**Uruchomienie projektu**

1. Uruchomić CarRental.API
   1. W solucji apliakcji projektem zaznaczonym jako startowy powinien być CarRental.API
   2. Uruchomić konsolę i wydać komendę: dotnet run
      1. ---------LUB--------
   3. Uruchomić wiersz poleceń w katalogu projektu
   4. Uruchomić komendę: dotnet run
2. Uruchomić CarRental.APP
   1. Uruchomić konsolę w katalogu projektu
   2. Uruchomić komendę: dotnet run
   3. Przejść pod wskazany przez środowisko adres: w tym przypadku: http://localhost:5233

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

**Wygląd apliakcji**

**Obraz zawierający zrzut ekranu, tekst, oprogramowanie, Strona internetowa

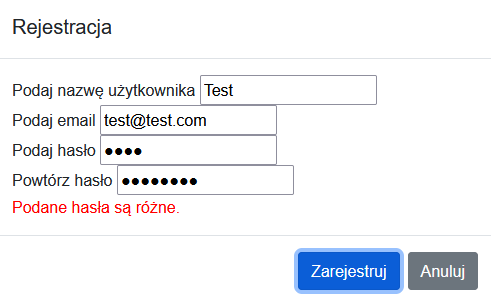
Opis wygenerowany automatycznie**Brak zalogowanego użytkownika

**Obraz zawierający tekst, numer, oprogramowanie, Czcionka

Opis wygenerowany automatycznie**

Rejestracja

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie****

W przypadku podania różnych haseł rejestracja nie uda się oraz zostanie wyświetlony komunikat

Zalogowany Admin

Obraz zawierający zrzut ekranu, tekst, oprogramowanie, System operacyjny

Opis wygenerowany automatycznie

Obraz zawierający tekst, numer, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie

Admin może dodać samochód

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, wyświetlacz

Opis wygenerowany automatycznie

Admin może usunąć samochód

Obraz zawierający zrzut ekranu, tekst, oprogramowanie, Czcionka

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Zalogowany użytkownik

W profilu posiada listę wynająć.

Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Opis wygenerowany automatycznie

Użytkownik może również usunąć wynajęty pojazd.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie

Użytkownik może dodać wynajem samochodu za pomocą przycisku Dodaj.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, numer

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

**Struktura aplikacji**

Aplikacja została stworzona w onion architecture.

Projekty:

* Frontend
  + CarRental.APP – Blazor Webserver Application
* Backend
  + CarRental.API – WebAPI
  + CarRental.Core – Library class
  + CarRental.Infrastructure – Library class

**Referencje**

CarRental.APP > CarRental.Infrastructure - aby uzyskać dostęp do obiektów DTO

CarRental.API > CarRental.Infrastructure

CarRental.Infrastructure > CarRental.Core Dodatkowo projekt

**Opis projektów**

Projekt CarRental.Core stanowi najgłębszą warstwę aplikacji. Zawiera modele encji oraz interfejsy repozytorium.

Projekt CarRental.Infrastructure jest najbardziej rozbudowany. Posiada implementację repozytoriów, interfejsy i implementacje serwisów, DTO (data transfer object) oraz część konfiguracji bazy danych.

Projekt CarRental.API posiada kontrolery, konfigurację serwisów w pliku Program.cs. Jest głownym projektem wykonywalnym udostępniającym API.

Projekt CarRental.APP stanowi frontend aplikacji. Dostarcza interfejs użytkownika umożliwiając logownanie i wyświetlanie danych. W zależności od uprawnień jakie posiada konto (admin, user) wyświetlane są odpowiednie funkcje.

**Zabezpieczenia**

Obiekty DTO pozwalają ukryć informacje, które nie powinny zostać wyświetlone w warstwie prezentacji np. hasło użytkownika albo czas aktualizacji danych.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Na przykładzie api pojazdów widać, że nie zostaje wyświetlone pole UpdatedAt.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, wyświetlacz

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

API użytkownika

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Aby zagwarantować bezpieczeństwo API nie udostępnia hasła.

