



Dokumentacja Projektu

MICHAŁ SAŁEK | GRUPA: L/23 N LAB5/2/PROG | INDEKS: 13809

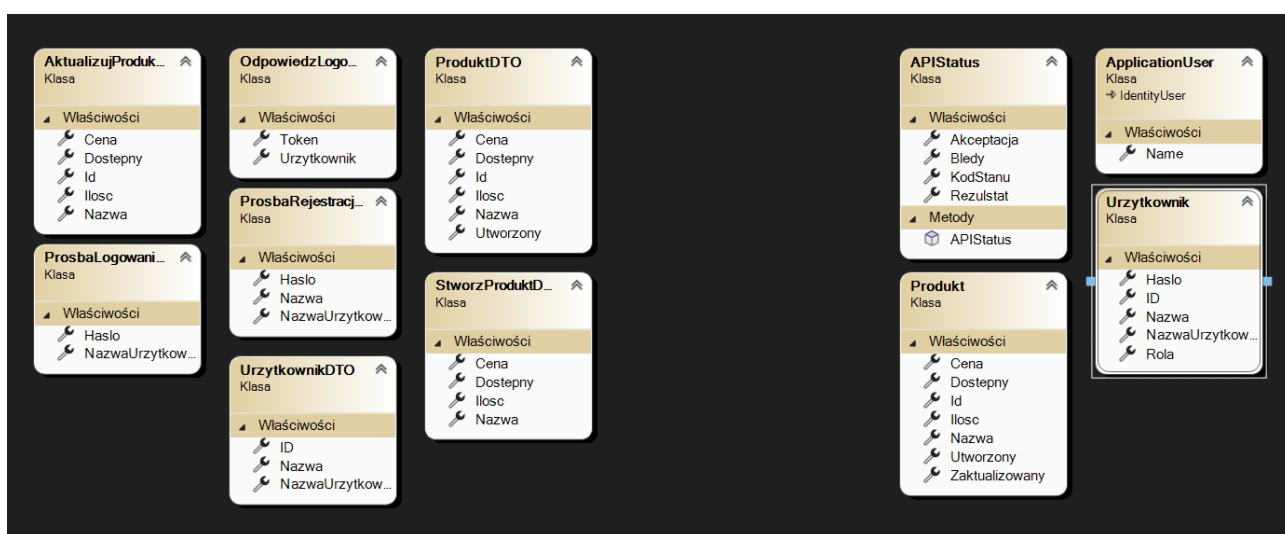
Opis

Założeniem aplikacji jest zaliczenie przedmiotu. Projekt jest to prosta aplikacja, umożliwiająca rejestrowanie użytkowników, logowania i tworzenie, odczytywanie, aktualizowanie i usuwanie elementów w bazie danych.

Rejestracja i logowanie odbywa się w bezpieczny sposób z udziałem autoryzacji oraz dostępem do ról, "admin", "customer"

Klasy Modelu

Po lewej klasy obiekt transferu danych. Po prawej pozostałe klasy modelu projektu



