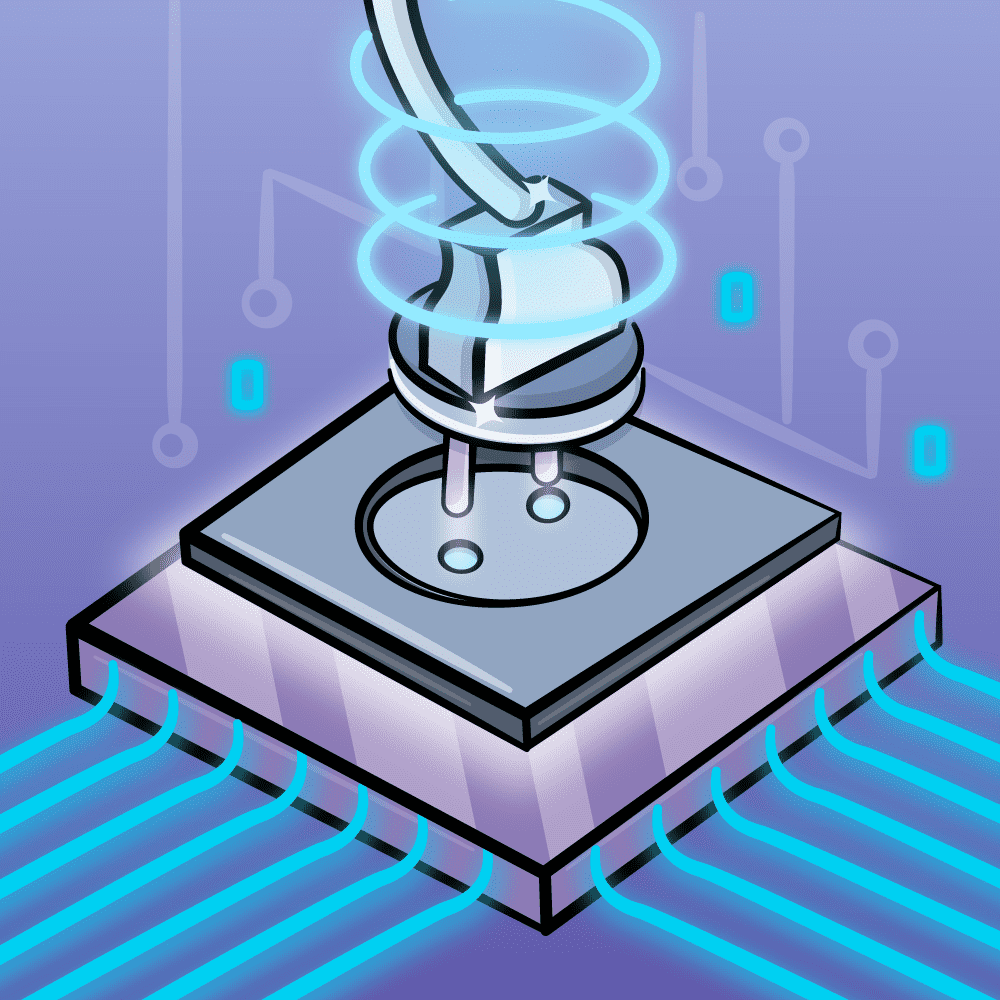
Maarten Jakobs

Hoe implanteer ik websockets in mijn project?



# Deelvragen:

## Wat is een websocket?

Websockets zijn speciaal ontworpen om toegepast te woeden in webbrowsers en webservers. Het websocket protocol lijkt heel erg op die van HTTP omdat de handshake van het WebSocketprotocol door HTTP-servers wordt geïnterpreteerd. Het protocol staat toe dat data op en neer gestuurd kan blijven worden.

## Waarom zou ik een websocket gebruiken?

Een websocket is handig als in een website data up-to-date moet blijven. Bijvoorbeeld bij een chat-applicatie waar de berichten op een willekeurig moment verstuurd kunnen worden.

