**HTTP Response Message**

1. Status line

* HTTP protocol version
* Status code
* Reason phrase

1. Message Response headers
2. Empty line
3. Message body aka payload (optional)

**HTTP Request Methods**

**A. Standard Methods**

1. **GET**

-Most commonly used HTTP method

-Used to request from the server the retrieval of the source identified by the request URI; the retrieved resource is returned in the message body as an entity.

-Can be combined with conditional and/or range request headers to effect conditional and/or partial resource retrieval

-Must be supported by all general-purpose servers.

1. **HEAD**

-Identical to GET, except the message body (entity) is not included in the response

-Used to retrieve metadata about the entity implied by request with transferring the entity itself

1. **POST**

-Request that the origin server accept the entity enclosed in the request as a new subordinate of the resource identified by the Request-URI in the Request-Line.

-Used in submitting HTML form data

1. **OPTIONS**

-Request for information about the communication options available on the request/response chain identified by the Request-URI. This method allows the client to determine the options and/or requirements associated with a resource, or the capabilities of a server, without implying a resource action or initiating resource retrieval.

1. **TRACE**

-Request the server to “echo” back to the client the received request

-Typically used for testing/diagnostics of the request chain

1. **PUT**

-Request the server to store the enclosed entity in the message under the specified request URI

1. **DELETE**

-Request the server to remove the resource identified by the request URI

1. **CONNECT**

-Reserved for use of tunneling proxy servers

**B. Extension Methods**

-e.g WebDAV (RFC 4918)

-Profind, propatch, mkcol, copy, move, click, lock, unlock

* **Safe Methods**

-method that doesn’t change on the resources on the client side (web)

-methods GEAT, HEAD, OPTIONS, TRACE

* **Idempotent Methods**

-methods GET, HEAD, PUT and DELETE share this property.

* **Cacheable Methods**

-response that we get can be cache for easily retrieving

**HTTP Message Headers (optional)**

**A. General Header Fields** – both for client and server

* Cache-control – our request can be cache / not
* Connection – used to control persistent connection
* Date – date when the request / response was held
* Pragma – old headers (HTTP 1.0); no cache control & connection (generic directive)
* Trailer – trailers available
* Transfer-encoding – chunk encoding
* Upgrade – possible to switch protocols
* Via – includes in the proxy headers which is the intersection
* Warning – used to indicate that there is something wrong / issue with the body

**B. Request Header** **Fields** – for request messages

* Accept – used for negotiating; mime types
* Accept-charset - mime types
* Accept-encoding - mime types
* Accept-Language – different languages
* Authorization – doing some protected resource
* Expect – 100 (Continue) or 417 (Expectation Failed)
* From – e-mail address of client (person generating request)
* Host – required header (virtual host)
* If-Match – conditional request; serves the service if it satisfy the condition
* If-Modified-Since - conditional request; serves the service if it satisfy the condition
* If-None-Match - conditional request; serves the service if it satisfy the condition
* If-Range - conditional request; serves the service if it satisfy the condition
* If-Unmodified-Since - conditional request; serves the service if it satisfy the condition
* Max-forward
* Proxy-Authorization
* Range – retrieve pieces of contents
* Referrer – where the request originated from
* User-Agent

**C. Response Header Fields** – for servers

* Accept-Range
* Age – age of the resource
* E-Tag – default configuration
* Location
* Proxy-Authenticate
* Retry-After – used conjunction with the error 503 (Service Unavailable)
* Server
* Vary
* WWW-Authenticate

**D. Entity Header Fields** – describes entity (payload)

* Allow – says what we can do in resource
* Content-encoding – how entity is encoded
* Content-language – specification language
* Content-length – length of the request in bytes
* Content-location – given a file you asked for
* Content-MD5 – digest algorithm; computes the hash of the resource
* Content-range – portion of range (resource)
* Content-type
* Expires – resource can be hash at a certain period of resource
* Last Modified – time stop of resource; used (conditional request)

**HTTP Status Codes**

**A. Informational (1xx)**

* 100 Continue – client is sending expect header
* 101 Switching Protocols – upgrading protocols

**B. Success (2xx)**

* 200 OK
* 201 Created
* 202 Accepted – already receive the request
* 203 Non-Authoritative Information – used by proxy
* 204 No Content
* 205 Reset Content – used to tell client to reset forms
* 206 Partial Content – range request

**C. Redirection (3xx)**

* 300 Multiple Choices – returned app if there are multiple resources
* 301 Moved Permanently – change bookmark
* 302 Found
* 303 See Other
* 304 Not Modified – conditional request
* 305 Use Proxy – go through a proxy
* 306 Temporary Redirect – shouldn’t make a change

**D. Client Error (4xx)**

* 400 Bad Request – request malform missing something
* 401 Unauthorized – accessing authorized page
* 402 Payment Required – allowing payment (e-commerce)
* 403 Forbidden – not allow to access resource
* 404 Not Found
* 405 Method not Allowed – put (not configured)
* 406 Not Acceptable – mime type / particular language
* 407 Proxy Authentication Required – redirected to proxy, authenticate first
* 408 Request Timeout – process of request time-out
* 409 Conflict
* 410 Gone – no longer available
* 411 Length Required - not specify how long data
* 412 Precondition Failed – put (already exist – non match)
* 413 Request Entity Too Large – can accept only to a maximum size
* 414 Request-URI Too Long – maximum size to URL (can’t handle by server)
* 415 Unsupported Media Type – doesn’t know how to handle content
* 416 Request Range Not Satisfiable
* 417 Expectation Failed – issuing expect header
* 417 Upgrade Required – upgrade first (http 2)

**E. Server Error (5xx)**

* 500 Internal Server Error - misconfiguration in server
* 501 Not Implemented – server doesn’t know how to handle request
* 502 Bad Gateway – there is a problem in gateway resource
* 503 Service Unavailable – maintenance work
* 504 Gateway Timeout - gateway (overloaded)
* 505 HTTP Version Not Supported - specify versions