API wypożyczeń

Obraz zawierający tekst, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie

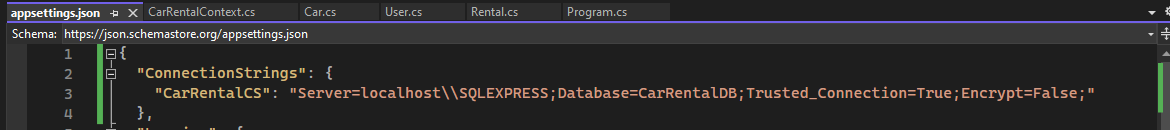
**Połączenie z bazą**

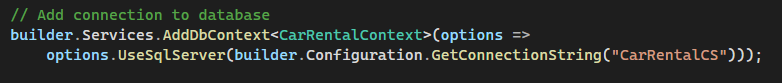
Połączenie z baza danych wykonane jest z użyciem Entity Framework Core.

**Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie**

Wymagało to konfiguracji pliku appsettings.json oraz utworzenia klasy CarRentalContext umożliwiającej dostęp do encji z bazy.



W pliku Program.cs projektu CarRental.API wstrzykiwanie zależności

**Wymagane pakiety**

CarRental.API:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

CarRental.Infrastructure

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Ikona komputerowa

Opis wygenerowany automatycznie

CarRental.APP

Obraz zawierający tekst, oprogramowanie, Oprogramowanie multimedialne, Czcionka

Opis wygenerowany automatycznie

**Tworzenie bazy danych na podstawie kodu apliakcji**

Polecenia wydawane z Package Manager Console

Obraz zawierający tekst, Oprogramowanie multimedialne, oprogramowanie, zrzut ekranu

Opis wygenerowany automatycznie

Wynikiem tego polecenia jest klasa Migrations

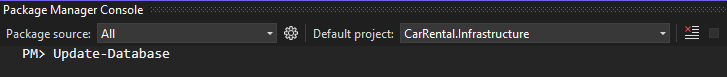
Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Na powyższym screenie widoczna jest również klasa CarRentalContext

CarRentalContext to klasa, która dziedziczy po DbContext i jest specyficznym typem kontekstu bazy danych. Kontekst bazy danych w Entity Framework Core jest główną klasą, która koordynuje funkcje EF dla modelu danych. Kontekst bazy danych to klasa, która zawiera metody do zapytań i zapisywania danych oraz zawiera DbSet<T> właściwości dla typów encji, które są zarządzane przez ten kontekst.

Aby uruchomić migrację na bazie danych, należy użyć polecenia:



Wynikiem komendy jest utworzona baza danych:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

**Repozytoria**

Implementacje repozytorium tworzone są w CarRental.Infrastructure.

Każda implementacja repozytorium posiada prywatne pole tylko do odczytu typu CarRentalContext o nazwie \_context.

To pole pozwala na dostęp do encji z bazy danych np.:

\_context.Rentals, \_context.Cars oraz \_context.Users

**Zapytania sql – wstawianie danych do bazy**

Tabela Cars

INSERT INTO [CarRentalDB].dbo.Cars (Id, Name, Description, PricePerDay, PricePerHour, UpdatedAt)

VALUES

('6ba7b810-9dad-11d1-80b4-00c04fd430c8', 'Audi A4 B8 - Sedan 2014r.', 'Wygodne auto, spełni oczekiwania każdego klienta.', 500, 50, '2023-06-28 00:00:00'),

('6ba7b810-9dad-11d1-80b4-00c04fd430c9', 'BMW M3 - Coupe 2015r.', 'Elegancki samochód sportowy dla prawdziwych entuzjastów.', 700, 70, '2023-06-28 00:00:00'),

('6ba7b810-9dad-11d1-80b4-00c04fd430ca', 'Mercedes-Benz E Class - Sedan 2018r.', 'Luksusowy i komfortowy samochód dla wymagających klientów.', 1000, 100, '2023-06-28 00:00:00'),

('6ba7b810-9dad-11d1-80b4-00c04fd430cb', 'Toyota Corolla - Sedan 2019r.', 'Niezawodny samochód dla rodziny.', 300, 30, '2023-06-28 00:00:00'),

('6ba7b810-9dad-11d1-80b4-00c04fd430cc', 'Porsche 911 - Coupe 2019r.', 'Szybki i mocny samochód dla miłośników prędkości.', 1200, 120, '2023-06-28 00:00:00'),

