HTTP *Status Codes*

After RFC 9110, *HTTP Semantics* (IETF, June 2022)

R. Fielding, M. Nottingham, J. Reschke, Editors · and the IANA Hypertext Transfer Protocol (HTTP) Status Code Registry

15

The status code of a response is a three-digit integer code that describes the result of the request and the semantics of the response, including whether the request was successful and what content is enclosed (if any). All valid status codes are within the range of 100 to 599, inclusive.

The first digit of the status code defines the class of response. The last two digits do not have any categorization role. There are five values for the first digit:

`1xx (Informational)`: The request was received, continuing process
`2xx (Successful)`: The request was successfully received, understood, and accepted
`3xx (Redirection)`: Further action needs to be taken in order to complete the request
`4xx (Client Error)`: The request contains bad syntax or cannot be fulfilled
`5xx (Server Error)`: The server failed to fulfill an apparently valid request

HTTP status codes are extensible. A client is not required to understand the meaning of all registered status codes, though such understanding is obviously desirable. However, a client MUST understand the class of any status code, as indicated by the first digit, and treat an unrecognized status code as being equivalent to the x00 status code of that class.

The status codes listed below are defined in this specification. The reason phrases listed here are only recommendations -- they can be replaced by local equivalents or left out altogether without affecting the protocol.

Informational 1xx

The 1xx (Informational) class of status code indicates an interim response for communicating connection status or request progress prior to completing the requested action and sending a final response.

`100` **Continue** `15.2.1`
The 100 (Continue) status code indicates that the initial part of a request has been received and has not yet been rejected by the server.

`101` **Switching Protocols** `15.2.2`
The 101 (Switching Protocols) status code indicates that the server understands and is willing to comply with the client's request, via the Upgrade header field (Section 7.8), for a change in the application protocol being used on this connection.

`102` **Processing** `RFC 2518`

`103` **Early Hints** `RFC 8297`

`104` **Upload Resumption Supported**
`draft-ietf-httpbis-resumable-upload-05`
`TEMPORARY - registered 2024-11-13, extension registered 2025-09-15, expires 2026-11-13`

`105-199` *Unassigned*

Successful 2xx

The 2xx (Successful) class of status code indicates that the client's request was successfully received, understood, and accepted.

`200` **OK** `15.3.1`
The 200 (OK) status code indicates that the request has succeeded.

`201` **Created** `15.3.2`
The 201 (Created) status code indicates that the request has been fulfilled and has resulted in one or more new resources being created.

`202` **Accepted** `15.3.3`
The 202 (Accepted) status code indicates that the request has been accepted for processing, but the processing has not been completed.

`203` **Non-Authoritative Information** `15.3.4`
The 203 (Non-Authoritative Information) status code indicates that the request was successful but the enclosed content has been modified from that of the origin server's 200 (OK) response by a transforming proxy (Section 7.7).

`204` **No Content** `15.3.5`
The 204 (No Content) status code indicates that the server has successfully fulfilled the request and that there is no additional content to send in the response content.

`205` **Reset Content** `15.3.6`
The 205 (Reset Content) status code indicates that the server has fulfilled the request and desires that the user agent reset the "document view", which caused the request to be sent, to its original state as received from the origin server.

`206` **Partial Content** `15.3.7`
The 206 (Partial Content) status code indicates that the server is successfully fulfilling a range request for the target resource by transferring one or more parts of the selected representation.

`207` **Multi-Status** `RFC 4918`

`208` **Already Reported** `RFC 5842`

`209-225` *Unassigned*

`226` **IM Used** `RFC 3229`

`227-299` *Unassigned*

Redirection 3xx

The 3xx (Redirection) class of status code indicates that further action needs to be taken by the user agent in order to fulfill the request.

`300` **Multiple Choices** `15.4.1`
The 300 (Multiple Choices) status code indicates that the target resource has more than one representation, each with its own more specific identifier, and information about the alternatives is being provided so that the user (or user agent) can select a preferred representation by redirecting its request to one or more of those identifiers.

`301` **Moved Permanently** `15.4.2`
The 301 (Moved Permanently) status code indicates that the target resource has been assigned a new permanent URI and any future references to this resource ought to use one of the enclosed URIs.

