

# TASK 1: 13<sup>TH</sup> MARCH 2021

## Question 1: HTTP Version History

### Answer:

Before knowing about the evolution, let's get to know about what is HTTP?

Most used internet protocol in the world is HTTP, which is abbreviated as **H**ypertext **T**ransfer **P**rotocol, which is an application layer protocol (which is the top-most layer in the 7-layer OSI model of networking) that was designed mainly to get HTML documents and sends it to the client, later it helps to transfer documents, videos, images, audio and much more.

#### HTTP/0.9 (1991)

Speaking about the evolution, the first initial version was HTTP/0.9, which is termed as “*The one-line protocol*”, since its requests do have a single line and one possible method called *GET* followed by the path to the resource but, this does not make any improvement as it is only the file. We could not send any other documents except HTML, as its sole purpose was fulfilled.

#### HTTP/1.0 (1996)

As the 0.9 version was much limited, so there came an extended version of HTTP/1.0. Most important upgradation was done by introducing HTTP headers, which hold the metadata of both the sending side as well as the receiving side too, that helped the ability to transfer other kinds of documents than plain HTML files. The connection will be terminated soon after the response completion.

#### HTTP/1.1 (1997)

HTTP/1.0 was more effective than 0.9 but still it faced a lot of interoperability issues, which made a pathway to the first and foremost standardised HTTP version with a lot of improvements and moreover with proper standardisation, which happened in the year 1997 that also clarified more ambiguities that earlier versions of HTTP/1.1.

Numerous improvements like

- a. Cache control mechanism (*allows a client or server to transmit a variety of directives in either requests or responses*)
- b. Chunked responses (*large amount of data were encoded into smaller pieces*)
- c. Keep-alive connections (*persistency in the connection, that allowed multiple requests to use a single connection*), and many more were introduced.

#### HTTP/2 (2015)

Later in 2015, a major revision in the HTTP network protocol was held by Google, who was behind the development of HTTP/2, which induced a major change in the latency time, which was effectively minimized compared to HTTP/1.1.

Let us see more wide differentiation of HTTP1.1 vs HTTP2 in the upcoming solution.

## **Question 2: Difference between HTTP/1.1 vs HTTP/2**

### **Answer:**

HTTP/1.1 faced issues like

#### 1. HOL (Head of Line Blocking)

Whenever a request is been received a TCP connection will be created between the server and client this TCP connection gets blocked till the response is received, and earlier 1 TCP connection was viable through HTTP/1.1 later it was increased to 6 TCP connections, but these meagre number of TCP connections are not at all sufficient for recent days heavy dynamic websites.

#### 2. Repetition of Header Data

HTTP is generally a stateless connection which means every request is independent of previous and upcoming request, each request must carry their own header data which cannot be reused.

### Improvements by HTTP/2.0

HTTP had a greater improvement in the later 2015 with more functionalities packed HTTP/2.0 which used single secured TCP connection pipeline with multiple streams, which was a major update regarding the minimal latency time.

HPACK was introduced which separated the header data and request data and been zipped that enables reuse of header data which need not to be repeated in every request. It also reduces the HTTP request size.

Additionally, HTTP/2.0 was built upon HTTP/1.1 so the migration does not affect vastly, as it leverages the use of browser caching this ultimately fast as of now.

**Question 3: List 5 difference between Browser JS (console) vs Node JS**

**Answer:**

<u>Browser JS</u>	<u>Node JS</u>
Most of the time the functionality is to communication with the DOM or any other web platform API.	Does not need any object communication provided by the browser.
Compatibility issues: May not be generic and usable for every web browser available, added to this version deprecation issues too.	It controls the environment, supports modern JavaScript versions irrespective of any configuration.
JavaScript is fast but the browsers can be slow.	Whereas, this a JavaScript runtime environment that executes JS code outside a web browser.
Does not have much effective API's compared to NodeJS.	Provides multiple modules and packages like filesystem access functionality
ES Modules are implemented.	CommonJS module system is been used here.

#### **Question 4: Abstract Working of JS engine (V8).**

##### **Answer:**

##### **Purpose JS Engine**

Parsing the human typed JavaScript code to machine understandable/executable code is the major function of the JS engine. Amongst various JS engine, V8 JS engine is more popular and used by Google in Chrome.

##### **V8 engine**

A multi-threaded program written in C++, that converts the JavaScript code into optimized machine code that employs a technique called '*Just in time*' JIT compilation. Working of V8

##### **Key Components:**

JS Interpreter – Ignition

Optimizing Compiler – Turbofan

1. Initially, the JS code is been passed to the Ignition interpreter and it does convert into machine code, this conversion is termed to be JIT compilation.
2. Profiler thread, monitors the hot segments ((segments that are hit multiple times Eg: For loop)
3. Turbofan is being responsible for the hot segments, this compiler compiles the code in much optimised way generating optimized machine code, with greater efficiency.

#### **Question 5: What happens when you type a URL in the address bar in the browser?**

##### **Answer:**

Entering a URL (eg: www.guvi.in) into a web browser,

1. The browser looks up the IP address (eg: 172.67.190.71) for the domain name via DNS (Domain Name System – simply phonebook of the Internet).
2. The browser sends a HTTP request to the server (place where the html,css,js and other webpage elements of the guvi.in website are stored).
3. The server sends back a HTTP response acknowledging that the address is been matched with request, as a response the index.html will be first fetch.
4. The browser begins rendering the HTML and later the other essential components like CSS, JS and other media files are fetched via repeated process.