Wojskowa Akademia Techniczna im. Jarosława Dąbrowskiego



Wydział Cybernetyki, kierunek informatyka - inżynieria systemów

Sprawozdanie z laboratorium z przedmiotu:

Standarty w projektowaniu systemów dialogowych

Temat laboratoriów:

Tworzenie systemu dialogowego w C#

Opracował: Radosław Relidzyński, Grupa: WCY23IX3S4

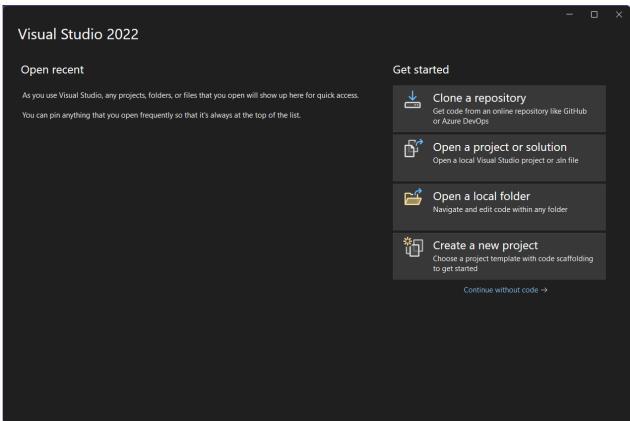
Spis treści

Wstęp	3
Krok 1: Instalacja narzędzi	3
Krok 2: Tworzenie nowego projektu botowego Krok 3: Konfiguracja projektu	4
Krok 4: Tworzenie podstawowych klas	4
Krok 5: Testowanie bota	5
Instalacja i uruchomienie emulatora	5
Uruchomienie bota	5
Krok 6: Rozszerzenie funkcjonalności	7
Uruchomienie bota po rozszerzeniach	8
Podsumowanie:	9

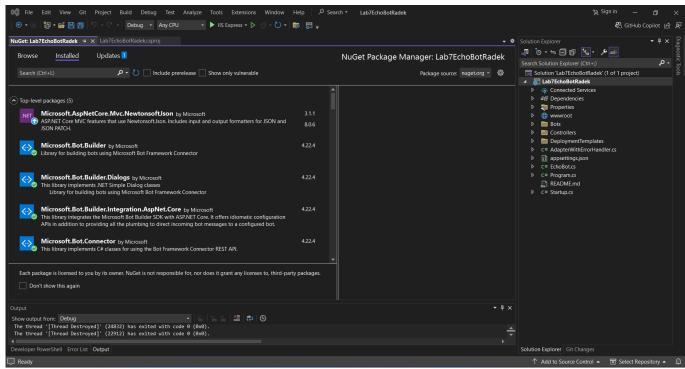
Wstęp

Zadaniem jest stworzenie systemu dialogowego przy pomocy Visual Studio z dodatkami do pracy ASP .NET Core oraz Bot Framework SDK.

