |  |  |  | Done |
| 2.2. | Паттерн "Репозиторий" |  |  |  | Done |
| **3.** | **Тестирование API** |  | Желательно | 3 | Done |
| 3.1. | С применением *HTTPClient* (или его аналогов) в WEB части (можно ASP.NET Core или других решениях) |  |  |  | Done in web with swagger |
| 3.2. | С применением *Postman* или других аналогах |  |  |  | Done |
| **4.** | **Реализация авторизации API (JWT или другие варианты)** |  | Обязательно | 8 | Done |
| **5.** | **Реализация Аутентификация и Авторизация в приложении с примененем Identity** |  | Желательно | 3 | Done |
| **ИТОГО** | | |  | 30 |  |

So uh I started a new ASP.Net Core Web Api project named KuroApi

I quess later development details is going to be in github but still idk abt it

**Time spent 2 hours**

**I came here after 1 day of doing other subjects,**

Created a new postgres database KuroApiDb and installed some packages so the fuckass .net core can access and talk to my postgres database

Yuh then I had to test that shi if its connect so I created 1st migration of that bih

**So uh imma separate ts by parts as listed in requirements:**

1. Implementing CRUD operations using the REST architectural style

1.1. Implementing validation (DataAnnotation, FluentAPI)

**IN CODE**

2. Implementing Dependency Injection

2.1. Logging (Seq, Serilog)

2.2. Repository pattern

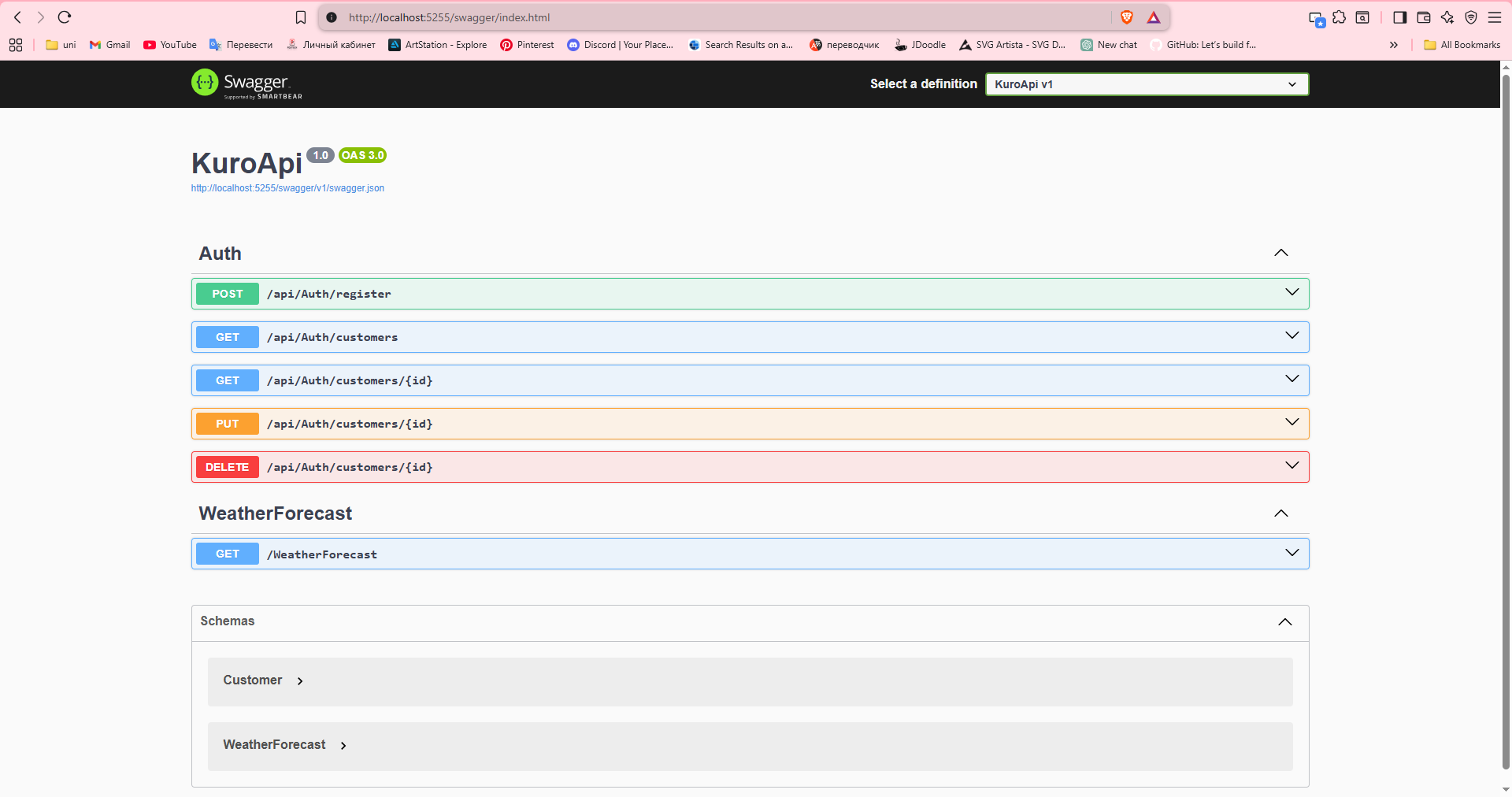
