Internet

-invented by Tim Berners-Lee

-jointly developed by W3C and the IETF

VERSION HISTORY

* HTTP 0.9 (1991)
* HTTP 1.0 (RFC 1945, May 1996)
* HTTP 1.1 (RFC 2068 Jan 1997, RFC 2616 Jun 1999), RFC 7230-7235 (Jun 2014)
* HTTP 2 (RFC 7540 May 2015)
* SPDY –Standard published by google
* HTTP runs on top of TCP/IP, using TCP port 80 by default, or TCP port 443 for HTTPS (HTTP over SSL/TLS)
* HTTP is based on client-server architecture
* HTTP uses a request-response standard protocol

-the client sends an HTTP request message to the server

- the server processes the request and replies with an HTTP response message

* HTTP is a stateless communications protocol
* Servers do not keep information about clients in between requests
* HTTP provides support for other functionalities like:
* Cache control
* Content media type (MIME) specification
* Language and character set specification
* Content/ transfer codings
* Content negotiation
* Client-server protocol negotiations
* Persistent connections
* Request pipelining
* Authentication/ authorization
* HTTP Resource Addressing
* HTTP resources are identified using URI (RFC 3986) or more specifically, HTTP URL’s
* URI –uniform resource identifier
* URN –uniform resource name
* URL- uniform resource locator
* Scheme(http or https)
* Authority
* user information or authentication credentials
* host- domain name (resolved to an IP address using DNS) of the server where the resource resides
* port number
* Path to resource (resolved to relative to the document root on the server)
* may refer to a state or dynamic resource
* Query (?)
* Typically provided as key = value pairs, with ampersand (&) separators between key/value pairs
* May be URL-encoded
* Fragment Identifier (#)
* Status Line
* HTTP protocol Version
* Status Code
* Reason Phrase
* HTTP Request Methods
* Request line (CRLF – terminated line consisting of three spaces separated values)

Standard Methods

* GET – transfer a current selected representation of the resource identified by the request URI, the retrieved resource is returned in the message body of the response as an entity
* HEAD – used to retrieved metadata about the entity implied by the request without transferring the entity itself (e.g to test for link validity or resource modification)

-like get, must be supported by general- purpose servers

* POST – perform resource- specific processing of the entities enclosed in the message body
* PUT – store the enclosed entity in the message body under the specified request URI
* DELETE – remove the resource
* Extension Methods

-e.g WebDAV (RFC 4918)

* PROFIND, PROPPATCH, MKCOL, COPY, MOVE, LOCK, UNLOCK
* Safe Methods, Idempotent methods and Cacheable methods

OPTIONS

* Communication options are available

TRACE

* Diagnostic purposes / testing of the request / response chain
* Request a loop-back to the client the received request message

CONNECT

* Request the establishment of a tunnel to the destination origin server
* Commonly used to connect from end-to-end
* HTTP Message Headers
* General Header Fields – client and servers
* Cache-control
* Connection
* Date
* Pragma –generic header used in older versions
* Trailer – headers at the end of the entity
* Transfer-coding
* Upgrade – converting to different versions
* Via- forward a message
* Warning
* Request Header Fields
* Negotiation requests
* Accept
* Accept-charset
* Accept-encoding
* Accept-language
* Authorization
* Proxy authentication
* Expect
* Informational
* Host
* From
* If-match
* If-modified-since
* If-none-match
* If-range
* If-unmodified-since
* Max-forwards
* Range
* Referer
* TE
* User agent
* Response Header Fields
* Accept Ranges
* Age
* ETag
* Location
* Proxy authenticate
* Retry after
* Server
* Vary
* WWW authenticate
* Entity Header Fields
* Allow
* Content-Encoding –compression
* Content-language –control how to read the content
* Content-length –how long is the payload in bytes
* Content-location
* Content MDS (Message Digest ver 500)[deprecated]
* Content-range -request content
* Content-type- image, text..etc.
* Expires –duration of the client keeping the content
* Last-Modified
* HTTP STATUS CODES
* Informational (1xx)
* 100 continue
* 101 switching protocol
* Success (2xx)
* 200 OK
* 201 Created
* 202 Accepted
* 203 Non-Authoritative- Information
* 204 No Content
* 205 Reset Content
* 206 Partial Content
* Redirection (3xx)
* 300 Multiple Choices
* 301 Moved Permanently
* 302 Found
* 303 See Other
* 304 Not Modified
* 305 Use Proxy
* 306 (unused)
* 307 Temporary Redirect
* Client Errors (4xx)
* 400 Bad Request
* 401 Unauthorized
* 402 Payment Required
* 403 Forbidden
* 404 Not Found
* 405 Method Not Allowed
* 406 Not Acceptable
* 407 Proxy Authentication
* 408 Request Time-out
* 409 Conflict
* 410 Gone
* 411 Length Required
* 412 Precondition Failed
* 413 Request Entity Too Large
* 414 Request URI Too Large
* 415 Unsupported Media Type
* 416 Requested Range Not Satisfiable
* 417 Expectation Failed
* 426 Upgrade Required
* Server Error (5xx)
* 500 Internal Server Error
* 501 Not Implemented
* 502 Bad Gateway
* 503 Service Unavailable
* 504 Gateway time-out
* 505 HTTP Version Not Supported