# Welke methodes voor Websockets in dot-net-core bestaan er al? (Available product analysis)

## Broadcasting with SignalR and .NET Core - TwilioSignalR

SignalR is een ‘helper’ library voor Websockets, de library past hier bij de ‘hub’ concept toe. Dit houd in dat zowel de server als de client naar elkaar calls kunnen maken. Voor SignalR is er ook een Angular library. SignalR wordt zelf door Microsoft aangeraden. Over SignalR zijn er veel voorbeelden en documentatie te vinden. De service kan je zelf hosten maar ook makkelijk te hosten bij Azure. De service is een gedeelte van ASP.NET

## Pusher

Pusher is een online service waarbij je een account moet aanmaken. In het account kan je een nieuwe pusher app maken. Je hebt de keuze uit Channels en Beams. Beams zijn gericht op het verzenden van notifications en met channels kan je eigen data verzenden. Via een API haalt hij dan de data van je server af. Wanneer er data binnen komt wordt er een eventTrigger aangeroepen en kan je zo bij de verstuurde data. Bij deze service ben je afhankelijk van de services van Pusher zelf. Pusher heeft een aantal tarieven van de Sandbox, die gratis is om mee te ontwikkelen en te testen tot de Growth Plus, dat het duurste pakket is en dan $1199,- kost per maand.

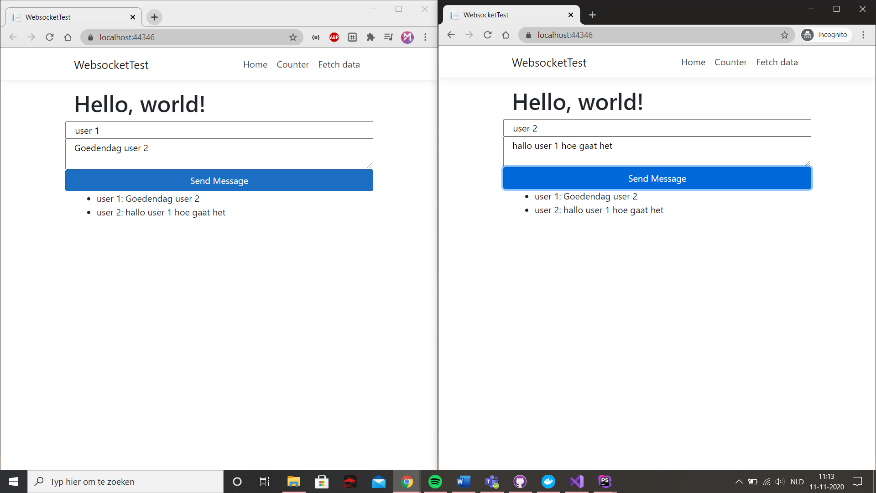
## Connecting a remote worldPubNub

PubNub is net zoals Pusher een online service die bestaat uit meerdere (aanpasbare) microservices. Deze services draaien op 14 verschillende servers op de wereld. Wanneer 1 server crasht neemt een andere server het over zodat je je data nog altijd realtime ontvangt. Er zijn verschillende services al kant en klaar zoals de In-app chat en de Alerts en Notifications. Het testen en ontwikkelen met PubNub is kosteloos maar wanneer je hem live wil zetten kost het $49,- per maand, ook is het mogelijk om custom pricing aan te vragen wanneer het pakket van $49,- niet genoeg bied.

# Waarvoor worden de Websockets in mijn project voor gebruikt? (Document analysis)

In mijn project moet via een websocket de bestelling lijst up-to-date blijven. Wanneer een gebruiker wil betalen kan die op een knop drukken om te betalen. Wanneer er op deze knop is gedrukt moet er bij de barman een melding komen met daarbij de totaal prijs berekend. Dit wil ik gaan bereiken via een of meerdere websockets. Het zou in principe ook via een REST-call kunnen die om de minuut het opnieuw binnen haalt maar dan zou het voor kunnen komen dat je als barman 50 seconde moet wachten op de melding dat iemand wil betalen.

# Prototype

Ik heb een prototype gemaakt met SignalR. In mijn prototype kunnen meerdere pagina’s met elkaar berichten sturen. Wanneer persoon 1 een bericht ontvangt ontvangen peronen 2 en 3 ze ook Mijn prototype is te vinden op: <https://github.com/Maarten14NL/Websockets-prototype>

Domein model toevoegen en diagrammetjes

# Conclusie

De websocket library die ik ga gebruiken is SignalR omdat het kosteloos is en veel voorbeeld code te vinden is op het internet. Daarnaast heb je bij SignalR de service zelf onder controle, dit heb je bij Pusher en PubNub niet. In mijn prototype heb ik al heel vlug en gemakkelijk SignalR kan toepassen in een programma.

# BronnenLijst

<https://dotnet.microsoft.com/apps/aspnet/signalr/>

<https://pusher.com/>

<https://www.pubnub.com/>