**Contents: HTML desc, version, components**

**HTTP , functionalities, version, Request Method Properties**

**Response,REQUEST LINE**

**WHAT IS HTML?**

(<https://www.w3.org/standards/webdesign/htmlcss>)

-abbreviation for Hyper Text Markup Language.

-First developed by Tim-Berners Lee in 1990.

-language for describing the structure of a webpage

-has evolved through various versions

**HTML Versions**

\*HTML

-First release of HTML. It was used to put up simple Web pages.

\*HTML 2.0(1995)

-HTML was standardized by IETF (Internet Engineering Task Force)

-Had most of the elements being used today but was missing some of the Netscape/Microsoft extensions, and did not support tables, or ALIGN attributes.

\*HTML 3.2(1997)

-Integrate support for TABLES, image, heading and other element ALIGN attributes.

-It was missing some of the Netscape/Microsoft extensions, such as FRAMEs, EMBED

and APPLET.

\*HTML 4.01(1999)

-Includes support for extra features(Internationalized documents, support for Cascading Style Sheets, extra TABLE, FORM, and JavaScript enhancements).

\*XHTML(2000)

-Extensible Markup Language

-Was conceived as a means of regaining the power and flexibility of SGML(Standard Generalized Markup Language) without most of its complexity. It removed many complex features of SGML that make the authoring and design of suitable software both difficult and costly.

\*HTML5(2004)

-WHATWG(Web Hypertext Application Technology Working Group) began working on a new version of HTML.

-Has new features:

=Reducing the overlap between HTML, CSS, and JavaScript

=Supporting rich media experiences while eliminating the need fir plugins such as Flash or Java

**HTML COMPONENTS** (https://www.w3resource.com/html/components-of-an-HTML-page.php)

\*Doctype

-indicates HTML document version

\*HTML Elements and Tags

-Text, image, video and other content attributes is contained in a HTML Elements.

\*HTML Comments

-Enables excluding some part of the code.

\*HTML Frames

-Enables loading of HTML pages with another HTML page in same browser window.

**HTTP** (https://developer.mozilla.org/en-US/docs/Web/HTTP)

-is an abbreviation of HyperText Transfer Protocol. It is an application-layer

protocol used for transmitting hypermedia documents like html. It is also used by

the world wide web. It defines the procedures and rules that the web servers follow.

**SOME FUNCTIONALITIES OF HTTP**

(<https://developer.mozilla.org/en-US/docs/Web/HTTP/Content_negotiation)>

\*Cache Control

-Used to specify directives for caching mechanisms in both request and responses.

-Used to define caching policies.

\*Language And Character Set Specification

\*Content/Transfer Coding

\*Content Negotiation

-Mechanism that is used for serving different representations of a resource at the same URI, so that the user agent can specify which is best suited for the user

\*Client-Server Protocol Negotiation

\*Persistent Connection

\*Request Pipelines-

\*Authentication/Authorization

**HTTP VERSIONS**

HTTP/ 0.9

Initial version of HTTP. A very simple with a single-line request and with a GET method only followed by the path of the resource.

HTTP/ 1.0

In HTTP/1.0 has more methods: GET, HEAD and POST.

It includes header fields:

* Allow
* Authorization
* Content-Encoding
* Content-Length
* Content-Type
* Date
* Expires
* From
* If-Modified-Since
* Last-Modified
* Location
* Pragma
* Referer
* Server
* User-Agent
* WWW-Authenticate

It has a status code line in top of the response.

1XX -HTTP/ 1.0 does not define this status code and not valid response to a HTTP/ 1.0 request.

2XX -this indicates if the client’s request was successfully received, understood and accepted.

200 OK

201 Created

202 Accepted

204 No content

3XX -this status code indicates that any action needs to be taken by user agent in order to perform the request.

300 Multiple Choices

301 Moved Permanently

302 Moved Temporarily

304 Not Modified

4XX -this is intended in case of the client having an error.

400 Bad Request

401 Unauthorized

403 Forbidden

404 Not Found

5XX-indicate cases in which the server is incapable of performing the request.

500 Internal Server Error

501 Not Implemented

502 Bad Gateway

503 Service Unavailable

And It can define 16 status code which was reserved number.

