HTTP

* Use to access resources on the World Wide Web.
* Invented by Sir Tim Berners-Lee.
* Jointly developed by W3C and IETF.

HTTP 0.9 (1991) – Get

HTTP 1.0 (RFC 1945, may 1996)

HTTP 1.1 (RFC 2068, Jan 1997)

HTTP 1.1 (RFC 2616, Jun 1999)

HTTP 1.1 (RFC 7540, Jun 2014)

HTTP 2 (RFC 7540, May 2015)

* Patterned after SPDY.
* 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 a client- server architecture
* Clients, a.k.a user agents (UA)
* Web browsers, web crawlers/spiders, ad other end user tools and applications
* NOTE: port number – unique number assigned by operating system (0-64K)
* IP address + port number = socket
* HTTPS – encrypted communication
* Servers – origin servers

-proxy servers, gateways, tunnels

* HTTP uses a request- response standard protocols.
* The client sends an HTTP request message to the servers.
* The server processes the request and replies with an HTTP response message.
* HTTP is a stateless communications protocols

Server do not keep information about clients in between requests. “SERVE AND FORGET”

* HTTP2 “PUSH”
* HTTP provides support for other functionalities, such as:
* Cache control
* Content media type (MIME) specification
* Language and character set specification
* Content/transfer codings
* Content negotiation
* Client server protocol negotiations

MIME – multipurpose Internet Mail Extensions.

HTTP Resource Addressing

* Routing the message
* HTTP resources are identified using URLs (RFC 3986) or more specifically, HTTP URLs

URN – naming without regard to where it is (Ex. ISBN)

URL – how to access (mechanism of finding)

* Scheme (HTTP or HTTPS) – protocol
* Authority

-user info or authentication credentials (deprecated)

-host (host name – where the resources)

-domain name (resolved to an IP address using DNS) of the server where the resource reside (or will be created)

-port number (default- 80)

* Path to resource (resolved relative to the document root on the server)
* May refer to a static or dynamic resource (static – html itself; dynamic – scripts)
* Query (starts with ? ) (Ex. Products.php?

HTTP request message

* Request line (CRLF terminated line connecting three spaces separated values)
* Method
* Request URI
* HTTP protocol version
* Message Headers (general, request and or entity headers)
* HTTP 1.1 requires at least the host request headers to be provided (1 header required)(Ex. Fieldname: value)

1. General headers fields – can be used by clients and server.
2. Request header fields – used by clients.
3. Response header fields – server
4. Entity header fields

* Empty line (CRLF)
* Message body a.k.a payload (optional) (message body – entity)

HTTP response message

* Status line (CRLF – line terminated line consisting of three space-separated values)
* HTTP protocols
* Status code
* Reason phase
* Information (1xx)
* Success (2xx)
* Redirection (3xx)
* Client error (4xx)
* Server error (5xx)
* Reason phase – consumption for human beings (Ex. Success 2xx – 200 OK)
* Message headers (general, response, and /or entity headers)
* Entity line (CRLF)
* Message body (optional) – have body more than request message.

HTTP Request methods (HTTP verbs)

* Standard methods
* Get
* Head
* Post
* Put
* Options
* Delete
* Trace
* Connect