

Exposé: Möglichkeiten zur nachträglichen Kommunikation zwischen Webbrowser und Server

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

Zusammenfassung

Die Kommunikation zwischen Webbrowser und Server wird in der Regel immer durch den Client (in der Regel ein Webbrowser) initiiert. Dieser fragt initial den Inhalt der 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.

ACM Reference Format:

Hendrik Wagner. 2022. Exposé: Möglichkeiten zur nachträglichen Kommunikation zwischen Webbrowser und Server. In *Hauptseminar II WiSe 22/23, gehalten von Kevin Linne, M.Sc.*. ACM, New York, NY, USA, 1 page. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 Motivation

Moderne Webanwendungen benötigen häufig die Fähigkeit, Daten vom Server an den Client zu senden. Soll zum Beispiel eine Chatanwendung realisiert werden, so muss der Server neue Nachrichten an alle verbundenen Clients senden können. Dies kann nicht ohne weiteres funktionieren, da unter HTTP nur der Client Daten anfragen oder versenden kann.

Es gibt mehrere Möglichkeiten, wie eine bidirektionale Kommunikation zwischen Client und Server realisiert werden kann. Um die richtige Lösung für ein bestimmtes Problem zu finden, ist es wichtig, die Funktionsweise sowie Vor- und Nachteile der einzelnen Methoden zu kennen. Auch sollten existierende Frameworks bekannt sein, um ggf. die Implementierung zu vereinfachen.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference'17, July 2017, Washington, DC, USA

© 2022 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM. . \$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

2 Aufgabenstellung und Abgrenzung

2.1 Aufgabenstellung

Die Seminararbeit soll sich mit der Frage beschäftigen, wie eine bidirektionale Kommunikation zwischen Client und Server realisiert werden kann. Um die Argumentationen zu stützen, soll eine Chatanwendung als Beispiel herangezogen werden, welche bestenfalls die verschiedenen Methoden implementiert. Um die Vergleiche zwischen den Methoden (Websockets, Polling, Long Polling) zu ermöglichen, werden diese zunächst genauer in ihrer Funktionsweise beschrieben. Danach werden die Vor- und Nachteile der einzelnen Methoden aufgezeigt und argumentiert, welche Methode für die Chatanwendung am besten geeignet ist. Zuletzt werden existierende Frameworks (socket.io, faye) vorgestellt, welche die Implementierung der verschiedenen Methoden vereinfachen.

2.2 Abgrenzung

Die Seminararbeit soll sich ausschließlich mit der Realisierung einer bidirektionalen Kommunikation zwischen Client und Server beschäftigen. Es gibt bereits eine große Vielfalt an Frameworks und Realisierungsmöglichkeiten, aus welchen die Seminararbeit eine Auswahl treffen soll. Vergleiche zwischen Methoden und zu Frameworks werden auf Performance und Implementierungsaufwand bezogen. Die Chatanwendung wird in ihrem Umfang minimal gehalten, notfalls nur als theoretisches Beispiel herangezogen. Auch werden Polling-Algorithmen nur begrenzt behandelt, da sie selten in der Praxis verwendet werden; Fortgeschrittene Algorithmen wie das Nyquist-Shannon-Abtasttheorem werden nicht behandelt.

3 Vorgehen

Zunächst werden die verschiedenen Methoden genauer erarbeitet um die Funktionsweise aufzuzeigen. Im Rahmen dessen wird die Chatanwendung zur Veranschaulichung als Beispiel erwähnt und ggf. implementiert. Danach werden die Vor- und Nachteile der einzelnen Methoden aufgezeigt und argumentiert, welche Methode für die Chatanwendung am besten geeignet ist und ob es Argumente gibt, die für eine andere Methode sprechen. Zuletzt werden existierende Frameworks vorgestellt, mit den klassischen Methoden verglichen und auch hier für die Chatanwendung eine Empfehlung gegeben.