**REQUEST**

**Request Message:**

**CRLF (Carriage Return Line Feed) -** contains three values which are separated by spaces.

**Request CRLF**

1. Method
2. Request Target
3. Protocol Version

**Process**:

* + - 1. Message Body
      2. Empty Line
      3. Payload

\*HTTP 1.0 has at least one required header which is request header

\*Message headers: General, Request, Entity Header

\*Payload - found after the header, blank line means empty payload

**Request Methods**

1. GET – retrieves, transfers a selected resource into the message body aka entity. Commonly used method (RFC 7231, Sec. 4.3.1) (RFC 2616, Sec. 9.3).
2. HEAD – identical to get but doesn’t get the resource. It has no payload, only headers and metadata (RFC 7231, Sec. 4.3.2) (RFC 2616, Sec. 9.4).
3. POST – non-safe and non-idempotent method. It is utilized for processing resources from the payload given by a request (RFC 7231, Sec. 4.3.3) (RFC 2616, Sec. 9.5).

* Typically used in HTML form data.
* Any data that is being submitted is already in the payload aka message body and the information will not be in the URL.

\*Search-friendly URL – doesn’t contain any query.

1. PUT – utilized to modify all existing models of the resource being targeted by basing it from the payload contained by the request (RFC 7231, Sec. 4.3.4) (RFC 2616, Sec. 9.6).

* Web servers, by default, do not allow put request since it may affect the information.

1. DELETE – logically removes the relationship existing between the resource being targeted and its existing functions. It does not necessarily delete the true resource hence it may be modified by a human intervening from the server which has the true copy (RFC 2616, Sec. 9.7) (RFC 7231, Sec 4.3.5).
2. OPTIONS – allows you to query a particular resource on the server itself. Its response is usually other request methods. It is another way of ensuring the integrity of a resource.

\*A request may be noted as an asterisk (\*) wherein it represents a global or general resource (RFC 7231, Sec. 4.3.7) (RFC 2616, Sec. 9.2).

1. TRACE – echoes back to the user or client what is happening to the request or received request message. It can manipulate the message (RFC 7231, Sec. 4.3.8) (RFC 2616, Sec. 9.8).

* Typically used for testing, diagnosing the request, response chain and troubleshooting.

\*Response chain – set of nodes from client to server.

1. CONNECT – set ups the connection to the requested server of the resource being targeted (RFC 7231, Sec. 4.3.6) (RFC 2616, Sec. 9.9).

\*Link Rot – linking to different pages but these pages aren’t controlled by you so some links may not exist anymore and if not updated, the website will degrade due to dead links. Solution: Update.

**Method Properties:**

1. Safe Methods – used only for read-only purposes. It can’t modify the resource but only retrieve it (RFC 7231, Sec 4.2.1).

* GET, HEAD, OPTIONS, TRACE

1. Idempotent Methods – it refers to the semantics of the method itself (RFC 7231, Sec 4.2.2).

* GET, HEAD, OPTIONS, TRACE, PUT, DELETE

1. Cacheable Methods – specifies that a certain resource given or sent by the server may be stored and retrieved for future purposes (RFC 7231, Sec 4.2.3).

* GET, HEAD, POST

**RESPONSE**

**Response Message:**

**Response CRLF:**

1. Protocol Version
2. Status Code
3. Reason Phrase

Example: HTTP/1.1 200 OK

**Process**:

1. Message Body
2. Empty Line
3. Payload

**Message Headers**

Categories:

1. General Header Fields – these are fields which can be utilized by both the client and server (RFC 2616, Sec. 4.5).

* Cache-Control – specifies the different rules and guidelines that must be observed and upheld by each and every cache-related operation in the request or response chain (RFC 2616, Sec. 14.9).
* Trailer – gives indication if a certain field in the header is made available in the message’s trailer which contains transfer-coding (RFC 2616, Sec. 14.40).
* Connection – gives the sender options for persistent connection (RFC 2616, Sec. 14.10).
* Upgrade – gives the client options on additional functional features of the communication protocols (RFC 2616, Sec. 14.42).
* Date – specifies what date and time the message was sent or received (RFC 2616, Sec. 14.18).
* Via – utilized by both gateways and proxies to specify certain protocols which will serve as bridging the user agent and server as well as between the server and client with regards to requests and responses, respectively (RFC 2616, Sec. 14.45).
* Pragma – identifies rules and guidelines that may correspond to the receiver present in the response or request chain. It originated from HTTP 1.0 and it also represents any new functionality not yet standardized i.e. not yet listed to any RFC’s (RFC 2616, Sec. 14.32).
* Warning – identifies supplementary information with regards to the state of a message sent or received (RFC 2616, Sec. 14.46).
* Transfer-encoding – specifies if the message body has been edited or transformed. This is important so that the message can be safely sent (RFC 2616, Sec. 14.41).

1. Request Header Fields - these are header fields which can only be seen in client-generated request messages (RFC 2616, Sec. 5.3).

