

BACHELOR PAPER

Term paper submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Engineering at the University of Applied Sciences Technikum Wien - Degree Program Computer Science

Clientside web communication options and their consequences for stateless service design

By: Peter Öttl

Student Number: 1910257146

Supervisor: Ing. Stefan Vietze, MSc.

Vienna, May 22, 2022

Abstract

More and more applications are using the advantages of cloud services and web applications. The advantages are that the application is just behind a tiny URL, can be used without an installation and can be accessed anywhere and from any device. Furthermore, these modern applications use HTTP to communicate with the server. This technology is limited to unidirectional communication. Therefore, only the server can send data to the client after a request. This happens after a user triggers an event, like a button click, a form submission or a simple page refresh.

With WebSockets or Server-Sent Events, the server can send data to the client without sending such a request. These features are provided with different technologies and network protocols. This thesis compares these technologies in terms of their advantages, disadvantages, and challenges. It also introduces the theory of TCP, HTTP and real-time HTTP.

The main goal is to integrate a real-time HTTP feature into the *all about apps go starter* project. The biggest challenges are integrating the real-time feature, keeping the application itself stateless, and using it with the current auth flow. Another requirement is to ensure that the application remains replicable and scalable and that all principles of twelve factor apps are kept. This was accomplished using Redis as a pub-sub service, WebSockets as a real-time communication protocol and a different auth endpoint for one-time auth tokens.

To conclude this thesis, with Redis PubSub and WebSockets, the server is capable of real-time communication features and the replication of the service is ensured.

Keywords: WebSocket, server-sent events, polling, client-server communication, Redis, Pub-sub, stateless service