



Kostenloses eBook

LERNEN

WebSockets

Free unaffiliated eBook created from
Stack Overflow contributors.

#websocket

Inhaltsverzeichnis

Über	1
Kapitel 1: Erste Schritte mit WebSockets	2
Bemerkungen.....	2
Examples.....	2
Installation oder Setup.....	2
Web-Socket-Client-Beispiel.....	2
Kapitel 2: Unterschiede zwischen WebSockets und AJAX	4
Einführung.....	4
Examples.....	4
Beispiel für eine Benachrichtigung.....	4
Credits	6



You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [websockets](#)

It is an unofficial and free WebSockets ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official WebSockets.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Kapitel 1: Erste Schritte mit WebSockets

Bemerkungen

In diesem Abschnitt erhalten Sie einen Überblick darüber, was WebSocket ist und warum ein Entwickler es verwenden möchte.

Es sollte auch alle großen Themen in WebSocket erwähnen und auf die verwandten Themen verweisen. Da die Dokumentation für WebSocket neu ist, müssen Sie möglicherweise erste Versionen dieser verwandten Themen erstellen.

Examples

Installation oder Setup

Detaillierte Anweisungen zum Einrichten oder Installieren der Web-Socket.

Web-Socket-Client-Beispiel

Dies ist ein Beispiel für einen Web-Socket-Client in Javascript.

Es:

1. Verbindet mit einem Live-Demo-Server.
2. Sendet eine Nachricht
3. Empfängt Nachricht (en).
4. Trennt nach einem Intervall die Verbindung.

```
var mySocket      = null;
var serverUrl     = 'wss://echo.websocket.org'; // wss: is ws: but using SSL.
var oWebSocket    = window.WebSocket || window.MozWebSocket;
if (oWebSocket) {
  mySocket = new oWebSocket (serverUrl);
  if (mySocket) {
    console.log (mySocket);
    mySocket.onopen      = onSocketOpen;
    mySocket.onclose     = onSocketClose;
    mySocket.onmessage   = onSocketMessage;
    mySocket.onerror     = onSocketError;

    setTimeout (closeSocket, 5000); // Be polite and free socket when done.
  }
}

function onSocketOpen (evt) {
  console.log ("Socket is now open.");
  mySocket.send ("Hello from my first live web socket!");
}

function onSocketClose (evt) {
  console.log ("Socket is now closed.");
}
```

```
}

function onSocketMessage (evt) {
    console.log ("Recieved from socket: ", evt.data);
}

function onSocketError (evt) {
    console.log ("Error with/from socket!");
    console.log (evt);
}

function closeSocket () {
    if (mySocket.readyState !== mySocket.CLOSED) {
        console.log ("Closing socket from our end (timer).");
        mySocket.close ();
    }
    else
        console.log ("Socket was already closed (timer).");
}
}
```

Wenn Sie diesen Code in einer Konsole ausführen, erhalten Sie ein Ergebnis wie:

```
WebSocket { url: "wss://echo.websocket.org/", readyState: 0, bufferedAmount: 0, onopen: null,
  onerror: null, onclose: null, extensions: "", protocol: "", onmessage: null,
  binaryType: "blob"
}
2 // <--- 2 was the socket ID in this instance.
Socket is now open.
Recieved from socket: Hello from my first live web socket!
Closing socket from our end (timer).
Socket is now closed.
```

Erste Schritte mit WebSockets online lesen: <https://riptutorial.com/de/websocket/topic/7958/erste-schritte-mit-websockets>

Kapitel 2: Unterschiede zwischen WebSockets und AJAX

Einführung

Ajax verwendet das HTTP-Protokoll und kann Anforderungen mithilfe von POST / GET-Methoden vom Client an den Server senden.

WebSocket ist selbst ein Protokoll zur Kommunikation zwischen Client und Server, das sich von HTTP unterscheidet.

Wenn Sie in Ajax eine Anfrage senden, sendet der Server eine Antwort für diese Anfrage und die Verbindung wird beendet.

Wenn Sie WebSockets verwenden, wenn Sie eine Verbindung mit dem Server herstellen, können Sie so oft zwischen Client und Server kommunizieren, wie Sie möchten.

Examples

Beispiel für eine Benachrichtigung

Ajax:

1. Neue Anfrage :

- Client sendet Anfrage: Server, Haben Sie eine neue Benachrichtigung?
- Server sendet Antwort: Client, NO
- Verbindung endet

2. Neue Anfrage :

- Client sendet Anfrage: Server, Haben Sie eine neue Benachrichtigung?
- Server sendet Antwort: Client, NO
- Verbindung endet

3. Neue Anfrage :

- Client sendet Anfrage: Server, Haben Sie eine neue Benachrichtigung?
- Server sendet Antwort: Client, NO
- Verbindung endet

4. Neue Anfrage :

- Client sendet Anfrage: Server, Haben Sie eine neue Benachrichtigung?
- Server sendet Antwort: Client, ja hier sind Sie
- Verbindung endet

Sie sehen, dass es eine Anzahl unbrauchbarer Anfragen gibt!

Web-Socket

- Verbindung hergestellt :

- Client sagt: Server, Haben Sie eine neue Benachrichtigung?
nach xx ist die Zeit vergangen
- Server sagt: Ja habe ich
nach xx ist die Zeit vergangen
- Server sagt: Ja habe ich
- Client sagt: Server, neue E-Mail senden, wenn ich habe
nach xx ist die Zeit vergangen
- Server sagt: Jetzt haben Sie eine neue E-Mail erhalten

Jetzt ist klar, dass wir mit Web-Socket keine unbrauchbaren Anfragen haben

Unterschiede zwischen WebSockets und AJAX online lesen:

<https://riptutorial.com/de/websocket/topic/8139/unterschiede-zwischen-websockets-und-ajax>

Credits

S. No	Kapitel	Contributors
1	Erste Schritte mit WebSockets	Brock Adams , Community , Farhad
2	Unterschiede zwischen WebSockets und AJAX	Farhad , jalanb