DHAMITH KUMARA

LIST OF ALL ERRORS

FULL ERROR LIST

400 (Bad request)
401 (Authorization required)
402 (Payment required)
403 (Forbidden)
404 (Not found)
405 (Method not allowed)
406 (Not acceptable)
407 (Proxy authentication required)
408 (Request Timeout)
409 (Conflict)
410 (Gone)
411 (Length required)
412 (Precondition failed)
413 (Request entity too large)
414 (Request URI too large)
415 (Unsupported media type)
416 (Request range not satisfiable)
417 (Expectation failed)

418 (I'm a teapot)
422 (Unprocessable entity)
425 (Too Early)
426 (Upgrade Required)
428 (Precondition Required)
429 Too Many Requests
431 (Request Header Fields Too Large)
451 Unavailable For Legal Reasons
500 (Internal server error)
501 (Not Implemented)
502 (Bad gateway)
503 (Service unavailable)
504 (Gateway timeout)
505 (HTTP version not supported)
506 (Variant also negotiates)
507 (Insufficient storage)
508 (Loop Detected)
510 (Not Extended)
511 (Network Authentication Required)

400 (BAD REQUEST)

The HyperText Transfer Protocol (HTTP) 400 Bad Request response status code indicates that the server cannot or will not process the request due to something that is perceived to be a client error (for example, malformed request syntax, invalid request message framing, or deceptive request routing).

401 (AUTHORIZATION REQUIRED)

The HyperText Transfer Protocol (HTTP) 401 Unauthorized response status code indicates that the client request has not been completed because it lacks valid authentication credentials for the requested resource.

402 (PAYMENT REQUIRED)

The HTTP 402 Payment Required is a nonstandard response status code that is reserved for future use. This status code was created to enable digital cash or (micro) payment systems and would indicate that the requested content is not available until the client makes a payment.
Sometimes, this status code indicates that the request cannot be processed until the client makes a payment. However, no standard use convention exists and different entities use it in different contexts.

403 (FORBIDDEN)

The HTTP 403 Forbidden response status code indicates that the server understands the request but refuses to authorize it.

This status is similar to 401, but for the 403 Forbidden status code, re-authenticating makes no difference. The access is tied to the application logic, such as insufficient rights to a resource.

404 (NOT FOUND)

The HTTP 404 Not Found response status code indicates that the server cannot find the requested resource. Links that lead to a 404 page are often called broken or dead links and can be subject to link rot.

A 404 status code only indicates that the resource is missing: not whether the absence is temporary or permanent. If a resource is permanently removed, use the 410 (Gone) status instead.

405 (METHOD NOT ALLOWED)

The HyperText Transfer Protocol (HTTP) 405 Method Not Allowed response status code indicates that the server knows the request method, but the target resource doesn't support this method.
The server must generate an Allow header field in a 405 status code response. The field must contain a list of methods that the target resource currently supports.

406 (NOT ACCEPTABLE)

The HyperText Transfer Protocol (HTTP) 406 Not Acceptable client error response code indicates that the server cannot produce a response matching the list of acceptable values defined in the request's proactive content negotiation headers, and that the server is unwilling to supply a default representation.

407 (PROXY AUTHENTICATION REQUIRED)

The HTTP 407 Proxy Authentication Required client error status response code indicates that the request has not been applied because it lacks valid authentication credentials for a proxy server that is between the browser and the server that can access the requested resource.

This status is sent with a Proxy-Authenticate header that contains information on how to authorize correctly.

408 (REQUEST TIMEOUT)

The HyperText Transfer Protocol (HTTP) 408 Request Timeout response status code means that the server would like to shut down this unused connection. It is sent on an idle connection by some servers, even without any previous request by the client.

A server should send the "close" Connection header field in the response, since 408 implies that the server has decided to close the connection rather than continue waiting.

This response is used much more since some browsers, like Chrome, Firefox 27+, and IE9, use HTTP pre-connection mechanisms to speed up surfing.

409 (CONFLICT)

The HTTP 409 Conflict response status code indicates a request conflict with the current state of the target resource.