('6ba7b810-9dad-11d1-80b4-00c04fd430cd', 'Mazda 3 - Sedan 2020r.', 'Wygodny i ekonomiczny samochód dla każdego.', 400, 40, '2023-06-28 00:00:00'),

('6ba7b810-9dad-11d1-80b4-00c04fd430ce', 'Ford Mustang - Coupe 2018r.', 'Prawdziwy amerykański muscle car dla fanów mocnych wrażeń.', 800, 80, '2023-06-28 00:00:00'),

('6ba7b810-9dad-11d1-80b4-00c04fd430cf', 'Hyundai i30 - Hatchback 2021r.', 'Bezpieczny i komfortowy samochód, idealny na długie podróże.', 350, 35, '2023-06-28 00:00:00'),

('6ba7b810-9dad-11d1-80b4-00c04fd430d0', 'Tesla Model 3 - Sedan 2022r.', 'Najnowsza technologia i komfort jazdy w jednym.', 900, 90, '2023-06-28 00:00:00'),

('6ba7b810-9dad-11d1-80b4-00c04fd430d1', 'Volvo XC60 - SUV 2021r.', 'Bezpieczny i przestronny samochód dla dużej rodziny.', 600, 60, '2023-06-28 00:00:00'),

('6ba7b810-9dad-11d1-80b4-00c04fd430d2', 'Volkswagen Golf - Hatchback 2020r.', 'Klasyczny i niezawodny samochód na co dzień.', 400, 40, '2023-06-28 00:00:00'),

('6ba7b810-9dad-11d1-80b4-00c04fd430d3', 'Nissan GTR - Coupe 2019r.', 'Japoński supercar dla fanów szybkiej jazdy.', 1100, 110, '2023-06-28 00:00:00'),

('6ba7b810-9dad-11d1-80b4-00c04fd430d4', 'Subaru Impreza - Sedan 2018r.', 'Sportowy samochód z napędem na 4 koła, idealny do jazdy w trudnych warunkach.', 450, 45, '2023-06-28 00:00:00'),

('6ba7b810-9dad-11d1-80b4-00c04fd430d5', 'Ferrari 488 - Coupe 2020r.', 'Samochód sportowy klasy premium dla prawdziwych koneserów.', 1300, 130, '2023-06-28 00:00:00'),

('6ba7b810-9dad-11d1-80b4-00c04fd430d6', 'Peugeot 208 - Hatchback 2021r.', 'Ekonomiczny i funkcjonalny samochód do jazdy po mieście.', 300, 30, '2023-06-28 00:00:00'),

('6ba7b810-9dad-11d1-80b4-00c04fd430d7', 'Jeep Wrangler - SUV 2022r.', 'Terenowy samochód dla miłośników przygód.', 750, 75, '2023-06-28 00:00:00'),

('6ba7b810-9dad-11d1-80b4-00c04fd430d8', 'Lamborghini Aventador - Coupe 2021r.', 'Luksusowy i mocny samochód dla osób szukających wyjątkowych emocji.', 2000, 200, '2023-06-28 00:00:00'),

('6ba7b810-9dad-11d1-80b4-00c04fd430d9', 'Dacia Duster - SUV 2022r.', 'Tani i solidny samochód dla każdego.', 250, 25, '2023-06-28 00:00:00'),

('6ba7b810-9dad-11d1-80b4-00c04fd430da', 'Lexus RX - SUV 2020r.', 'Komfortowy samochód dla osób ceniących luksus.', 850, 85, '2023-06-28 00:00:00'),

('6ba7b810-9dad-11d1-80b4-00c04fd430db', 'Renault Clio - Hatchback 2021r.', 'Kompaktowy i wygodny samochód do jazdy w miejskich warunkach.', 350, 35, '2023-06-28 00:00:00');

Tabela Users

-- Dodaj pierwszego użytkownika jako admina

INSERT INTO Users (Id, Role, Name, Email, Password, CreatedAt)

VALUES ('4f77f5ef-57d6-4b0d-88e7-35e0ad5d4e71', 'admin', 'Admin', 'admin@example.com', 'Admin123', '2023-06-28');

