

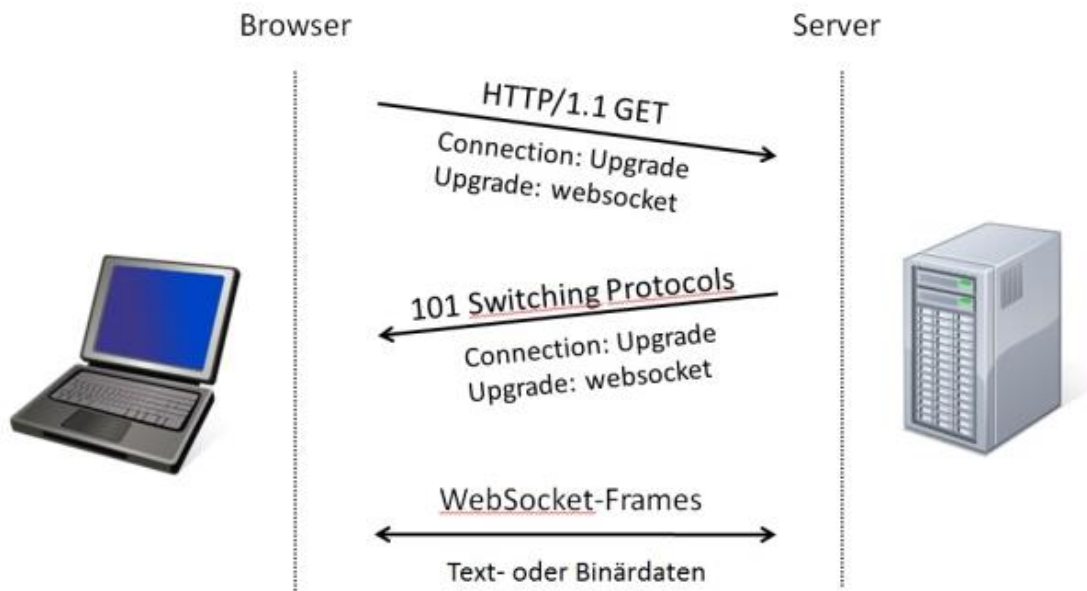
Spring: WebSockets (Events und Chat im Spiel)

Alexander Hildebrandt und Tobias Herzog

Problem: Der Server ist nicht in der Lage Kommunikation zum Client zu initiieren.

WebSockets bieten die Möglichkeit einer bidirektionalen Verbindung zwischen Server und Client

Dies erlaubt eine direkte Kommunikation des Servers mit dem Client



<https://entwickler.de/online/web/server-push-und-bidirektionale-kommunikation-mit-websockets-136313.html>

Der Client baut eine normale HTTP Verbindung zum Server auf. In dieser stellt er die Anfrage auf ein Upgrade der HTTP Verbindung auf einen WebSocket. Der Server antwortet daraufhin mit einer Bestätigung der Anfrage, woraufhin eine WebSocket Verbindung aufgebaut wird und Daten zwischen Client und Server ausgetauscht werden können.

Vorteile von WebSockets:

- Reale zweiseitige Kommunikation (kein Workaround)
- Erhöhte Kommunikationseffizienz
- Simple API
- Durch TCP sehr stabil

Nachteile von WebSockets:

- Von älteren Browsern nicht unterstützt
- Proxy-Server bereiten Probleme
- Eine hohe Anzahl an Gleichzeitig geöffneten WebSockets sorgt zu Problemen
- WebSockets und HTTP/2 funktionieren nicht zusammen
- Load-Balancing ist schwer

Browser die WebSockets unterstützen sind:

- Google Chrome
- Safari
- Opera, ab Version 10.70
- Firefox, ab Version 4
- Internet Explorer, ab Version 10.0

Vorgehensweise für das Benutzen von WebSockets:

- WebSockets werden in Java-Klassen deklariert, konfiguriert und kontrolliert.
- Im HTML Code wird nun eine Verbindung über WebSockets gestartet, damit Server und User sich Nachrichten schicken können.
- Bei Bedarf wird außerdem ein Subprotokoll über die WebSockets gelegt, um z.B. im Fall von STOMP Nachrichten per Abonnements verschicken zu können
- In den Java-Klassen wird dann zusätzlich beschrieben, was bei eingehenden Nachrichten passiert (z.B. das Starten einer Begrüßungs-Methode) und wann bzw. an wen der Server seine Nachrichten sendet (z.B. nach 3 Sekunden an alle Abonnenten)