KNOW YOUR HTTP STATUS

HTTP status codes are returned in the response. They each consist of a three digit numerical code and a text description (the text description is simply advisory, and may be translated to other languages). The codes are categorized such that general classes of errors have the same most-significant digit, so if a client does not recognize a status code it may still infer the code's category.

1XX Informational

Status Code	Description
100 Continue	The server has received the request headers, and the client should begin to send the request body
101 Switching Protocols	The server has received a request to switch protocols and is doing so
102 Processing (WebDAV)	Indicates that the server has received the request, used by WebDAV to avoid timeouts for long-running requests

2XX Success

Status Code	Description
200 OK	Standard response for successful HTTP requests
201 Created	The request was successful, *and* a new resource has been created
202 Accepted	The request was accepted for processing, but the job hasn't actually been completed. It is possible that the request will be rejected once processing takes place
203 Non-Authoritative Information	The request was successfully processed, but the returned information may be from an untrusted third party
204 No Content	The request was processed and no content is being returned
206 Partial Content	Only part of the request body is being delivered (for example, when resuming an interrupted download)
207 Multi-Status (WebDAV)	The message body is an XML document and may contain multiple status codes per sub-requests
208 Already Reported (WebDAV)	The response has already been enumerated in a previous reply and will not be reported again

3XX Redirection

Status Code	Description
300 Multiple Choices	Indicates that there are multiple locations which the client may follow
301 Moved Permanently	The resource has moved permanently, and all future requests should use the give URL instead
302 Found	The resource has been found (or moved temporarily). The HTTP/1.0 specification requires that the redirect uses the same verb, but in practice clients use a GET as in a 303. See 303 and 307
303 See Other	The resource has been found, and should be accessed using a GET method. Added in HTTP/1.1 to clarify the ambiguity in the behavior of status 302. See 302 and 307
304 Not Modified	The resource has not been modified since the last time the client has cached it
305 Use Proxy	The resource should be accessed through a specified proxy
307 Temporary Redirect	The request should be repeated with the same request method at the given address. Added in HTTP/1.1 to clarify the ambiguity in the behavior of status 302. See 302 and 303

4XX Client Error

Status Code	Description
400 Bad Request	The request can not be fulfilled because the request contained bad syntax
401 Unauthorized	The client needs to authenticate in order to access this resource
402 Payment Required	This code is intended to be used for a micropayment system, but the specifics for this system are unspecified and this code is rarely used
403 Forbidden	The client is not allowed to access this resource. Generally, the client is authenticated and does not have sufficient permission
404 Not Found	The resource was not found, though its existence in the future is possible
405 Method Not Allowed	The method used in the request is not supported by the resource
406 Not Acceptable	The server can not generate content which is acceptable to the client according to the request's "Accept" header
407 Proxy Authentication Required	The client must authenticate with the proxy
408 Request Timeout	The client did not complete its request in a reasonable timeframe
409 Conflict	The request could not be completed due to a conflict in state (for example, attempting to update a resource when it has changed since last access)
410 Gone	The resource is gone, and will always be gone; the client should not request the resource again
411 Length Required	The request is missing its "Content-Length" header, which is required by this resource
412 Precondition Failed	The server can not meet preconditions specified in the client request
413 Request Entity Too Large	The request body is larger than the server will process
414 Request-URI Too Long	The request URI is too long for the server to process
415 Unsupported Media Type	The server can not process the request body because it is of an unsupported MIME type
416 Requested Range Not Satisfiable	The client has asked for portion of a file that the server can not supply (ie, a range of bytes outside the size of the requested file)
417 Expectation Failed	The server can not meet the requirements of the "Expect" header in the request
418 I'm a teapot (HTCPCP)	Returned by teapots implementing the HyperText Coffee Pot Control Protocol
420 Enhance Your Calm (Twitter)	The client is being rate-limited; a reference to cannabis culture
422 Unprocessable Entity (WebDAV)	The server can not process the request due to semantic errors
423 Locked (WebDAV)	The resource is currently locked
424 Failed Dependency (WebDAV)	The request failed because of a previously-failed request
429 Too Many Requests	The client is being rate-limited
431 Request Header Fields Too Large	Either a single request header is too large, or all the header fields as a group are too large
444 No Response (Nginx)	Used in Nginx logs. Indicates that the server closed the connection without sending any response whatsoever
449 Retry With (Microsoft)	The request should be retried after performing some action
450 Blocked by Windows Parental Controls (Microsoft)	Windows Parental Controls are turned on and are blocking access to the resource
451 Unavailable For Legal Reasons (Internet Draft)	Intended to be used when a resource is being censored or blocked; a reference to Fahrenheit 451

5XX Server Error

Status Code	Description
500 Internal Server Error	A generic server error message, for when no other more specific message is suitable
501 Not Implemented	The server can not process the request method
502 Bad Gateway	The server is a gateway or proxy, and received a bad response from the upstream server (such as a socket hangup)
503 Service Unavailable	The resource is temporarily unavailable, usually because it is overloaded or down for maintenance
504 Gateway Timeout	The server is a gateway or proxy, and the upstream server did not respond in a reasonable timeframe
505 HTTP Version Not Supported	The server does not support the request's specified HTTP version
507 Insufficient Storage (WebDAV)	The server is out of storage space and can not complete the request
508 Loop Detected (WebDAV)	The server has detected an infinite loop while processing the request
509 Bandwidth Limit Exceeded	A convention used to report that bandwidth limits have been exceeded, and not part of any RFC or spec