

Web Sockets – krótko i na temat przykładzie SignalR

Michał „Azkel” Smyk
smyk.it

14.11.2018

Ja

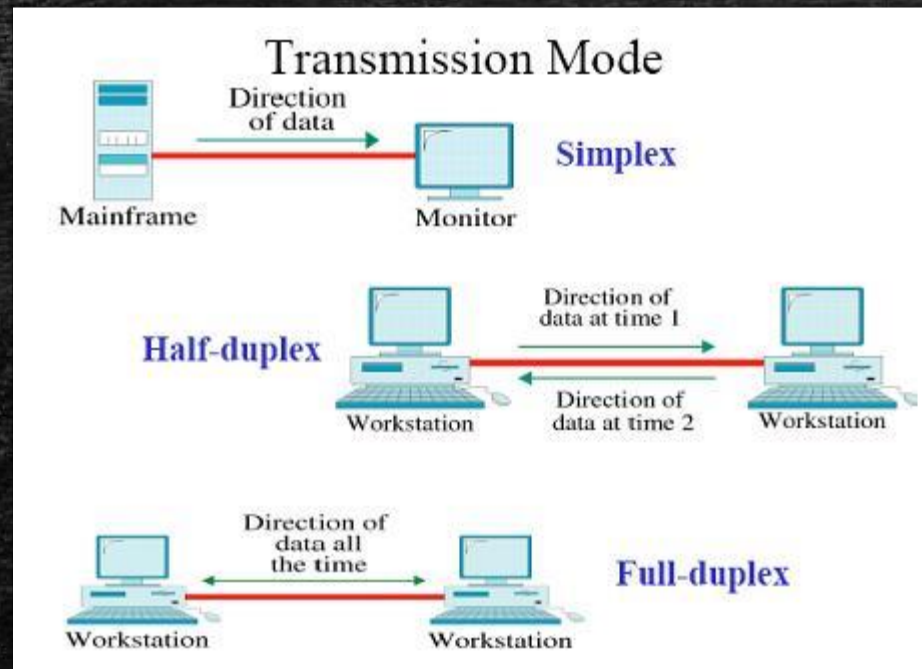
- Programista .NET (aktualny pracodawca: Unit4 Poland)
- mgr inż. (specjalizacja Inteligentne Systemy Informatyczne na W8, PWr)
- Kiedyś prezesowałem w ASI
- Twórca imprez (kilkanaście [konwentów mangowych](#) i [Sesje Linuksowe 11-15](#))
- [Microsoft Certified Solutions Associate](#) - Web Applications
- [Bawiłem się .NETem na Linuxie zanim to było modne](#)

Na początek pogadajmy o
dupach

Połączenia typu Duplex

Rodzaje połączeń

Przykłady



- Simplex: np. fale radiowe
- Half-duplex: dowolny request HTTP
- Duplex – Web Sockety

Web Sockets

- Zapewniają połączenie full-duplex między przeglądarką a serwerem
- Łatwe do implementacji, ~~duża gama dostępnych technologii*~~
- Przykładowe zastosowania:
 - Chat
 - Powiadomienia na Facebooku
 - Gry przeglądarkowe

* - no prawie, szczegóły na kolejnym slajdzie

Dostępne technologie

Dostępne języki

- Tools per Language
 - Agnostic
 - C
 - C++
 - C#
 - Erlang
 - Go
 - Haskell
 - Java VM
 - Clojure
 - Java
 - Scala
 - Node.js
 - Perl
 - PHP
 - Python
 - Ruby

Przykładowe biblioteki Back-endowe

- .NET ([SignalR](#))
- PHP ([Ratchet](#))*
- Java ([ServerSocket](#) w pakiecie *java.net*)*
- Python ([websockets](#))*
- Ruby ([websocket](#))*
- Pewnie jakieś inne technologie umożliwiające Web Development

* - pierwsze wyniki z Googla, więc nie wypowiem się czy warto czy nie

SignalR vs SignalR Core

SignalR

- github.com/SignalR/SignalR
- .NET Framework
- Windows only
- 4 developerów

SignalR Core

- github.com/aspnet/SignalR
- .NET Standard (aktualnie 2.0)
- Linux / Windows / macOS
- 1 developer (twórca SignalR)

Obie wersje są rozwijane równocześnie, niezależnie od siebie.