Krok 1: Instalacja narzędzi



Krok 2: Tworzenie nowego projektu botowego Krok 3: Konfiguracja projektu



Krok 4: Tworzenie podstawowych klas

```
€ GitHub Copilot 🖻 🎘
EchoBot.cs 🖘 🗙 AdapterWithErrorHandler.cs
                                                                                                                                                                   ・ 『 OnMessageActivityAsync(ITurnContext<IMessageActivity> turnC・ 中 。 同 で 「 っ メ
                                                                                - %Lab7EchoBotRadek.EchoBot
                      adek

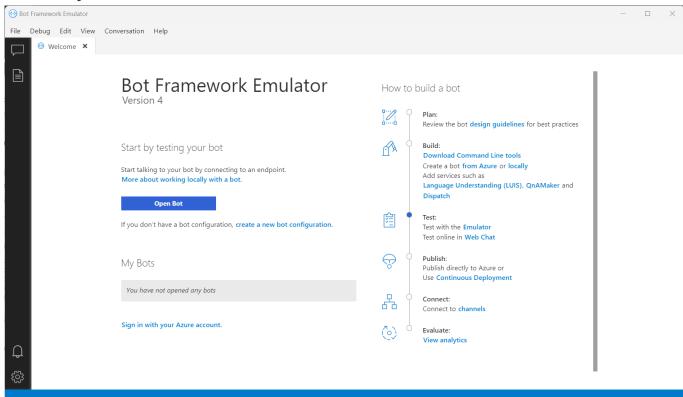
"using Microsoft.Bot.Builder;

using Microsoft.Bot.Schema;
using System.Collections.Generic;
using System.Threading;
using System.Threading.Tasks;
                                                                                                                                                                                                                                                               Search Solution Explorer (Ctrl+;)

☐ Solution 'Lab7EchoBotRadek' (1 of 1 project)
                                                                                                                                                                                                                                                                 ∨namespace Lab7EchoBotRadek
                                   Oreferences
protected override async Task OnMessageActivityAsync(ITurnContext<IMessageActivity>
turnContext, CancellationToken cancellationToken)
                                                                                                                                                                                                                                                                    C# AdapterWithErrorHandler.cs
that appsettings.json
C# EchoBot.cs
                                          var replyText = $"Echo: {turnContext.Activity.Text}";
await turnContext.SendActivityAsync(MessageFactory.Text(replyText, replyText),
cancellationToken);
                                    protected override async Task OnMembersAddedAsync(IList<ChannelAccount>
membersAdded, ITurnContext<IConversationUpdateActivity> turnContext,
CancellationToken cancellationToken)
                                                      await turnContext.SendActivityAsync(MessageFactory.Text("Hello and welcome!"),
cancellationToken);
 Show output from: Debug
The thread '[Thread Destroyed]' (24832) has exited with code 0 (0x0).
The thread '[Thread Destroyed]' (22912) has exited with code 0 (0x0).
  Developer PowerShell Error List Output
                                                                                                                                                                                                                                                                     ↑ Add to Source Control 🔺 🔟 Select Repository 🔺
```

Krok 5: Testowanie bota

Instalacja i uruchomienie emulatora

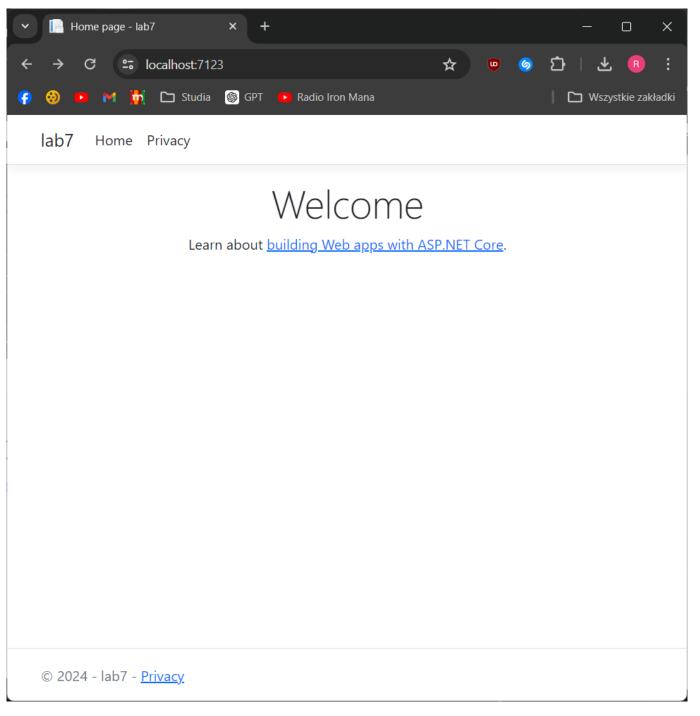


Uruchomienie bota

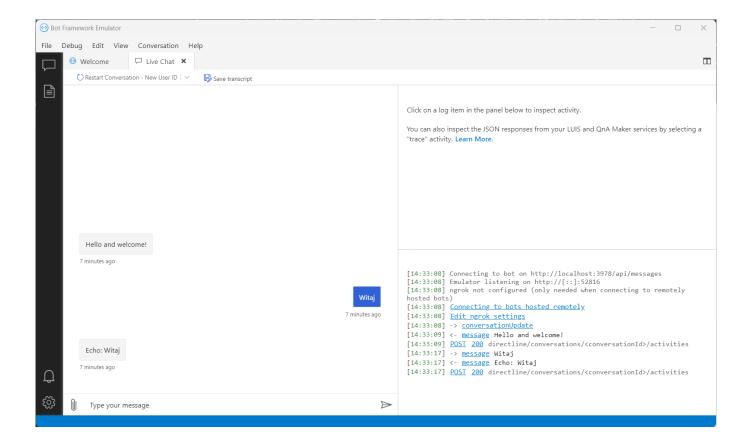
Uprzednio należy dodać te 2 linijki do "ConfigureServices":

services.AddSingleton<ICredentialProvider, ConfigurationCredentialProvider>(); services.AddSingleton<IBotFrameworkHttpAdapter, AdapterWithErrorHandler>();

