

|  |
| --- |
| Werkstuk webapplicaties SignalR  Ouassim Abdellaoui &  Graduaat Informatica Programmeren  Webapplicaties |

Inhoudstafel

[Inhoudstafel 3](#_Toc129462551)

[1. Inleiding 4](#_Toc129462552)

[2. Aanpak 4](#_Toc129462553)

[1.1 Verwerking van behandelde research 4](#_Toc129462554)

[1.2 Manier van werken 4](#_Toc129462555)

[1.3 Bronnen 4](#_Toc129462556)

[3. Wat is SignalR? 4](#_Toc129462557)

[2.1 Belangrijkste features SignalR 5](#_Toc129462559)

[2.1.1 Verbindingsbeheer gebeurt automatisch 5](#_Toc129462560)

[2.1.2 Berichten naar alle, specifieke of groep actieve gebruikers 5](#_Toc129462561)

[2.1.3 Schaalt naar trafiek 5](#_Toc129462562)

[2.1.4 SignalR Hub Protocol 6](#_Toc129462563)

[4. Reflectie 6](#_Toc129462564)

# Inleiding

In deze opdracht gaan wij het hebben over SignalR.

We beginnen eerst met onze aanpak. Hoe zijn we tot dit werkstuk gekomen? Welke research hebben we allemaal gedaan en hoe hebben we deze informatie behandeld?

Daarna gaan we dieper op in wat is SignalR juist is. Waarvoor wordt het gebruikt en hoe wordt dit geïmplementeerd binnen ASP.Net? We proberen de materie zo goed mogelijk over te brengen zodat onze medestudenten dit ook kunnen gebruiken in hun toekomstige projecten. Aan de hand van beeldmateriaal hopen wij dat SignalR net iets duidelijker wordt.

Eindigen doen we met een reflectie. Hoe verliep de samenwerking in onze groep?

Wat we allemaal hebben bijgeleerd en of er eventueel ook dingen zijn die we beter anders hadden aangepakt.

# Aanpak

## Verwerking van behandelde research

## Manier van werken

## Bronnen

<https://learn.microsoft.com/nl-nl/aspnet/core/tutorials/signalr?WT.mc_id=dotnet-35129-website&view=aspnetcore-7.0&tabs=visual-studio>

<https://learn.microsoft.com/nl-nl/aspnet/core/signalr/introduction?WT.mc_id=dotnet-35129-website&view=aspnetcore-7.0>

<https://learn.microsoft.com/en-us/aspnet/signalr/overview/getting-started/introduction-to-signalr>

<https://learn.microsoft.com/en-us/aspnet/core/signalr/scale?view=aspnetcore-7.0>

<https://www.tutorialspoint.com/what-is-signalr-and-how-to-use-it>

<https://www.c-sharpcorner.com/UploadFile/abhijmk/what-why-and-how-about-signalr/>

<https://kb.2pintsoftware.com/help/signalr-and-connection-management>

<https://en.wikipedia.org/wiki/WebSocket#:~:text=WebSocket%20is%20a%20computer%20communications,as%20RFC%206455%20in%202011>.

<https://scientificprogrammer.net/2022/09/24/sending-messages-to-individual-signalr-clients-or-groups-of-clients/>

<https://scientificprogrammer.net/2022/05/07/signalr-on-net-6-the-complete-guide/>

<https://infinum.com/blog/scaling-out-your-own-signalr-chat-application/>

<https://medium.com/yemeksepeti-teknoloji/what-is-server-sent-events-sse-and-how-to-implement-it-904938bffd73>

https://www.codemag.com/article/1807061/Build-Real-time-Applications-with-ASP.NET-Core-SignalR

# Wat is SignalR?

## Intro

SignalR is een open-source bibliotheek dat het gemakkelijk maakt om real-time functionaliteit naar apps te brengen. Door middel van webfunctionaliteit wordt het mogelijk gemaakt om servercode meteen naar cliënten te pushen.

Enkele goede voorbeelden hiervan zijn de gamingindustrie, sociale media, veilingsites en GPS. Een bekend voorbeeld waarbij SignalR wordt gebruikt zijn Google documenten. Als 2 mensen dezelfde spreadsheet openen en persoon A begint te typen, kan persoon B dit meteen zien zonder zijn webpagina te verversen.

## Ontstaan

SignalR is in 2011 in het leven geroepen door David Fowler en Damian Edwards. Het werd toegevoegd in het ASP.Net project en uitgebracht als deel van ASP.Net in 2013.

In 2013 was het WebSocket protocol juist gestandaardiseerd maar ondersteunde de meeste browsers dit nog niet. WebSocket maakt het mogelijk om data tegelijkertijd in twee richtingen te sturen, hier zo meteen meer over.

Voor SignalR was er niet echt een goede methode voor real-time communicatie.

