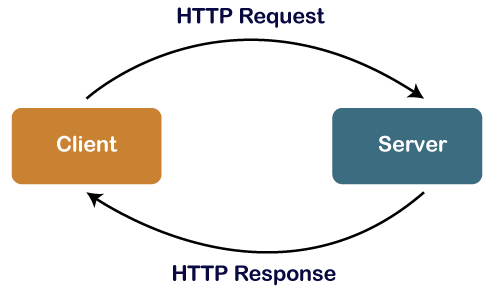
HTTP Status Codes

The HTTP status code determines whether the request made by the client has been successfully completed or not. The server's status codes are provided in response to the client's request to the server. In short, we can say that when the client makes a request, then the HTTP status codes sent by the server allow the clients to know about whether the request was a success, a failure, or something in-between.

**Let's understand the HTTP status code in detail**.

When the browser sends a request to the server, the server responds back with an [HTTP](https://www.javatpoint.com/http) status code of three digits long.



The status codes are divided into five classes. The first digit of the [HTTP](https://www.javatpoint.com/http-tutorial) status code defines the class out of these five classes while the last two digits do not have any role of categorization. The IANA (Internet Assigned Numbers Authority) is an organization that contains the official registry of HTTP status codes. The following are the five classed defined by the IANA standard:

1. 1xx- informational
2. 2xx- Success
3. 3xx- Redirection
4. 4xx- Client error
5. 5xx- Server error

The 100 blocks are informational requests, 200 blocks are success requests, 300 blocks are for redirects, 400 blocks will be for client errors, and 500 blocks will be server errors.

1xx informational response

The codes which are starting with **one** will give some information while the connection is still in progress. This class of status code defines that the request has been received, and processing has been started. An informational response is issued to inform the client that the request processing has been started and waits for a final response. The response consists of only the status line and the optional header fields. It is terminated by an empty line.

**100 continue**

The 100 continue is an informational status response code that indicates till now everything is OK, and the client should continue sending the request body. If the request is finished, then ignore this status code.

A client should send the **Expect: 100- continue** as a header in an initial request so that the server can check the request's header, the client will receive the **100 continue HTTP status code** in response before sending the request body.

**100 switching protocol**

The client has requested the server to switch the protocol. The server sends the 100 switching protocol as a response code to inform the client that it is switching to the protocol as requested by the client, which sent the message that includes the upgrade request header. In short, we can say that server responds to the upgrade request header.

102 processing

The 102 processing status code is sent by the server to inform the client that it has accepted the complete request and processing the request, but still, no response is available.

2xx Success

The 2xx class means that the request made by the client was received, understood, and accepted. Or we can say that the 2xx http status code represents that the Http request was successful.

**200 OK**

Whenever the server sends back the 200 OK status code in response to the request made by the client is accepted and successful. The actual response with 200 OK might be dependent on the HTTP request method. The following methods can be sent in the HTTP request header:

**Get method:** The **Get()** method is used when we want something from the server. For example, we requested some resources from the server; then, the requested resource would be sent in the response body.

**Head method:** The **head()** method is used only when we are requesting the headers. The entity-header fields of the requested resource sent in the response body without any message.

**Post method:** If we send the **post()** method to the server, we are requesting some new resources. An entity describing the result of the action is transmitted in the message body.

**201 Created**

When the server sends back the 201 created as the response code, which says that the request made by the client to the server to create a new resource was successful. The server returns the information regarding the location of the newly created resource in the header field. In short, we can say that the 201 Created response is used to provide the URI in the location header field of the newly created resource.

**202 Accepted**

The server sends 202 accepted in response to inform the client that we have accepted the request and will do the request processing later. This response is used when there is heavy computational processing required or request needs to be processed in the near future. For example, a client sends a request to server for a NEFT where NEFT is used to transfer the money from one account to another account in another bank. NEFT request is processed three times a day, i.e., one in the morning, second in the afternoon, and third in the evening. In such cases, a 202 accepted response is sent that says your request for transfer money is accepted, but I will do it later.

**203 Non-Authoritative Information**

The server uses another response code, i.e., 203 Non-Authoritative Information, which lets the client know that the proxy server is sitting between the client and the server. The proxy may or may not change the header information.

**204 No Content**

The 204 No Content is used when the client's request is successful, but the server has nothing to send in the response body. Instead of simply saying 204, the server says 204 that your request is successful. In this response, the message body is empty.

**205 Reset Content**

Sometimes the client also gets the 205 Reset Content [HTTP](https://www.javatpoint.com/computer-network-http) response, and this response is used in cases when the server wants client viewport to be reset. For example, the client creates a form through which we can create a category, and we submit it, the server sends back the 205 response. This response means that the client needs to reset the input category to blank or to reset the input category for the name of the category that was earlier used.

**206 Partial Content**

The client can also receive the **206 partial content response,** and this response means that the server has fulfilled the partial GET request, not the complete one. The client can use the **Range Header field** that indicates the desired range, so 206 can navigate to a bigger response. The client can also include the If-Range header field to make the request conditional.

**207 Multi-Status**

The 207 Multi-Status is a new addition in the response code. For example, the client sends a request to the server; then the server prepares a response. To prepare a response, the server needs to connect with three different machines, i.e., database system, file system, and the cache system. Let's suppose that the database system is down, and it sends 500; in such cases, the server cannot say that it was successful or not successful, so it sends back 207 responses. The multi-status code is included in a response body as a text/xml or application/xml.