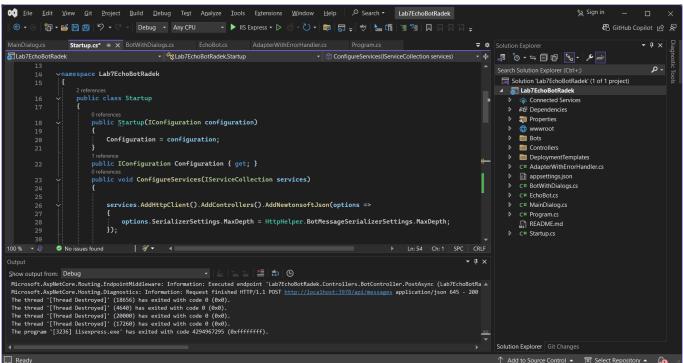
Okno potwierdzające uruchomienie bota:



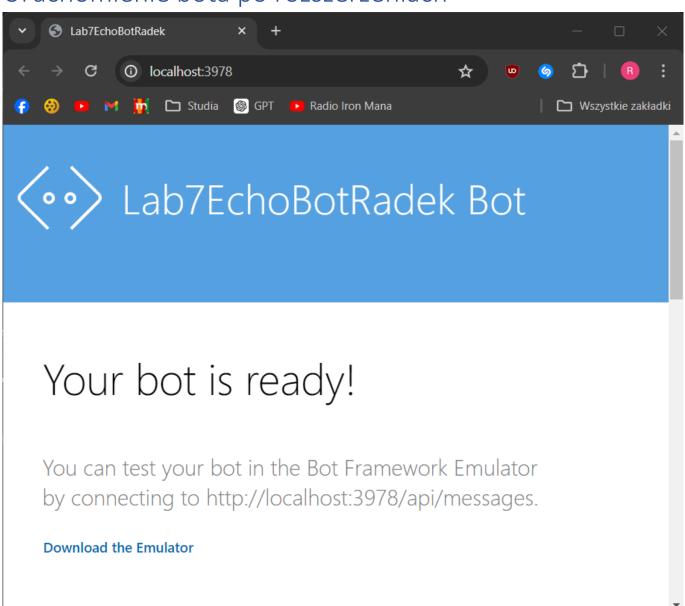
Połączenie z botem w emulatorze:

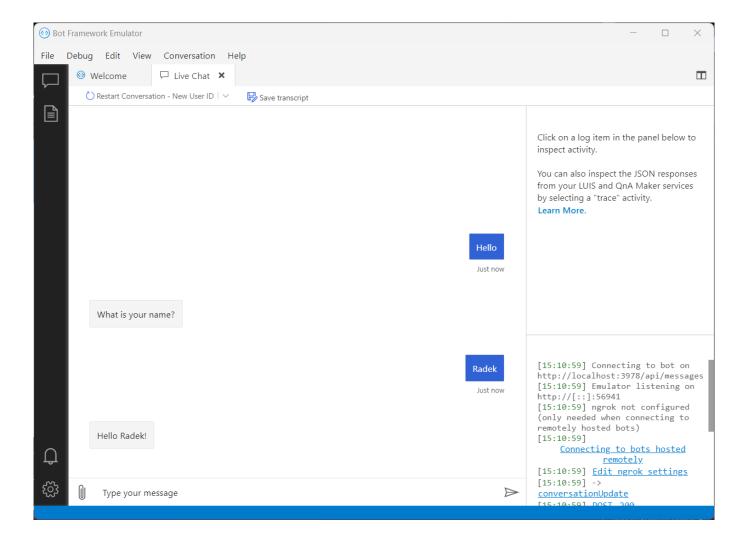


Krok 6: Rozszerzenie funkcjonalności



Uruchomienie bota po rozszerzeniach





Podsumowanie:

W ramach ćwiczenia udało się stworzyć chatbota w C# w ramach Bot Framework SDK. Udało się skutecznie połączyć z nim przy pomocy bot framework emulator.