Dostępne technologie - SignalR

SignalR jako odbiorca

- [.NET](#)
- [Java](#)
- [JavaScript](#)
- [C++](#)

SignalR Core jako odbiorca

- [.NET](#)
- [Java](#) (8+)
- [JavaScript](#)

Brak wsparcia, ale ponoć działa:

- [C++](#)
- [Swift](#)

Jak to działa w SignalR?

Klient

- Wysyła do Serwera subskrypcję
- Tworzy Event Listenery
- Nasłuchuje eventów i reaguje na nie za pomocą metod z Listenerów

Serwer (Hub)

- Nasłuchuje Subskrypcji
- Może sortować użytkowników na grupy
- Wysyła informację, że dany event został wywołany wraz z paczką danych

Disclaimer

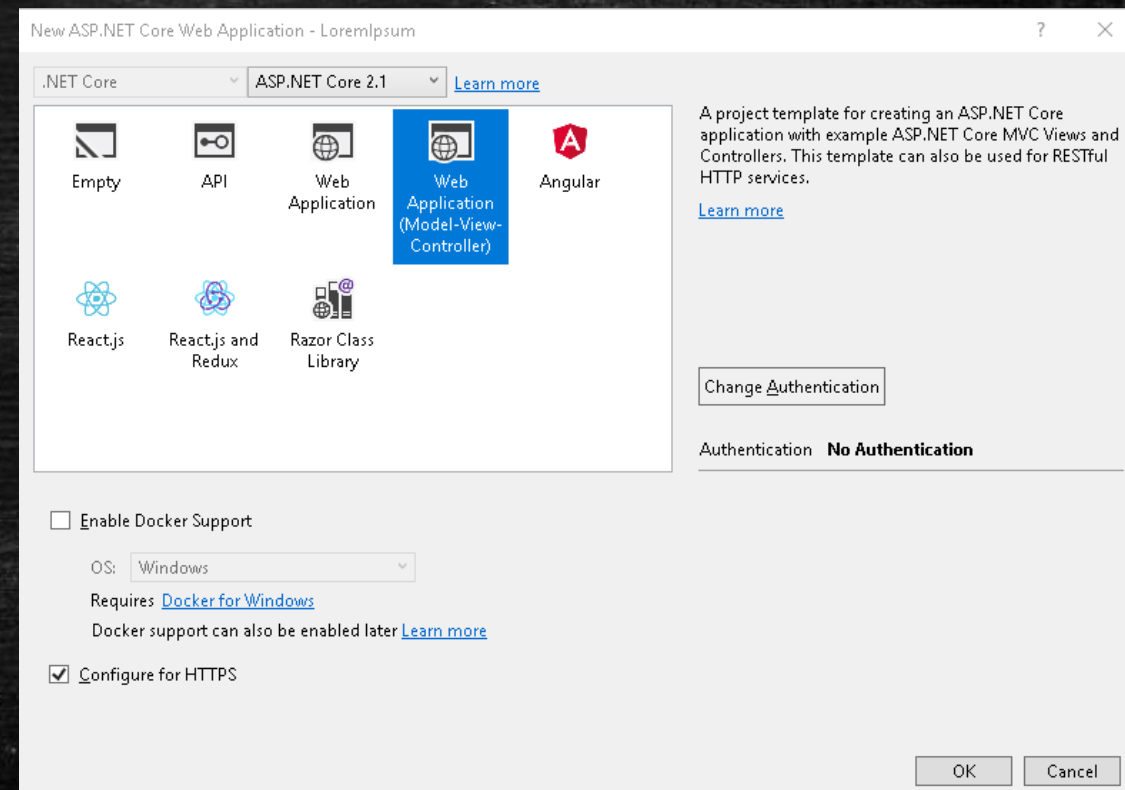
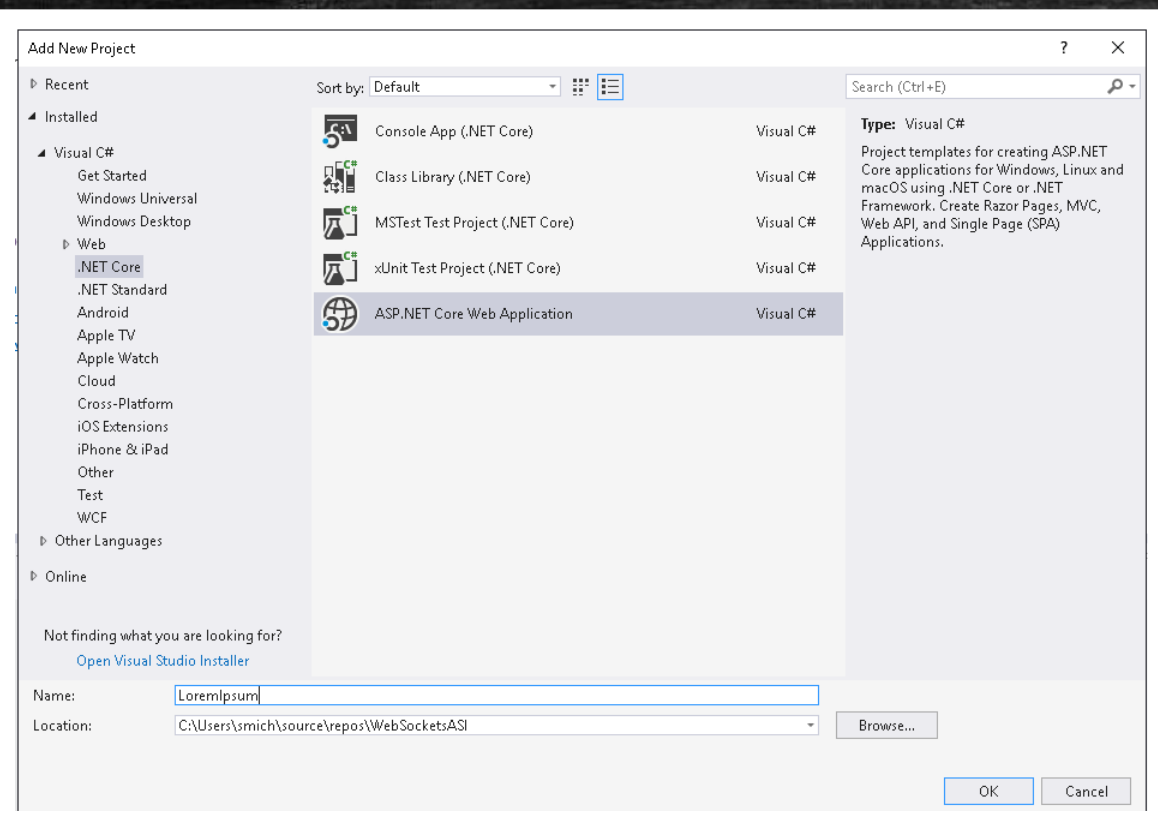
Generalnie przykłady kodu są lepiej widoczne na jasnym tle.

Stąd też kolor środowiska na kolejnych slajdach też jest biały.

