PRO Szkolenie Techniczne 3

28.06.2025

Hubert Michna, w67259

**Temat**: Aplikacja serwisu hotelowego:

* Serwis Klienta
* Serwis Rezerwacji
* Serwis Promocji i Zniżek

# **Stos Technologiczny**

Kilka projektów typu Web API napisanych w języku C#, przygotowanych w Visual Studio 2022. Wykorzystanie serwera baz danych MS SQL Server. Został wykorzystany framework EntityFramework. Do testowania i sprawdzania poprawności działania programu został wykorzystany Swagger oraz Postman. Do tworzenie diagramów został wykorzystany program draw.io.

# **Instrukcja Uruchamiania**

Aby uruchomić ten projekt należy:

1. Pobrać całość projektu z GitHuba, [link](https://github.com/Hubsik456/PRO_Szkolenie_Techniczne_3).
2. Otworzyć projekt i podmienić connection string na odpowiedni (jeśli jest taka potrzeba) w plikach, metoda „OnConfiguring”:
   1. ClientDbContext.cs (projekt „Hotel”)
   2. ReservationDbContext.cs (projekt „Hotel.Rezerwacja”)
   3. DiscountDbContext.cs (projekt „Hotel.Zniżka”)
3. W Visual Studio, z górnego menu wybrać „Narzędzia”/”Menadżer Pakietów NuGet”/”Konsola Menadżera Pakietów”.
4. W konsoli wpisać „Update-Database”, ten krok należy wykonać osobno, dla każdego z projektów, przełącznik „Projekt domyślny”.
5. W eksploratorze rozwiązania, kliknąć PPM na całym rozwiązaniu, wybrać właściwości, w „Konfiguruj projekty start-upowe” ustwić na trzech projektach z API akcje na „Uruchom”. Wtedy podczas następnego kroku, zostaną uruchomione na raz wszystkie trzy API.  
   Obraz zawierający tekst, zrzut ekranu, wyświetlacz, oprogramowanie

   Zawartość wygenerowana przez AI może być niepoprawna.
6. Kliknąć przycisk „F5”. Należy przeczytać i zaakceptować wyskakujące komunikaty (takie jak np. informacje o niezweryfikowanym certyfikacie). W trzech osobnych oknach przeglądarki zostaną otwarte Swaggery z informacjami i możliwością testowania API.

# **Diagram Bazy Danych**

Baza danych składa się z trzech osobnych baz danych, które znajdują się w jednym schemacie, po jednym na każdy z projektów Web API. Każda z tabel reprezentuje daną klasę:

* Projekt „Hotel”  
  Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

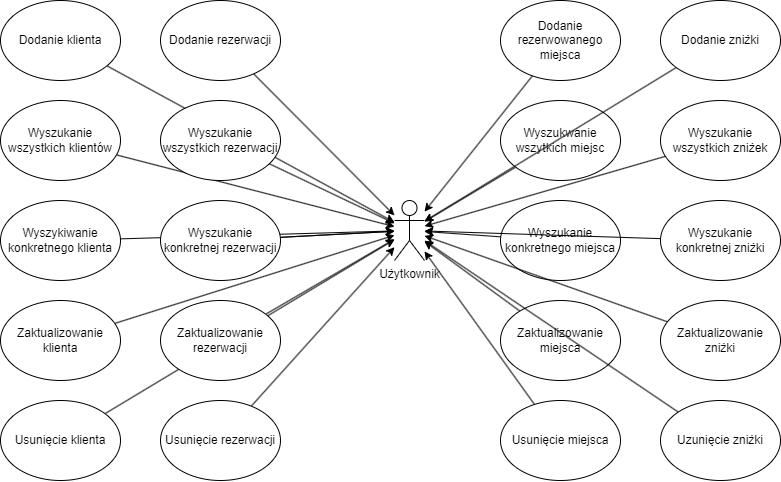
  Zawartość wygenerowana przez AI może być niepoprawna.
* Projekt „Hotel.Rezerwacja”  
  Obraz zawierający tekst, Czcionka, numer, zrzut ekranu

  Zawartość wygenerowana przez AI może być niepoprawna.
* Projekt „Hotel.Zniżka”  
  Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

  Zawartość wygenerowana przez AI może być niepoprawna.

