1. HTTP1.1 AND HTTP 2

The HTTP1.1 had been introduced in 1997 as an internet communication standard since then a few revisions took place. In 2015 HTTP 2 came into use for better performance and security. HTTP 2 achieve this by techniques such as compression, multiplexing, and prioritization. HTTP1.1 keeps requests and responses in plain text format on the other hand HTTP 2 use a binary framing layer to convert these messages into binary.

As opposed to HTTP1.1, which must make use of multiple TCP connections to lessen the effect of HOL blocking, HTTP 2 establishes a single connection object between the two machines. The requests and responses can run in parallel without blocking the messages behind them, a process called multiplexing.This allows the client to construct multiple streams in parallel, these streams only need to make use of a single TCP connection. Having a single persistent connection per origin improves upon HTTP1.1 by reducing the memory and operational cost.HTTP 2 use encryption connection only which increases security. A server can send a resource to a client along with the requested HTML page, providing the resource before the client asks for it .This process is called server push which helps to load website faster. HTTP 2 also compresses headers. As these are often similar among a set of requests, this removes duplication and overhead of data transmitted.

1. HTTP VERSION HISTORY

  HTTP is the foundation of data communication for the World wide web. Tim Berners Lee and his team in 1989. The initial version of HTTP had no version number; it has been later called HTTP 0.9 to differentiate it from the later versions. There were no HTTP headers, meaning that only HTML files could be transmitted, but no other type of documents. There were no status or error codes: in case of a problem, a specific HTML file was send back with the description of the problem contained in it, for human consumption.

HTTP 1.0 was the first documented in 1996.A status code line is sent at the beginning of the response, allowing the browser itself to understand the success or failure of the request .The notion of HTTP headers has been introduced, both for the requests and the responses, allowing metadata to be transmitted and making the protocol extremely flexible and extensible.With the help of the new HTTP headers, the ability to transmit other documents than plain HTML files has been added.

HTTP 1.1 clarified ambiguities and introduced in1997. Pipelining has been added, allowing to send a second request before the answer for the first one is fully transmitted, lowering the latency of the communication. Content negotiation, including language, encoding, or type, has been introduced, and allows a client and a server to agree on the most adequate content to exchange. Thanks to the Host header, the ability to host different domains at the same IP address now allows server colocation.HTTP 1.1 was first published in January 1997. This protocol has been extremely stable over more than 15 years.

In 2015 HTTP 2 came into use for better performance and security.Which was also compatible for HTTP 1.1. HTTP 2 achieve this by techniques such as compression, multiplexing, and prioritization. HTTP didn't stop evolving upon the release of HTTP 2.The adoption of HTTP 2 hints at a bright future for the protocol.

3.JAVA SCRIPT Vs NODE JS

JAVA SCRIPT NODE JS

|  |  |
| --- | --- |
| Javascript is a programming language that is used for writing scripts on the website. | NodeJS is a Javascript runtime environment. |
| Javascript can only be run in the browsers. | NodeJS code can be run outside the browser. |
| It is basically used on the client-side. | It is mostly used on the server-side. |
| Javascript is capable enough to add HTML and play with the DOM. | Nodejs does not have capability to add HTML tags. |
| Javascript is used in frontend development. | Nodejs is used in server-side development. |

4 **.Steps for what happens when we enter a URL**

1. Browser checks cache for DNS entry to find the corresponding IP address of website.  
   It looks for following cache. If not found in one, then continues checking to the next until found.
   * Browser Cache
   * Operating Systems Cache
   * Router Cache
   * ISP Cache
2. If not found in cache, ISP’s (Internet Service Provider) DNS server initiates a DNS query to find IP address of server that hosts the domain name.  
   The requests are sent using small data packets that contain information content of request and IP address it is destined for.
3. Browser initiates a TCP connection with the server using synchronize(SYN) and acknowledge(ACK) messages.
4. Browser sends an HTTP request to the web server. GET or POST request.
5. Server on the host computer handles that request and sends back a response. It assembles a response in some format like JSON, XML and HTML.
6. Server sends out an HTTP response along with the status of response.
7. Browser displays HTML content