HTTP response status codes

Prepared by

Shreya C N

HTTP Status Codes

When a client makes a request to an HTTP server — and the server successfully receives the request — the server must notify the client if the request was successfully handled or not.

HTTP accomplishes this with five categories of status codes:

- 100-level (Informational) – server acknowledges a request
- 200-level (Success) – server completed the request as expected
- 300-level (Redirection) – client needs to perform further actions to complete the request
- 400-level (Client error) – client sent an invalid request
- 500-level (Server error) – server failed to fulfil a valid request due to an error with server.

100-level Informational: These codes indicate that the server has received the request and is continuing to process it, but the client should wait for further information. These errors don't require any action from the client. The server is just informing the client that it has received the request and is continuing to process it.

- 100 Continue: The server has received the request headers and the client should proceed with the request.
- 101 Switching Protocols: The server has accepted the request and is switching to a different protocol (e.g., switching from HTTP to WebSocket).
- 102 Processing: The server is still processing the request but has received and understood the client's headers, so the client can continue to send the request body.

200-level Success: These codes indicate that the server has successfully received, understood, and processed the request. These errors indicate that the request was successful, so no action is required from the client.

- 200 OK: The request was successful, and the server is returning the requested data.
- 201 Created: The request was successful, and the server has created a new resource as a result.

- 202 Accepted: The request has been accepted for processing but may not be complete yet.
- 204 No Content: The request was successful, but there is no data to return in the response.
- 206 Partial Content: The server is returning a partial response, indicating that the client should request the remaining data with a new range header.

300-level Redirection: These codes indicate that the requested resource is not available at the requested location and the client should try again with a different location or resource. These errors indicate that the requested resource is not available at the requested location and the client should try again with a different location or resource. The client should follow the redirection and try again with the new location.

- 300 Multiple Choices: The requested resource has multiple options available, and the client should choose one.
- 301 Moved Permanently: The requested resource has permanently moved to a new location, and the client should update their links to point to the new location.
- 302 Found: The requested resource has temporarily moved to a new location, and the client should follow the redirection.
- 304 Not Modified: The requested resource has not been modified since the client last requested it, so the server is returning a cached version.
- 307 Temporary Redirect: The server sends this response to direct the client to get the requested resource at another URI with the same method that was used in the prior request. This has the same semantics as the 302 Found HTTP response code, with the exception that the user agent must not change the HTTP method used: if a POST was used in the first request, a POST must be used in the second request.
- 308 Permanent Redirect: This means that the resource is now permanently located at another URI, specified by the Location: HTTP Response header. This has the same semantics as the 301 Moved Permanently HTTP response code, with the exception that the user agent must not change the HTTP method used: if a POST was used in the first request, a POST must be used in the second request.

400-level Client Error: These codes indicate that there was an error with the client's request. These errors indicate that there was an error with the client's request. The client should review the request and make sure it's valid. Here are some steps to take to resolve some of the most common 4xx errors:

- 400 Bad Request: The server could not understand the client's request due to malformed syntax or missing information. Review the request syntax and make sure all required fields are included.
- 401 Unauthorized: The client must authenticate themselves to access the requested resource. Authenticate yourself with the server by providing valid credentials.
- 402 Payment Required Experimental: This response code is reserved for future use. The initial aim for creating this code was using it for digital payment systems, however this status code is used very rarely, and no standard convention exists.
- 403 Forbidden: The client does not have permission to access the requested resource. Make sure you have permission to access the requested resource.
- 404 Not Found: The requested resource could not be found on the server. Double-check the requested URL to make sure it's correct and the resource exists.
- 405 Method Not Allowed: The requested method is not supported for the requested resource. Make sure you're using the correct HTTP method for the resource.
- 406 Not Acceptable: This response is sent when the web server, after performing server-driven content negotiation, doesn't find any content that conforms to the criteria given by the user agent.
- 407 Proxy Authentication Required: This is like 401 Unauthorized but authentication is needed to be done by a proxy.
- 408 Request Timeout: The server timed out while waiting for the client to complete the request. Check your internet connection and try again.
- 409 Conflict: The request conflicts with the current state of the server. Review the request and make sure it's not conflicting with another request or resource.

- 410 Gone: The requested resource is no longer available on the server. The requested resource is no longer available, so there's no resolution for this error.
- 411 Length Required: Server rejected the request because the Content-Length header field is not defined, and the server requires it.
- 412 Precondition Failed: The client has indicated preconditions in its headers which the server does not meet.
- 413 Payload Too Large: Request entity is larger than limits defined by server. The server might close the connection or return and Retry-After header field.
- 414 URI Too Long: The URI requested by the client is longer than the server is willing to interpret.
- 415 Unsupported Media Type: The media format of the requested data is not supported by the server, so the server is rejecting the request.
- 416 Range Not Satisfiable: The range specified by the Range header field in the request cannot be fulfilled. It's possible that the range is outside the size of the target URI's data.
- 417 Expectation Failed: This response code means the expectation indicated by the Expect request header field cannot be met by the server.
- 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.
- 422 Unprocessable Content (WebDAV): The request was well-formed but was unable to be followed due to semantic errors.
- 423 Locked (WebDAV): The resource that is being accessed is locked.
- 424 Failed Dependency (WebDAV): The request failed due to failure of a previous request.
- 425 Too Early Experimental: Indicates that the server is unwilling to risk processing a request that might be replayed.
- 426 Upgrade Required: 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 in a 426 response to indicate the required protocol(s).

- 428 Precondition Required: The origin server requires the request to be conditional. This response is 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: The client has sent too many requests in each amount of time, and the server is throttling the requests. Wait for some time and try again later or reduce the frequency of requests.

500-level Server Error: These codes indicate that there was an error with the server. This error is usually caused by a server-side issue. Contact the server administrator for assistance.

- 500 Internal Server Error: The server encountered an error while processing the request. This error is usually caused by a server-side issue. Contact the server administrator for assistance.
- 501 Not Implemented: The server does not support the requested method for the requested resource.
- 502 Bad Gateway: The server received an invalid response from an upstream server while processing the request. This error is usually caused by a proxy or gateway server that is misconfigured or down. Contact the server administrator for assistance.
- 503 Service Unavailable: The server is temporarily unavailable, usually due to maintenance or high traffic. The server is temporarily unavailable, usually due to maintenance or high traffic. Wait for some time and try again later.
- 504 Gateway Timeout: The server timed out while waiting for a response from an upstream server. This error is usually caused by a proxy or gateway server that is not responding. Contact the server administrator for assistance.
- 505 HTTP Version Not Supported: The HTTP version used in the request is not supported by the server.
- 506 Variant Also Negotiates: The server has an internal configuration error: the chosen variant resource is configured to engage in transparent content negotiation itself and is therefore not a proper end point in the negotiation process.

- 507 Insufficient Storage (WebDAV): The method could not be performed on the resource because the server is unable to store the representation needed to successfully complete the request.
- 508 Loop Detected (WebDAV): The server detected an infinite loop while processing the request.
- 509 Bandwidth Limit Exceeded: The server has exceeded its bandwidth limit, and the client should try again later. This error occurs when the server has exceeded its bandwidth limit. Wait for some time and try again later.
- 510 Not Extended: Further extensions to the request are required for the server to fulfil it.
- 511 Network Authentication Required: Indicates that the client needs to authenticate to gain network access.