

# HTTP – Hyper Text Transfer Protocol

## What is HTTP

HTTP – is an application layer protocol that allows web-based applications to communicate and exchange data

It is based on TCP/IP

Usage: Originally designed to fetch HTML documents and send it to client.

## Difference between HTTP 1.1 vs HTTP2

### Shortcomings in HTTP 1.1

- ➔ **Head of line Blocking** – HTTP 1.1 has fixed number of TCP connections – 6 TCP connections per hostname. This means, at the max, only 6 parallel requests can be made to the server. **A new request has to wait for the previous request on the same connection to complete before the client can make a new request.**

Example: (for 1 TCP connection): Consider, client has parsed the index.html file and has found that it needs, STYLE.CSS & SCRIPT.JS files to load the web page. In this case, client can't make parallel requests requesting CSS & JS at the same time. First it has to make a request for STYLE.CSS sheet and has to wait for the request to complete before the client can make a new request for SCRIPT.JS.

- ➔ **Header Repetition** – Since HTTP 1.1 is stateless, each request which is sent has the header data. This leads to the repetition of headers. Also, the headers cannot be compressed in HTTP 1.1

### Solutions for the above shortcomings in HTTP 2.0

- ➔ Single TCP connection is used
- ➔ Multiple streams are created in the above single TCP connection in which every HTTP request is sent as stream. So, client can make a parallel request to the server and doesn't have to wait for the previous request on the same connection to complete
- ➔ TLS is mandatory for HTTP 2.0
- ➔ HPACK – Header data is separated from the actual payload and allows compression
- ➔ PUSH – Push allows to send mandatory resources in advance  
Example -> whenever a client requests for index.html, server can use PUSH and send the script.js, style.css files along with the response for the initial request

Blog Link: <https://muthu-rama-krishnan.medium.com/>

## HTTP Version History

Version	Year Released	Feature
HTTP 0.9	1991	Originally designed for transferring HTML documents Had only GET, POST, HEAD methods
HTTP 1.0	1996	PUT, DELETE, LINK and UNLINK methods were introduced
HTTP 1.1	1997	CORS, Keep Alive were introduced.

## Browser JS VS Nodejs

### Browser JS

- ➔ It is a programming language used in the client browsers that allows you to implement complex features on web pages – displaying dynamic content like timely content updates, animated 2D/3D graphics, control multimedia, animate images etc
- ➔ It is a programming language which runs on the browser's JS engine
- ➔ It is the only programming language which is used for client-side scripting
- ➔ Since it is browser dependent, each browser has different JavaScript engines to parse and run the code
- ➔ Example – V8 engine, Chakra, Trident, Spider monkey

### Node JS

- ➔ Node is just a Run-Time environment which enables the JavaScript to run on the server.
- ➔ Node JS runs the same JavaScript code on the server and handles all the requests made by the client
- ➔ Node uses V8 as the JavaScript engine to parse and run your JavaScript code. It is same engine which is used in the google chrome browser
- ➔ Unlike, client-side scripting, Node JS is not the only server-side solution for creating applications. We have several options for server-side programming like -> C, C++, java (spring), python etc. While JavaScript is the only programming language which is used for client-side scripting

## What happens when you type a URL in the address bar in the browser?

- ➔ **Initial Typing**
- ➔ **Figures out the protocol it should use HTTP or HTTPS**
- ➔ **DNS Lookup**  
**Browser makes a call to the DNS server (Domain Name Server) to perform IP lookup ->**  
where DNS takes URL as the input and provides the corresponding IP address
- ➔ **Once the IP is retrieved, a TCP connection is established**
- ➔ Data transmission occurs. (i.e Based on the client's request, server serves the connection)