HTTP/1.1 Status Codes by Category

This page has been tested and conforms to WCAG 2.0 Accessibility Guidelines

About the HTTP protocol ...

The HTTP Protocol (*Hyper-Text Transfer Protocol*) is a part of the TCP/IP suite of protocols and it is the communications protocol that is used to request and serve web pages or other HTTP documents. This page describes all the HTTP/1.1 response Status Codes and also provides a demonstration of the use of the HTTP protocol.

About HTTP Status Codes ...

Most of you, during your travels around the Web, will have seen an 'HTTP 404 Not Found' response page, perhaps when trying to follow a broken link from a search engine. This is the nearest that most people will come to directly interacting with the HTTP protocol. The '404' is an HTTP status code and their is a long list of other HTTP status codes, most of which you will never come across unless you are writing server-side programs that use the HTTP protocol to communicate between two servers. The only exception to this that I can think of is if you are a webmaster using 'error redirection' on your web server.

The current HTTP protocol (June, 2000) is HTTP/1.1 and this page lists all the HTTP/1.1 status codes together with a corresponding reason phrase and full description. If you are a student of the Web then it can be very interesting to read through the descriptions to help gain an insight into how the HTTP protocol works.

Most of the information on this page has been taken from W3c's rfc2068 (*Request For Comment*) and a link is provided at the bottom of this page. A primary reason for our presenting the information here is to present it in a more easily understandable format. RFC's are not noted for their brevity or clarity and they appear never to use one word when a hundred words can be used instead. They are written much like legal documents and it is our intention is to make the information more easily understood.

Status Code Categories ...

HTTP/1.1 status codes are split into five different 'Classes of Response' or categories and we start by listing these categories together with a description of each category. We then list the individual values of the numeric status codes defined in HTTP/1.1, and an example corresponding 'reason-phrase' description is shown with each status code. The 'reasons-phrases' descriptions are only recommended -- they may be replaced by local equivalents without affecting the protocol (*Ed. presumably, this relates to different human languages*).

Family	Category	Description
1xx	Informational	*REQUEST RECEIVED, CONTINUING PROCESS.* This class of status code indicates a provisional response, consisting only of the Status-Line and optional headers, and is terminated by an empty line. Since HTTP/1.0 did not define any 1xx status codes, servers MUST NOT send a 1xx response to an HTTP/1.0 client except under experimental conditions.
2xx	Success	*THE ACTION WAS SUCCESSFULLY RECEIVED, UNDERSTOOD, AND ACCEPTED.* This class of status code indicates that the client's request was successfully received, understood, and accepted.
3xx	Redirection	*FURTHER ACTION MUST BE TAKEN IN ORDER TO COMPLETE THE REQUEST.* This class of status code indicates that further action needs to be taken by the user agent in order to fulfill the request. The action required MAY be carried out by the user agent without interaction with the user if and only if the method used in the second request is GET or HEAD. A user agent SHOULD NOT automatically redirect a request more than 5 times, since such redirections usually indicate an infinite loop.
4xx	Client Error	*THE REQUEST CONTAINS BAD SYNTAX OR CANNOT BE FULFILLED.* The 4xx class of status code is intended for cases in which the client seems to have erred. Except when responding to a HEAD request, the server SHOULD include an entity containing an explanation of the error situation, and whether it is a temporary or permanent condition. These status codes are applicable to any request method. User agents SHOULD display any included entity to the user. Note: If the client is sending data, a server implementation using TCP should be careful to ensure that the client acknowledges receipt of the packet(s) containing the response, before the server closes the input connection. If the client continues sending data to the server after the close, the server's TCP stack will send a reset packet to the client, which may erase the client's unacknowledged input buffers before they can be read and interpreted by the HTTP application.

5xx	Server Error	*THE SERVER FAILED TO FULFILL AN APPARENTLY VALID REQUEST.* Response status codes beginning with the digit "5" indicate cases in which the server is aware that it has erred or is incapable of performing the request. Except when responding to a HEAD request, the server SHOULD include an entity containing an explanation of the error situation, and whether it is a temporary or permanent condition. User agents SHOULD display any included entity to the user. These response codes are applicable to any request method.

1xx family - informational ...

