HTTP Status Codes

Enjoy this cheat sheet at its fullest within Dash, the macOS documentation browser.

1xx Informational

100 Continue

The server has received the request headers, and that the client should proceed to send the request body.

101 Switching protocols

The requester has asked the server to switch protocols and the server is acknowledging that it will do so.

102 Processing

The server has received and is processing the request, but no response is available yet.

2xx Success

200 OK

The standard response for successful HTTP requests.

201 Created

The request has been fulfilled and a new resource has been created.

202 Accepted

The request has been accepted but has not been processed yet. This code does not guarantee that the request will process successfully.

203 Non-authoritative information

HTTP 1.1. The server successfully processed the request but is returning information from another source.

204 No content

The server accepted the request but is not returning any content. This is often used as a response to a `DELETE` request.

205 Reset content

Similar to a `204 No Content` response but this response requires the requester to reset the document view.

206 Partial content

The server is delivering only a portion of the content, as requested by the client via a range header.

207 Multi-status

The message body that follows is an XML message and can contain a number of separate response codes, depending on how many sub-requests were made.

WebDAV - RFC 4918

208 Already reported

The members of a DAV binding have already been enumerated in a previous reply to this request, and are not being included again.

WebDAV - RFC 5842

226 IM used

The server has fulfilled a GET request for the resource, and the response is a representation of the result of one or more instance-manipulations applied to the current instance.

RFC 3229

3xx Redirection

300 Multiple choices

There are multiple options that the client may follow.

301 Moved permanently

The resource has been moved and all further requests should reference its new URI.

302 Found

The HTTP 1.0 specification described this status as "Moved Temporarily", but popular browsers respond to this status similar to behavior intended for `303`. The resource can be retrieved by referencing the returned URI.

303 See other

The resource can be retrieved by following other URI using the `GET` method. When received in response to a `POST`, `PUT`, or `DELETE`, it can usually be assumed that the server processed the request successfully and is sending the client to an informational endpoint.

304 Not modified

The resource has not been modified since the version specified in `If-Modified-Since` or `If-Match` headers. The resource will not be returned in response body.

305 Use proxy

HTTP 1.1. The resource is only available through a proxy and the address is provided in the response.

306 Switch proxy

Deprecated in HTTP 1.1. Used to mean that subsequent requests should be sent using the specified proxy.

307 Temporary redirect

HTTP 1.1. The request should be repeated with the URI provided in the response, but future requests should still call the original URI.

308 Permanent redirect

Experimental. The request and all future requests should be repeated with the URI provided in the response. The HTTP method is not allowed to be changed in the subsequent request.

308 Resume Incomplete (Google)

This code is used in the Resumable HTTP Requests Proposal to resume aborted PUT or POST requests

4xx Client Error

400 Bad request

The request could not be fulfilled due to the incorrect syntax of the request.

401 Unauthorized

The requester is not authorized to access the resource. This is similar to `403` but is used in cases where authentication is expected but has failed or has not been provided.

402 Payment required

Reserved for future use. Some web services use this as an indication that the client has sent an excessive number of requests.

403 Forbidden

The request was formatted correctly but the server is refusing to supply the requested resource. Unlike `401`, authenticating will not make a difference in the server's response.

404 Not found

The resource could not be found. This is often used as a catch-all for all invalid URIs requested of the server.

405 Method not allowed

The resource was requested using a method that is not allowed. For example, requesting a resource via a `POST` method when the resource only supports the `GET` method.

406 Not acceptable

The resource is valid, but cannot be provided in a format specified in the `Accept` headers in the request.

407 Proxy authentication required

Authentication is required with the proxy before requests can be fulfilled.

408 Request timeout

The server timed out waiting for a request from the client. The client is allowed to repeat the request.

409 Conflict

The request cannot be completed due to a conflict in the request parameters.

410 Gone

The resource is no longer available at the requested URI and no redirection will be given.

411 Length required

The request did not specify the length of its content as required by the resource.

412 Precondition failed

The server does not meet one of the preconditions specified by the client.

413 Request entity too large

The request is larger than what the server is able to process.

414 Request-URI too long

The URI provided in the request is too long for the server to process. This is often used when too much data has been encoded into the URI of a `GET` request and a `POST` request should be used instead.

415 Unsupported media type

The client provided data with a media type that the server does not support.

416 Requested range not satisfiable

The client has asked for a portion of the resource but the server cannot supply that portion.

417 Expectation failed

The server cannot meet the requirements of the Expect request-header field.

418 I'm a teapot

Any attempt to brew coffee with a teapot should result in the error code "418 I'm a teapot". The resulting entity body MAY be short and stout.

HTCPCP - RFC 2324

421 Misdirected request

The request was directed at a server that is not able to produce a response. This can be sent by a server that is not configured to produce responses for the combination of scheme and authority that are included in the request URI.

HTTP/2 - RFC 7540

422 Unprocessable entity

The request was formatted correctly but cannot be processed in its current form. Often used when the specified parameters fail validation errors.