Conflicts are most likely to occur in response to a PUT request. For example, you may get a 409 response when uploading a file that is older than the existing one on the server, resulting in a version control conflict.

410 (GONE)

The HyperText Transfer Protocol (HTTP) 410 Gone client error response 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.

If you don't know whether this condition is temporary or permanent, a 404 status code should be used instead.

411 (LENGTH REQUIRED)

The HyperText Transfer Protocol (HTTP) 411 Length Required client error response code indicates that the server refuses to accept the request without a defined Content-Length header.

412 (PRECONDITION FAILED)

The HyperText Transfer Protocol (HTTP) 412 Precondition Failed client error response code indicates that access to the target resource has been denied. This happens with conditional requests on methods other than GET or HEAD when the condition defined by the If-Unmodified-Since or If-None-Match headers is not fulfilled. In that case, the request, usually an upload or a modification of a resource, cannot be made and this error response is sent back.

413 (REQUEST ENTITY TOO LARGE)

The HTTP 413 Payload Too Large response status code indicates that the request entity is larger than limits defined by server; the server might close the connection or return a Retry-After header field.

414 (REQUEST URI TOO LARGE)

The HTTP 414 URI Too Long response status code indicates that the URI requested by the client is longer than the server is willing to interpret.

415 (UNSUPPORTED MEDIA TYPE)

The HTTP 415 Unsupported Media Type client error response code indicates that the server refuses to accept the request because the payload format is in an unsupported format.

The format problem might be due to the request's indicated Content-Type or Content-Encoding, or as a result of inspecting the data directly.

416 (REQUEST RANGE NOT SATISFIABLE)

The HyperText Transfer Protocol (HTTP) 416 Range Not Satisfiable error response code indicates that a server cannot serve the requested ranges. The most likely reason is that the document doesn't contain such ranges, or that the Range header value, though syntactically correct, doesn't make sense.

The 416 response message contains a Content-Range indicating an unsatisfied range (that is a '*') followed by a '/' and the current length of the resource. E.g. Content-Range: bytes */12777

417 (EXPECTATION FAILED)

The HTTP 417 Expectation Failed client error response code indicates that the expectation given in the request's Expect header could not be met.

418 (I'M A TEAPOT)

The HTTP 418 I'm a teapot client error response code indicates that the server refuses to brew coffee because it is, permanently, a teapot. A combined coffee/tea pot that is temporarily out of coffee should instead return 503. This error is a reference to Hyper Text Coffee Pot Control Protocol defined in April Fools' jokes in 1998 and 2014. Some websites use this response for requests they do not wish to handle, such as automated queries.

422 (UNPROCESSABLE ENTITY)

The HyperText Transfer Protocol (HTTP) 422 Unprocessable Entity response status code indicates that the server understands the content type of the request entity, and the syntax of the request entity is correct, but it was unable to process the contained instructions.

425 (TOO EARLY)

The HyperText Transfer Protocol (HTTP) 425 Too Early response status code indicates that the server is unwilling to risk processing a request that might be replayed, which creates the potential for a replay attack.

426 (UPGRADE REQUIRED)

The HTTP 426 Upgrade Required client error response 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.

The server sends an Upgrade header with this response to indicate the required protocol(s).

428 (PRECONDITION REQUIRED)

The HTTP 428 Precondition Required response status code indicates that the server requires the request to be conditional.

Typically, this means that a required precondition header, such as If-Match, is missing.

When a precondition header is not matching the server side state, the response should be 412 Precondition Failed.

429 (TOO MANY REQUESTS)

The HTTP 429 Too Many Requests response status code indicates the user has sent too many requests in a given amount of time ("rate limiting").

A Retry-After header might be included to this response indicating how long to wait before making a new request.

431 (REQUEST HEADER FIELDS TOO LARGE)

The HTTP 431 Request Header Fields Too Large response status code indicates that the server refuses to process the request because the request's HTTP headers are too long. The request may be resubmitted after reducing the size of the request headers.

