



# Web Sockets

**tutorialspoint**

SIMPLY EASY LEARNING

[www.tutorialspoint.com](http://www.tutorialspoint.com)



<https://www.facebook.com/tutorialspointindia>



<https://twitter.com/tutorialspoint>

## About the Tutorial

---

Web sockets are defined as a two-way communication between the servers and the clients, which mean both the parties, communicate and exchange data at the same time.

This protocol defines a full duplex communication from the ground up. Web sockets take a step forward in bringing desktop rich functionalities to the web browsers. It represents an evolution, which was awaited for a long time in client/server web technology.

## Audience

---

This tutorial has been prepared for anyone who has a basic knowledge of Protocols and understanding of HTTP. After completing this tutorial, you will find yourself at a moderate level of expertise in understanding what makes Web Sockets different from the traditional HTTP request/response pattern.

## Prerequisites

---

Before you start proceeding with this tutorial, we are assuming that you are already aware about the basics of JavaScript and understanding of the HTTP protocol. If you are not well aware of these concepts, then we will suggest you to go through our short tutorials on JavaScript and HTTP.

## Copyright & Disclaimer

---

© Copyright 2016 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at [contact@tutorialspoint.com](mailto:contact@tutorialspoint.com)

## Table of Contents

---

About the Tutorial .....	i
Audience.....	i
Prerequisites.....	i
Copyright & Disclaimer .....	i
Table of Contents .....	ii
<b>UNIT 1: WEB SOCKETS – INTRODUCTION.....</b>	<b>1</b>
<b>1. Web Sockets – Overview.....</b>	<b>2</b>
Web Sockets – Definition .....	2
Description of Web Socket Protocol.....	3
<b>2. Web Sockets – Duplex Communication.....</b>	<b>4</b>
Polling .....	4
Long Polling .....	4
Streaming .....	4
AJAX.....	4
HTML5 .....	5
Why Do We Need Web Sockets?.....	5
<b>3. Web Sockets – Functionalities.....</b>	<b>6</b>
<b>4. Web Sockets – Implementation .....</b>	<b>9</b>
<b>UNIT 2: WEB SOCKETS – ROLES.....</b>	<b>10</b>
<b>5. Web Sockets – Events and Actions.....</b>	<b>11</b>
Web Sockets – .....	11
Events .....	11
Web Sockets – .....	12
Actions.....	12
<b>6. Web Sockets – Opening Connections .....</b>	<b>14</b>
<b>7. Web Sockets – Handling Errors .....</b>	<b>17</b>
Checking Network Availability .....	17
<b>8. Web Sockets – Send &amp; Receive Messages .....</b>	<b>21</b>
Strings.....	21
JSON (JavaScript Object Notation) .....	22
XML.....	22
ArrayBuffer .....	22
<b>9. Web Sockets – Closing a Connection .....</b>	<b>25</b>
<b>UNIT 3: JAVASCRIPT &amp; WEB SOCKETS API .....</b>	<b>28</b>
<b>10. Web Socket Server – Working.....</b>	<b>29</b>
Connecting to the Web Server .....	29
Creating a Web Socket Server Instance.....	29

