Hypertext Transfer Protocol

* Application layer communications protocol used to access resources (hypertext/hypermedia) on the World Wide Web
* Invented by Tim Berners – lee at CERN in 1989
* Jointly developed by the W3C(World Wide Web Consortium) and the IETF (Internet Engineering Task Force)
* Version history:
  + HTTP 0.9 (1991)
  + HTTP 1.0 (RFC 1945 may 1996)
  + HTTP 1.1 (RFC 2068 Jan 1997 June 1999), RFC 7230 – 7235
  + HTTP 2 (RFC 7540 May 2015)

Pipelining – to make a request and wait for the response to come in before requesting again

HTTP FUNDAMENTALS

* HTTP typically runs on the 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
  + Servers:
    - Origin servers
    - Proxy servers, gateways, tunnels
  + Clients, a.k.a. user agents (UA):
    - Web browsers, web crawlers/spiders, other end user tools and applications
* HTTP encrypts
* 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
* HTTP provides support for other functionalities such as:
  + Cache
  + Content media type (MIME) specification
  + Language and character set specification

HTTP REQUEST METHOD

* STANDARD METHOD
  + GET – transfer a current selected
  + HEAD - identical to get, accept that the server does not send a message
  + POST – request that the target resource perform resource – specific processing of the presentation enclosed in the message body of the request
    - Query – typically provide as key = value pairs, with ampersand (&)separators between key/value pairs
    - Path – to resource (resolved relative to the document root on the server)
    - Fragment identifier
  + PUT – create or replace the state of the target source with the state defined by the presentation enclosed in the request message payload.
  + DELETE – remove the association between the target resources and its current functionality
  + OPTIONS – allows to query about the server on what to do , Request information about the communication, Target may be \*
  + TRACE – request a remote, application – level loop –back of the request message (ie; request the recipient to echo back to the client the request message). Typically used for testing/diagnostics of the request/response chain
  + CONNECT – requests the establishment of a tunnel to the destination origin server, and if successful, there after restrict its behavior to blind – forwarding of packets, in both directions, until tunnel is closed.
* METHOD PROPERTIES
  + Safe Method – client does not request, and does not expect, any state change on the origin server as a result of applying the method to a target source. GET,HEAD, OPTIONS, TRACE.
  + Idempotent method – intended effect on the server of multiple identical requests with the methods is the same as the effect of a single such request. GET, HEAD, OPTIONS, TRACE, PUT DELETE
  + Cacheable method – indicates that the response to a method is allowed to be stored for future use. GET, HEAD, POST

HTTP MESSAGE HEADERS

* General Header Fields – used by client and servers
* Request Header Fields – generated by clients
* Response Header Fields- only response messages
* Entity Header Fields - refers to entity

HTTP STATUS CODES

* Information (1xx)
* Success (2xx)
* Redirection (3xx)
* Client Error (4xx)
* Server Error (5xx)