Twee manieren die wel gebruikt werden waren AJAX polling en server-sent events, beide redelijk omslachtig tegenover SignalR. Bij AJAX (long) polling is het constant maken van HTTP calls tot je het verwacht antwoord krijgt, hier heb je dus redelijk wat werklast op je systeem. Met server-sent events kun je berichten naar cliënten sturen als deze real-time informatie nodig hebben. Denk hier aan alarmen, meldingen, nieuws, sportscores,…

## Belangrijkste features SignalR

### Verbindingsbeheer gebeurt automatisch

SignalR gebruikt een API die instaat voor het verbindingsbeheer. Hiervoor wordt WebSocket gebruikt. WebSocket is communicatieprotocol dat het mogelijk maakt om in realtime data te sturen van en naar de server. Communicatie is consistent, bidirectioneel en de data kan tegelijk in twee richtingen gestuurd worden.

Bij het gebruik van SignalR zijn er veel functionaliteiten die je normaal extra moet installeren (denk hier bijvoorbeeld aan oudere systemen). Door het gebruik van WebSocket worden deze automatisch geïmplementeerd.

WebSocket begint als HTTP protocol en doet eerst een handshake. Een handshake is een signaal tussen twee toestellen of programma’s om een link op te zetten tussen beide. Daarna wordt de HTTP Upgrade header gebruikt en wordt de HTTP een WebSocket protocol.

Verschillend met HTTP is dat WebSocket een open connectie houdt. Gebruiker en server kunnen data versturen en ontvangen zonder dat de gebruiker hier eerst een verzoek voor moet sturen.

### Berichten naar alle, specifieke of groep actieve gebruikers

SignalR gebruikt hubs om te communiceren tussen gebruikers en servers. Een hub is een pijplijn die toelaat dat een gebruiker en server methodes kunnen oproepen bij elkaar. Dit wordt automatisch gedaan door SignalR, het zorgt ervoor dat de machines met elkaar verbinden en er zo communicatie mogelijk is tussen server en gebruiker of vice versa. Dit gebeurt in JSON of MessagePack, wat een compactere vorm van JSON is.

Deze hubs worden gebruikt op de serverzijde voor het versturen van berichten naar gebruikers. Het volstaat om een klasse aan te maken die over erft van Hub. Hierin kunnen verschillende methodes, tasks, interfaces,… aangemaakt worden.

Voor berichten te sturen zijn er 3 methoden: SendMessage gebruikt de property Clients.All om naar alle gebruikers een bericht te sturen. SendMessageToCaller heeft Clients.Caller om te antwoorden op de gebruiker die juist een bericht heeft gestuurd. Als laatste is er nog SendMessageToGroup die SignalR Users groups gebruikt.

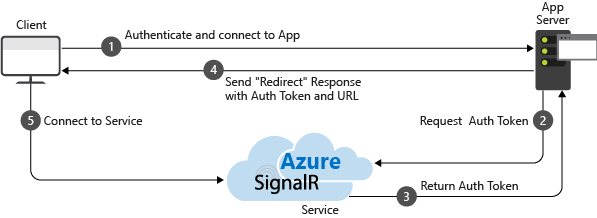
### Schaalt naar trafiek

Een app dat SignalR gebruikt moet steeds alle verbindingen bijhouden, wat problemen geeft bij meerdere servers. Stel één server wil een bericht sturen naar alle gebruikers. Dat bericht wordt alleen verzonden naar de gebruikers die met die ene server zijn verbonden. Er zijn twee opties om dit probleem op te lossen: Azure SignalR Service en Redis backpane.

Diagram

Description automatically generated

**Azure SignalR Service:** Azure verhelpt dit probleem door bij elke connectie van de gebruiker een omleiding te leggen naar de service. De gebruiker verbindt met de service zelf waardoor elke server maar een beperkt aantal connecties moet bijhouden.



**Redis backpane:** Redis werkt werkt met een publiceer/abonneer manier. Wanneer een gebruiker verbindt wordt deze informatie doorgegeven naar de “backplane” van Redis. Wanneer een server dan informatie wil sturen naar alle gebruikers wordt dit via het backplane gedaan.

Timeline

Description automatically generated with medium confidence

### SignalR Hub Protocol

SignalR maakt gebruik van hubs om de communicatie van de server naar cliënten en van cliënten naar server te doen. In de code definieer je methoden die via de server de aangeroepen kunnen worden, SignalR handelt zelf de routing af.

Hieronder is een voorbeel van zo’n hub. Door middel van de methode “SendMessage” kan een gebruiker een bericht sturen. Door de methode ReceiveMessage wordt dit dan verstuurd naar alle cliënten die verbonden zijn.

Text

Description automatically generated

In de code van de webpagina is dan de methode “ReceiveMessage” toegevoegd. Deze heeft dezelfde naam als degene die in de hub is aangemaakt. Het luistert naar berichten waarin de ReceiveMessage wordt opgeroepen.

Text

Description automatically generated

# Reflectie