



FREE eBook

LEARNING WebSockets

Free unaffiliated eBook created from
Stack Overflow contributors.

#websocket

Table of Contents

About	1
Chapter 1: Getting started with WebSockets	2
Remarks.....	2
Examples.....	2
Installation or Setup.....	2
web-socket Client example.....	2
Chapter 2: Differences between WebSockets and AJAX	4
Introduction.....	4
Examples.....	4
example getting notification.....	4
Credits	6

About

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

Chapter 1: Getting started with WebSockets

Remarks

This section provides an overview of what websocket is, and why a developer might want to use it.

It should also mention any large subjects within websocket, and link out to the related topics. Since the Documentation for websocket is new, you may need to create initial versions of those related topics.

Examples

Installation or Setup

Detailed instructions on getting web-socket set up or installed.

web-socket Client example

This is an example of a web-socket client in javascript.

It:

1. Connects to a live demo server.
2. Sends a message.
3. Receives message(s).
4. Disconnects after an interval.

```
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).");
}
```

If you run that code in a console you get a result like:

```
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.
```

Read [Getting started with WebSockets](https://riptutorial.com/websocket/topic/7958/getting-started-with-websockets) online: <https://riptutorial.com/websocket/topic/7958/getting-started-with-websockets>

Chapter 2: Differences between WebSockets and AJAX

Introduction

Ajax uses the HTTP Protocol and can send requests using POST/GET methods from Client to Server.

WebSocket is itself a protocol to communicate between Client and Server, distinct from HTTP.

In Ajax when you send a request , server sends response for that request and connection ends.

Using WebSockets when you establish a connection with server , then you can communicate between client and server as much you want and it keeps connection alive.

Examples

example getting notification

Ajax :

1. New request :
 - Client sends request : Server , Do you have new notification ?
 - Server sends response : Client , NO
 - connection ends
2. New request :
 - Client sends request : Server , Do you have new notification ?
 - Server sends response : Client , NO
 - connection ends
3. New request :
 - Client sends request : Server , Do you have new notification ?
 - Server sends response : Client , NO
 - connection ends
4. New request :
 - Client sends request : Server , Do you have new notification ?
 - Server sends response : Client , Yes here you are
 - connection ends

you see that there is number of useless requests !

web-socket

- connection established :
- client says : Server , Do you have new notification ?

after xx time passed

- Server says : Yes I have

after xx time passed

- Server says : Yes I have
- Client says : Server , send new email if I have

after xx time passed

- Server says : now you got new email

now the it's clear that with web-socket we won't have useless requests

Read Differences between WebSockets and AJAX online:

<https://riptutorial.com/websocket/topic/8139/differences-between-websockets-and-ajax>

Credits

S. No	Chapters	Contributors
1	Getting started with WebSockets	Brock Adams , Community , Farhad
2	Differences between WebSockets and AJAX	Farhad , jalanb