

TASK 1

1. Difference between HTTP1 and HTTP2

HTTP - Generally the work of http is to enable the communication between the browser and the server (ie.,Hyper text transfer protocol) and to process text commands to complete the request response cycle. But Http1 can upload only a single request at a time and hence a better version had to come, where HTTP2 was released in 2015.

Differences

- Http1 is located at port 80, where as Http2 uses the port 443.
- Http sends the information in the textual form and Http2 converts the same information into the binary format and then does the transmission.
- Http1 can allow only a single request at a time that resulted in clogging when multiple responses were fetched on a single tcp connection. Where as Http2 allows the concept of multiplexing, that ensures safer transmission of multiple signals over a single connection.
- Hence when compared to Http1, this newer version results in reduced latency and better packet streaming management.
- Http2 makes the loading of a page 14 percent faster comparatively.
- Http2, ie the newer version has many additions onto the older version such as, allows push responses, header compression and faster transmission.

2. HTTP version history

- Invented by Tim Berners-Lee at CERN in the years 1989–1991, HTTP (Hypertext Transfer Protocol) is the underlying communication protocol of the World Wide Web.
- HTTP functions as a request–response protocol in the client–server computing model.
- HTTP standards are developed by the Internet Engineering Task Force (IETF) and the World Wide Web Consortium (W3C), culminating in the publication of a series of Requests for Comments (RFCs). HTTP has four versions — HTTP/0.9, HTTP/1.0, HTTP/1.1, and HTTP/2.0.
- Today the version in common use is HTTP/1.1 and the future will be HTTP/2.0.

Version	Type	Methods supported	Response type	Connection nature	Headers
HTTP 0.9	Initial and a simple client server model	Get only	Hypertext only	Terminates immediately after response.	No HTTP headers, and cannot transfer other content files.
HTTP 1.0	Browser friendly protocol	Get, Head, Post	Not limited to hypertexts, can transfer scripts and style sheets also.	Terminates immediately after response.	Headers include information about requests and responses.
HTTP 1.1 (currently in use)	Standardised protocol	Get, head, post, put, delete, trace, options	Can transfer many types of content.	Long lived connection	Keep alive headers (allows connection to stay active until an event occurs) and upgrade headers.
HTTP 2.0	Bi- directional protocol	Supports same as above versions	Supports a multiplexed response	Supports same as above versions	Supports same as above versions

3. Differences between Browser Js and Node Js

	Browser JS	Node JS
1.	Brendon Eich developed the original JS.	This was later developed by Ryan Dahl.
2.	Browser javascript executes js outside the browser.	He made it possible to execute js even outside of the browser, by Node js.
3.	Extensively used in front end programming.	Node Js makes it possible to use it even in the back end side.
4.	Due to the faster movement of Js, we can get stuck on the web due to it's slower upgradations.	But that is not the case with Node js, it gets automatically updated.
5.	Browser Js are not headless softwares.	Node js have headless softwares.
6.	Browsers don't have "require" predefined.	"Require" object is predefined in Node which is used to include modules in the app.

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

URL - Uniform Resource locator is the address of a given unique resource on the web.

Steps :

1. We enter a url on the address bar on the browser.
2. Then the browser looks for the IP address for the domain name using DNS - Domain Name System that converts the URL into an IP address.
3. Then the browser sends a HTTP request to the server.
4. The server then sends a response back to the browser which will consist of three files - HTML, CSS and Javascript.
5. Then the Rendering Engine starts rendering the files to generate DOM and SSOM tree for better reference by parsing them and the Javascript engine generates AS tree from the js file.
6. Then a render tree is generated and is fetched to the browser which converts all these files into meaningful information that is user understandable. The browser also sends further requests required if any, for any additional objects like images in Html, Css and Js.
7. Then the page is loaded and is visible to the user.