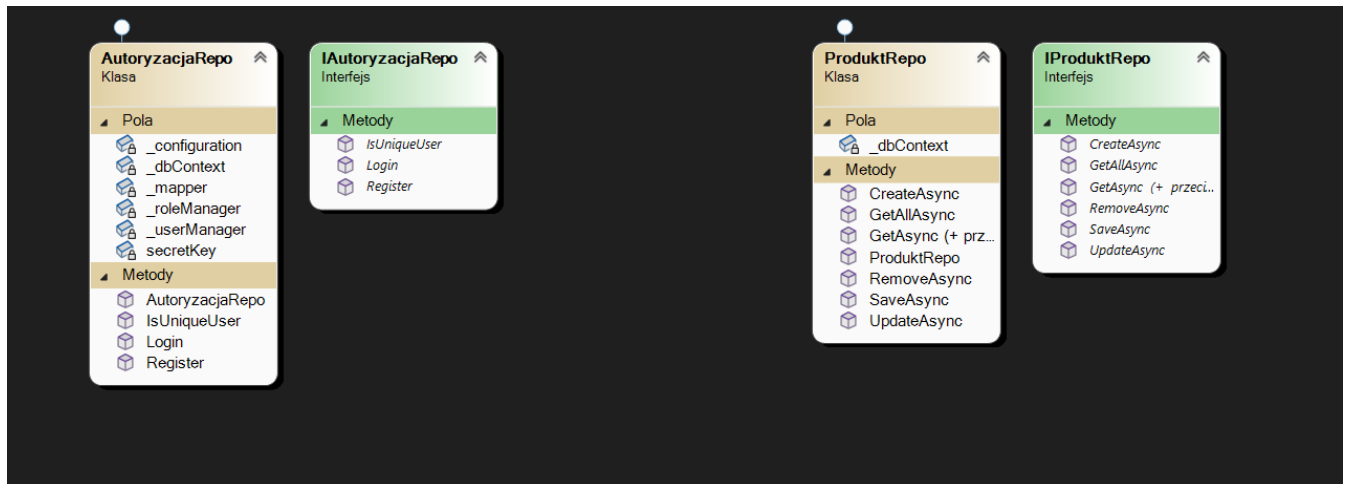
Klasy walidacji oraz końcówek

Po lewej klasy walidacji aktualizacji oraz tworzenia elementów



Klasy repozytorium

Po lewej klasa i interfejs autoryzacji odpowiadający za logowanie i rejestrację, po prawej klasa i interfejs odpowiadający za działania na tabeli produktów.



Działanie aplikacji

Do uruchomienia aplikacji należy wykonać zaktualizowanie naszej bazy poleceniem `dotnet ef database update`

W terminalu w głównym katalogu projektu. Baza zostanie utworzona z migracji znajdującej się w folderze migracje.

Należy również w pliku `appsettings.json` dodać nazwę swojego serwera. Jest on wymagany do działania aplikacji

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "DefaultConnection": "Server=LAPTOP-9A79R96U;Database=ProjectAPI;TrustServerCertificate=true;Trusted_Connection=True;MultipleActiveResultSets=true"
  },
  "ApiSettings": {
    "Secret": "Weryfikacja Tokenu JWT Sekret Token Token Token"
  }
}
```

Widok swagger

Na początku możemy dodać użytkownika rejestrujące się. Bez autoryzacji po zalogowaniu nie będziemy mogli np. usunąć rekordu, dostaniemy kod błędu 401, natomiast będziemy mogli wyświetlić produkt po id

DELETE

/api/product/{id}

Parameters

Cancel

Name	Description
id * required	
integer(\$int32)	
(path)	1

ExecuteClear

Responses

Curl

```
curl -X 'DELETE' \
'https://localhost:7223/api/product/1' \
-H 'accept: */*'
```

Request URL

```
https://localhost:7223/api/product/1
```

Server response

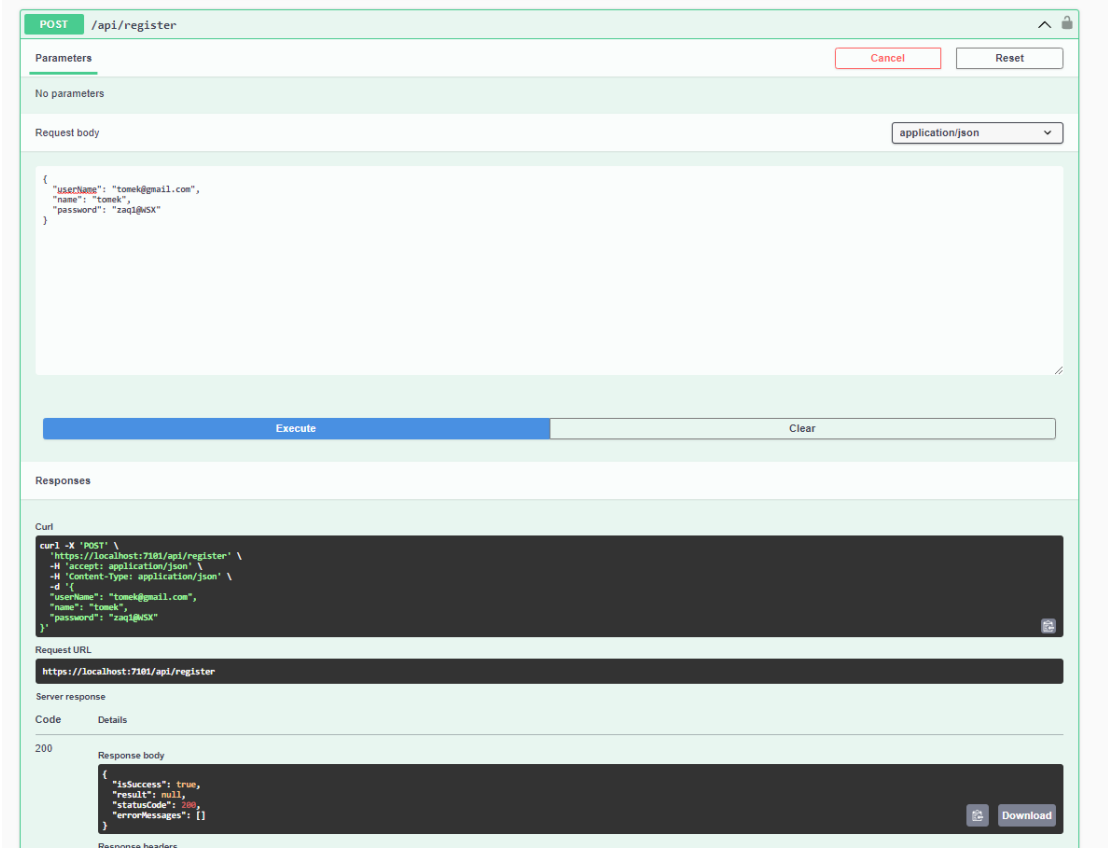
Code	Details
401	
Undocumented	Error: response status is 401

Response headers