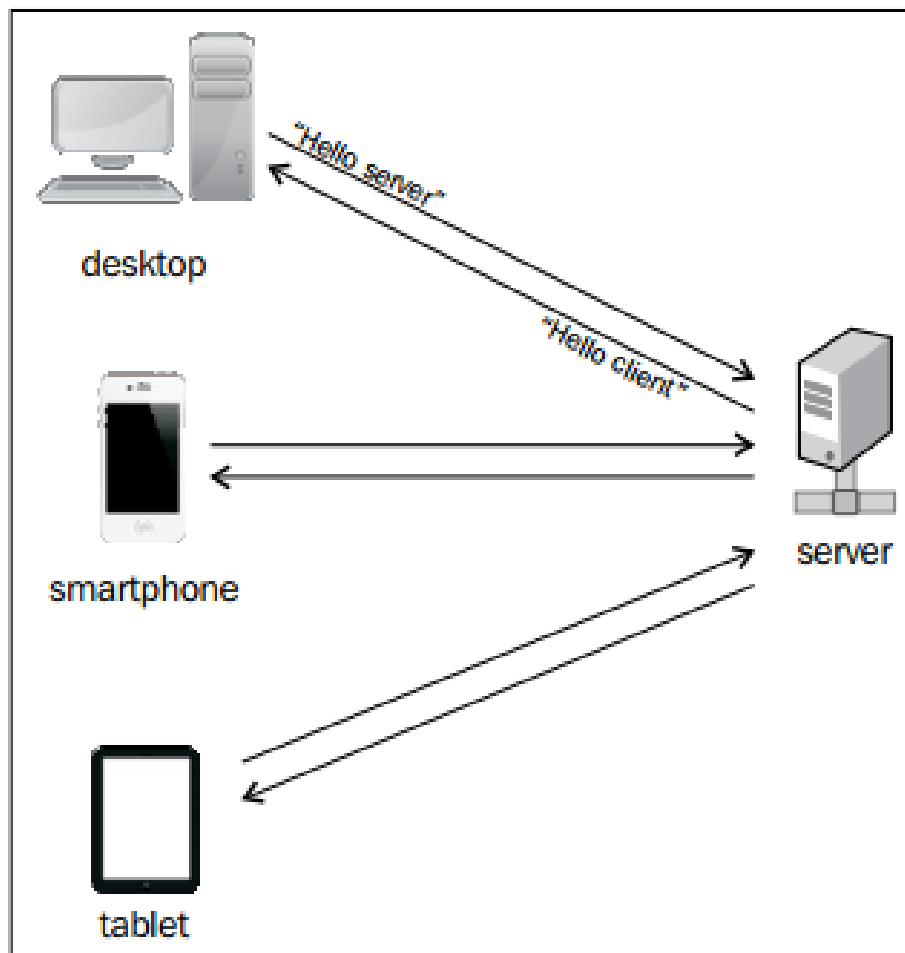
OnOpen Event .....	30
OnClose Event .....	30
OnMessage Event .....	30
Send () Method.....	31
<b>11. Web Socket as API .....</b>	<b>32</b>
API – Definition.....	32
Advantages of Web Socket.....	32
<b>12. Web Sockets – JavaScript Application .....</b>	<b>35</b>
<b>UNIT 4: WEB SOCKETS – IMPLEMENTATION .....</b>	<b>41</b>
<b>13. Web Sockets – Communicating with Server .....</b>	<b>42</b>
Bringing Sockets to the Web .....	42
Communicating with the Server .....	43
<b>14. Web Sockets – Security .....</b>	<b>45</b>
Denial of Service .....	45
Man-in-the-middle .....	45
XSS .....	45
WebSocket Native Defense Mechanisms.....	46
<b>15. Web Sockets – Mobile APP .....</b>	<b>47</b>
Why Mobile Matters?.....	47
Native Mobile App Vs Mobile Website.....	47

# Unit 1: Web Sockets – Introduction

# 1. Web Sockets – Overview

In literal terms, handshaking can be defined as gripping and shaking of right hands by two individuals, as to symbolize greeting, congratulations, agreement or farewell. In computer science, handshaking is a process that ensures the server is in sync with its clients. Handshaking is the basic concept of Web Socket protocol.

The following diagram shows the server handshake with various clients:



## Web Sockets – Definition

Web sockets are defined as a two-way communication between the servers and the clients, which mean both the parties communicate and exchange data at the same time.

The key points of Web Sockets are **true concurrency** and **optimization of performance**, resulting in more responsive and rich web applications.

## Description of Web Socket Protocol

---

This protocol defines a full duplex communication from the ground up. Web sockets take a step forward in bringing desktop rich functionalities to the web browsers. It represents an evolution, which was awaited for a long time in client/server web technology.

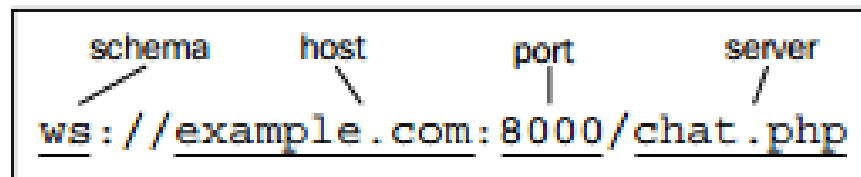
The main features of web sockets are as follows:

- Web socket protocol is being standardized, which means real time communication between web servers and clients is possible with the help of this protocol.
- Web sockets are transforming to cross platform standard for real time communication between a client and the server.
- This standard enables new kind of the applications. Businesses for real time web application can speed up with the help of this technology.
- The biggest advantage of Web Socket is it provides a two-way communication (full duplex) over a single TCP connection.