451 (UNAVAILABLE FOR LEGAL REASONS)

The HyperText Transfer Protocol (HTTP) 451 Unavailable For Legal Reasons client error response code indicates that the user requested a resource that is not available due to legal reasons, such as a web page for which a legal action has been issued.

500 (INTERNAL SERVER ERROR)

The HyperText Transfer Protocol (HTTP) 500 Internal Server Error server error response code indicates that the server encountered an unexpected condition that prevented it from fulfilling the request.

This error response is a generic "catch-all" response. Usually, this indicates the server cannot find a better 5xx error code to response. Sometimes, server administrators log error responses like the 500 status code with more details about the request to prevent the error from happening again in the future.

501 (NOT IMPLEMENTED)

The HyperText Transfer Protocol (HTTP) 501 Not Implemented server error response code means that the server does not support the functionality required to fulfill the request.

501 is the appropriate response when the server does not recognize the request method and is incapable of supporting it for any resource. The only methods that servers are required to support (and therefore that must not return 501) are GET and HEAD.

502 (BAD GATEWAY)

The HyperText Transfer Protocol (HTTP) 502 Bad Gateway server error response code indicates that the server, while acting as a gateway or proxy, received an invalid response from the upstream server.

503 (SERVICE UNAVAILABLE)

The HyperText Transfer Protocol (HTTP) 503 Service Unavailable server error response code indicates that the server is not ready to handle the request.

Common causes are a server that is down for maintenance or that is overloaded. This response should be used for temporary conditions and the Retry-After HTTP header should, if possible, contain the estimated time for the recovery of the service.

504 (GATEWAY TIMEOUT)

The HyperText Transfer Protocol (HTTP) 504 Gateway Timeout server error response code indicates that the server, while acting as a gateway or proxy, did not get a response in time from the upstream server that it needed in order to complete the request.

505 (HTTP VERSION NOT SUPPORTED)

The HyperText Transfer Protocol (HTTP) 505 HTTP Version Not Supported response status code indicates that the HTTP version used in the request is not supported by the server.

506 (VARIANT ALSO NEGOTIATES)

The HyperText Transfer Protocol (HTTP) 506 Variant Also Negotiates response status code may be given in the context of Transparent Content Negotiation. This protocol enables a client to retrieve the best variant of a given resource, where the server supports multiple variants.

The Variant Also Negotiates status code indicates an internal server configuration error in which the chosen variant is itself configured to engage in content negotiation, so is not a proper negotiation endpoint.

507 (INSUFFICIENT STORAGE)

The HyperText Transfer Protocol (HTTP) 507 Insufficient Storage response status code may be given in the context of the Web Distributed Authoring and Versioning (WebDAV) protocol.

It indicates that a method could not be performed because the server cannot store the representation needed to successfully complete the request.

508 (LOOP DETECTED)

The HyperText Transfer Protocol (HTTP) 508 Loop Detected response status code may be given in the context of the Web Distributed Authoring and Versioning (WebDAV) protocol.

It indicates that the server terminated an operation because it encountered an infinite loop while processing a request with "Depth: infinity". This status indicates that the entire operation failed.

510 (NOT EXTENDED)

The HyperText Transfer Protocol (HTTP) 510 Not Extended response status code is sent in the context of the HTTP Extension Framework.

In that specification a client may send a request that contains an extension declaration, that describes the extension to be used. If the server receives such a request, but any described extensions are not supported for the request, then the server responds with the 510 status code.

511
(NETWORK AUTHENTICATION REQUIRED)

The HTTP 511 Network Authentication Required response status code indicates that the client needs to authenticate to gain network access.

This status is not generated by origin servers, but by intercepting proxies that control access to the network. Network operators sometimes require some authentication, acceptance of terms, or other user interaction before granting access (for example in an internet café or at an airport). They often identify clients who have not done so using their Media Access Control (MAC) addresses.

DHAMITH KUMARA

THANK YOU!

I HOPE YOU'VE LEARNED SOMETHING NEW FROM THIS

Follow me on