



Introduction to Web Sockets



What are Web Sockets?

- Part of HTML5 set of standards,
- Allows server to push data to client,
- Allow client to send/request data from server
- Uses a persistent connection,
- Less overhead than HTTP, web sockets uses data or text frames as its protocol,
- Replaces solutions such as:
 - Comet,
 - Long polling
 - Short polling



What are Web Sockets?

- Full duplex communication channel over a single tcp connection,
- Requires web browsers and servers to support web sockets,
- Uses standard HTTP/HTTPS ports,
- Uses HTTP protocol to request an upgrade to the web socket protocol,
- No more cross domain issues,
- Ping/Pong frames for keep-alive, hearbeat,
- URI –
 - `ws://html5.co.za/chat`
 - `Wss://html5.co.za/chat`



Web Socket Protocol

```
Request URL: ws://localhost:8888/
Request Method: GET
Status Code: 101 Switching Protocols
▼ Request Headers view source
  Cache-Control: no-cache
  Connection: Upgrade
  Cookie: COOKIE_SUPPORT=true; GUEST_LANGUAGE_ID=en_GB
  Host: localhost:8888
  Origin: null
  Pragma: no-cache
  Sec-WebSocket-Extensions: x-webkit-deflate-frame
  Sec-WebSocket-Key: nKs+yyXBZxP9dIGuKl8tHg==
  Sec-WebSocket-Version: 13
  Upgrade: websocket
  User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/29.0.1547.76 Safari/537.36
▼ Response Headers view source
  Connection: Upgrade
  Sec-WebSocket-Accept: 2tD04/UraOzLpC5Hw/noHZeIUvo=
  Upgrade: websocket
```



Browser Support

Web Sockets - Candidate Recommendation

Bidirectional communication technology for web apps

Usage stats:	Global
Support:	69.72%
Partial support:	3.25%
Total:	72.97%

Show all versions	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Blackberry Browser	IE Mobile
								2.1		
						3.2		2.2		
						4.0-4.1		2.3		
						4.2-4.3		3.0		
	8.0	22.0				5.0-5.1		4.0		
	9.0	23.0	28.0	5.1		6.0-6.1		4.1	7.0	
Current	10.0	24.0	29.0	6.0	16.0	7.0	5.0-7.0	4.2	10.0	10.0
Near future	11.0	25.0	30.0	7.0	17.0					
Farther future			31.0							

Notes [Known issues \(1\)](#) [Resources \(6\)](#) [Feedback](#)

[Edit on GitHub](#)

Partial support refers to the websockets implementation using an older version of the protocol and/or the implementation being disabled by default (due to security issues with the older protocol).



Web Socket Applications

- Client side javascript
 - Uses standard W3C based API
- Server Side
 - Platform dependent
- Note: There are many non-standard compliant client and server libraries available to mimic web sockets.
 - Be aware that the server side implementation must support the client side protocol



Server Support

- Apache does not yet have support
 - Many plugins/libraries/standalone servers for different technologies,
 - PHP – Ratchet
- Java – Glassfish/Tomcat – Support in Java EE 7
- Javascript – Node.js - many libraries not all standard compliant and not all good.

Client Web Socket API

```
[Constructor(DOMString url, optional (DOMString or DOMString[]) protocols)]
interface WebSocket : EventTarget {
  readonly attribute DOMString url;

  // ready state
  const unsigned short CONNECTING = 0;
  const unsigned short OPEN = 1;
  const unsigned short CLOSING = 2;
  const unsigned short CLOSED = 3;
  readonly attribute unsigned short readyState;
  readonly attribute unsigned long bufferedAmount;

  // networking
      attribute EventHandler onopen;
      attribute EventHandler onerror;
      attribute EventHandler onclose;
  readonly attribute DOMString extensions;
  readonly attribute DOMString protocol;
  void close([Clamp] optional unsigned short code, optional DOMString reason);

  // messaging
      attribute EventHandler onmessage;
      attribute DOMString binaryType;
  void send(DOMString data);
  void send(Blob data);
  void send(ArrayBuffer data);
  void send(ArrayBufferView data);
};
```




Events

- Message Event
 - Data -
 - String,
 - Blob,
 - ArrayBuffer
- Close Event
 - code – unsigned long – code provided by server
 - reason – string – reason for close
 - wasClean – boolean – was connection cleanly closed

Simple Demo



- Send message to server
- Illustrate the basic structure of client side javascript
- Use node.js on the server side

Chat Demo



- Show interaction between two clients,
 - Node.js on the server side.