-- Dodaj pozostałych użytkowników jako "user"

INSERT INTO Users (Id, Role, Name, Email, Password, CreatedAt)

VALUES

('2f03883a-276b-4321-96a2-162a09a4646a', 'user', 'Jan Kowalski', 'jan.kowalski@example.com', 'Jan123', '2023-06-28'),

('fbd236b1-8e83-4e43-84fb-d0416a066d76', 'user', 'Anna Nowak', 'anna.nowak@example.com', 'Anna123', '2023-06-28'),

('758c7156-5ef8-42a5-937f-ee017e06a6af', 'user', 'Marek Wojcik', 'marek.wojcik@example.com', 'Marek123', '2023-06-28'),

('7b34e5d0-3cb0-4e11-82b7-d5c8e1b0d88a', 'user', 'Ewa Kowalczyk', 'ewa.kowalczyk@example.com', 'Ewa123', '2023-06-28'),

('d8a10234-55be-4bfe-862e-d31e2a7e2d20', 'user', 'Piotr Lewandowski', 'piotr.lewandowski@example.com', 'Piotr123', '2023-06-28'),

('26d1da9d-4d6c-4e4e-95c5-092cd5328740', 'user', 'Zofia Kaminska', 'zofia.kaminska@example.com', 'Zofia123', '2023-06-28'),

('86d06f6b-6f08-4a75-ae18-9e3fb1ab1578', 'user', 'Tomasz Zielinski', 'tomasz.zielinski@example.com', 'Tomasz123', '2023-06-28'),

('47c6f4b5-5c56-43b7-a8c5-8319925be383', 'user', 'Maria Szymanska', 'maria.szymanska@example.com', 'Maria123', '2023-06-28'),

('2b2f0e76-5c6b-4cb5-952f-73b2c8a0a1c3', 'user', 'Krzysztof Wasik', 'krzysztof.wasik@example.com', 'Krzysztof123', '2023-06-28'),

('3d4b017e-7f1a-4e78-92d4-0c0b1b30112c', 'user', 'Barbara Bak', 'barbara.bak@example.com', 'Barbara123', '2023-06-28'),

('b8c85b1d-cdc0-4532-86c1-c9d4191236b1', 'user', 'Adam Kaczmarek', 'adam.kaczmarek@example.com', 'Adam123', '2023-06-28'),

('91c6daa1-dfbe-4324-a9d2-40b3e70b0e35', 'user', 'Hanna Pawlak', 'hanna.pawlak@example.com', 'Hanna123', '2023-06-28'),

('e5e77d5a-38e6-4e97-84e2-d94b5323e9eb', 'user', 'Andrzej Michalski', 'andrzej.michalski@example.com', 'Andrzej123', '2023-06-28'),

('e8b8fd3b-1d44-41d2-9e22-eb2d3cb33f6a', 'user', 'Magdalena Zajac', 'magdalena.zajac@example.com', 'Magdalena123', '2023-06-28'),

('b45bb277-3d9c-41f1-aa8b-7c42d369b1f9', 'user', 'Robert Adamski', 'robert.adamski@example.com', 'Robert123', '2023-06-28'),

('4b61d724-9b95-4d79-8b3f-d2ff7c312638', 'user', 'Monika Wrobel', 'monika.wrobel@example.com', 'Monika123', '2023-06-28'),

('a1b3ce11-875f-494a-a8dd-4fcf6434d37f', 'user', 'Wojciech Nowicki', 'wojciech.nowicki@example.com', 'Wojciech123', '2023-06-28'),

('a697e91d-bb29-4c26-81a9-7f1d209c05e7', 'user', 'Katarzyna Jankowska', 'katarzyna.jankowska@example.com', 'Katarzyna123', '2023-06-28'),

('da6e2b92-15f4-48fc-8d7c-63a3312c1c2f', 'user', 'Michal Wróbel', 'michal.wrobel@example.com', 'Michal123', '2023-06-28'),

('bebc7f7d-65cd-41d2-bb5f-3c872b2637df', 'user', 'Dorota Kaczor', 'dorota.kaczor@example.com', 'Dorota123', '2023-06-28');