**IN CODE**

3. API testing

3.1. Using HTTPClient (or its equivalent) in the web part (ASP.NET Core or other solutions are possible)

3.2. Using Postman or other equivalents

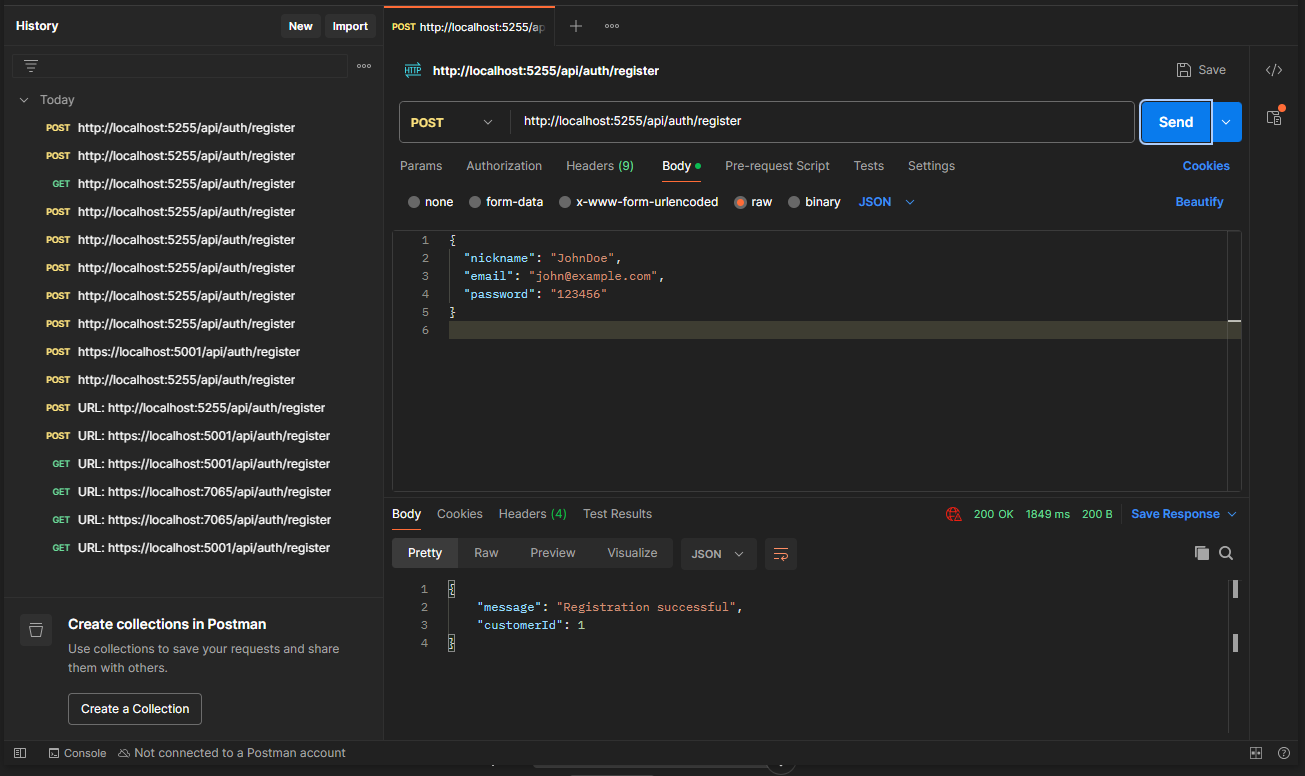
Web

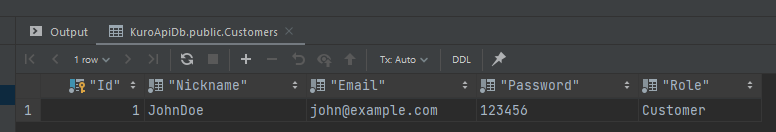


And postman ⬇

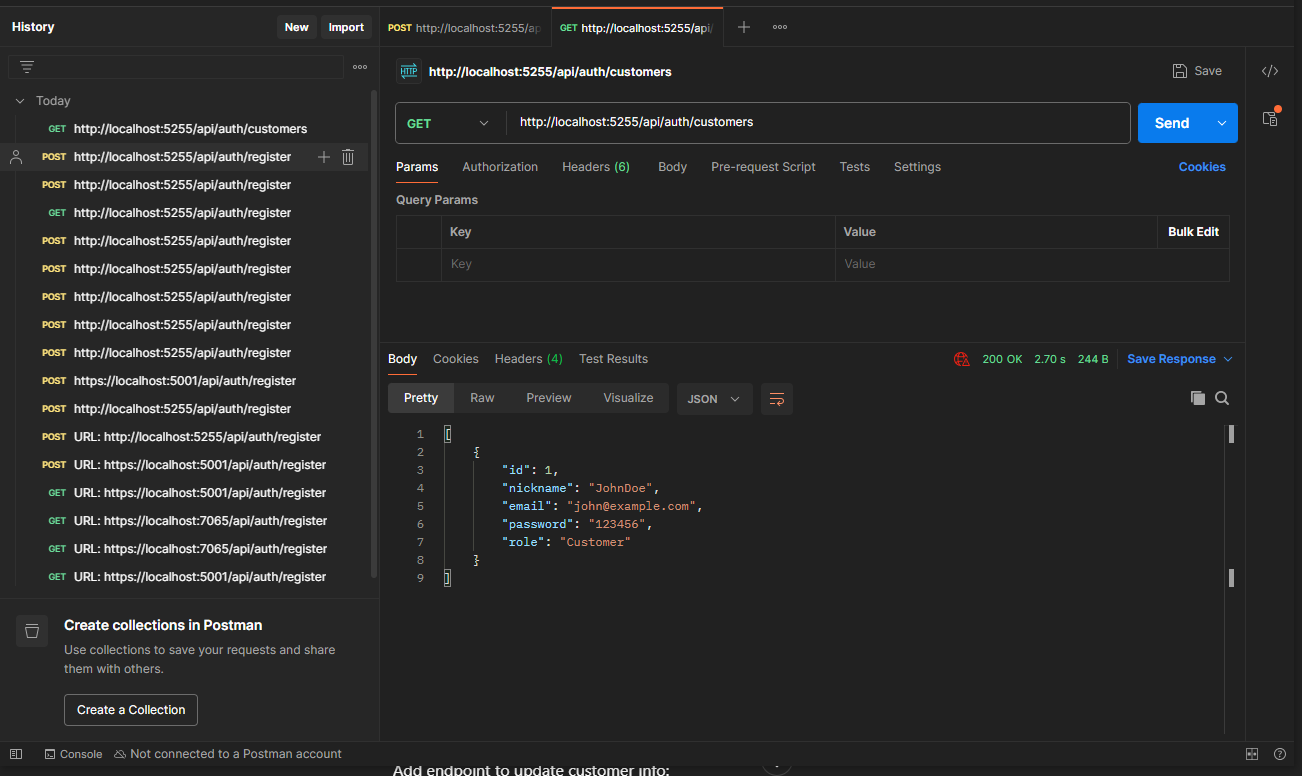
**Crud**

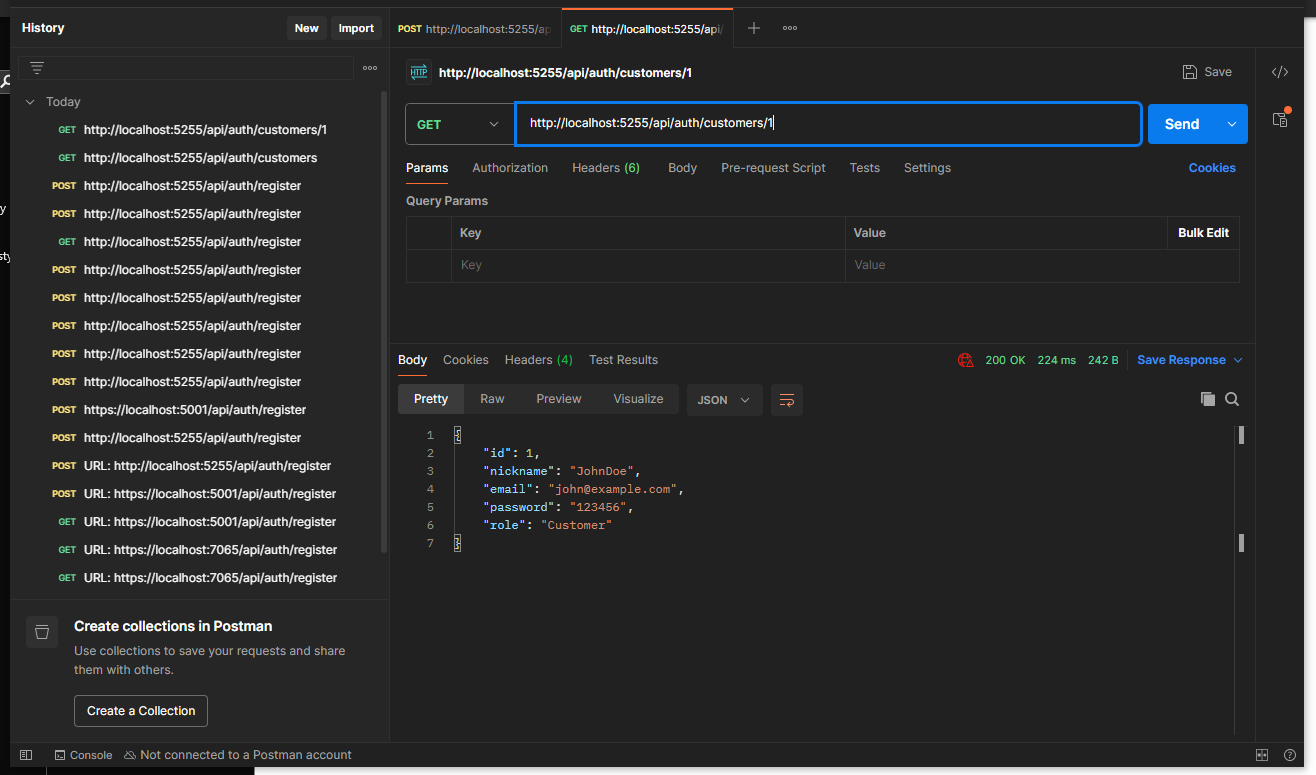
Create





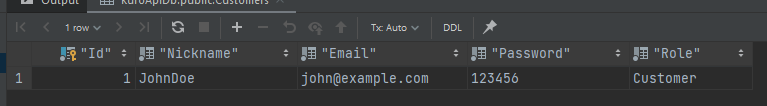
Read read all customer, and read customer by id