```
content-length: 0
date: Sun, 30 Apr 2023 17:04:47 GMT
server: Kestrel
www-authenticate: Bearer
```

Responses

Rejestracja użytkownika. Identity przyjmuje hasła z minimum 1 małą, wielką literą, znakiem specjalnym oraz cyfrą, minimum 8 znaków



Results Messages

	Id	Name	UserName	NormalizedUserName	Email	NormalizedEmail	EmailConfirmed	PasswordHash
1	620fc407-1486-492f-b019-3aa640ac5894	marek	marek@gmail.com	MAREK@GMAIL.COM	marek@gmail.com	MAREK@GMAIL.COM	0	AQAAAAIAAYagAAAAEF
2	98a4c861-22cd-433b-8045-f96495d8d5ff	lolek	lolek@gmail.com	LOLEK@GMAIL.COM	lolek@gmail.com	LOLEK@GMAIL.COM	0	AQAAAAIAAYagAAAAEH
3	e36e1942-5693-4cba-a1a7-744d410bb0e9	tomek	romek@gmail.com	ROMEK@GMAIL.COM	romek@gmail.com	ROMEK@GMAIL.COM	0	AQAAAAIAAYagAAAAEH
4	f53321f3-99bd-4037-8355-5bb7cde83838	tomek	tomek@gmail.com	TOMEK@GMAIL.COM	tomek@gmail.com	TOMEK@GMAIL.COM	0	AQAAAAIAAYagAAAAEH
5	fc279012-57db-444b-9a21-d3b97dc0b078	bolek	bolek@gmail.com	BOLEK@GMAIL.COM	bolek@gmail.com	BOLEK@GMAIL.COM	0	AQAAAAIAAYagAAAAEH

Aby zarejestrować użytkownika o innej roli należy w pliku zmienić role na admin lub centromer

```
ApplicationDbContext.cs    ProduktySD.cs    AutoryzacjaRepo.cs    x
ProjectAPI    ProjectAPI.Repozytorium.AutoryzacjaRepo    Register(ProszbaRejestracjiDTO prosbaDTO)
82    UserName = prosbaDTO.NazwaUrzytkownika,
83    Name = prosbaDTO.Nazwa,
84    NormalizedEmail = prosbaDTO.NazwaUrzytkownika.ToUpper(),
85    Email = prosbaDTO.NazwaUrzytkownika
86    };
87
88    try
89    {
90        var rezultat = await _userManager.CreateAsync(urzytkownikObj, prosbaDTO.Haslo);
91        if (rezultat.Succeeded)
92        {
93            if (!_roleManager.RoleExistsAsync("admin").GetAwaiter().GetResult())
94            {
95                await _roleManager.CreateAsync(new IdentityRole("admin"));
96                await _roleManager.CreateAsync(new IdentityRole("customer"));
97            }
98            await _userManager.AddToRoleAsync(urzytkownikObj, "admin");
99
100        var uzytkownik = _dbContext.ApplicationUsers.FirstOrDefault(u => u.UserName == prosbaDTO.UserName);
101        return _mapper.Map<UrzytkownikDTO>(uzytkownik);
102    }
103    catch (Exception e)
104    {
105    }
106
107    }
108    return null;
109    }
110    }
111 }
```

Po zalogowaniu na stworzonego użytkownika otrzymujemy kod autoryzacyjny.

Responses

Curl

