HTTP STATUS CODES

1. Informational 1xx

   Indicates a provisional response.

   1.1. 100 Continue

   Response used to inform the client that the initial part of the request has been received and has not yet been rejected by the server.

   1.2. 101 Switching Protocols

2. Successful 2xx

   Indicates that the client's request was successfully received, understood, and accepted.

   2.1. 200 OK

   A certain request has succeeded. The information returned with the response is dependent on the method used in the request like:

   - GET, entity corresponding to the requested resource are sent in the response
   - HEAD, entity-header fields corresponding to the requested resource are sent in the response without any message body.
   - POST, entity describing or containing the result of the action.
   - TRACE, entity containing the request message as received by the end server.

   2.2. 201 Created

   Indicates that a request has been fulfilled and resulted in a new resource being created.

   2.3. 202 Accepted

   Indicates that a request has been accepted for processing, but the processing has not been completed. The request might or might not eventually be acted upon, as it might be disallowed when processing takes place. The 202 response is intentionally non-committal. Its purpose is to allow a server to accept a request for some other process without requiring that the user agent's connection to the server persist until the process is completed.

   2.4. 203 Non-Authoritative Information

   Indicates that the returned metainformation in the entity-header is not the definitive set as available from the origin server, but is gathered from a local or a third-party copy.

   2.5. 204 No Content

   Indicates that the server has fulfilled the request but does not need to return an entity-body, and might want to return updated information.

   2.6. 205 Reset Content

   Indicates that the server has fulfilled the request and the user agent should reset the document view which cause the request to be sent. This response is primarily intended to allow input for actions to take place via user input, followed by a clearing of the form in which the input is given so that the user can easily initiate another input action.

   2.7. 206 Partial Content

   Indicates that the server has fulfilled the partial GET request for the resource. This response must include header fields like Content-Range, date, Etag/Content-Location and Expires, Cache-control and vary.

3. Redirection 3xx

   This class of status codes indicates tha further action needs to be taken by the user agent in order to fulfil the request.

3.1. 300 Multiple Choice

Indicates that the requested resource corresponds to any one of a set of representations, each with its own specific location, and agent-driven negotiation information is being provide so that the user agent can select a preferred representation and redirect its request to that location.

3.2. 301 Moved Permanently

Indicates that the requested resource has been assigned a new permanent URI and any future references to this resource should use one of the returned URIs. This response is cacheable unless indicated otherwise. If this response is received in response to a request other that GET or HEAD, the user agent must not automatically redirect the request unless it can be confirmed by the user, since this might change the conditions under which the request was issued.

3.3. 302 Found

Indicates that the requested resource resides temporarily under a different URI. Since the redirection might be altered in occasion, the client should continue to use the Request-URI for future requests.

3.4. 303 See Other

Indicates that the response to the request can be found under a different URI and should be retrieved using a GET method on that resource. This method exists primarily to allow the output of a POST activated script to redirect the user agent to as selected resource. This response must not be cached, but the response to the second (redirected) request might be cacheable.

3.5. 304 Not Modified

Indicates that the client has performed a conditional GET request and access is allowed, but the document has not been modified, the server should respond with this status code. This response must not contain a message-body, and thus is always terminated by the first empty line after the header fields.

3.6. 305 Use Proxy

Indicates that the requested resource must be accessed through the proxy given by the Location field. The Location field gives the URI of the proxy. The recipient is expected to repeat this single request via the proxy. This response must only be generated by origin servers.

3.7. 306 (Unused)

This status code is no longer used, and the code was reserved.

3.8. 307 Temporary Redirect

Indicates that the requested resource resides temporarily under a different URI. Since he redirection may be altered on occasion, the client should continue to use the Request-URI for future requests.

4. Client Error 4xx

This class of status codes is intended for cases in which the client seems to have erred. These status codes are applicable to any request method.

4.1. 400 Bad Request

Indicates that the request could not be understood by the server due to malformed syntax. The client should not repeat the request without modifications.

4.2. 401 Unauthorized

Indicates that the request requires user authentication.

4.3. 402 Payment Required
This code is reserved for future use.

4.4. 403 Forbidden
Indicates that the server understood the request, but is refusing to fulfil it.

4.5. 404 Not Found
Indicates that the server has not found anything matching the Request-URI.

4.6. 405 Method Not Allowed
Indicates that the method specified in the Request-Line is not allowed or the resource identified by the Request-URI.

4.7. 406 Not Acceptable
Indicates that the resource identified by the request is only capable of generating response entities which have content characteristics not acceptable according to the accept headers sent in the request.

4.8. 407 Proxy Authentication Required
This code is similar to 401, but indicates that the client must first authenticate itself with the proxy. The proxy must return a Proxy-Authenticate header field containing a challenge applicable to the proxy for the requested resource.

4.9. 408 Request Timeout
Indicates that the client did not produce a request within the time that that the server was prepared to wait.

4.10. 409 Conflict
Indicates that request could not be completed due to a conflict with the current state of the resource.

4.11. 410 Gone
Indicates that the request resource is no longer available at the server and no forwarding address is known.

4.12. 411 Length Required
Indicates that the server refuse to accept the request without a defined Content-Length.

4.13. 412 Precondition Failed
Indicates that the precondition given in one or more of the request-header fields evaluated to false when it was tested on the server.

4.14. 413 Request Entity Too Large
Indicates that the server is refusing to process a request because the request entity is larger than that the server is willing or able to process.

4.15. 414 Request-URI Too Long
Indicates that the server is refusing to service that request because the Request-URI is longer that the server is willing to interpret.

4.16. 415 Unsupported Media Type
Indicates that the server is refusing to service the request because the entity of the request is in format not supported by the requested resource for the requested method.

4.17. 416 Requested Range Not Satisfiable
Indicates that a server should return a response with this status code I a request includes a Range request-header field and none of the range-specifier value in this field overlap the

current extent of the selected resource, and the request did not include an If-Range request-header field.

4.18. 417 Expectation Failed

Indicates that the expectation given in an Expect request-header field could not be met by this server, or, if the server is a proxy, the server has unambiguous evidence that the request could not be met by the next-hop server.

5. Server Error 5xx

Class of status codes indicating cases on which the server is aware that it has erred or is incapable of performing the request.

5.1. 500 Internal Server Error

Indicates that the server encountered an unexpected condition which prevented it from fulfilling the request.

5.2. 501 Not Implemented

Indicates that the server does not support the functionality required to fulfil the request. This is the appropriate response when the server does not recognize the request method and is not capable of supporting it for any resource.

5.3. 502 Bad Gateway

Indicates that the server, while acting as a gateway or proxy, received an invalid response from the upstream server it accessed in attempting to fulfil the request.

5.4. 503 Service Unavailable

Indicates that the server is currently unable to handle the request due to a temporary overloading or maintenance of the server.

5.5. 504 Gateway Timeout

Indicates that a server currently acting as a gateway or proxy did not receive a timely response from the upstream server specified by the URI or some other auxiliary server it need to access in attempting to complete the request.

5.6. 505 HTTP Version Not Supported

Indicates that the server does not support the HTTP protocol version that was used in the request message.