

WebSockets - Functionalities

Web Socket represents a major upgrade in the history of web communications. Before its existence, all communication between the web clients and the servers relied only on HTTP.

Web Socket helps in dynamic flow of the connections that are persistent full duplex. Full duplex refers to the communication from both the ends with considerable fast speed.

It is termed as a game changer because of its efficiency of overcoming all the drawbacks of existing protocols.

Web Socket for Developers and Architects

Importance of Web Socket for developers and architects –

- Web Socket is an independent TCP-based protocol, but it is designed to support any other protocol that would traditionally run only on top of a pure TCP connection.
- Web Socket is a transport layer on top of which any other protocol can run. The Web Socket API supports the ability to define sub-protocols: protocol libraries that can interpret specific protocols.
- Examples of such protocols include XMPP, STOMP, and AMQP. The developers no longer have to think in terms of the HTTP request-response paradigm.
- The only requirement on the browser-side is to run a JavaScript library that can interpret the Web Socket handshake, establish and maintain a Web Socket connection.
- On the server side, the industry standard is to use existing protocol libraries that run on top of TCP and leverage a Web Socket Gateway.

The following diagram describes the functionalities of Web Sockets –



Web Socket connections are initiated via HTTP; HTTP servers typically interpret Web Socket handshakes as an Upgrade request.

Web Sockets can both be a complementary add-on to an existing HTTP environment and can provide the required infrastructure to add web functionality. It relies on more advanced, full duplex protocols that allow data to flow in both directions between client and server.

Functions of Web Sockets

Web Sockets provide a connection between the web server and a client such that both the parties can start sending the data.

The steps for establishing the connection of Web Socket are as follows –

- The client establishes a connection through a process known as Web Socket handshake.
- The process begins with the client sending a regular HTTP request to the server.
- An Upgrade header is requested. In this request, it informs the server that request is for Web Socket connection.
- Web Socket URLs use the **ws** scheme. They are also used for secure Web Socket connections, which are the equivalent to HTTPS.

A simple example of initial request headers is as follows –

```
GET ws://websocket.example.com/ HTTP/1.1
Origin: http://example.com
Connection: Upgrade
Host: websocket.example.com
Upgrade: websocket
```