# **Diagram przypadków użycia**

Poniżej znajduje się diagram przypadków użycia.



# **Opis Kluczowych Elementów Back-Endu**

W każdym z trzech projektów („Hotel”, „Hotel.Rezerwacja” i „Hotel.Zniżka”) można wyróżnić najważniejsze wspólne elementy, które określają działanie i logikę programu:

* **Entities** – W folderach „Entities” znajdują się klasy, które reprezentują tabele w bazie danych oraz obiekty które są wykorzystywane w API.
* **Servicy** – W folderach „Services” znajdują się serwisy, które udostępniają faktyczne metody, z których korzystają kontrolery; tutaj znajdują się np. definicje „GetByID”.
* **Controllery** – W folderach „Controller” znajdują się kontrolery, w których znajdują się endpointy i związana z nimi logika.

W projekcie „Hotel” znajduje się klasa przechowujące informacje o klientach.  
W projekcie „Hotel.Rezerwacja” znajdują się klasy przechowujące informacje o rezerwacjach i miejscach, które są rezerwowane.  
W projekcie „Hotel.Zniżka” znajduję się klasa przechowująca informacje o zniżkach dla danej rezerwacji

# **Testy jednostkowe**

W dodatkowym projekcie „Hotel.Testy” znajdują się trzy klasy z testami jednostkowymi dla poszczególnych klas reprezentujących tabele z bazy danych.

Poniżej znajduje się zrzut ekranu z pomyślnym wynikiem testów jednostkowych.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Zawartość wygenerowana przez AI może być niepoprawna.

# **Testy integracyjne**

Testy integracyjne zostały przeprowadzone przy użycia Postmana, poniżej znajdują się zrzuty ekranów z niektórych testów.

## **Hotel – Klienci, GET (Wszystko)**

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Zawartość wygenerowana przez AI może być niepoprawna.

## **Hotel – Klienci, GET (ID)**

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Zawartość wygenerowana przez AI może być niepoprawna.

## **Hotel – Klienci, POST**

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Zawartość wygenerowana przez AI może być niepoprawna.

## **Hotel – Klienci, PUT**

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Zawartość wygenerowana przez AI może być niepoprawna.

## **Hotel – Klienci, DELETE**

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Zawartość wygenerowana przez AI może być niepoprawna.

# **Opis Przypadków Testowych**

Poniżej znajduje się kilka przykładowych przypadków testowych dla API „Hotel”:

1. Mając dane o kliencie, którego chcemy przesłać do API  
   Kiedy wyślemy żądanie na endpoint „POST” z treścią żądania  
   Wtedy zostanie dodany nowy klient.
2. Mając informacje o tym, że chcemy wyszukać wszystkich klientów poprzez API  
   Kiedy wyślemy żądanie na endpoint „GET”.  
   Wtedy zostanie zwrócony JSON z danymi wszystkich klientów.
3. Mając ID konkretnego klienta, którego chcemy wyszukać poprzez API  
   Kiedy wyślemy żądanie na endpoint „GET” z podanym ID  
   Wtedy zostanie zwrócony JSON z danymi klienta.
4. Mając ID klienta, którego chcemy zaktualizować poprzez API  
   Kiedy wyślemy żądanie na endpoint „PUT” z podanym ID i treścią żądania  
   Wtedy zostanie zaktualizowany klient w API.
5. Mając ID klienta, którego chcemy usunąć z API  
   Kiedy wyślemy żądanie na endpoint „DELETE” z podanym ID  
   Wtedy zostanie usunięty klient w API.

# **Literatura**

* <https://learn.microsoft.com/pl-pl/aspnet/core/tutorials/first-web-api?view=aspnetcore-9.0&tabs=visual-studio> –Tworzenie Web API, z dnia 07.07.2025
* <https://plawgo.pl/2019/08/06/postman-asserty-w-automatycznych-testach/> – Postman, z dnia 07.07.2025
* <https://learn.microsoft.com/pl-pl/ef/> – Entity Framework, z dnia 07.07.2025
* <https://learn.microsoft.com/en-us/visualstudio/test/getting-started-with-unit-testing?view=vs-2022&tabs=dotnet%2Cmstest> – Testy jednostkowe, z dnia 07.07.2025