## URL

HTTP has its own set of schemas such as http and https. Web socket protocol also has similar schema defined in its URL pattern.

The following image shows the Web Socket URL in tokens.



## Browser Support

The latest specification of Web Socket protocol is defined as **RFC 6455** – a proposed standard.

**RFC 6455** is supported by various browsers like Internet Explorer, Mozilla Firefox, Google Chrome, Safari, and Opera.

# 2. Web Sockets – Duplex Communication

Before diving to the need of Web sockets, it is necessary to have a look at the existing techniques, which are used for **duplex communication** between the server and the client. They are as follows:

1. Polling
2. Long Polling
3. Streaming
4. Postback and AJAX
5. HTML5

## Polling

---

Polling can be defined as a method, which performs periodic requests regardless of the data that exists in the transmission. The periodic requests are sent in a synchronous way. The client makes a periodic request in a specified time interval to the client. The response of the server includes available data or some warning message in it.

## Long Polling

---

Long polling, as the name suggests, includes similar technique like polling. The client and the server keep the connection active until some data is fetched or timeout occurs. If the connection is lost due to some reasons, the client can start over and perform sequential request.

Long polling is nothing but performance improvement over polling process, but constant requests may slow down the process.

## Streaming

---

It is considered as the best option for real-time data transmission. The server keeps the connection open and active with the client until and unless the required data is being fetched. In this case, the connection is said to be open indefinitely. Streaming includes HTTP headers which increases the file size, increasing delay. This can be considered as a major drawback.

## AJAX

---

AJAX is based on Javascript's **XmlHttpRequest** Object. It is an abbreviated form of Asynchronous Javascript and XML. **XmlHttpRequest** Object allows execution of the Javascript without reloading the complete web page. AJAX sends and receives only a portion of the web page.

The code snippet of AJAX call with **XmlHttpRequest** Object is as follows:



```

var xmlhttp;
if (window.XMLHttpRequest) {
  xmlhttp = new XMLHttpRequest();
} else {
  // code for IE6, IE5
  xmlhttp =new ActiveXObject("Microsoft.XMLHTTP");
}

```

The major drawbacks of **AJAX** in comparison with **Web Sockets** are:

- They send HTTP headers, which makes total size larger.
- The communication is half-duplex.
- The web server consumes more resources.

## HTML5

HTML5 is a robust framework for developing and designing web applications. The main pillars include **Mark-up**, **CSS3** and **Javascript** APIs together.

The following diagram shows HTML5 components:



The code snippet given below describes the declaration of HTML5 and its doctype.

```
<!DOCTYPE html>
```

## Why Do We Need Web Sockets?

Internet was conceived to be a collection of Hypertext Mark-up Language (HTML) pages linking one another to form a conceptual web of information. During the course of time, static resources increased in number and richer items, such as images and began to be a part of the web fabric.

Server technologies advanced which allowed dynamic server pages - pages whose content was generated based on a query.

Soon, the requirement to have more dynamic web pages lead to the availability of Dynamic Hypertext Mark-up Language (DHTML). All thanks to JavaScript. Over the following years, we saw **cross frame communication** in an attempt to avoid page reloads followed by **HTTP Polling** within frames.

However, none of these solutions offered a truly standardized cross browser solution to real-time bi-directional communication between a server and a client.

This gave rise to the need of Web Sockets Protocol. It gave rise to full-duplex communication bringing desktop-rich functionality to all web browsers.