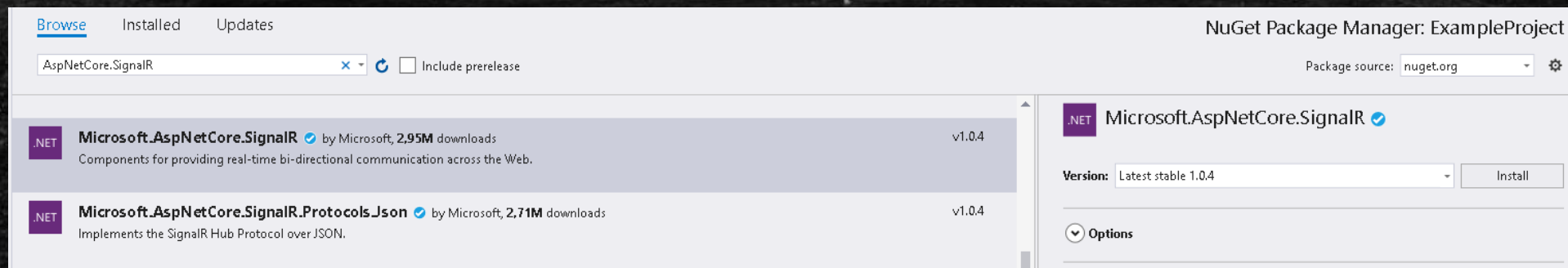
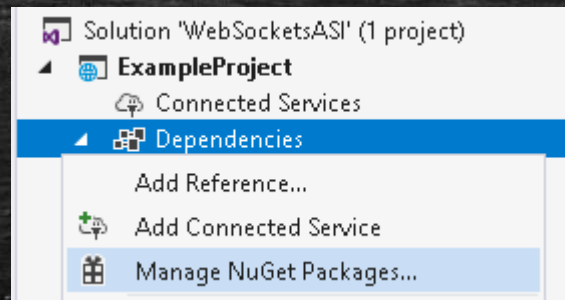
Nie jestem potworem, to tylko na potrzeby prezentacji.



Przykład – jak stworzyć własny chat



Przykład – jak zacząć



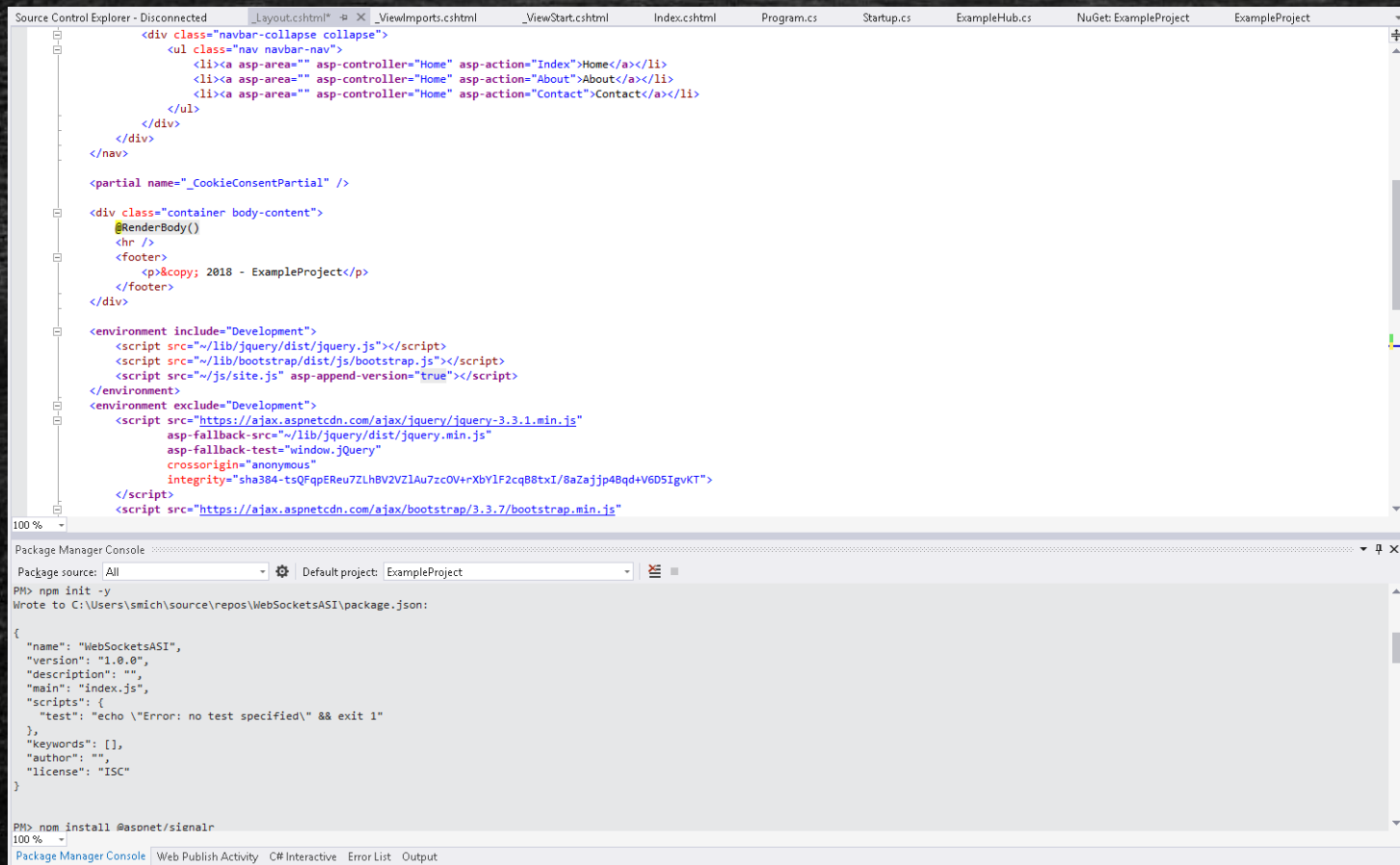
Przykładowy Hub

```
1 using Microsoft.AspNetCore.SignalR;
2 using System;
3 using System.Threading.Tasks;
4
5 namespace ExampleProject.Hubs
6 {
7     public class ExampleHub : Hub
8     {
9         ///
10        public override async Task OnConnectedAsync()
11        {
12            // Jeżeli dołączył na kanał, wyślij do wszystkich informację, że dołączył.
13            await Clients.All.SendAsync("SendToClient", $"{Context.ConnectionId} joined");
14        }
15
16        public Task SendToServer(string message)
17        {
18            // Wyślij wiadomość od użytkownika.
19            return Clients.All.SendAsync("SendToClient", $"{Context.ConnectionId}: {message}");
20        }
21
22        public override async Task OnDisconnectedAsync(Exception exception)
23        {
24            // Jeżeli wyszedł z kanału, wyślij do wszystkich informację, że wyszedł.
25            await Clients.All.SendAsync("SendToClient", $"{Context.ConnectionId} left.");
26        }
27    }
28 }
```

