1. Differences between HTTP1.1 AND 2

|  |  |
| --- | --- |
| HTTP/1.1 | HTTP/2 |
| 1. It was introduced in 1997. | 1) It was introduced in 2015. |
| 1. It supports connection reuse i.e. for every TCP connection there could be multiple requests and responses, and pipelining where the client can request several resources from the server at once. However, pipelining was hard to implement due to issues such as head-of-line blocking and was not a feasible solution. | 2) Uses multiplexing,where over a single TCP connection resources to be delivered are  interleaved and arrive at the client almost at the same time. It is done using streams which can be prioritized, can have dependencies and individual flow control. It also provides a feature called server push that allows the server to send data that the client will need but has not yet requested. |
| 1. Introduces a warning header field to carry additional information about the status of a message. Can define 24 status codes, error reporting is quicker and more efficient. | 3) Underlying semantics of HTTP such as headers, status codes remain the same. |
| 1. It is relatively secure since it uses digest authentication,NTLM authentication. | 4) Security concerns from previous versions will continue to be seen in HTTP/2. However, it is better equipped to deal with them due to new TLS features like connection error of type Inadequate\_Security. |
| 1. Expands on the caching support by using additional headers like cache-control, conditional headers like If-Match and by using entity tags. | 5) HTTP/2 does not change much in terms of caching. With the server push feature if the client finds the resources are already present in the cache, it can cancel the pushed stream. |
| 1. Spiriting, concatenating, in-lining, domain sharing are some of the optimizations used as a workaround to the ‘six connections per host rule. | 6) Removes the need for unnecessary optimization hacks. |
| 1. Text based protocol that is in the readable form. | 7) It is a binary protocol (HTTP requests are sent in the form of 0s and 1s).Needs to be converted back from binary in order to read it. |

2) Differences Between GET and POST

Both GET and POST are HTTP methods to make a request to a server

|  |  |
| --- | --- |
| GET | POST |
| 1. GET is used to request data from a specified resource. | 1) POST is used to send data to a server to create/update a resource. |
| 1. GET requests remain in the browser history. | 2) POST requests do not remain in the browser history. |
| 1. GET requests can be bookmarked. | 3) POST requests cannot be bookmarked. |
| 1. GET requests can be cache. | 4) POST requests are never cache. |
| 1. GET requests are only used to request data without modification. | 5) POST is used to update a resource. |
| 1. GET requests have length restrictions. | 6) POST requests have no length restrictions. |
| 1. GET parameters are in the URL. | 7) POST parameters are in the body. |
| 1. GET requests shouldn't change the server. | 8)POST requests are OK to change the server. |
| 1. GET can be send only with limited data. | 9) POST can send unlimited data in a request. |
| 1. GET is insecure as data is visible in the URL. | 10)POST is secure as data is sent in a body and not exposed. |
| 1. GET is mostly used for view purposes. | 11)POST is mainly used for update purposes. |