# 3. Web Sockets – 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.

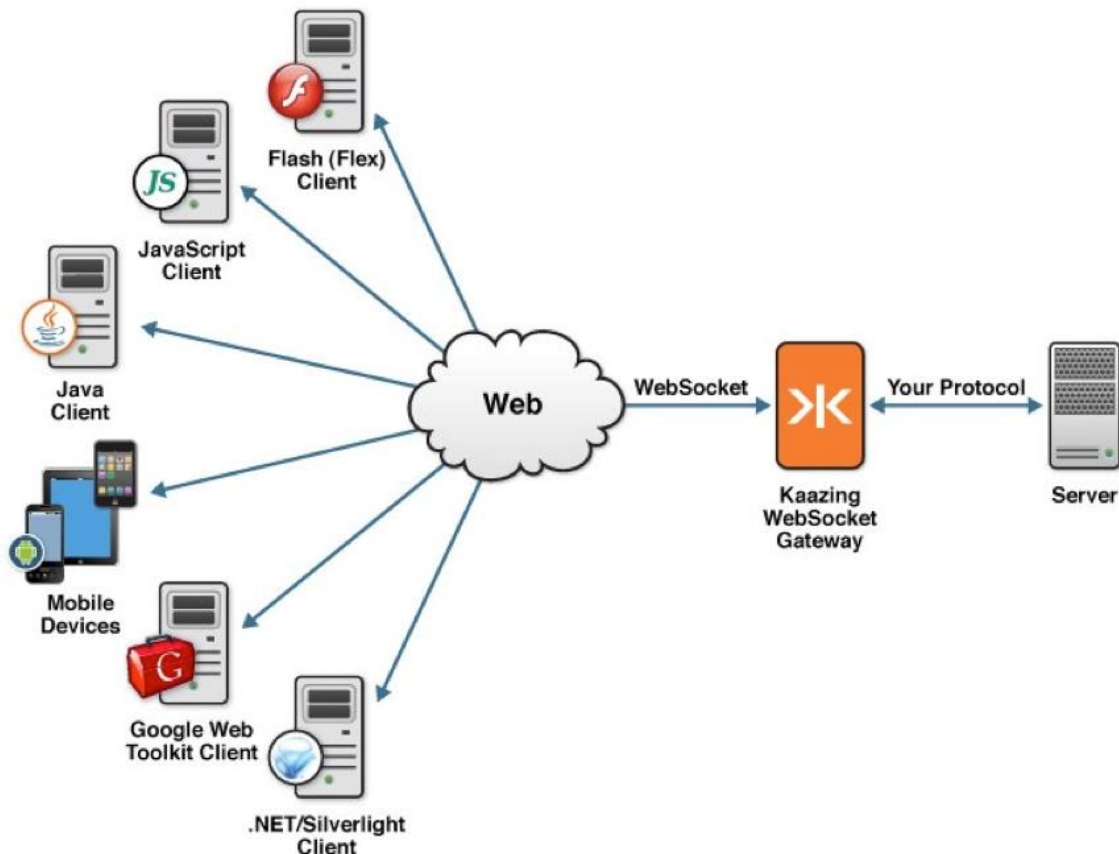
It is termed as a game changer because of its efficiency of overcoming all the drawbacks of existing protocols.

## 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:

1. The client establishes a connection through a process known as Web Socket handshake.
2. The process begins with the client sending a regular HTTP request to the server.

3. An Upgrade header is requested. In this request, it informs the server that request is for Web Socket connection.
4. Web Socket URLs use the **ws** scheme. They are also used for secure Web Socket connections, which are the equivalent to HTTPS.

