



---

[HTML](#) [CSS](#) [JavaScript](#) [TypeScript](#) [jQuery](#) [AngularJS](#) [ReactJS](#) [Next.js](#) [React Native](#) [NodeJS](#) [Expre](#)

---

# What is web socket and how it is different from the HTTP?

Last Updated : 22 Aug, 2024

---

HTTP and WebSocket are both communication protocols used in client-server communication.

## HTTP protocol

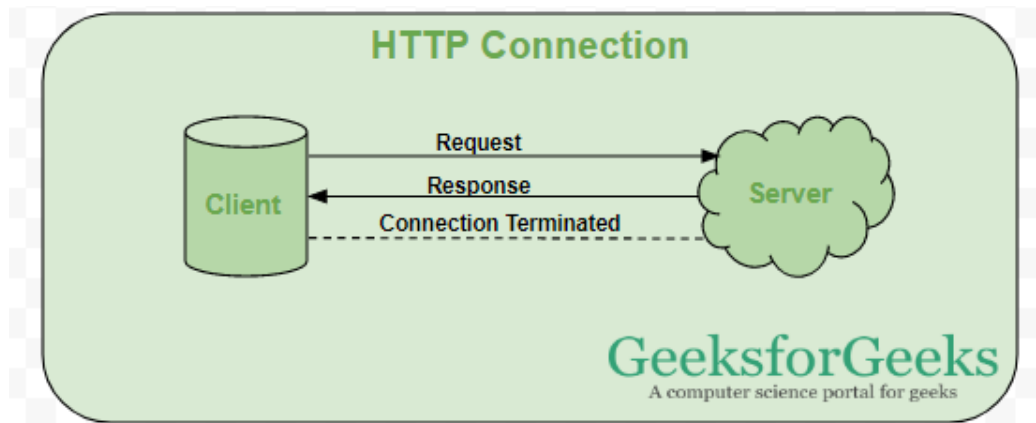
HTTP is unidirectional where the client sends the request and the server sends the response. Let's take an example when a user sends a request to the server this request goes in the form of HTTP or HTTPS, and after receiving a request server sends the response to the client, each request is associated with a corresponding response, and after sending the response the connection gets closed, each HTTP or HTTPS request establish a new connection to the server every time and after getting the response the connection gets terminated by itself.

HTTP protocols are important to learn if you are working with websites and if you wish to learn web development then you can check out our [Full Stack Web development course](#) in which we have covered all these points.

HTTP is a stateless protocol that runs on top of TCP which is a connection-oriented protocol it guarantees the delivery of data packet transfer using the three-way handshaking methods and re-transmits the lost packets.

HTTP can run on top of any reliable connection-oriented protocol such as TCP, or SCTP. When a client sends an HTTP request to the server, a TCP connection is open between the client and server and after getting the response the TCP connection gets terminated, each HTTP request opens a separate TCP connection to the server, e.g. if the client sends 10 requests to the server then all 10 separate TCP connection will be opened and will get close after getting the response/fallback.

HTTP message information encoded in ASCII, each HTTP request message is composed of HTTP protocol version(HTTP/1.1, HTTP/2), HTTP methods (GET/POST, etc.), HTTP headers (content type, content length), host information, etc. and the body which contain the actual message which is being transferred to the server. HTTP headers vary from 200 bytes to 2 KB in size, the common size of HTTP headers is 700-800 bytes. When a web application uses more cookies and other tools on the client side that expand the storage features of the agent it reduces the HTTP header payload.



