30-11-2021

Doomen,Jens J.J.W.

Research rapport

# Inleiding

In mijn applicatie wordt gebruik gemaakt van websockets om te kunnen chatten. Het komt alleen voor dat als er twee berichten snel achter elkaar worden gestuurd dat de tweede niet verstuurd kan worden omdat er dan nog een “Connecting state” .

# Hoe zorg ik dat de WS-Server sneller de connectie terug zet na elk bericht?

## Hoe meet ik de latency tussen de connectie met een server na een bericht?

Er zijn op internet meerde tools beschikbaar om latency te meten en een network map uit te zetten. Twee voorbeelden hiervan zijn Solarwinds en Datadog. Bijde tools zijn instaat om bij alle connecties de performance te laten zien en het volledige path weer te geven. Op deze manier is het snel en makkelijk te bepalen waar in het syteem de “spike” ligt en dus de latency hoog is Ik ga kiezen voor Datadog om mijn latency te testen omdat hier ook makkelijk te zien is of er fouten bij de client of bij de server liggen.

(avaiable product analysis)

# Inleiding

Voor mijn eigen project maak ik een chat app. Ik wilde eerst voor elke message een get toevoegen en een post voor het sturen. Dit ging in de praktijk natuurlijk niet zo lekker. En mijn database werd dus ook snel te vol. Vooral als ik meerdere gesprekken in de toekomst wil toevoegen en tegelijk wil laten draaien.

# Welke methodes worden vaak gebruikt in chat applicaties?

Dit is mijn hoofdvraag en doormiddel van de deelvragen. “Welke methodes zijn er?” en “Welke methode valt het beste in mijn applicatie?”

## Welke methodes zijn er?

De meeste chat apps maken gebruik van Websockets. Websockets zijn een communicatie protocol die over een TCP connectie werken. Het protocol zorgt voor communicatie tussen een client en een de web server. Op deze manier kunnen berichten naar meerder clients gestuurd worden zonder dat de client een er naar vraagt met een get, de connectie blijft ook open staan.

<https://en.wikipedia.org/wiki/WebSocket>

Verder kan ik werken met REST, ik kan een request sturen naar de server die een bericht ophaalt en dan een response terug stuurt naar mijn client. Het nadeel is dat er dan eerst een request gestuurd moet worden vanuit de client, ook wordt dit proces steeds herhaald ook al is er geen nieuwe data beschikbaar.

Een variant wat ik hierop kan zetten is long polling. Bij long polling wordt er niet steeds een response terug gestuurd maar blijft de connectie open totdat de data veranderd en dan wordt er een response gestuurd

<https://ably.com/blog/websockets-vs-long-polling>

(avaiable product analysis)

## Welke methode valt het beste in mijn applicatie?