**HEADERS**

* The frame (type = 0x1) is used for *opening* a stream, and it is the one who carries the header block fragment. These are the state which the HEADERS Frames can be sent into a stream: IDLE, RESERVED (LOCAL), OPEN, and HALF-CLOSED (REMOTE). While the End Header, the frame (type=0x4) is set when the bit 2 indicates that a certain frame have the entire header block and there’s no any CONTINUATION FRAMES followed.

**STREAMS**

* These are identified by an unsigned 31-bit integer type value. The streams initiated by the client should be odd-numbered streams identifiers, and if the streams are initiated by the servers it should be even-numbered stream identifiers. The Stream Identifiers of (0x0) or ZERO is commonly used for connection control messages and this identifier could not use for establishing a new stream.

**REQUEST METHODS**

* It is the label of a desired action to be performed on the given resources. It is also called HTTP verbs but they can be also nouns, each of these methods has its own semantics on doing its task, but those which has a similar features on their semantics are shared via groupings.

**AUTHENTICATION**

* The framework that can be used by a server for challenging its clients when requesting and it is also used by client by providing the authenticated information. This is how the authentication framework flows: The server-side will respond 401 (UNAUTHORIZE) status to the client and gives the information on how to authenticate with the “WWW-Authenticate response header” that contains at least 1 challenge. In client-side it will prompt for password to the user for issuing the right authorization header, then request to the server, and a client that wants to authenticate itself with a server must include the Authorization request header field with the proper request.

**CONTENT NEGOTIATION**

* Commonly HTTP responses consist an interpretation of a human user information, making it as an entity, because naturally, it is desirable if the corresponding request is supplied by “best-available” entity. But we know that all servers and caches do not have same preferences of what is the “best” for the user, same as the user agents they do not have the equal capability of rendering all the given entity types. Because of that issue the HTTP has provisions for several techniques for “content-negotiation it is the process for selecting the best entity for the given response when there are bunch of available representation”.
* Two kinds of CONTENT-NEGOTIATION:
* **Server-Driven –** its algorithm is used for the selection of best representation for a response. The selection is depend on the available representation of the given response (Information it may vary: Content-coding, Language), contents of header fields in a request message, and other information that is related to the request.
* **Agent-Driven –** The feedback of the user agent after receiving initial response from the origin server is the information used for the selection (can be automatic if the user is capable or manually if the user selected from the generated menu) of the best representation for a response. The selection is depends on the available representation of the response included within header field or even the entity-body of an initial response. Each representation is identified via URI (Uniform Resource Identifier).

References:

Belshe, M., Peon, R., Thomson, M. (Ed), (2015, May), Hypertext Transfer Protocol Version 2, Retrieved from <https://lists.w3.org/Archives/Public/ietf-http-wg/2015AprJun/0436.html>.

Scholz, F.(2017, Jun 7), HTTP Request Methods, Retrieved from <https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>

Mills, C. (2018, Jan 26), HTTP Authentication, Retrieved from <https://developer.mozilla.org/en-US/docs/Web/HTTP/Authentication>

Content-Negotiation, (n.d), Retrieved from https://www.w3.org/Protocols/rfc2616/rfc2616-sec12.html