Jan. 19, 2018

WWW

* Collection of web resources and applications in which access the Internet
* Information system
* 1989-1990
* 30 year old technology
* Established by Sir Tim Berners-Lee
* Purely text before
* Hypertext into hypermedia
* Client-server type architecture

Internet

* Global network of networks (interconnection)
* Hosts, media, protocols – can serve as a platform to build applications

URL

* Way to address a specific resource, website, webpage, etc.

DNS

* Change the IP address into a specific hostname/domain name

Web debugging

* Intercepts the process/procedure/request within the browser

Differences of HTTP 1.0, HTTP1.1, HTTP 2.0

HTTP 1.0 (RFC 1945)

* November 1996
* Does not require a host header, but can be added
* Persistent connections – need to open a new connection for each request/response pair
* Does not understand the 100 (Continue) response
* Compression – includes the Content-Encoding header, which indicates end-to-end content-coding used for message
* Can define 16 status codes
* Authentication is unsafe as it is not encrypted

HTTP 1.1 (RFC 2616/RFC 7230-7235)

* Early 1997
* Has a required host header by specification
* Persistent connections – allows to have persistent connections which means that you can have more than one request/response on the same HTTP connection
* Introduces the OPTIONS method – the HTTP client can use this method to determine the abilities of the HTTPS server
* Caching – expands on the caching support a lot by using “entity tag”, and also adds the If-Unmodified-Since, If-Match, If-None-Match conditional headers
* Includes the 100 (Continue), to inform the client that the request should be transmitted
* Compression – carefully specifies the Accept-Encoding header, used by a client to indicate what content-coding it can handle and which one it prefers
* Generally an upgrade of the limitations of HTTP 1.0
* Can define 24 status codes
* Safe as it uses a checksum of username, password and one time value

HTTP 2.0 (RFC 7540)

* Early 2010
* Allows multiplexing – allowing multiple requests at the same time, and therefore speeding up the process of delivering web pages and content
* Binary instead of textual – binary protocols are more efficient to parse, more compact “on the wire”, and most importantly they are much less error-prone compared to textual protocols
* Server push – allows the server to avoid delays of request-response by “pushing” the responses it thinks the client will need into its cache

Jan. 23, 2018

HTTP

* Hypertext Transfer Protocol
* Invented by Sir Tim Berners-Lee at CERN (1989)
* Protocol to access resources (hypertext/hypermedia) on the WWW
* Jointly developed by the W3C (World Wide Web Consortium) and the IETF (Internet Engineering Task Force)

Version History

HTTP 0.9

* 1991
* Only GET method
* Single connection

HTTP 1.0

* Still single connection
* Can access other character sets

HTTP 1.1

* Current mainstream that is in used
* OPTIONS, GET, HEAD, POST, PUT, DELETE, TRACE, CONNECT
* Can have persistent connection

HTTP 2

* More storage
* Better bandwidth
* Concurrency and multiplexing
* Server push – anticipate what the client needs
* Header compression

HTTP Fundamentals

* HTTP typically runs on top of TCP/IP (Transmission Control Protocol/Internet Protocol)
* HTTPS is based on a client-server architecture
* Servers: origin servers, proxy servers, gateways, tunnels
* Clients: web browsers, web crawlers/spiders
* HTTP is a stateless communications protocol – it does not get any information

Jan. 30, 2018

HTTP support for other functionalities

* Cache control
* Content media type (MIME) specification
* Language and character set specification
* Content/transfer coding
* Content negotiation – can get other formats preferred by the client
* Client-server protocol negotiations
* Persistent connections – starts at HTTP 1.1
* Request pipelining/multiplexing – simultaneously (requests)
* Authentication/authorization

HTTP Resource Addressing

* HTTP resources are identified using URIs
* Scheme – http or https
* Authority – host (domain name), port number (80 default port number), user information or authentication credentials (deprecated)
* Path to resource – resolved relative to the document root on the server (can be static or dynamic resource)
* Query – typically provided as key=value pairs, with ampersand (&) separators between key/value (starts with (?) and may be URL-encoded)
* Fragment identifier – subsection in the document (starts with (#))
* Ex. http (scheme) :// usr:pwd@server.org:81 (authority) /info/profile.php (path to resource) ?id=1234 (query) #addr (fragment identifier)

HTTP Request Message

* Request line – method, request target, protocol version
* Message headers
* Empty line (CRLF)
* Message body/payload (optional)

HTTP Response Message

* Response line – protocol version, status code
* Message headers
* Empty line (CRLF)
* Message boy/payload (optional)