**Evolution of HTTP**

HTTP was developed by Tim Berners-Lee and his team at CERN in 1989 as part of the 4 building blocks of Mesh or what is now known as the World Wide Web. The following are the version of HTTP and some improvements.

**HTTP/0.9**

* HTTP/0.9 is the first version of HTTP. It is a simple protocol used for raw data transfer across the Internet.
* Method names are case sensitive. Some of the specified methods of HTTP 0.9 are the following:
  + GET - used to retrieve the data associated with the specified URL
  + HEAD - same as GET but only returns HTTP headers
  + PUT - specifies that the data in the body section should be stored under the specified URL
  + DELETE - used to delete the data associated with the specified URL
* Unresolved Point - The client has no way of knowing what data formats the server is prepared to accept (HTTP 1992)

|  |  |  |  |
| --- | --- | --- | --- |
| Some Status Codes | | | |
| Success 2xx | Redirection 3xx | Client Error 4xx | Server 5xx |
| OK 200  Created 201  Accepted 202 | Moved 301  Found 302 | Bad Request 400  Unauthorized 401 | Internal Error 500  Not Implemented 501 |

**HTTP/1.0 (RFC1945)**

* HTTP/1.0 is an improved protocol which allowed messages to be in the form of MIME-like messages. This means that documents, other than texts, like images and sounds can be inserted in a message. Messages also contained informations about the data transferred and modifiers on the request/response semantics. (RFC2616)
* *1xx Informational*

Another class of status code was introduced, however, it was intended for future use.

* HTTP/1.0 only allowed one request to be serviced in a single connection.

**HTTP/1.1 (RFC2616)**

* HTTP/1.1 added request pipelining, but this only partially addressed request concurrency and still suffers from head-of-line blocking.
* Support for persistent connection - the connection is now long lived unlike from the previous versions which terminates the connection after the response.
* Support for pipelining - allows the clients to make multiple request without waiting for the response in the first request

**HTTP/2 (RFC7540)**

* Support for prioritization of request - this allows more important request to be serviced as a priority
* Multiplexing of requests - each request/response exchange have its own stream
* Server push - allows the server to send data to the client which it thinks the client need