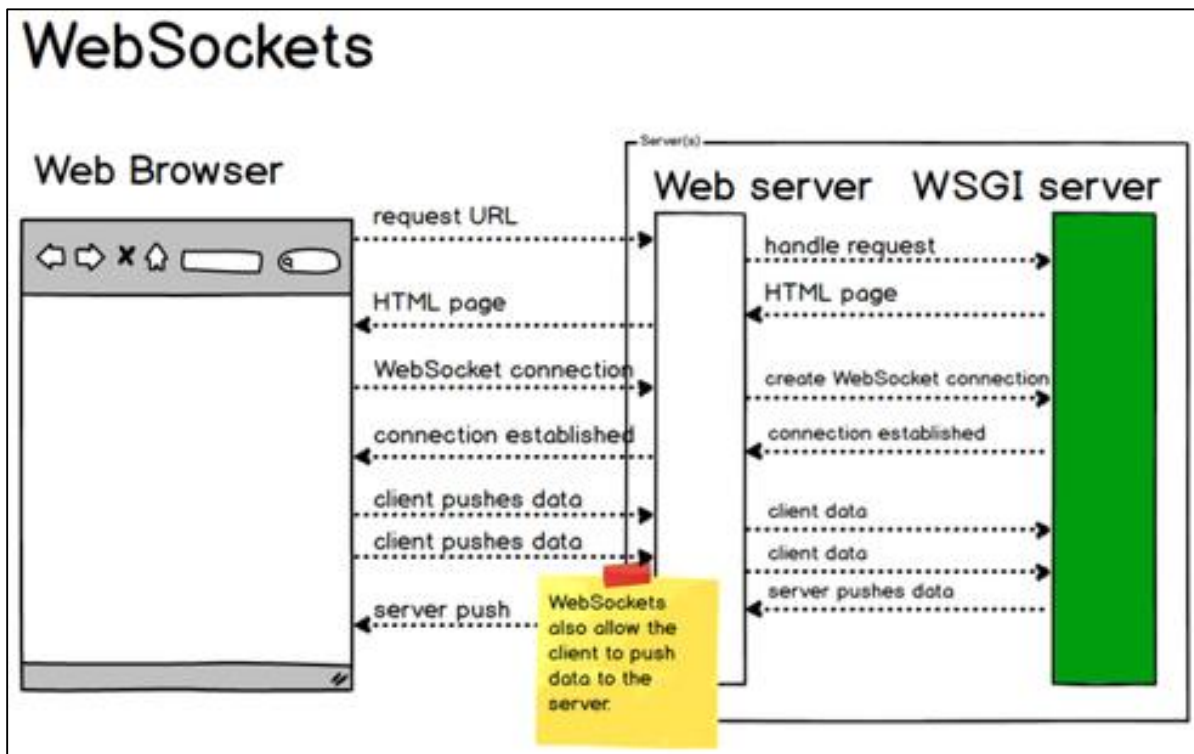
A simple example of initial request headers is as follows:

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

# 4. Web Sockets – Implementation

Web Sockets occupy a key role not only in the web but also in the mobile industry. The importance of Web Sockets is given below.

- Web Sockets as the name indicates, are related to the web. Web consists of a bunch of techniques for some browsers; it is a broad communication platform for vast number of devices, including desktop computers, laptops, tablets and smart phones.
- HTML5 app that utilizes Web Sockets will work on any HTML5 enabled web browser.
- Web socket is supported in the mainstream operating systems. All key players in the mobile industry provide Web Socket APIs in own native apps.
- Web sockets are said to be a full duplex communication. The approach of Web Sockets works well for certain categories of web application such as chat room, where the updates from client as well as server are shared simultaneously.



Web Sockets, a part of the HTML5 specification, allow full duplex communication between web pages and a remote host. The protocol is designed to achieve the following benefits, which can be considered as the key points:

- **Reduce unnecessary network traffic and latency** using full duplex through a single connection (instead of two).

- **Streaming through proxies and firewalls**, with the support of upstream and downstream communication simultaneously.

## Unit 2: Web Sockets – Roles



# 5. Web Sockets – Events and Actions

It is necessary to initialize the connection to the server from client for communication between them. For initializing the connection, creation of Javascript object with the URL with the remote or local server is required.

```
var socket = new WebSocket(" ws://echo.websocket.org ");
```

The URL mentioned above is a public address that can be used for testing and experiments. The websocket.org server is always up and when it receives the message and sends it back to the client.

This is the most important step to ensure that application works correctly.

## Web Sockets – Events

---

There are four main Web Socket API **events**:

- Open
- Message
- Close
- Error

Each of the events are handled by implementing the functions like **onopen**, **onmessage**, **onclose** and **onerror** functions respectively. It can also be implemented with the help of **addEventListener** method.

The brief overview of the events and functions are described as follows:

### Open

Once the connection has been established between the client and the server, the open event is fired from Web Socket instance. It is called as the initial handshake between client and server. The event, which is raised once the connection is established, is called **onopen**.

### Message

Message event happens usually when the server sends some data. Messages sent by the server to the client can include plain text messages, binary data or images. Whenever the data is sent, the **onmessage** function is fired.

### Close

**Close** event marks the end of the communication between server and the client. Closing the connection is possible with the help of **onclose** event. After marking the end of communication with the help of **onclose** event, no messages can be further transferred

between the server and the client. Closing the event can happen due to poor connectivity as well.

## Error

**Error** marks for some mistake, which happens during the communication. It is marked with the help of **onerror** event. **Onerror** is always followed by termination of connection. The detailed description of each and every event is discussed in further chapters.

End of ebook preview  
If you liked what you saw...  
Buy it from our store @ <https://store.tutorialspoint.com>