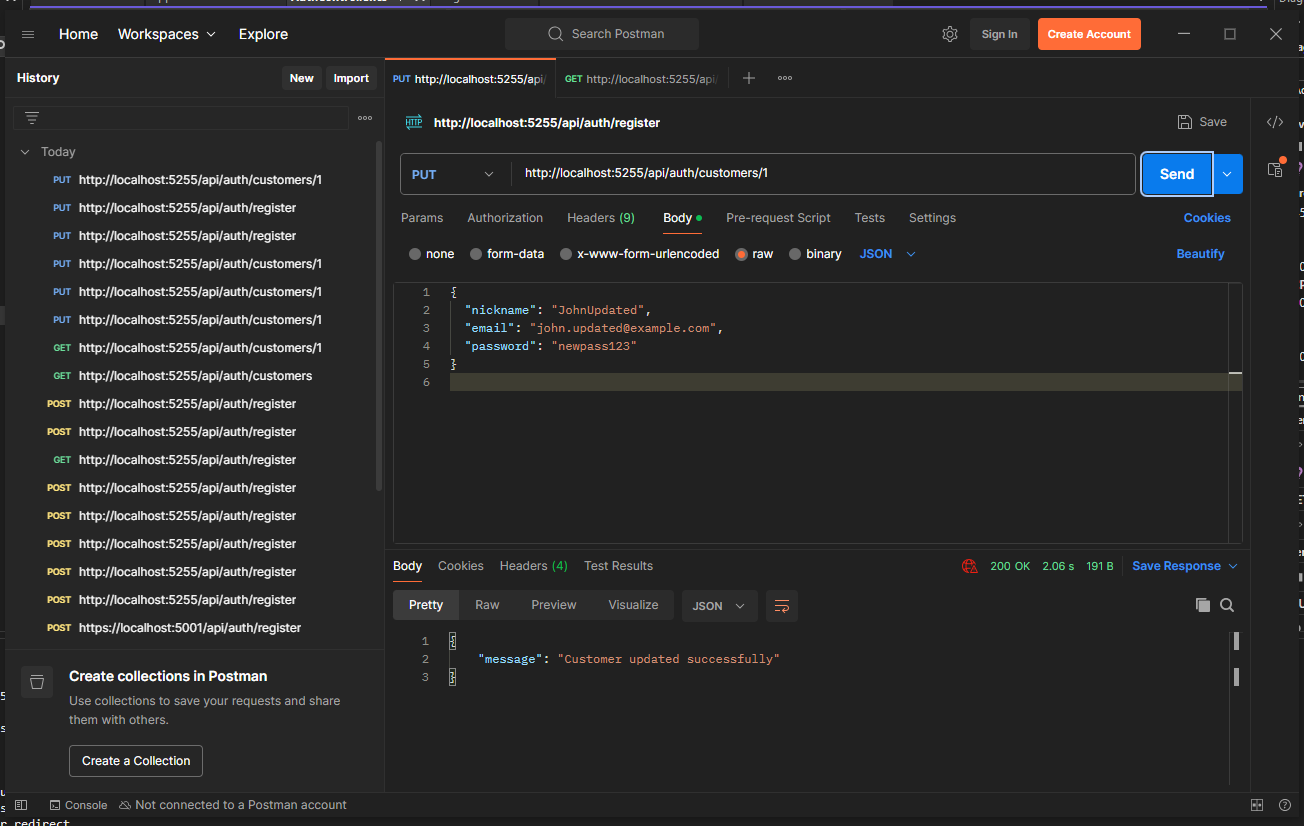


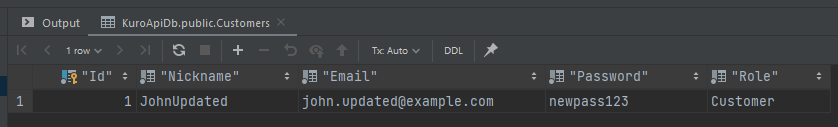
Update

Before



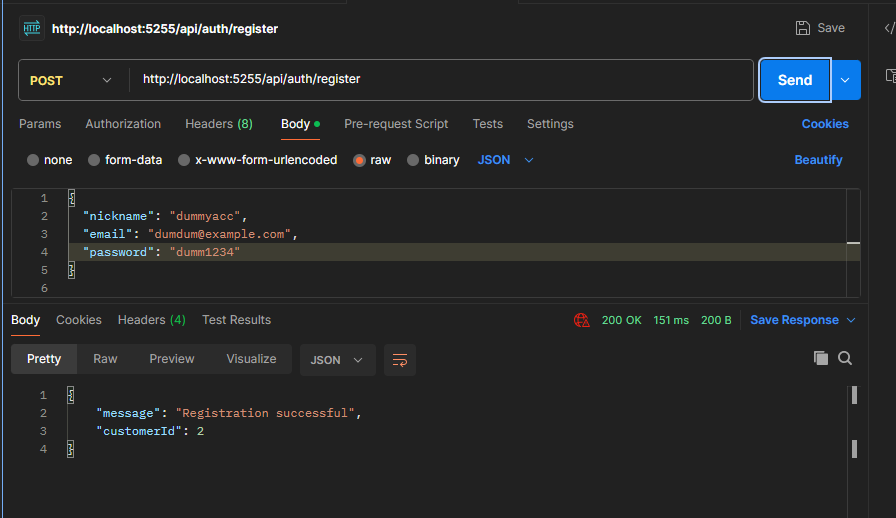
After

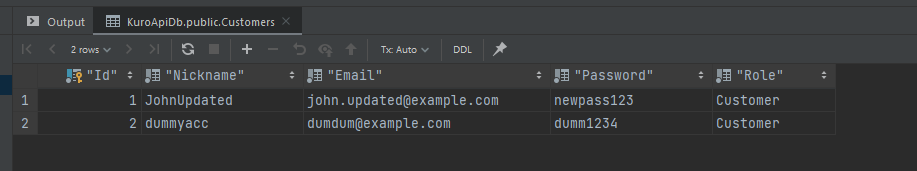




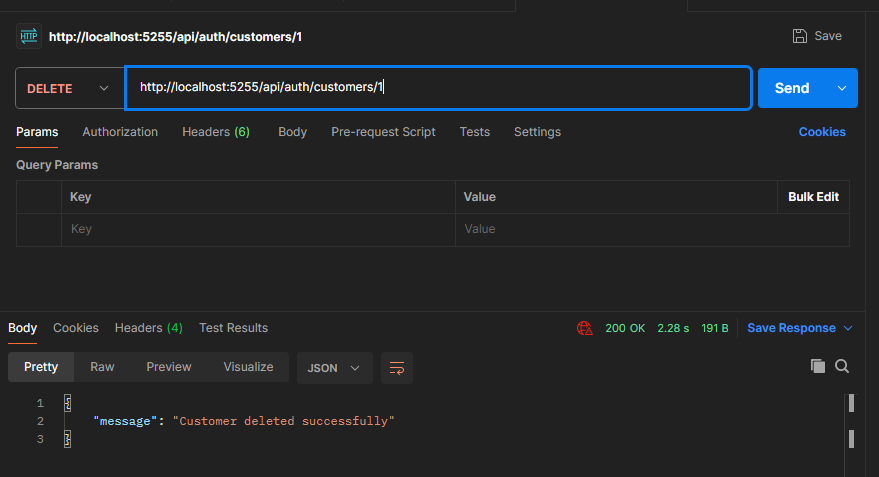
Delete

Added dummy acc with id 2 to delete it after

