|  |  |
| --- | --- |
| HTTP 1 | HTTP 2 |
| 1. The Hypertext Transfer Protocol (HTTP) is an application-level protocol with the lightness and speed necessary for distributed, collaborative, hypermedia information systems.  2. HTTP has been in use by the World-Wide Web global information initiative since 1990.  3.  This specification reflects common usage of the protocol referred to as "HTTP/1.0".  4.  This specification describes the features that seem to be consistently implemented in most HTTP/1.0 clients and servers.  5. The specification is split into two sections. Those features of HTTP for which implementations are usually consistent are described in the main body of this document.  6. Practical information systems require more functionality than simple retrieval, including search, front-end update, and annotation.  7.  HTTP allows an open-ended set of methods to be used to indicate the purpose of a request.  8. The HTTP protocol is based on a request/response paradigm. A client establishes a connection with a server and sends a request to the server in the form of a request method, URI, and protocol version, followed by a MIME-like message containing request modifiers, client information, and possible body content.  9. The server responds with a status line, including the message's protocol version and a success or error code, followed by a MIME-like message containing server information, entity metainformation, and possible body content.  10. HTTP only presumes a reliable transport; any protocol that provides such guarantees can be used, and the mapping of the HTTP/1.0 request and response structures onto the transport data units of the protocol in question is outside the scope of this specification. | 1. HTTP/2 will make our applications faster, simpler, and more robust — a rare combination — by allowing us to undo many of the HTTP/1.1 workarounds previously done within our applications and address these concerns within the transport layer itself.  2. The primary goals for HTTP/2 are to reduce latency by enabling full request and response multiplexing, minimize protocol overhead via efficient compression of HTTP header fields, and add support for request prioritization and server push.  3.  To implement these requirements, there is a large supporting cast of other protocol enhancements, such as new flow control, error handling, and upgrade mechanisms, but these are the most important features that every web developer should understand and leverage in their applications.  4. HTTP/2 does not modify the application semantics of HTTP in any way. All the core concepts, such as HTTP methods, status codes, URIs, and header fields, remain in place.  5.  Instead, HTTP/2 modifies how the data is formatted (framed) and transported between the client and server, both of which manage the entire process, and hides all the complexity from our applications within the new framing layer.  6.  HTTP/2 continued to coevolve in parallel, with SPDY acting as an experimental branch that was used to test new features and proposals for the HTTP/2 standard.  7. In early 2015 the IESG reviewed and approved the new HTTP/2 standard for publication. Shortly after that, the Google Chrome team announced their schedule to deprecate SPDY and NPN extension.  8. We plan to remove support for SPDY in early 2016, and to also remove support for the TLS extension named NPN in favor of ALPN in Chrome at the same time.  9. Server developers are strongly encouraged to move to HTTP/2 and ALPN.  10.  As a result, the HTTP/2 standard is one of the best and most extensively tested standards right out of the gate. |
|  |  |