WebDAV - RFC 4918

423 Locked

The requested resource was found but has been locked and will not be returned.

WebDAV - RFC 4918

424 Failed dependency

The request failed due to a failure of a previous request.

WebDAV - RFC 4918

426 Upgrade required

The client should repeat the request using an upgraded protocol such as TLS 1.0.

428 Precondition required

The origin server requires the request to be conditional.

Additional HTTP Status Codes - RFC 6585

429 Too many requests

The user has sent too many requests in a given amount of time ("rate limiting").

Additional HTTP Status Codes - RFC 6585

431 Request header fields too large

The server is unwilling to process the request because its header fields are too large.

Additional HTTP Status Codes - RFC 6585

440 Login Timeout (Microsoft)

A Microsoft extension. Indicates that your session has expired.

444 No Response (Nginx)

Used in Nginx logs to indicate that the server has returned no information to the client and closed the connection (useful as a deterrent for malware).

449 Retry With (Microsoft)

A Microsoft extension. The request should be retried after performing the appropriate action.

450 Blocked by Windows Parental Controls (Microsoft)

A Microsoft extension. This error is given when Windows Parental Controls are turned on and are blocking access to the given webpage.

451 Unavailable For Legal Reasons

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 1953 dystopian novel *Fahrenheit 451*, where books are outlawed.

An HTTP Status Code to Report Legal Obstacles - RFC 7725

451 Redirect (Microsoft)

Used in Exchange ActiveSync if there either is a more efficient server to use or the server cannot access the users' mailbox.

494 Request Header Too Large (Nginx)

Nginx internal code similar to 431 but it was introduced earlier in version 0.9.4 (on January 21, 2011).

495 Cert Error (Nginx)

Nginx internal code used when SSL client certificate error occurred to distinguish it from 4XX in a log and an error page redirection.

496 No Cert (Nginx)

Nginx internal code used when client didn't provide certificate to distinguish it from 4XX in a log and an error page redirection.

497 HTTP to HTTPS (Nginx)

Nginx internal code used for the plain HTTP requests that are sent to HTTPS port to distinguish it from 4XX in a log and an error page redirection.

498 Token expired/invalid (Esri)

Returned by ArcGIS for Server. A code of 498 indicates an expired or otherwise invalid token.

499 Client Closed Request (Nginx)

Used in Nginx logs to indicate when the connection has been closed by client while the server is still processing its request, making server unable to send a status code back.

499 Token required (Esri)

Returned by ArcGIS for Server. A code of 499 indicates that a token is required (if no token was submitted).

5xx Server Error

500 Internal server error

A generic status for an error in the server itself.

501 Not implemented

The server cannot respond to the request. This usually implies that the server could possibly support the request in the future — otherwise a `4xx` status may be more appropriate.

502 Bad gateway

The server is acting as a proxy and did not receive an acceptable response from the upstream server.

503 Service unavailable

The server is down and is not accepting requests.

504 Gateway timeout

The server is acting as a proxy and did not receive a response from the upstream server.

505 HTTP version not supported

The server does not support the HTTP protocol version specified in the request.

506 Variant also negotiates

Transparent content negotiation for the request results in a circular reference.

507 Insufficient storage

The user or server does not have sufficient storage quota to fulfill the request.

WebDAV - RFC 4918

508 Loop detected

The server detected an infinite loop in the request.

WebDAV - RFC 5842

509 Bandwidth Limit Exceeded (Apache bw/limited extension)

This status code is not specified in any RFCs. Its use is unknown.

510 Not extended

Further extensions to the request are necessary for it to be fulfilled.

511 Network authentication required

The client must authenticate with the network before sending requests.

RFC 6585

520 Unknown Error (Microsoft / CloudFlare)

This status code is not specified in any RFC and is returned by certain services, for instance Microsoft Azure and CloudFlare servers: "The 520 error is essentially a "catch-all" response for when the origin server returns something unexpected or something that is not tolerated/interpreted (protocol violation or empty response)."

521 Web Server Is Down (CloudFlare)

The origin server has refused the connection from CloudFlare.

522 Connection Timed Out (CloudFlare)

CloudFlare could not negotiate a TCP handshake with the origin server.

523 Origin Is Unreachable (CloudFlare)

CloudFlare could not reach the origin server; for example, if the DNS records for the origin server are incorrect.

524 A Timeout Occurred (CloudFlare)

CloudFlare was able to complete a TCP connection to the origin server, but did not receive a timely HTTP response.

525 SSL Handshake Failed (CloudFlare)

CloudFlare could not negotiate a SSL/TLS handshake with the origin server.

526 Invalid SSL Certificate (CloudFlare)

CloudFlare could not validate the SSL/TLS certificate that the origin server presented.

527 Railgun Error (CloudFlare)

The request timed out or failed after the WAN connection has been established.

Notes

- Based on cheat.errtheblog.com and List of HTTP status codes on Wikipedia.

You can modify and improve this cheat sheet here