Response codes of the Hypertext Transfer Protocol This is a list of Hypertext Transfer Protocol (HTTP) response status codes. Status codes are issued by a server in response to a client's request made to the server. It includes codes from IETF Request for Comments (RFCs), other specifications, and some additional codes used in some common applications of the HTTP. The first digit of the status code specifies one of five standard classes of responses. The optional message phrases shown are typical, but any human-readable alternative may be provided, or none at all. Unless otherwise stated, the status code is part of the HTTP standard.[1] The Internet Assigned Numbers Authority (IANA) maintains the official registry of HTTP status codes.[2] All HTTP response status codes are separated into five classes or categories. The first digit of the status code defines the class of response, while the last two digits do not have any classifying or categorization role. There are five classes defined by the standard: 1xx informational response – 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 fulfil an apparently valid request 1xx informational response An informational response indicates that the request was received and understood. It is issued on a provisional basis while request processing continues. It alerts the client to wait for a final response. The message consists only of the status line and optional header fields, and is terminated by an empty line. As the HTTP/1.0 standard did not define any 1xx status codes, servers must not[note 1] send a 1xx response to an HTTP/1.0 compliant client except under experimental conditions.
4xx client errors
404 error on Wikimedia This class of status code is intended for situations in which the error seems to have been caused by the client. 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.
400 Bad Request The server cannot or will not process the request due to an apparent client error (e.g., malformed request syntax, size too large, invalid request message framing, or deceptive request routing).
401 Unauthorized
Similar to 403 Forbidden, but specifically for use when authentication is required and has failed or has not yet been provided. The response must include a WWW-Authenticate header field containing a challenge applicable to the requested resource. See Basic access authentication and Digest access authentication. 401

semantically means "unauthorised", the user does not have valid authentication credentials for the target resource. Some sites incorrectly issue HTTP 401 when an IP address is banned from the website (usually the website domain) and that specific address is refused permission to access a website.[citation needed]
402 Payment Required
Reserved for future use. The original intention was that this code might be used as part of some form of digital cash or micropayment scheme, as proposed, for example, by GNU Taler, but that has not yet happened, and this code is not widely used. Google Developers API uses this status if a particular developer has exceeded the daily limit on requests. Sipgate uses this code if an account does not have sufficient funds to start a call. Shopify uses this code when the store has not paid their fees and is temporarily disabled. Stripe uses this code for failed payments where parameters were correct, for example blocked fraudulent payments.
403 Forbidden The request contained valid data and was understood by the server, but the server is refusing action. This may be due to the user not having the necessary permissions for a resource or needing an account of some sort, or attempting a prohibited action (e.g. creating a duplicate record where only one is allowed). This code is also typically used if the request provided authentication by answering the WWW-Authenticate header field challenge, but the server did not accept that authentication. The request should not be repeated
404 Not Found The requested resource could not be found but may be available in the future. Subsequent requests by the client are permissible.
405 Method Not Allowed A request method is not supported for the requested resource; for example, a GET request on a form that requires data to be presented via POST, or a PUT request on a read-only resource.
406 Not Acceptable The requested resource is capable of generating only content not acceptable according to the Accept headers sent in the request. See Content negotiation.
407 Proxy Authentication Required The client must first authenticate itself with the proxy.
408 Request Timeout The server timed out waiting for the request. According to HTTP specifications: "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 Indicates that the request could not be processed because of conflict in the current state of the resource, such as an edit conflict between multiple simultaneous updates.
410 Gone Indicates that the resource requested was previously in use but is no

longer available and will not be available again. This should be used when a resource has been intentionally removed and the resource should be purged. Upon receiving a 410 status code, the client should not request the resource in the future. Clients such as search engines should remove the resource from their indices. Most use cases do not require clients and search engines to purge the resource, and a "404 Not Found" may be used instead.
411 Length Required The request did not specify the length of its content, which is required by the requested resource. 412 Precondition Failed The server does not meet one of the preconditions that the requester put on the request header fields.
413 Payload Too Large The request is larger than the server is willing or able to process. Previously called "Request Entity Too Large".
414 URI Too Long The URI provided was too long for the server to process. Often the result of too much data being encoded as a query-string of a GET request, in which case it should be converted to a POST request. Called "Request-URI Too Long" previously.
415 Unsupported Media Type The request entity has a media type which the server or resource does not support. For example, the client uploads an image as image/svg+xml, but the server requires that images use a different format.
416 Range Not Satisfiable The client has asked for a portion of the file (byte serving), but the server cannot supply that portion. For example, if the client asked for a part of the file that lies beyond the end of the file. Called "Requested Range Not Satisfiable" previously. 417 Expectation Failed The server cannot meet the requirements of the Expect request-header field.
418 I'm a teapot (RFC 2324, RFC 7168) This code was defined in 1998 as one of the traditional IETF April Fools' jokes, in RFC 2324, Hyper Text Coffee Pot Control Protocol, and is not expected to be implemented by actual HTTP servers. The RFC specifies this code should be returned by teapots requested to brew coffee. This HTTP status is used as an Easter egg in some websites, such as Google.com's "I'm a teapot" easter egg. Sometimes, this status code is also used as a response to a blocked request, instead of the more appropriate 403 Forbidden. 421 Misdirected Request The request was directed at a server that is not able to produce a response (for example because of connection reuse). 422 Unprocessable Content The request was well-formed (i.e., syntactically correct) but could not be processed. 423 Locked (WebDAV; RFC 4918) The resource that is being accessed is locked. 424 Failed Dependency (WebDAV; RFC 4918) The request failed because it depended on another request and that request failed (e.g., a PROPPATCH). 425 Too Early (RFC 8470) Indicates that the server is unwilling to risk processing a request that might be replayed. 426 Upgrade Required The client should switch to a different protocol such

as TLS/1.3, given in the Upgrade header field. 428 Precondition Required (RFC 6585) The origin server requires the request to be conditional. Intended to prevent the 'lost update' problem, where a client GETs a resource's state, modifies it, and PUTs it back to the server, when meanwhile a third party has modified the state on the server, leading to a conflict.
429 Too Many Requests (RFC 6585) The user has sent too many requests in a given amount of time. Intended for use with rate-limiting schemes. 431 Request Header Fields Too Large (RFC 6585) The server is unwilling to process the request because either an individual header field, or all the header fields collectively, are too large.
451 Unavailable For Legal Reasons (RFC 7725) A server operator has received a legal demand to deny access to a resource or to a set of resources that includes the requested resource. The code 451 was chosen as a reference to the novel Fahrenheit 451 (see the Acknowledgements in the RFC).