

Möglichkeiten zur asynchronen Kommunikation zwischen Webbrowser und Server

Hendrik Wagner
hendrik.wagner@mni.thm.de
Technische Hochschule Mittelhessen
Gießen, Hessen, Germany

Zusammenfassung

Die Kommunikation zwischen Webbrowser und Server wird klassischerweise durch den Client (in der Regel ein Webbrowser) initiiert. Dieser fragt den Inhalt einer Website an, und kann später weitere Inhalte laden oder Daten übermitteln. Der Server kann aber keine Daten nachträglich an den Client senden, wenn dieser diese nicht aktiv anfragt. In diesem Artikel werden verschiedene Möglichkeiten zur fortlaufenden Kommunikation zwischen Client und Server vorgestellt und analysiert. Dabei werden die Vor- und Nachteile der einzelnen Methoden aufgezeigt.

INHALTSVERZEICHNIS

Abstract	1
Inhaltsverzeichnis	1
1 Einleitung	1
2 Vorstellung der Kommunikationsvarianten	1
2.1 Klassisches Polling	1
2.2 Long Polling	1
2.3 WebSockets	1
3 Beispielhafte Implementierungen	2
3.1 Implementierung von Polling	2
3.2 Implementierung von Long Polling	2
3.3 Implementierung von WebSockets	2
4 Vergleiche zwischen Implementierungen	2
5 Vorstellung von Frameworks	2
5.1 socket.io	2
5.2 faye	2
6 Fazit	2
Literatur	2

1 Einleitung

Gilt es, eine Website mit bidirektionaler bzw. asynchroner Kommunikation, insbesondere mit vom Server bereitgestellten Aktualisierungen, zu realisieren, so muss eine Auswahl zwischen verschiedenen Ansätzen gewählt werden. Ein solcher Fall kann eine Chatanwendung sein, in welcher der Server von einem Client erhaltene Nachrichten an andere Clients weitersendet. Ein weiteres Beispiel wäre eine rechenintensive Anwendung, in der eine Berechnung nach einiger Zeit erfolgt und der Server das Ergebnis an den Client melden soll. Da der Server unter HTTP nicht ohne weiteres eine Nachricht an einen Client senden kann, muss dieser die Nachricht anfordern - diese simple Art des Nachrichtempfangs nennt man *Polling*, also die wiederholte Abfrage nach neuen Nachrichten. Neben diesem Ansatz gibt es *WebSockets*, welche eine TCP/UDP-ähnliche Kommunikation im Web ermöglichen.

2 Vorstellung der Kommunikationsvarianten

2.1 Klassisches Polling

Unter dem HTTP-Protokoll gibt es eine Reihe von Anfragen, die der Client an den Server senden kann [1].
HTTP Protokoll?

2.2 Long Polling

Eine Optimierung des Polling-Konzepts ist das sog. *Long Polling*, manchmal auch als *Streaming* bekannt. Dabei wird die Antwort auf eine HTTP-Anfrage zurückgehalten, bis der Server eine Nachricht versenden will [2]. Ein Risiko dieser Methode sind Timeouts und viele offene, ungenutzte Verbindungen. [3].

2.3 WebSockets

WebSockets sind ein vom IETF¹ entwickeltes Protokoll, welches 2011 als Standard veröffentlicht wurde [4].

Websockets haben einen simplen Handshake

Notabel ist auch die Reduktion in Overhead, der bei Polling durch HTTP-Header entsteht. Üblicherweise sind HTTP-Header zwischen 200 und 2.000 Bytes groß [5]. Dies kann einen ernsthaften Effekt auf die Netzwerkbelastung haben, besonders wenn die Polling- oder Nachrichtenfrequenz hoch

¹Internet Engineering Task Force.

ist. So würden 1.000 Nutzer, die sekundlich Nachrichten anfragen, allein durch die Header etwa 6 Mbps an Netzwerkdurchsatz verursachen [6]. Dieser Netzwerkdurchsatz entfällt durch Websockets.

Wie realisiert?

3 Beispielhafte Implementierungen

Blabla wir machen einen Chat.

3.1 Implementierung von Polling

...

3.1.1 Polling Client.

3.1.2 Polling Server.

3.2 Implementierung von Long Polling

...

3.3 Implementierung von WebSockets

3.3.1 WebSocket Client.

3.3.2 WebSocket Server.

4 Vergleiche zwischen Implementierungen

Vor- und Nachteile Performance?

... Die Latenzen überschneiden sich mit Messungen getätigt im Rahmen eines Papers von Pimentel und Nickerson in 2012[7].

5 Vorstellung von Frameworks

Es gibt Frameworks, die das Implementieren von asynchroner Kommunikation erleichtern.

5.1 socket.io

Was ist das? Wie ist es besser oder schlechter für die Implementierung als Chat-App? Performance?

5.2 faye

6 Fazit

Literatur

- [1] "HTTP request methods" [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>
- [2] "Long polling," Dec. 2021. [Online]. Available: <https://javascript.info/long-polling>
- [3] P. Saint-Andre, S. Loreto, S. Salsano, and G. Wilkins, "Known Issues and Best Practices for the Use of Long Polling and Streaming in Bidirectional HTTP," Internet Engineering Task Force, Request for Comments RFC 6202, Apr. 2011, num Pages: 19. [Online]. Available: <https://datatracker.ietf.org/doc/rfc6202>
- [4] A. Melnikov and I. Fette, "The WebSocket Protocol," Internet Engineering Task Force, Request for Comments RFC 6455, Dec. 2011, num Pages: 71. [Online]. Available: <https://datatracker.ietf.org/doc/rfc6455>
- [5] "SPDY: An experimental protocol for a faster web." [Online]. Available: <https://www.chromium.org/spdy/spdy-whitepaper/>
- [6] P. Lubbers and F. Greco, "HTML5 WebSocket - A Quantum Leap in Scalability for the Web." [Online]. Available: <https://web.archive.org/web/20210812191126/https://websocket.org/quantum.html>
- [7] V. Pimentel and B. G. Nickerson, "Communicating and Displaying Real-Time Data with WebSocket," *IEEE Internet Computing*, vol. 16, no. 4, pp. 45–53, Jul. 2012, conference Name: IEEE Internet Computing.