TDE
DIGITAL CREATIVES IN SPORTS

Real Time Communication Methods

Improving the Grand Prix experience for F1 viewers at home

S8 Graduation FHICT

4 Sept 2023 - 16 Jan 2024

By Jordi Franssen

1 November 2023

www.tde.nl

Introduction

The tool will need live timing data to visualize the positions of the Formula 1 cars in ThreeJS. There are a few methods to get this data from the server to the client. In this document, investigate what method will provide the best solution for my tool. I'll do this by interviewing an expert on the topic and comparing the methods the expert suggested.

Expert interview

To find out what method to use as a real time communication method for my app, I interviewed my company coach John, who is an expert on the topic. After a small introduction of what I wanted to investigate, I asked John the following question:

"What is the best method to get data from the server to the client for the tool I'm going to build?"

John argued that there are three possible methods to consider. Websockets, REST-APIs and Eventsourcing. All three have their advantages and disadvantages. Websockets are really fast and have low latency, but are quite complex to set up. REST-APIs are much easier to set up, but require pulls from the client with a regular interval which could be difficult for the server to handle considering data needs to be pulled with a really short interval. If I would want to pull the REST-API's endpoint with the same interval as the FIA API updates its information, which is at around 230 milliseconds, a REST-API is not an option. Especially considering that multiple users are using the tool at the same moment during a race. Another thing to note is that REST-API's are not actually real time, as the client pulls with an interval and not when new data is available. At last, EventSourcing could provide a solution as well.

Communication Method Comparison

The expert interview resulted in three interesting options to consider. Below, I compared these three methods by enumerating the pros and cons that have an effect on the tool I'm going to make.

REST-APIs

Pros:

- Easy to implement and widely known with other developers.

Cons:

- REST-APIs are not real-time as polling is required to get updates.
- Data needs to be polled at a high frequency interval which makes it inefficient. This is also difficult for the server as many clients will be polling at the same time during a racing event.
- It could be possible that data isn't always received in the same order when the endpoint is called at such a high frequency interval. In some cases, a response is still on its way to the client, while another request has already been fulfilled.

Websockets

Pros:

- Extremely fast with almost no latency.
- No need for consistent polling as the server pushes updates when information changes.

Cons:

- Much more difficult to implement.

EventSourcing

Pros:

- Easy to implement on both the server and the client.
- Automatic reconnection after a few seconds if the connection is lost.
- Efficient as the connection stays open so updates will happen as the server receives new information.

Cons:

- Most browsers limit the number of eventsourcing connections to 6 per domain.

Source:

EventSource - Web APIs | MDN. (2023, February 26).
https://developer.mozilla.org/en-US/docs/Web/API/EventSource

Conclusion

First, I assumed that Websockets would be the best option. However, I wasn't familiar with SSE. After some research online, I found that SSE is the best method for my tool because it's easier to implement than websockets. Websockets might be faster after all, but the difference in latency really doesn't matter for the user if it's only a few milliseconds. REST-APIs are definitely not an option as the client will poll with an interval that's too difficult to handle for the server.

Summary

Data needs to be processed by a server before it gets sent to the client. There are multiple real time communication methods to do this. Therefore, I need to know which one works best for the tool I'm going to make.

I conducted an expert interview with John to determine which methods to compare to find the best option. With this information, I conducted library research online to find pros and cons for each method and concluded which one is the best to implement in my tool.

I found that Server Send Events (SSE) works best as SSE is fast and easy to implement. It also allows only one-way traffic. This works for my tool as the clients don't have to send any information back to the server.

Learning Outcome Clarification

- Learning Outcome 1: Professional Duties
- Learning Outcome 2: Situation-Orientation
- Learning Outcome 4: Investigative Problem Solving
- Learning Outcome 6: Targeted Interaction

This deliverable is a professional duty on a bachelor level in the activities of Analysis and Advise as I analyzed what the best possible real-time communication method is and advised which one to use in my project. Therefore, Learning Outcome 1: Professional Duties applies.

This deliverable is relevant for one or more persons and creates value as it's important that the correct communication method is used to get data from the server to the client. I also worked in a methodological and structured way as I first interviewed an expert to determine which options to further investigate and later compared these options to come to a dependable solution. Therefore, Learning Outcome 2: Situation-Orientation applies.

I used a variety of research strategies, methods and activities to come to my conclusion. Therefore, Learning Outcome 4: Investigative Problem Solving applies.

I communicated appropriately with partners in my project as I conducted an interview with an expert to achieve the right impact and execution. Therefore, Learning Outcome 6: Targeted Interaction applies.