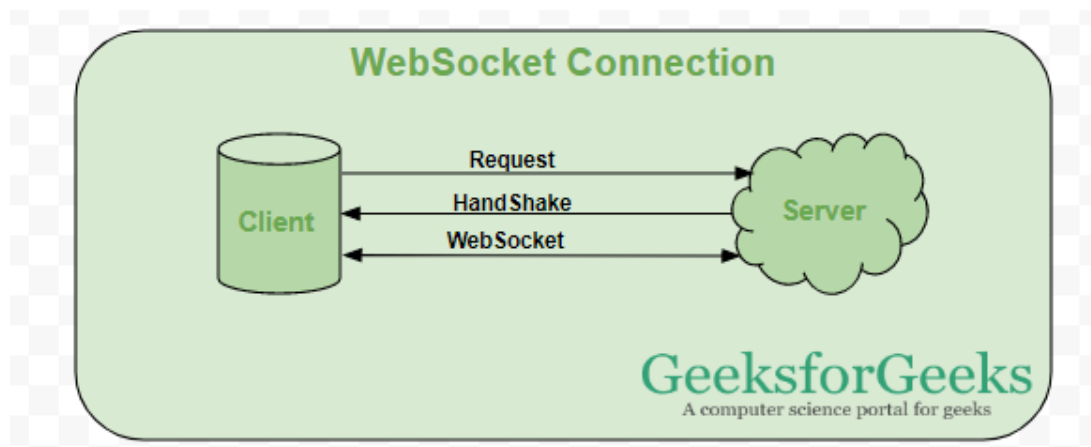
## WebSocket

WebSocket is bidirectional, a full-duplex protocol that is used in the same scenario of client-server communication, unlike HTTP which starts from **ws://** or **wss://**. It is a stateful protocol, which means the connection between client and server will stay alive until it gets terminated by either party (client or server). After closing the connection by either of the client or server, the connection is terminated from both ends.

Let's take an example of client-server communication, there is the client which is a web browser, and a server, whenever we initiate the connection between client and server, the client-server makes the handshaking and decides to create a new connection and this connection will keep alive until terminated by any of them. When the connection is established and alive the communication takes place using the same connection channel until it is terminated.

This is how after client-server handshaking, the client-server decides to keep a new connection alive, this new connection will be known as WebSocket. Once the communication link is established and the connections are opened, message exchange will take place in bidirectional mode until the connection persists between client-server. If anyone of them (client-server) dies or decide

to close the connection then it is closed by both the party. The way in which the socket works is slightly different from how HTTP works, the status code 101 denotes the switching protocol in WebSocket.



## When can a web socket be used?

- **Real-time web application:** Real-time web application uses a web socket to show the data at client end, which is continuously being sent by the backend server. In WebSocket, data is continuously pushed/transmitted into the same connection which is already open, that is why WebSocket is faster and improves the application performance.  
e.g. in a trading website or bitcoin trading, for displaying the price fluctuation and movement data is continuously pushed by the backend server to the client end by using a WebSocket channel.
- **Gaming application:** In a Gaming application, you might focus on that, data is continuously received by the server, and without refreshing the UI, it will take effect on the screen, UI gets automatically refreshed without even establishing the new connection, so it is very helpful in a Gaming application.
- **Chat application:** Chat applications use WebSockets to establish the connection only once for exchange, publishing, and broadcasting the message among the subscribers. It reuses the same WebSocket connection, for sending and receiving the message and for one-to-one message transfer.

## When not to use WebSocket?

WebSocket can be used if we want any real-time updated or continuous streams of data that are being transmitted over the network but if we want to fetch old data, or want to get the data only once to process it with an application we should go with **HTTP protocol**, old data which is not required very frequently or fetched only once can be queried by the simple HTTP request, so in this scenario, it's better not use WebSocket.

***Note:** RESTful web services are sufficient to get the data from the server if we are loading the data only once.*

## Differences between HTTP and WebSocket Connection

WebSocket Connection	HTTP Connection
WebSocket is a bidirectional communication protocol that can send the data from the client to the server or from the server to the client by reusing the established connection channel. The connection is kept alive until terminated by either the client or the server.	The HTTP protocol is a unidirectional protocol that works on top of TCP protocol which is a connection-oriented transport layer protocol, we can create the connection by using HTTP request methods after getting the response HTTP connection get closed.
Almost all the real-time applications like (trading, monitoring, notification) services use WebSocket to receive the data on a single communication channel.	Simple RESTful application uses HTTP protocol which is stateless.
All the frequently updated applications used WebSocket because it is faster than HTTP Connection.	It is used when we do not want to retain a connection for a particular amount of time or reuse the connection for transmitting data; An HTTP connection is slower than WebSockets.

**Note:** Depending on your project you have to choose where it will be WebSocket or HTTP Connection.

Want to be a Software Developer or a Working Professional looking to enhance your **Software Development Skills**? Then, master the concepts of Full-Stack Development. Our [Full Stack Development - React and Node.js](https://www.geeksforgeeks.org/full-stack-development-react-and-node-js/)

[Course](#) will help you achieve this quickly. Learn everything from **Front-End to Back-End Development** with hands-on **Projects** and real-world examples. This course enables you to build scalable, efficient, dynamic web applications that stand out. Ready to become an expert in Full-Stack? Enroll Now and Start Creating the Future!

A Arpit... [+ Follow](#)



201

### Previous Article

Servers, streams and sockets in Node

### Next Article

## Similar Reads

### Node.js http.ClientRequest.socket Property

The http.ClientRequest.socket is an inbuilt application programming interface of class ClientRequest within HTTP module which is used to get a Proxy object tha...