`302` **Found** `15.4.3`
The 302 (Found) status code indicates that the target resource resides temporarily under a different URI.

`303` **See Other** `15.4.4`
The 303 (See Other) status code indicates that the server is redirecting the user agent to a different resource, as indicated by a URI in the Location header field, which is intended to provide an indirect response to the original request.

`304` **Not Modified** `15.4.5`
The 304 (Not Modified) status code indicates that a conditional GET or HEAD request has been received and would have resulted in a 200 (OK) response if it were not for the fact that the condition evaluated to false.

`305` **Use Proxy** `15.4.6`
The 305 (Use Proxy) status code was defined in a previous version of this specification and is now deprecated (Appendix B of [RFC7231]).

`306` **(Unused)** `15.4.7`
The 306 status code was defined in a previous version of this specification, is no longer used, and the code is reserved.

`307` **Temporary Redirect** `15.4.8`
The 307 (Temporary Redirect) status code indicates that the target resource resides temporarily under a different URI and the user agent MUST NOT change the request method if it performs an automatic redirection to that URI.

`308` **Permanent Redirect** `15.4.9`
The 308 (Permanent Redirect) status code indicates that the target resource has been assigned a new permanent URI and any future references to this resource ought to use one of the enclosed URIs.

`309-399` *Unassigned*

Client Error 4xx

The 4xx (Client Error) class of status code indicates that the client seems to have erred.

`400` **Bad Request** `15.5.1`
The 400 (Bad Request) status code indicates that the server cannot or will not process the request due to something that is perceived to be a client error (e.g., malformed request syntax, invalid request message framing, or deceptive request routing).

`401` **Unauthorized** `15.5.2`
The 401 (Unauthorized) status code indicates that the request has not been applied because it lacks valid authentication credentials for the target resource.

`402` **Payment Required** `15.5.3`
The 402 (Payment Required) status code is reserved for future use.

`403` **Forbidden** `15.5.4`
The 403 (Forbidden) status code indicates that the server understood the request but refuses to fulfill it.

`404` **Not Found** `15.5.5`
The 404 (Not Found) status code indicates that the origin server did not find a current representation for the target resource or is not willing to disclose that one exists.

`405` **Method Not Allowed** `15.5.6`
The 405 (Method Not Allowed) status code indicates that the method received in the request-line is known by the origin server but not supported by the target resource.

`406` **Not Acceptable** `15.5.7`
The 406 (Not Acceptable) status code indicates that the target resource does not have a current representation that would be acceptable to the user agent, according to the proactive negotiation header fields received in the request (Section 12.1), and the server is unwilling to supply a default representation.

`407` **Proxy Authentication Required** `15.5.8`
The 407 (Proxy Authentication Required) status code is similar to 401 (Unauthorized), but it indicates that the client needs to authenticate itself in order to use a proxy for this request.

`408` **Request Timeout** `15.5.9`
The 408 (Request Timeout) status code indicates that the server did not receive a complete request message within the time that it was prepared to wait.

`409` **Conflict** `15.5.10`
The 409 (Conflict) status code indicates that the request could not be completed due to a conflict with the current state of the target resource.

`410` **Gone** `15.5.11`
The 410 (Gone) status code indicates that access to the target resource is no longer available at the origin server and that this condition is likely to be permanent.

`411` **Length Required** `15.5.12`
The 411 (Length Required) status code indicates that the server refuses to accept the request without a defined Content-Length (Section 8.6).

`412` **Precondition Failed** `15.5.13`
The 412 (Precondition Failed) status code indicates that one or more conditions given in the request header fields evaluated to false when tested on the server (Section 13).

`413` **Content Too Large** `15.5.14`
The 413 (Content Too Large) status code indicates that the server is refusing to process a request because the request content is larger than the server is willing or able to process.

`414` **URI Too Long** `15.5.15`
The 414 (URI Too Long) status code indicates that the server is refusing to service the request because the target URI is longer than the server is willing to interpret.

`415` **Unsupported Media Type** `15.5.16`
The 415 (Unsupported Media Type) status code indicates that the origin server is refusing to service the request because the content is in a format not supported by this method on the target resource.

`416` **Range Not Satisfiable** `15.5.17`
The 416 (Range Not Satisfiable) status code indicates that the set of ranges in the request's Range header field (Section 14.2) has been rejected either because none of the requested ranges are satisfiable or because the client has requested an excessive number of small or overlapping ranges (a potential denial of service attack).

`417` **Expectation Failed** `15.5.18`
The 417 (Expectation Failed) status code indicates that the expectation given in the request's Expect header field (Section 10.1.1) could not be met by at least one of the inbound servers.

`418` **(Unused)** `15.5.19`
[RFC2324] was an April 1 RFC that lampooned the various ways HTTP was abused; one such abuse was the definition of an application-specific 418 status code, which has been deployed as a joke often enough for the code to be unusable for any future use.

`419-420` *Unassigned*

`421` **Misdirected Request** `15.5.20`
The 421 (Misdirected Request) status code indicates that the request was directed at a server that is unable or unwilling to produce an authoritative response for the target URI.

`422` **Unprocessable Content** `15.5.21`
The 422 (Unprocessable Content) status code indicates that the server understands the content type of the request content (hence a 415 (Unsupported Media Type) status code is inappropriate), and the syntax of the request content is correct, but it was unable to process the contained instructions.

`423` **Locked** `RFC 4918`

`424` **Failed Dependency** `RFC 4918`

`425` **Too Early** `RFC 8470`

`426` **Upgrade Required** `15.5.22`
The 426 (Upgrade Required) status code indicates that the server refuses to perform the request using the current protocol but might be willing to do so after the client upgrades to a different protocol.

`427` *Unassigned*

`428` **Precondition Required** `RFC 6585`

`429` **Too Many Requests** `RFC 6585`

`430` *Unassigned*

`431` **Request Header Fields Too Large** `RFC 6585`

`432-450` *Unassigned*

`451` **Unavailable For Legal Reasons** `RFC 7725`

`452-499` *Unassigned*

Server Error 5xx

The 5xx (Server Error) class of status code indicates that the server is aware that it has erred or is incapable of performing the requested method.

`500` **Internal Server Error** `15.6.1`
The 500 (Internal Server Error) status code indicates that the server encountered an unexpected condition that prevented it from fulfilling the request.

`501` **Not Implemented** `15.6.2`
The 501 (Not Implemented) status code indicates that the server does not support the functionality required to fulfill the request.

`502` **Bad Gateway** `15.6.3`
The 502 (Bad Gateway) status code indicates that the server, while acting as a gateway or proxy, received an invalid response from an inbound server it accessed while attempting to fulfill the request.

`503` **Service Unavailable** `15.6.4`
The 503 (Service Unavailable) status code indicates that the server is currently unable to handle the request due to a temporary overload or scheduled maintenance, which will likely be alleviated after some delay.

`504` **Gateway Timeout** `15.6.5`
The 504 (Gateway Timeout) status code indicates that the server, while acting as a gateway or proxy, did not receive a timely response from an upstream server it needed to access in order to complete the request.

`505` **HTTP Version Not Supported** `15.6.6`
The 505 (HTTP Version Not Supported) status code indicates that the server does not support, or refuses to support, the major version of HTTP that was used in the request message.

`506` **Variant Also Negotiates** `RFC 2295`

`507` **Insufficient Storage** `RFC 4918`

`508` **Loop Detected** `RFC 5842`

`509` *Unassigned*

`510` **Not Extended**
`RFC 2774 · Status change of HTTP experiments to Historic`
`OBSOLETED`

`511` **Network Authentication Required** `RFC 6585`

`512-599` *Unassigned*

Codes and reason phrases: IANA HTTP Status Code Registry, updated 2025-09-15, CC0 1.0. Definitions: first sentences quoted unmodified from RFC 9110, Section 15, by the IETF; section numbers refer to RFC 9110. Copyright (c) 2022 IETF Trust and the persons identified as the document authors. Compilation: MIT.

Codes without a definition are registered by the document named after their reason phrase.

`rfc-editor.org/rfc/rfc9110 · iana.org/assignments/http-status-codes`