* Accept – indicates what types of media are allowed to be made available for the response (RFC 2616, Sec. 14.1).
* Accept-Charset - indicates what types of set of characters are allowed to be made available for the response (RFC 2616, Sec. 14.2).
* Accept-Encoding – analogous with the method Accept with the exception that it limits the coding of the content that are made available to the response (RFC 2616, Sec. 14.3).
* Accept-Language – limits the preferred natural languages (RFC 2616, Sec. 14.4).
* Authorization – contains the credentials of the user-agent (RFC 2616, Sec. 14.8).
* Expect – expectations of the server which are indicated by the client (RFC 2616, Sec. 14.20).
* From – contains the email address of the user (RFC 2616, Sec. 14.22).
* Host – indicates the Internet host as well as the resource’s number of the port that is being queried (RFC 2616, Sec. 14.23).
* If-Match – used to compare resources through their entity tags. If it matches, the response is a GET request (RFC 2616, Sec. 14.24).
* If-Modified-Since – used to compare resources especially if the resource may have been updated and if not, the server will not receive any representation of the resource (RFC 2616, Sec. 14.25).
* If-None-Match – opposite of the IF-Match header (RFC 2616, Sec. 14.26).
* If-Range – sends parts of the entity if it is not changed, else, it sends the whole entity (RFC 2616, Sec. 14.27).
* If-Unmodified-Since – another way to compare resources and works as If-Modified-Since header’s polar opposite (RFC 2616, Sec. 14.28).
* Max-Forwards – delimits the number of either gateways or proxies that are able to send request by making use of the TRACE and OPTIONS methods (RFC 2616, Sec. 14.31).
* Proxy-Authorization – authorizes the client with the use of proxy which will require the client to authenticate itself (RFC 2616, Sec. 14.34).
* Range – size (RFC 2616, Sec. 14.35).
* Referer – not a typo in this circumstance. It contains the URI address of the resource (RFC 2616, Sec. 14.36).
* TE – lists the accepted transfer-encoding’s extension (RFC 2616, Sec. 14.39).
* User-Agent – lists the user-agent’s information (RFC 2616, Sec. 14.43).

1. Response Header Fields – these are header fields which can only be seen on the server’s side. This requires authentication (RFC 2616, Sec. 6.2).

* Accept-Ranges – lists the acceptable range of requests of resource as indicated by the server (RFC 2616, Sec. 14.5).
* Age – estimated time duration it took for the server to process and create the response and send it to the receiver (RFC 2616, Sec. 14.6).
* ETag – means entity tag which is used to identify which entity a certain resource belongs to (RFC 2616, Sec. 14.19).
* Location – utilized for diverting to another address and not the address which was requested (RFC 2616, Sec. 14.30).
* Proxy-Authenticate – challenges the credentials contained in the Proxy-Authentication (RFC 2616, Sec. 14.33).
* Retry-After – may be utilized for the response 503, Service Unavailable, to let the client know the amount of time the service is unavailable (RFC 2616, Sec. 14.37).
* Server – lists the information with regards to the server (RFC 2616, Sec. 14.38).
* Vary – specifies available header fields which correspond to requests (RFC 2616, Sec. 14.44).
* WWW-Authenticate – a definite inclusion in response 401, Unauthorized, messages. It challenges the credentials with regards to the URI requested (RFC 2616, Sec. 14.47).

1. Entity Header Fields – gives description to the entity which is in the payload (RFC 2616, Sec. 7.1).

* Allow – indicates to the receiver the valid methods applicable to a resource (RFC 2616, Sec. 14.7).
* Content-Encoding – gives information on the type of media of the entity-body (RFC 2616, Sec. 14.11).
* Content-Language – the natural language of which the entity is presented as (RFC 2616, Sec. 14.12).
* Content-Length – corresponds to the entity’s size according to its body or length of body (RFC 2616, Sec. 14.13).
* Content-Location – where the resource is located (URI) (RFC 2616, Sec. 14.14).
* Content-MD5 – used for message integrity check (MIC) contained in the entity-body (RFC 2616, Sec. 14.15).
* Content-Range – specifies the entity-body’s total length (RFC 2616, Sec. 14.16).
* Content-Type – corresponds to what kind of media the body of the entity is (RFC 2616, Sec. 14.17).
* Expires – corresponds to the date and time the response will expire (RFC 2616, Sec. 14.21).
* Last-Modified – date and time the entity was last changed (RFC 2616, Sec. 14.29).
* Extension-header – this is the message header (RFC 2616, Sec. 7.1).
* For a Full list go to iana.org <a href=http://www.iana.org/assignments/message-headers/message-headers.xhtml</a>

References

RFC2616 Hypertext Transfer Protocol -- HTTP/1.1. R. Fielding, J. Gettys, J.

Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee. June 1999.

(Format: TXT=422317, PS=5529857, PDF=550558 bytes) (Obsoletes

RFC2068) (Obsoleted by RFC7230, RFC7231, RFC7232, RFC7233, RFC7234,

RFC7235) (Updated by RFC2817, RFC5785, RFC6266, RFC6585) (Status:

DRAFT STANDARD) (DOI: 10.17487/RFC2616)

RFC7231 Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content. R.

Fielding, Ed., J. Reschke, Ed.. June 2014. (Format: TXT=235053

bytes) (Obsoletes RFC2616) (Updates RFC2817) (Status: PROPOSED

STANDARD) (DOI: 10.17487/RFC7231)