Code	Reason-Phrase	Description
100	Continue	The client may continue with its request. This interim response is used to inform the client that the initial part of the request has been received and has not yet been rejected by the server. The client SHOULD continue by sending the remainder of the request or, if the request has already been completed, ignore this response. The server MUST send a final response after the request has been completed.
101	Switching Protocols	The server understands and is willing to comply with the client's request, via the Upgrade message header field (section 14.41), for a change in the application protocol being used on this connection. The server will switch protocols to those defined by the response's Upgrade header field immediately after the empty line which terminates the 101 response. The protocol should only be switched when it is advantageous to do so. For example, switching to a newer version of HTTP is advantageous over older versions, and switching to a real-time, synchronous protocol may be advantageous when delivering resources that use such features.

2xx family - success ...

Code	Reason-Phrase	Description

200	OK	The request has succeeded. The information returned with the response is dependent on the method used in the request, for example: GET an entity corresponding to the requested resource is sent in the response; HEAD the entity-header fields corresponding to the requested resource are sent in the response without any message-body;
201	Created	The request has been fulfilled and resulted in a new resource being created. The newly created resource can be referenced by the URI(s) returned in the entity of the response, with the most specific URL for the resource given by a Location header field. The origin server MUST create the resource before returning the 201 status code. If the action cannot be carried out immediately, the server should respond with 202 (Accepted) response instead.
202	Accepted	The request has been accepted for processing, but the processing has not been completed. The request MAY or MAY NOT eventually be acted upon, as it MAY be disallowed when processing actually takes place. There is no facility for re-sending a status code from an asynchronous operation such as this. The 202 response is intentionally non-committal. Its purpose is to allow a server to accept a request for some other process (perhaps a batch-oriented process that is only run once per day) without requiring that the user agent's connection to the server persist until the process is completed. The entity returned with this response SHOULD include an indication of the request's current status and either a pointer to a status monitor or some estimate of when the user can expect the request to be fulfilled.
203	Non-Authoritative Information	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. The set presented MAY be a subset or superset of the original version. For example, including local annotation information about the resource MAY result in a superset of the metainformation known by the origin server. Use of this response code is not required and is only appropriate when the response would otherwise be 200 (OK).
204	No Content	The server has fulfilled the request but there is no new information to send back. If the client is a user agent, it SHOULD NOT change its document view from that which caused the request to be sent. This response is primarily intended to allow input for actions to take place without causing a change to the user agent's active document view. The response MAY include new metainformation in the form of entity-headers, which SHOULD apply to the document currently in the user agent's active view. The 204 response MUST NOT include a message-body, and thus is always terminated by the first empty line after the header fields.
205	Reset Content	The server has fulfilled the request and the user agent SHOULD reset the document view which caused 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. The response MUST NOT include an entity.

206	Partial Content	The server has fulfilled the partial GET request for the resource. The request must have included a Range header field (section 14.36) indicating the desired range. The response MUST include either a Content-Range header field (section 14.17) indicating the range included with this response, or a multipart/byteranges Content-Type including Content-Range fields for each part. If multipart/byteranges is not used, the Content-Length header field in the response MUST match the actual number of OCTETs transmitted in the message-body. A cache that does not support the Range and Content-Range headers MUST NOT cache 206 (Partial) responses.

3xx family - redirection ...

