HTTP FUNDAMENTALS

-HTTP is an abbreviation of HyperText Transfer Protocol. It is an application-layer protocol used for transmitting hypermedia documents like HTML. It runs in a client-server computing model where the client requests for a specific web content and the server provides it. Moreover, it defines the procedures and rules that the client and server follows for transmitting information resources.

**HTTP VERSIONS**

HTTP/ 0.9

The initial version of the HTTP. Client requests consist only of a single request line (<Request Method> <path of the document>). It supports only the GET method for retrieving web resources. The web resources supported in this version are HTML files only. Lastly the TCP/IP connection between the server and the client is terminated immediately after the document has been transmitted.

HTTP/ 1.0

Added support to header fields for both request and response headers (e.g. HTTP version number, Status Code, Content-type) which allows the transmission of not only HTML files but also of scripts, stylesheets, and Multimedia content. This version of HTTP also allowed support for the HEAD and POST methods.

HTTP/1.1

This standardized protocol was published in early 1997. It introduced many improvements from the past version. Persistent connections (Keep-Alive), pipelining, content negotiation, etc. There were also new methods supported in this version, namely, PUT, DELETE, TRACE, and OPTIONS.

HTTP/2.0

The latest version of HTTP. The specifications for this version was published in May 2015 by the IETF. The primary goals for this version are to minimize protocol overhead, reduce latency by multiplexing, and add support for server push and request prioritization.

**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.
    - Status Code Overview
      * 100: Continue
      * 101: Switching Protocols
      * 200: OK
      * 201: Created
      * 202: Accepted
      * 203: Non-Authorative Information
      * 204: No Content
      * 205: Reset Content
      * 206: Partial Content
      * 300: Multiple Choices
      * 301: Moved Permanently
      * 302: Found
      * 303: See Other
      * 304: Not Modified
      * 305: Use Proxy
      * 307: Temporary Redirect
      * 400: Bad Request
      * 401: Unauthorized
      * 402: Payment Required
      * 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: Payload Too Large
      * 414:URI Too Long
      * 415: Unsupported Media Type
      * 416: Range Not Satisfiable
      * 417: Expectation Failed
      * 426: Upgrade Required
      * 500: Internal Server Error
      * 501: Not Implemented
      * 502: Bad Gateway
      * 503: Service Unavailable
      * 504: Gateway Timeout
      * 505: HTTP Version Not Supported