HTTP/1.1

This is a standardized protocol which was published in early 1997. It introduced many improvements from the past version. For the methods, there were many added:

* OPTIONS
* GET
* HEAD
* POST
* PUT
* PATCH
* COPY
* MOVE
* DELETE
* LINK
* UNLINK
* TRACE
* WRAPPED

There are some Status Codes that were maintained from the previous version and there are additional:

Informational 1xx

100 Continue

101 Switching Protocols

Successful 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

307 Temporary Redirect

Client Error 4xx

400 Bad Request

401 Unauthorized

402 Payment Required-reserved for future use.

403 Forbidden

404 Not Found

405 Method Not Allowed

406 Not Acceptable

407 Proxy Authentication Required

408 Request Timeout

409 Conflict

410 Gone

411 Length Required

412 Precondition Failed

413 Request Entity Too Large

414 Request-URI Too Long

415 Unsupported Media Type

416 Requested Range Not Satisfiable

417 Expectation Failed

Server Error 5xx

500 Internal Server Error

501 Not Implemented

502 Bad Gateway

503 Service Unavailable

504 Gateway Timeout

505 HTTP Version Not Supported

It has 24 status codes. For the header field, it has Field extensibility, field order, whitespace, field Parsing, field limits and field value components- common syntax component (tokens, quoted string, and comment).

Header fields:

* Accept
* Accept-Charset
* Accept-Encoding
* Accept-language
* Accept-Ranges
* Age
* Allow
* Authorization
* Cache-control
* Connection
* Content-Encoding
* Content-Language
* Content-Location
* Content-MD5
* Content-Range
* Content-Type
* Date
* ETag
* Expect
* Expires
* From
* Host
* If-Match
* If-Modified-Since
* If-None-Match
* If-Range
* If-Unmodified-Since
* Last-Modified
* Location
* Max-Forwards
* Pragma
* Proxy-Authenticate
* Proxy-Authorization
* Range
* Referer
* Retry-After
* Server
* TE
* Trailer
* Transfer-Encoding
* Upgrade
* User-Agent
* Vary
* Via
* Warning
* WWW-Authenticate

Pipelining has been included for lowering the latency of communication. Also has Transfer coding contains Chunked transfer codings, compression codings, TE and Trailers. Additional cache control mechanism introduced. Content negotiation added on this version and Caching.

HTTP/2.0

The latest version of HTTP which its specification was published in May 2015. By the Internet Engineering Task Force (IETF) revised the HTTP/1.1, it was developed the next generation of the application protocol, HTTP/2.0. This HTTP version is a binary protocol which is more efficient to parse, they are less error-prone. It also introduced multiplexing, via a single TCP connection, multiple HTTP request can be sent and responses will be received asynchronously. HTTP/ 2.0 includes frames and streams, error codes -for the stream and connection error, and header compression.

**HTTP Request Method Properties**

Safe Methods

* Are considered safe when their request semantics are read-only. This means that there should be no change of state to the server.
* The methods considered safe are the GET, HEAD, TRACE, and OPTIONS methods.

Idempotent Methods

* If multiple identical requests are sent to the server, the effect will be the same as the first issuance of such request.
* The methods considered idempotent are the following:
  + PUT, DELETE, and the safe request methods.

Cacheable Methods

* As the term implies, request methods are termed to be cacheable if server responses to those requests are allowed to be stored for reuse.(LINK RFC 7231)
* Only the GET, HEAD, and POST methods are cacheable.

**HTTP Response**

Status Line

* Consists of the protocol version, status code and its corresponding Reason-Phrase
  + <Protocol>/<Version> <Space> <Status Code> <Space> <Reason Phrase>

Status Code and Reason Phrase

* + The status code is a 3-digit integer result code of the attempt to understand and satisfy the request (link RFC 2616).
    - Fist-digit = defines the class of response
      * 1xx: Informational
      * 2xx: Success
      * 3xx: Redirection
      * 4xx: Client error
      * 5xx: Server error
  + The reason phrase is a short textual description of the associated status code.

**HTTP REQUEST LINE** (http://toolsqa.com/client-server/http-request/)

-Specifies the method token(GET, PUT…) followed by the request URI then the HTTP Protocol that is being used