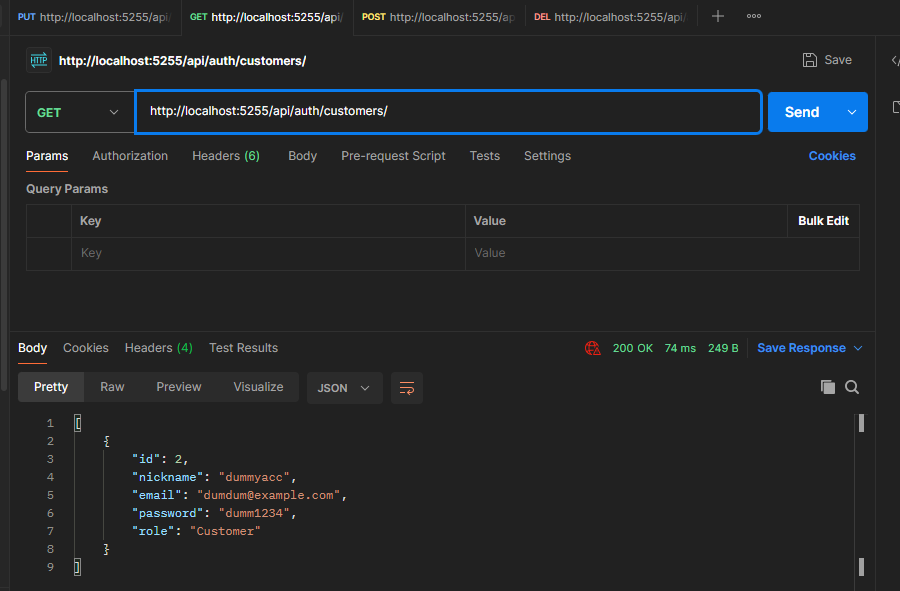




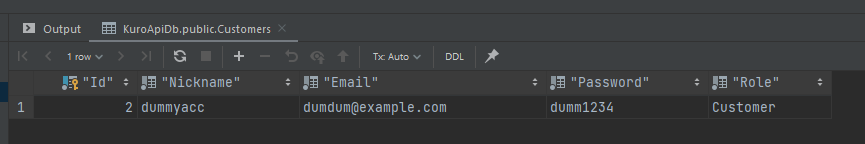
Now we delete id 1



And check for all users

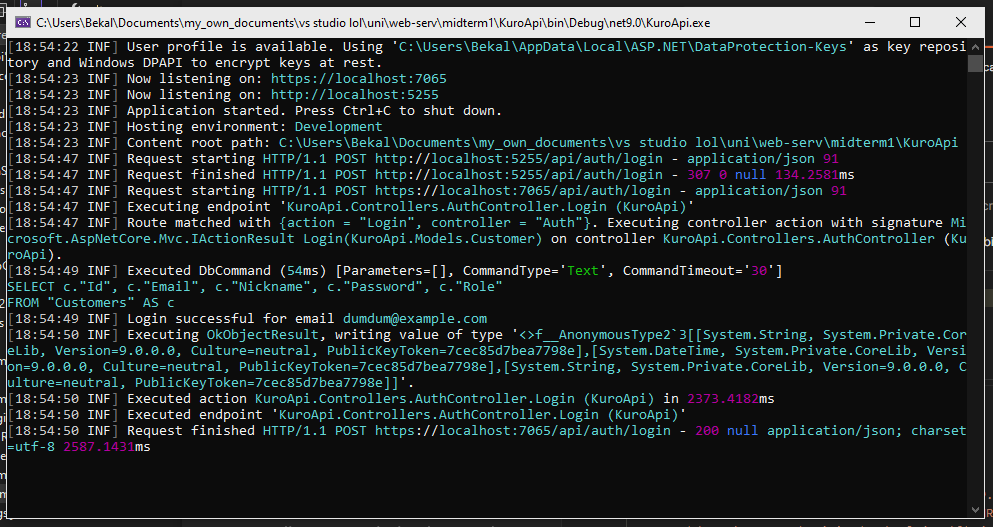


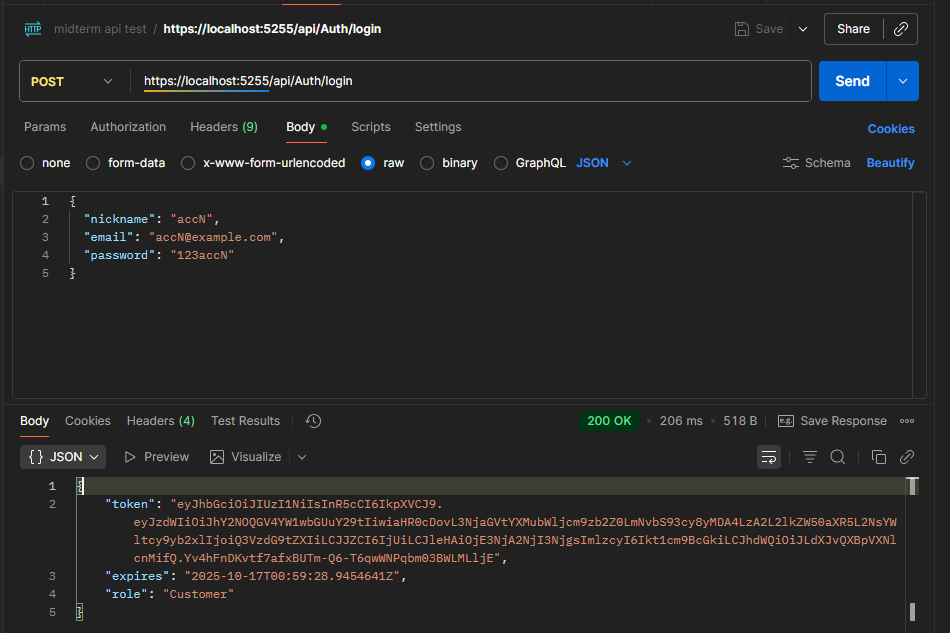
In db too