Startup.cs – niech SignalR uruchamia się przy starcie

```
1 using System;
2 using ExampleProject.Hubs;
3 using Microsoft.AspNetCore.Builder;
4 using Microsoft.AspNetCore.Hosting;
5 using Microsoft.AspNetCore.Http;
6 using Microsoft.AspNetCore.Mvc;
7 using Microsoft.Extensions.Configuration;
8 using Microsoft.Extensions.DependencyInjection;
9
10 namespace ExampleProject
11 {
12     public class Startup
13     {
14         (... - Not-important for presentation
15         // This method gets called by the runtime. Use this method to add services to the container.
16         public void ConfigureServices(IServiceCollection services)
17         {
18             services.AddSignalR(options =>
19             {
20                 // Faster pings for testing
21                 options.KeepAliveInterval = TimeSpan.FromSeconds(5);
22             });
23         (... Not-important for presentation
24         }
25
26         // This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
27         public void Configure(IApplicationBuilder app, IHostingEnvironment env)
28         {
29             (... - Not-important for presentation
30
31             app.UseSignalR(routes =>
32             {
33                 routes.MapHub<ExampleHub>("/example");
34             });
35             // WAŻNE - UseMvc musi być dopiero po UseSignalR!
36             app.UseMvc(routes =>{..});
37         }
38     }
39 }
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
```

Barbarzyńskie dodawanie bibliotek part 1



The screenshot shows the Source Control Explorer window with several files open. The main window displays the code for a navigation bar and a footer. The Package Manager Console shows the following commands and output:

```
PH> npm init -y
Wrote to C:\Users\smich\source\repos\WebSocketsASI\package.json:

{
  "name": "WebSocketsASI",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}

PH> npm install @aspnet/signalr
100 %
```

