

# Aplikacja Tasker



Hubert Jarząbek  
Krzysztof Krawczyk  
K52

# Spis treści

1. Wprowadzenie
2. Konfiguracja
3. Design
4. Funkcjonalności
5. Kod źródłowy

# Wprowadzenie

W natłoku codziennych obowiązków, łatwo zapomnieć o rzeczach, które trzeba jeszcze zrobić. Dzięki naszej stronie internetowej z łatwością będziesz mógł zaplanować swój dzień. Korzystając z Tasker nie zapomnimy już o rzeczach, które mamy jeszcze do zrobienia. W Tasker będziesz mógł zrobić swoją własną checkliste, która z pewnością przypadnie Ci do gustu. Aplikacja będzie darmowa dla wszystkich, a jej prostota i wygoda w obsłudze sprawia, że nawet największy technologiczny troglodyta poradzi sobie w jej obsłudze.

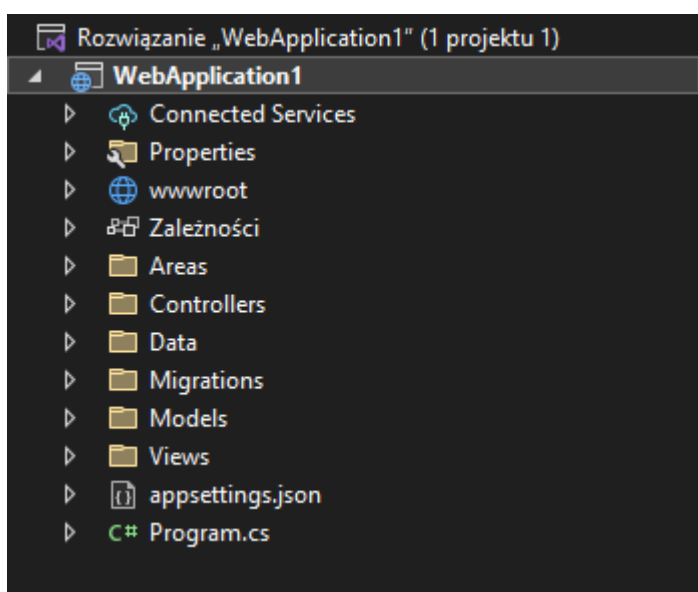
Aplikacja pozwala rozplanować sobie swoją listę obowiązków. W Tasker będzie możliwość rozpisania sobie listy obowiązków. Aplikacja pozwala na stworzenie nowej kilku oddzielnych list zadań a w niej będziemy mogli: dodawać swoje notatki, edytować swoje notatki, usuwać swoje notatki, przeglądać utworzone notatki a także je przeglądać.

# Konfiguracja

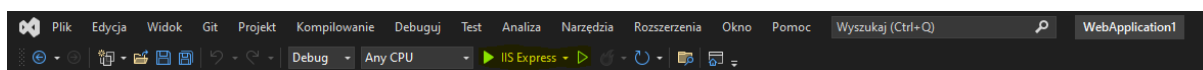
## 1. Konfiguracja projektu

Aby poprawnie odpalić projekt na maszynie lokalnej, należy pobrać repozytorium z GitHub: <https://github.com/Swaggin/apka-todo>.

Następnym krokiem jest otwarcie projektu z wykorzystaniem programu Visual Studio 2022. Powinniśmy otrzymać następujący schemat plików:



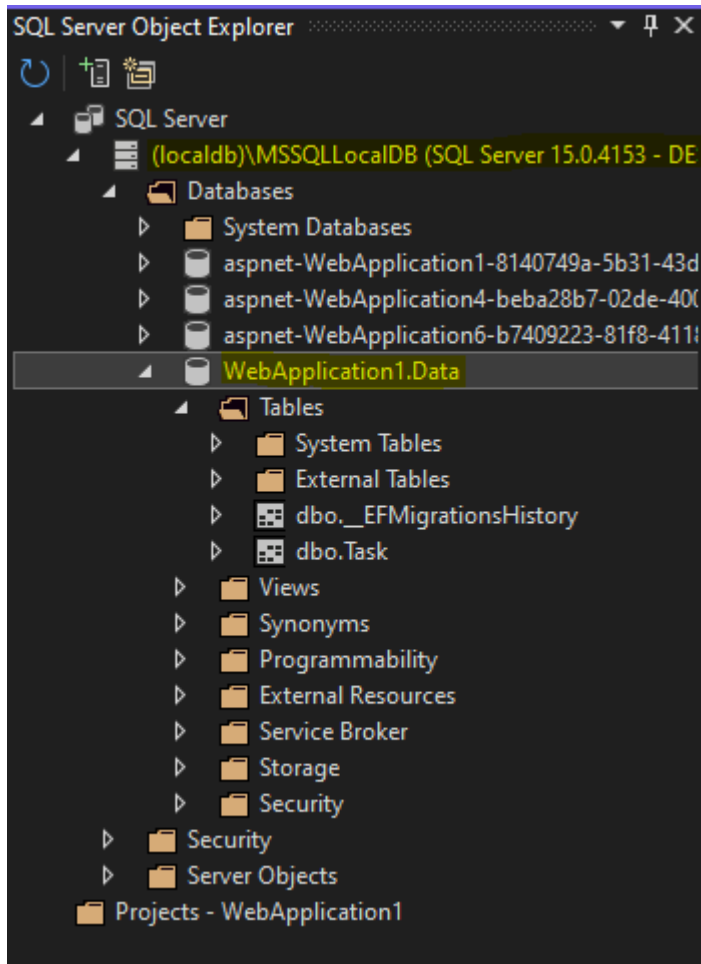
Aby móc podejrzeć stronę lokalnie, należy wybrać opcję IIS Express na pasku akcji:



Projekt powinien się odpalić bez żadnych problemów. Najważniejszymi katalogami tego projektu są kolejno: **Controllers** (odpowiedzialne za działanie strony głównej oraz funkcjonowanie widoku związanego z dodawaniem, usuwaniem oraz edycją każdego taska), **Models** (odpowiedzialne za utworzenie modelu Taska oraz inicjacja kolumn w tabeli bazy danych) oraz **Views** (odpowiedzialne za wyświetlanie danych na odpowiednich widokach).

## 2. Konfiguracja bazy danych

Baza danych projektu powinna utworzyć się sama przy próbie odpalenia projektu. W zakładce SQL Server Object Explorer powinniśmy posiadać bazę:



Najważniejszą częścią bazy danych jest tabela **dbo.Task**, w której przechowywane są identyfikatory oraz treść każdego z tasków:

Name	Data Type	Allow Nulls	Default
Id	int	<input type="checkbox"/>	
Content	nvarchar(MAX)	<input type="checkbox"/>	

### 3. Migracja projektu na Azure Cloud

Aplikacja działająca lokalnie została zmigrowana na osobny adres URL dostępny w sieci pod adresem: <https://webapplication120230107184240.azurewebsites.net/>.

Wszelkie pliki oraz konfiguracje, włącznie z bazą danych zostały przeniesione na serwer Azure, następnie została utworzona osobna baza danych, którą należało podłączyć w miejsce aktualnej bazy lokalnej. Aby tego dokonać, w panelu administracyjnym serwera należało utworzyć nową bazę danych, odblokować lokalny adres IP komputera, na którym projekt został postawiony.

#### Reguły zapory

Zezwalaj określonym publicznym internetowym adresom IP na dostęp do zasobu. [Dowiedz się więcej](#)

+ Dodaj swój adres IPv4 klienta [redacted].66 + Dodaj regułę zapory

Nazwa reguły	Początkowy adres IPv4	Końcowy adres IPv4	
ClientIPAddress_2023-1-7_19-7-9	[redacted].66	[redacted].66	

Ostatnim krokiem do poprawnego podpięcia bazy jest podmiana string'a odpowiedzialnego za połączenie ze starego, lokalnego adresu na nowy, dostępny z poziomu widoku **Publish** w Visual Studio 2022:

```
"ConnectionStrings": {
  "DefaultConnection": "Data Source=tcp:webapplication1dbservercv2137.database.windows.net,1433;Initial Catalog=WebApplication1_db;User Id=webapproot@webapplication1dbservercv2137;Password=;MultipleActiveResultSets=true",
  "WebApplicationDbContext": "Server=(localdb)\\mssqllocaldb;Database=WebApplication1_Data;Trusted_Connection=True;MultipleActiveResultSets=true",
  "WebApplicationDbContext": "Data Source=tcp:webapplication1dbservercv2137.database.windows.net,1433;Initial Catalog=WebApplication1_db;User Id=webapproot@webapplication1dbservercv2137;Password=;MultipleActiveResultSets=true"
},
"Logging": {
  "LogLevel": {
    "Default": "Information",
    "Microsoft.AspNetCore": "Warning"
  }
},
"AllowedHosts": "*"
}
```

# Design

Oprawa wizualna aplikacji jest przejrzysta oraz intuicyjna. Interfejs składa się z nagłówka oraz głównej sekcji strony.

Z poziomu nagłówka użytkownik ma możliwość przejścia do ustawień użytkownika oraz całkowitego wylogowania się z bieżącej sesji.

Główna sekcja strony zawiera grupy zadań w postaci kafelków, w skład których wchodzi m. in. opcja dodawania zadań oraz lista zadań, dzięki której użytkownik będzie mógł decydować o każdym elemencie z listy. Każdy element w liście posiada dwie akcje - edycja treści zadania oraz usunięcie.

Design z założenia zaprojektowany jest pod kątem mobile-first, natomiast jest w pełni obsługiwany na desktop. Domyślnie wyświetlany jest widok mobilny.

Lista		
<a href="#">Stwórz</a>		
Zadanie		
Wyjść z psem	<a href="#">Edytuj</a>	<a href="#">Usuń</a>
Wynieść śmieci	<a href="#">Edytuj</a>	<a href="#">Usuń</a>

# Funkcjonalności

## 1. Podstawowe zasady działania aplikacji:

- Rejestracja użytkownika
- Logowanie
- Zarządzanie kontem
- Stworzenie listy zadań
- Edytowanie listy zadań
- Przeglądanie listy zadań



## 2. Diagram użycia:



# Kod źródłowy i demo

1. GitHub - repozytorium: <https://github.com/Swaggin/apka-todo/tree/main>
2. Demo Live: <https://webapplication120230107184240.azurewebsites.net/>