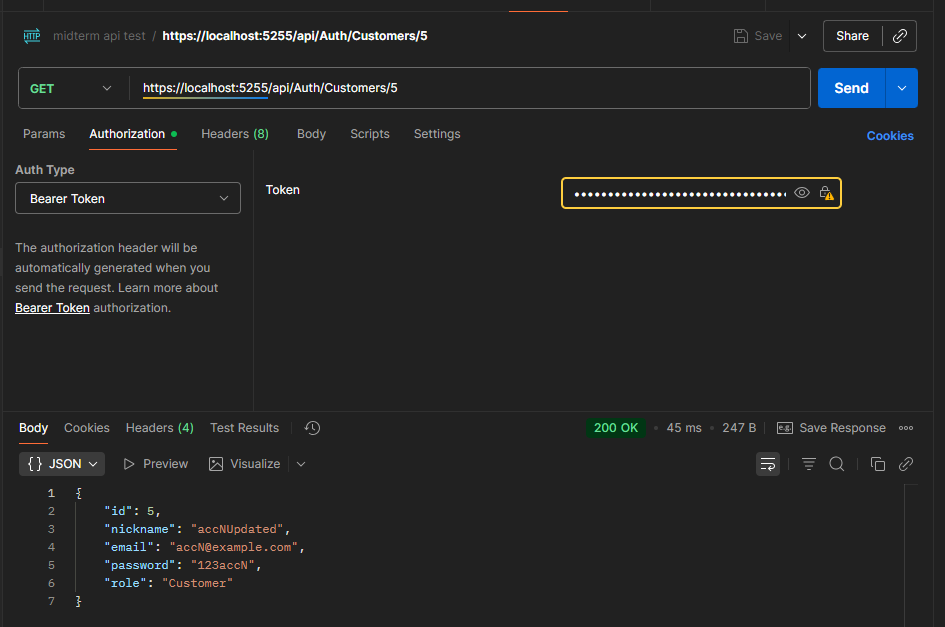


4. Implementing API authorization (JWT or other options)

**IN CODE**

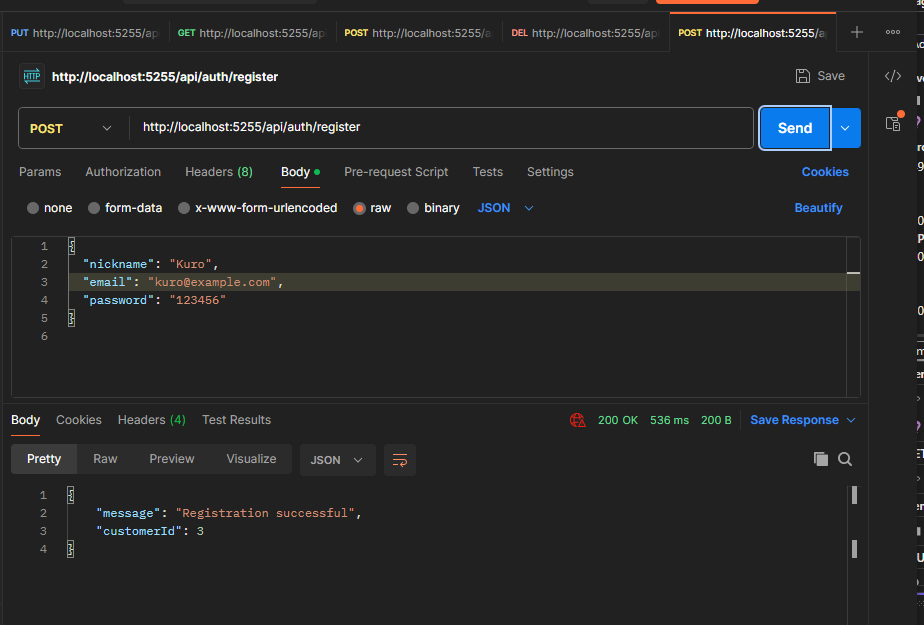


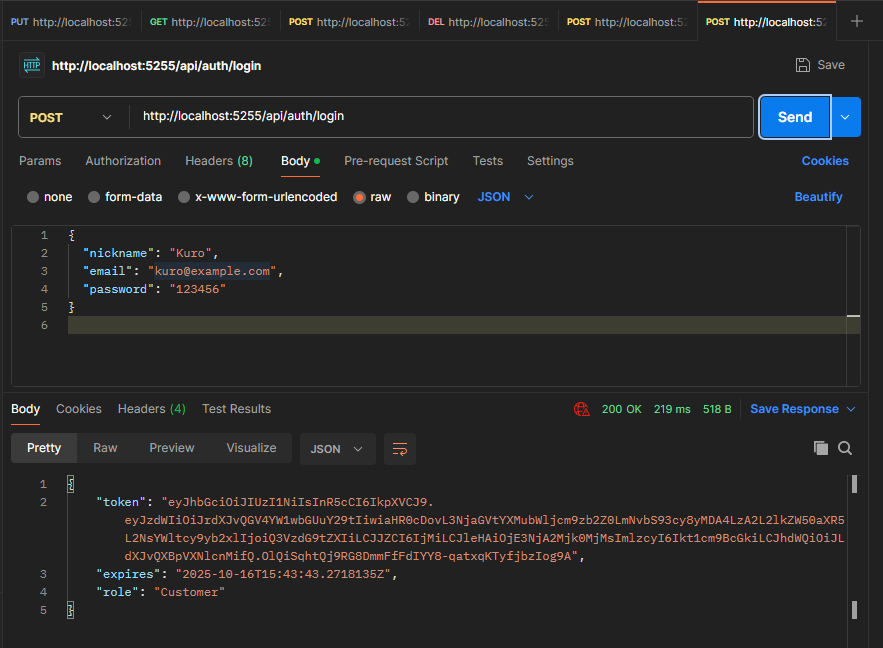
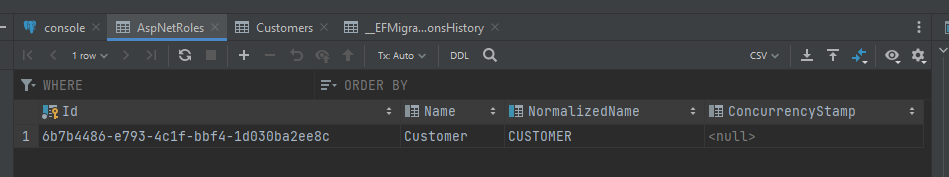




5. Implementing authentication and authorization in the application using Identity

IN CODE

Identity registration  


Identity login with generated jwt token:  
  
  
oh and here are some additional screenshots from the database:  


**Time spend 8hrs**

**The rest and more of the info is in GitHub repo.**