2 min read

### Node.js http.ServerResponse.socket Api

The httpServerResponse.socket is an inbuilt application programming interface of class ServerResponse within http module which is used to get the reference of...

2 min read

### Web-Socket in Node

In Node.js WebSocket enables real-time, bidirectional communication between the clients and servers. Traditional HTTP follows a request-response model whi...

4 min read

### Why are HTTP cookies used by Node.js for sending and receiving HTTP...

The HTTP protocol is one of the most important protocols of the Application layer of the OSI model. The main use of this protocol is for sending hypertext...

5 min read

### Difference between HTTP/2 and HTTP/1.1

HTTP stands for hypertext transfer protocol & it is used in client-server communication. By using HTTP user sends the request to the server & the serve...

3 min read

### Will the 'error' event ever be emitted on 'http.IncomingMessage' in a node....

In this article, we will see how to handle the 'error' event that is emitted on 'http.IncomingMessage' in a node.js http.request. In Node.js, the...

3 min read

### Difference Between Web 1.0, Web 2.0, and Web 3.0

Web 1.0 was all about fetching, and reading information. Web 2.0 is all about reading, writing, creating, and interacting with the end user. It was famously...

8 min read

### Different types of module used for performing HTTP Request and Respons...

HTTP's requests and responses are the main fundamental block of the World Wide Web. There are many approaches to perform an HTTP request and...

3 min read

### Explain Different Types of HTTP Request

Hypertext Transfer Protocol (HTTP) defines a variety of request methods to describe what action is to be done on a certain resource. The most often used...

5 min read

### Different kinds of HTTP requests

HTTP (Hypertext Transfer Protocol) specifies a collection of request methods to specify what action is to be performed on a particular resource. The most...

5 min read

Article Tags :

[GBlog](#)

[Web Technologies](#)



Corporate & Communications Address:-  
A-143, 9th Floor, Sovereign Corporate  
Tower, Sector- 136, Noida, Uttar Pradesh  
(201305) | Registered Address:- K 061,  
Tower K, Gulshan Vivante Apartment,  
Sector 137, Noida, Gautam Buddh  
Nagar, Uttar Pradesh, 201305



## Company

[About Us](#)  
[Legal](#)  
[In Media](#)  
[Contact Us](#)  
[Advertise with us](#)  
[GFG Corporate Solution](#)  
[Placement Training Program](#)  
[GeeksforGeeks Community](#)

## DSA

[Data Structures](#)  
[Algorithms](#)  
[DSA for Beginners](#)  
[Basic DSA Problems](#)  
[DSA Roadmap](#)  
[Top 100 DSA Interview Problems](#)  
[DSA Roadmap by Sandeep Jain](#)  
[All Cheat Sheets](#)

## Web Technologies

[HTML](#)  
[CSS](#)  
[JavaScript](#)  
[TypeScript](#)  
[ReactJS](#)  
[NextJS](#)  
[Bootstrap](#)  
[Web Design](#)

## Languages

[Python](#)  
[Java](#)  
[C++](#)  
[PHP](#)  
[GoLang](#)  
[SQL](#)  
[R Language](#)  
[Android Tutorial](#)  
[Tutorials Archive](#)

## Data Science & ML

[Data Science With Python](#)  
[Data Science For Beginner](#)  
[Machine Learning](#)  
[ML Maths](#)  
[Data Visualisation](#)  
[Pandas](#)  
[NumPy](#)  
[NLP](#)  
[Deep Learning](#)

## Python Tutorial

[Python Programming Examples](#)  
[Python Projects](#)  
[Python Tkinter](#)  
[Web Scraping](#)  
[OpenCV Tutorial](#)  
[Python Interview Question](#)  
[Django](#)



## Computer Science

Operating Systems  
Computer Network  
Database Management System  
Software Engineering  
Digital Logic Design  
Engineering Maths  
Software Development  
Software Testing

## System Design

High Level Design  
Low Level Design  
UML Diagrams  
Interview Guide  
Design Patterns  
OOAD  
System Design Bootcamp  
Interview Questions

## School Subjects

Mathematics  
Physics  
Chemistry  
Biology  
Social Science  
English Grammar  
Commerce  
World GK

## DevOps

Git  
Linux  
AWS  
Docker  
Kubernetes  
Azure  
GCP  
DevOps Roadmap

## Interview Preparation

Competitive Programming  
Top DS or Algo for CP  
Company-Wise Recruitment Process  
Company-Wise Preparation  
Aptitude Preparation  
Puzzles

## GeeksforGeeks Videos

DSA  
Python  
Java  
C++  
Web Development  
Data Science  
CS Subjects

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved