**HTTP HEADRS** – offers information about any request or response that were made or about any information sent over the message body

4 Types of HTTP Headers:

*General-Header:* Can be used for both request and response messages

*Client Request-Header:* Can be only used for request messages

*Server Response-Header:* Can be only used for response messages

*Entity-Header*: Offers information about the meta information found in the entity-body

If there is no body present, it will offer information about the resource being

requested

**1. GENERAL HEADERS**

*Cache-Control*

Used to specify the instruction that needs to be followed by the caching system.

*Connection*

Used by the sender to specify the options for the connection being made.

*Date*

Shows the HTTP date/time stamps in Greenwich Mean Time (GMT)

*Pragma*

*Trailer*

*Transfer-Encoding*

It shows the information about what type the message body has been transformed to in order to be able to safely transfer from the sender to the recipient.

*Upgrade*

Used by the client to specify what other communication protocol it can support and would be change to if server finds the need to switch the protocols.

*Via*

Used by the gateways and proxies to know the information about the protocols and recipients

*Warning*

Shows any other information about the status of the message which may not have been included in the messages

**2. CLIENT REQUEST HEADERS**

*Accept*

Shows the acceptable media types for the response.

*Accept-Charset*

Shows the acceptable character set for the response.

*Accept-Encoding*

Shows the acceptable content-coding for the response.

*Accept-Language*

Shows the preferred natural languages for the response

*Authorization*

Contains the authentication information of the user agent that is trying to access a resource

*Cookie*

Contains information stored for the URL

*Expect*

Shows the particular response that Is being expected by the client

*From*

Shows the e-mail address of the human user who made the request

*Host*

Shows the host and port number of the resource that is being requested

*If-Match*

It will ask the server to perform a certain method that was requested when the given value matches an entity in ETag

*If-Modified-Since*

The entity from a requested URL will not be returned if nothing has been modified since the time that was specified in the field. Instead, the “304 Not Modified” status code will be returned.

*If-None-Match*

It will ask the server to perform a certain method that was requested when the given value does not match any entity in Etag.

*If-Range*

Used to ask for a particular portion of a certain entity that has been missing

*If-Unmodified-Since*

It will ask the server to perform a certain method that was requested when the requested resource was not modified since the time specified in the field

*Max-Forward*

Will limit number of proxies or gateways that the request can be forwarded to

*Proxy-Authorization*

When a server requires an authentication, this allows the client to be able to identify itself or the user

*Range*

Shows information in what the partial range of the content was specified by a request

*Referer*

This allows a user or client to specify the URI of a certain resource

*TE*

It will tell what extension transfer-coding it will be accepting in the response and also whether or not it will be accepting any trailer fields that is in a chunked transfer-coding

*User-Agent*

Shows the information of the user who made the request

**3. SERVER RESPONSE HEADERS**

*Accept-Ranges*

Shows the acceptance of range requests for a certain resource by a server

*Age*

Shows the estimated amount of time since the generation of the response at the origin server

*ETag*

Shows the value of the entity tag for the variant that was requested

*Location*

This is used to redirect the recipient to another location that is not the Request-URI

*Proxy-Authenticate*

Included in the “407 Proxy Authentication Required” response

*Retry-After*

Together with the “503 Service Unavailable” response, it shows how long the service will be not available to the client that made the request

*Server*

Shows the information about the software that was used by the origin server that handled the request

*Set-Cookie*

Shows information (name/value) to retain for the URL

*Vary*

Shows that the entity has available multiple sources and can vary depending on the request headers that was specified

*WWW-Authenticate*

Together with the “401 Unauthorized” response, it shows at least one challenge in which it shows what parameters and authentication schemes are applicable to the Request-URI

**4. ENTITY HEADERS**

*Allow*

Show a list of the methods that are supported by resource that is being identified by the Request-URI

*Content-Encoding*

Used to show what the characteristic of the entity that is being identified by the Request-URI is or also known as the content-coding

*Content-Language*

Shows what the language of the entity is and also be able to differentiate entities that will go with the user’s preferred language

*Content-Length*

Shows what the size of the entity-body that was sent to the recipient is in decimal number of OCTETs if used in the HEAD method. But if used in the GET method, it will only show the size of the entity-body that would’ve been sent.

*Content-Location*

Used to provide the resource location when the entity that has been requested is accessible in another location other than the resource’s URI

*Content-MD5*

Checks the integrity of the message and provide MD5 digest of the entity

*Content-Range*

When a partial entity-body is sent, this is used to specify in what part of the full entity-body should the partial body should go to

*Content-Type*

Shows what media type is the entity-body that was sent if used in the HEAD method. But if used in the GET method, it will show what media type is the entity-body that is supposed to be sent.

*Expires*

Shows when and/or what time the response will be considered stale

*Last-Modified*

Shows the date and time at which the entity or resource was last modified