```
curl -X 'POST' \
'https://localhost:7223/api/login' \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
  "nazwaUzytkownika": "tomek@gmail.com",
  "haslo": "zaq!@eSX"
}'
```

Request URL

```
https://localhost:7223/api/login
```

Server response

CodeDetails

200

Response body

```
{
  "akceptacja": true,
  "resultat": {
    "uzytkownik": {
      "id": "f53321f3-99bd-4037-8355-5bb7cde83838",
      "nazwaUzytkownika": null,
      "nazwo": null,
      "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1bmVudWVmbmF7ZS16InRvbmVrIiwicm95ZS16InRvbmVrIiwiaWF0IjEwNjYyOTQ0ODQ0OTQ3LjCjE0eA10JEZDOWMmZk3NDcsIm1hdHc1ZGMTY4Jjg3NDw6N3B8.3dv67NFXfBMV7eATAKK8yKagphw2FjP0oOd4YjFZ9sk"
    },
    "kodStatus": 200,
    "body": []
  }
}
```

Response headers

```
content-type: application/json; charset=utf-8
date: Sun, 30 Apr 2023 17:15:46 GMT
server: Kestrel
```

Wprowadzamy token tak jak na schemacie

Available authorizations

Bearer (apiKey)

Wpisz 'Bearer' [spacja] a następnie token uwierzytleniania
"Bearer 123456"

Name: Authorization

In: header

Value:

Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXLTUxMjM0NTY3ODk0In0=

Authorize

Close

Po zalogowaniu jako customer możemy, zrobić wszystko poza pobraniem listy wszystkich produktów. Jest to opcja zablokowana przez barierę

ProduktKoncowki

GET/api/product

< >

Cancel

No parameters

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
https://localhost:7223/api/product' \
-H 'accept: application/json' \
-H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1bmVudWUiOmFZTSZlbnRvbmVrIlwic9sZSI6ImNlc3RvbGVyTiwibmVjcnxkYgOOC1MTQwLClleHAIOJE2ODMWNzkSNDAsInldCIGMY4Hjg3NTe0PHB. Y4U3LvPRjmfs7awsfcd'
```

Request URL

```
https://localhost:7223/api/product
```

Server response

Code	Details
403 <i>Undocumented</i>	Error: response status is 403 <p>Response headers<pre>content-length: 0 date: Sun, 30 Apr 2023 17:20:07 GMT server: Kestrel</pre></p>

Responses

Code	Description	Links
------	-------------	-------

Po zalogowaniu jako admin możemy dokonać pobrania danych

Curl

```
curl -X 'GET' \
  'https://localhost:7223/api/product' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1bmIudG9mbmFtZSI6ImlhcnVrIiwicm9sZSI6ImFkbWU1Iiwibm9mljowIjgyODc1NDc4LjIeHA1OjE2ODM0ODAyMTgsIm1hdC16MTY4Ijg3NTQxOjB9.Nwq3jJNyp03RckZ4_1x41of'
```

Request URL

```
https://localhost:7223/api/product
```

Server response

Code

Details

200

Response body

```
{
  "akceptacja": true,
  "rezultat": [
    {
      "id": 1,
      "nazwa": "Kubek",
      "cena": 10,
      "ilosc": 10,
      "dostepny": true,
      "utworzony": null,
      "zaktualizowany": null
    },
    {
      "id": 2,
      "nazwa": "Dlugopis",
      "cena": 20,
      "ilosc": 0,
      "dostepny": false,
      "utworzony": null,
      "zaktualizowany": null
    }
  ],
  "kodStanu": 200,
  "bledy": []
}
```

Response headers

```
content-type: application/json; charset=utf-8
date: Sun, 30 Apr 2023 17:26:09 GMT
server: Kestrel
```

Responses

Code

Description

Links