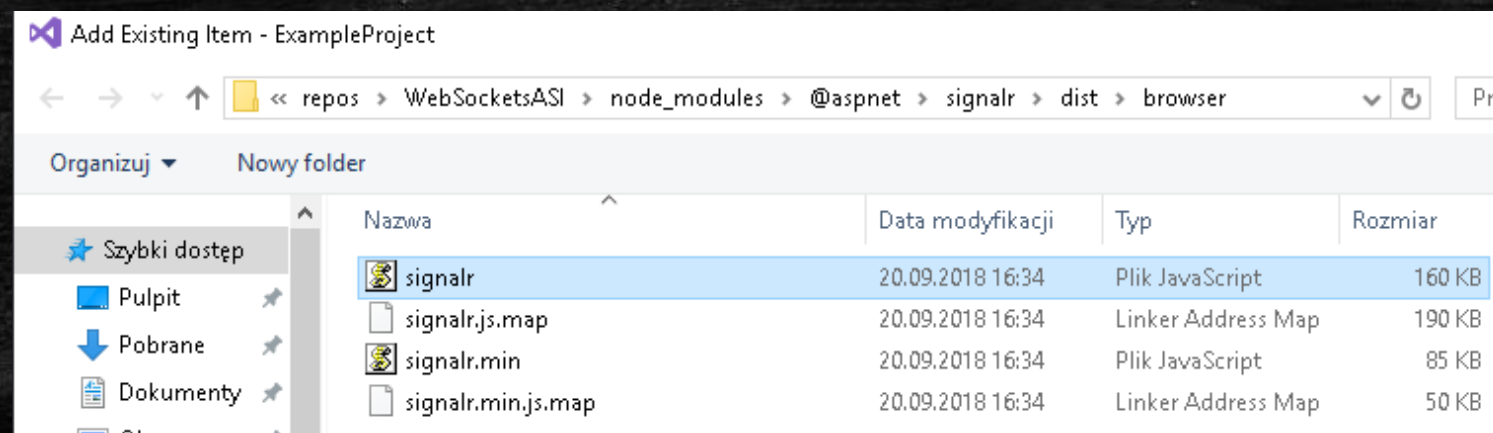
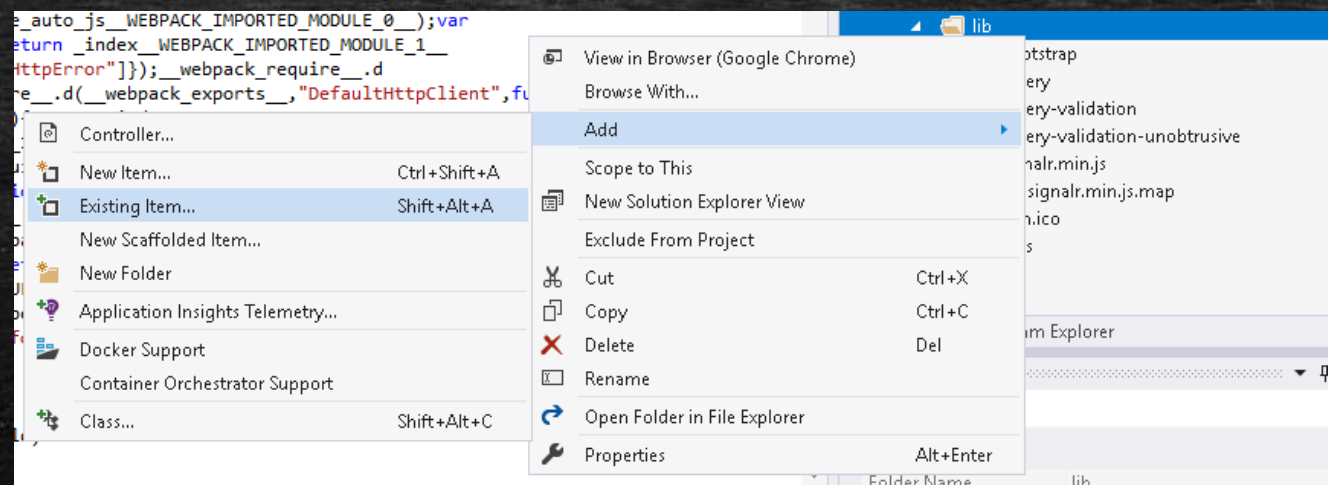
Nowy projekt NPM:
npm init -y

Zainstaluj klienta JS
SignalR:

**npm install
@aspnet/signalr**

Żeby to zadziało trzeba mieć NPMa zainstalowanego z Visual Studio, bez tego ani rusz.