Code	Reason-Phrase	Description
300	Multiple Choices	The requested resource corresponds to any one of a set of representations, each with its own specific location, and agent-driven negotiation information (section 12) is being provided so that the user (or user agent) can select a preferred representation and redirect its request to that location. Unless it was a HEAD request, the response SHOULD include an entity containing a list of resource characteristics and location(s) from which the user or user agent can choose the one most appropriate. The entity format is specified by the media type given in the Content-Type header field. Depending upon the format and the capabilities of the user agent, selection of the most appropriate choice may be performed automatically. However, this specification does not define any standard for such automatic selection. If the server has a preferred choice of representation, it SHOULD include the specific URL for that representation in the Location field; user agents MAY use the Location field value for automatic redirection. This response is cachable unless indicated otherwise.
301	Moved Permanently	The requested resource has been assigned a new permanent URI and any future references to this resource SHOULD be done using one of the returned URIs. Clients with link editing capabilities SHOULD automatically re-link references to the Request-URI to one or more of the new references returned by the server, where possible. This response is cachable unless indicated otherwise. If the new URI is a location, its URL SHOULD be given by the Location field in the response. Unless the request method was HEAD, the entity of the response SHOULD contain a short hypertext note with a hyperlink to the new URI(s). If the 301 status code is received in response to a request other than 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. Note: When automatically redirecting a POST request after receiving a 301 status code, some existing HTTP/1.0 user agents will erroneously change it into a GET request.
302	Moved Temporarily	The requested resource resides temporarily under a different URI. Since the redirection may be altered on occasion, the client SHOULD continue to use the Request-URI for future requests. This response is only cachable if indicated by a Cache-Control or Expires header field. If the new URI is a location, its URL SHOULD be given by the Locationfield in the response. Unless the request method was HEAD, the entity of the response SHOULD contain a short hypertext note with a hyperlink to the new URI(s). If the 302 status code is received in response to a request other than 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. Note: When automatically redirecting a POST request after receiving a 302 status code, some existing HTTP/1.0 user agents will erroneously change it into a GET request.
303	See Other	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 a selected resource. The new URI is not a substitute reference for the originally requested resource. The 303 response is not cachable, but the response to the second (redirected) request MAY be cachable. If the new URI is a location, its URL SHOULD be given by the Location field in the response. Unless the request method was HEAD, the entity of the response SHOULD contain a short hypertext note with a hyperlink to the new URI(s).
304	Not Modified	If 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. The response MUST NOT contain a message-body. If the conditional GET used a strong cache validator (see section 13.3.3), the response SHOULD NOT include other entity-headers. Otherwise (i.e., the conditional GET used a weak validator), the response MUST NOT include other entity-headers; this prevents inconsistencies between cached entity-bodies and updated headers. If a 304 response indicates an entity not currently cached, then the cache MUST disregard the response and repeat the request without the conditional. If a cache uses a received 304 response to update a cache entry, the cache MUST update the entry to reflect any new field values given in the response. The 304 response MUST NOT include a message-body, and thus is always terminated by the first empty line after the header fields.
305	Use Proxy	The requested resource MUST be accessed through the proxy given by the Location field. The Location field gives the URL of the proxy. The recipient is expected to repeat the request via the proxy.

4xx family - client error ...

Code	Reason-Phrase	Description
400	Bad Request	The request could not be understood by the server due to malformed syntax. The client SHOULD NOT repeat the request without modifications.
401	Unauthorized	The request requires user authentication. The response MUST include a WWW-Authenticate header field (section 14.46) containing a challenge applicable to the requested resource. The client MAY repeat therequest with a suitable Authorization header field (section 14.8). If the request already included Authorization credentials, then the 401 response indicates that authorization has been refused for those credentials. If the 401 response contains the same challenge as the prior response, and the user agent has already attempted authentication at least once, then the user SHOULD be presented the entity that was given in the response, since that entity MAY include relevant diagnostic information. HTTP access authentication is explained in section 11.
402	Payment Required	This code is reserved for future use.
403	Forbidden	The server understood the request, but is refusing to fulfill it. Authorization will not help and the request SHOULD NOT be repeated. If the request method was not HEAD and the server wishes to make public why the request has not been fulfilled, it SHOULD describe the reason for the refusal in the entity. This status code is commonly used when the server does not wish to reveal exactly why the request has been refused, or when no other response is applicable.
404	Not Found	The server has not found anything matching the Request-URI. No indication is given of whether the condition is temporary or permanent. If the server does not wish to make this information available to the client, the status code 403 (Forbidden) can be used instead. The 410 (Gone) status code SHOULD be used if the server knows, through some internally configurable mechanism, that an old resource is permanently unavailable and has no forwarding address.
405	Method Not Allowed	The method specified in the Request-Line is not allowed for the resource identified by the Request-URI. The response MUST include an Allow header containing a list of valid methods for the requested resource.