Barbarzyńskie dodawanie bibliotek part 2



Widok – Index.cshtml

```
<h1>Chat</h1>
<form id="sendmessage"> <!-- Formularz do wysłania wiadomości -->
  <input type="text" id="data" />
  <button type="button" id="send">Wyślij</button>
</form>

<ul id="messages"></ul> <!-- Lista naszych wiadomości! -->
@section Scripts { <!-- Takie fajne ASP.Netowe coś, co wstrzykuje zawarty wewnątrz kod w miejsce w _Layout.cshtml gdzie jest napisane RenderSection("Scripts", required: false) -->
  <script src="~/lib/signalr.min.js"></script>

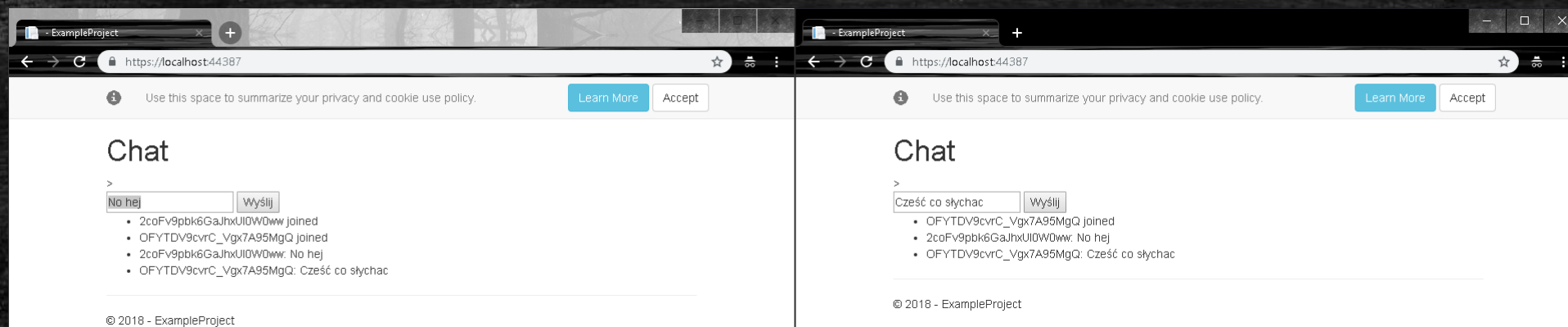
  <script>
    const connection = new signalR.HubConnectionBuilder()
      .withUrl("/example") // Zadeklarowany wcześniej w Startup.cs adres Huba
      .build();

    connection.on('SendToClient', data => {
      let child = document.createElement('li'); // Jeżeli przyjdzie z Huba metoda "SendToClient"
      child.innerText = data; // Stwórz nowy element listy
      document.getElementById('messages').appendChild(child); // Wypełnij go otrzymanym z Huba tekstem
    }); // Doklej na koniec listy wiadomości

    $('#send').on('click', function (event) {
      connection.send('SendToServer', $("#data").val()); // Jeżeli naciśnięty zostanie przycisk "Wyślij"
    }); // Wyślij zawartość pola tekstowego o id "data" do metody "SendToServer" Huba

    connection.start().catch(err => console.error(err.toString())); // Nawiąż połączenie z Hubem - jak coś pójdzie nie tak, wyrzuć do konsoli błąd
  </script>
}
```

Rezultat końcowy



Linki

- Prezentacja: <https://smyk.it/files/WebSockets.pdf>
- Repo: <https://github.com/Azkel/SignalR-ASI>
- SignalR Core: <https://github.com/aspnet/SignalR>

Dzięki za uwagę.

Pytania?