406	Not Acceptable	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. Unless it was a HEAD request, the response SHOULD include an entity containing a list of available entity characteristics and location(s) from which the user or user agent can choose the one most appropriate. The entity format is specified by the media type given in the Content-Type header field. Depending upon the format and the capabilities of the user agent, selection of the most appropriate choice may be performed automatically. However, this specification does not define any standard for such automatic selection. Note: HTTP/1.1 servers are allowed to return responses which are not acceptable according to the accept headers sent in the request. In some cases, this may even be preferable to sending a 406 response. User agents are encouraged to inspect the headers of an incoming response to determine if it is acceptable. If the response could be unacceptable, a user agent SHOULD temporarily stop receipt of more data and query the user for a decision on further actions.
407	Proxy Authentication Required	This code is similar to 401 (Unauthorized), but indicates that the client MUST first authenticate itself with the proxy. The proxy MUST return a Proxy-Authenticate header field (section 14.33) containing a challenge applicable to the proxy for the requested resource. The client MAY repeat the request with a suitable Proxy-Authorization header field (section 14.34). HTTP access authentication is explained in section 11.
408	Request Timeout	The client did not produce a request within the time that the server was prepared to wait. The client MAY repeat the request without modifications at any later time.
409	Conflict	The request could not be completed due to a conflict with the current state of the resource. This code is only allowed in situations where it is expected that the user might be able to resolve the conflict and resubmit the request. The response body SHOULD include enough information for the user to recognize the source of the conflict. Ideally, the response entity would include enough information for the user or user agent to fix the problem; however, that may not be possible and is not required. Conflicts are most likely to occur in response to a PUT request. If versioning is being used and the entity being PUT includes changes to a resource which conflict with those made by an earlier (third-party) request, the server MAY use the 409 response to indicate that it can't complete the request. In this case, the response entity SHOULD contain a list of the differences between the two versions in a format defined by the response Content-Type.
410	Gone	The requested resource is no longer available at the server and no forwarding address is known. This condition SHOULD be considered permanent. Clients with link editing capabilities SHOULD delete references to the Request-URI after user approval. If the server does not know, or has no facility to determine, whether or not the condition is permanent, the status code 404 (Not Found) SHOULD be used instead. This response is cachable unless indicated otherwise. The 410 response is primarily intended to assist the task of web maintenance by notifying the recipient that the

		resource is intentionally unavailable and that the server owners desire that remote links to that resource be removed. Such an event is common for limited-time, promotional services and for resources belonging to individuals no longer working at the server's site. It is not necessary to mark all permanently unavailable resources as "gone" or to keep the mark for any length of time -- that is left to the discretion of the server owner.
411	Length Required	The server refuses to accept the request without a defined Content-Length. The client MAY repeat the request if it adds a valid Content-Length header field containing the length of the message-body in the request message.
412	Precondition Failed	The precondition given in one or more of the request-header fields evaluated to false when it was tested on the server. This response code allows the client to place preconditions on the current resource metainformation (header field data) and thus prevent the requested method from being applied to a resource other than the one intended.
413	Request Entity Too Large	The server is refusing to process a request because the request entity is larger than the server is willing or able to process. The server may close the connection to prevent the client from continuing the request. If the condition is temporary, the server SHOULD include a Retry-After header field to indicate that it is temporary and after what time the client may try again.
414	Request-URI Too Long	The server is refusing to service the request because the Request-URI is longer than the server is willing to interpret. This rare condition is only likely to occur when a client has improperly converted a POST request to a GET request with long query information, when the client has descended into a URL "black hole" of redirection (e.g., a redirected URL prefix that points to a suffix of itself), or when the server is under attack by a client attempting to exploit security holes present in some servers using fixed-length buffers for reading or manipulating the Request-URI.
415	Unsupported Media Type	The server is refusing to service the request because the entity of the request is in a format not supported by the requested resource for the requested method.

5xx family - server error ...

Code	Reason-Phrase	Description

500	Internal Server Error	The server encountered an unexpected condition which prevented it from fulfilling the request.
501	Not Implemented	The server does not support the functionality required to fulfill 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.
502	Bad Gateway	The server, while acting as a gateway or proxy, received an invalid response from the upstream server it accessed in attempting to fulfill the request.
503	Service Unavailable	The server is currently unable to handle the request due to a temporary overloading or maintenance of the server. The implication is that this is a temporary condition which will be alleviated after some delay. If known, the length of the delay may be indicated in a Retry-After header. If no Retry-After is given, the client SHOULD handle the response as it would for a 500 response. Note: The existence of the 503 status code does not imply that a server must use it when becoming overloaded. Some servers may wish to simply refuse the connection.
504	Gateway Timeout	The server, while acting as a gateway or proxy, did not receive a timely response from the upstream server it accessed in attempting to complete the request.
505	HTTP Version Not Supported	The server does not support, or refuses to support, the HTTP protocol version that was used in the request message. The server is indicating that it is unable or unwilling to complete the request using the same major version as the client, as described in section 3.1, other than with this error message. The response SHOULD contain an entity describing why that version is not supported and what other protocols are supported by that server.