**208 Already Reported**

The 208 is similar to 207, but it is used only for WebDAV. WebDAV stands for Web Distributed Authoring and Versioning. WebDAV is an extension of Hypertext Transfer Protocol (HTTP) that allows clients to perform remote web content authoring operations on the distributed net.

3xx Redirection

The 3xx block is a class used for redirection. It indicates that the client needs to take some additional measures to complete the request. Here Redirection means URL forwarding is a process of providing more one URL to the page. The [HTTP](https://www.javatpoint.com/http-interview-questions) provides HTTP redirect response for such kind of operations.

**The redirects are mainly divided into three categories:**

* Permanent redirects
* Temporary redirects
* Special redirects

Permanent redirections

The permanent redirections are those redirections in which the new [URL](https://www.javatpoint.com/url-full-form) replaces the existing URL. This means that the original URL is no longer exists and replaced by the new one.

**The following are the status codes of the permanent redirections:**

**301 Moved permanently**

The 301 [HTTP](https://www.javatpoint.com/http-full-form) status code tells the browser that the uri of the resource has been changed permanently to the new uri specified in the 'Location' response header. If we make a new request in the browser, then subsequently browser will not make another request to the original link.

For example,

1. Client request:
2. GET /index.php HTTP/1.1
3. Host: www.example.org
4. Server response:
5. HTTP/1.1 301 Moved Permanently
6. Location: http://www.example.org/index.asp

In the above example, when the user manually types the url, i.e., **www.example.org/index.php**, then the browser will make the call to the **www.example.org/index.asp** as the server responds with a 301 response code having text Moved Permanently and the new location is **www.example.org/index.asp**.

The http response status code 301 Moved Permanently is used for permanent URL redirection, which means that the current links are updated. The new URL is provided in the Location field included with a response. The 301 HTTP status code is also used for upgrading the users from HTTP to HTTPS.

RFC 2616 states that if a 301 status code is received in response to other than GET or HEAD, the client must ask the user before redirecting. In other words, we can say that the 301 response status code does not inform the browsers to automatically redirect the POST method request to the GET method request. But some browsers mistakenly redirect the POST method request to the GET method request.

**308 Permanent Redirect**

The 308 status code came in the HTTP standard in April 2015, as mentioned in the RFC7538 document specification for the 308 status code. The 308 is an HTTP response status code, which is similar to the 301 status code, which means that the requested resource has been replaced by the new uri as specified in the Location header included within the response. The 308 rarely indicates an actual problem, but it mainly occurs due to the changes that occur in the server's behavior or configuration. No other status codes indicate the permanent redirect. The 308 is the only one that indicates the permanent redirect. It also avoids the automatic redirect from the POST method request to the GET method request.

All modern browsers detect the 308 Permanent Response and perform the redirection automatically. The server sends a 308 HTTP status code that includes the location header as a response where the location header defines the requested resource location. For example, if the client makes the POST request, then the webserver will configure to redirect to the different uri.

Temporary Redirections

The temporary redirection means the requested resource has assigned a new uri temporarily specified in the location header.

**The following are the HTTP status codes used for the temporary redirects:**

**302 Found**

The 302 Found informs the browser that the uri of the resource has been changed temporarily. When the client makes a request, the browser redirects it to the new location only. The next time when the request comes, then the browser will call the original uri.

**303 See other**

The server sends the 303 response code to the client to get the requested resource at another **uri with GET request, that's why the name of this code is see other. The 303 HTTP response** code informs the browser to redirect to another page but not to the newly uploaded resources. The response is sent only when the client has made the PUT or POST request. The method that displays the requested page is a GET method.

**307 Temporary Redirect**

The 307 is similar to 302; the only difference is that it tells the browser that the next request should be made with the same verb as the original. For example, the POST request is made to the original link then it should be redirected to the new link by a POST only.

307 is similar to 302 as both the responses tell the client to temporarily redirect to the different of the requested resource. The only difference between the 307 and 302 is that in 307, the client must use the same request again. For example, if the client issues the POST request as an original request, then the client should follow the same POST request in the next request.

Special Redirections

**The special redirections include the following HTTP status codes:**

**300 Multiple choices**

When the server sends 300 multiple choices, HTTP response indicating that the request has multiple responses. The [browser](https://www.javatpoint.com/browsers) can choose either of them. There is no standard way of choosing among these multiple responses.

**304 Not Modified**

The server generates the 304 Not Modified response only when the file is not changed since it was last accessed. This response basically increases the speed of the user's browsing experience. If the page that the user is accessing is not modified, then the client will show the cached data which is stored locally to the user, so that the client does not make the request to the webserver for the file. The 304 response is sent in response to the client's conditional validation request to indicate that the client's copy is available in a cache.

When the client makes a Conditional Validation request, then the client sends the Last-Modified date of its copy with the help of If-Modified-Since header and the cached copy's ETag identifier with the help of If-None-match identifier. The server checks these headers to find whether the latest file is available in the cache or not. If it is the latest, then the server sends back the 304 HTTP response. In this case, the client can use the file from the cache instead sending the request to the server for the file. If the server finds an outdated file in a cache, it then sends back the 200 OK response and a new response body.

4xx Client error

The server sends the 4xx HTTP response status codes to the client when some errors occur in the request made by the client. This response indicates that the browser has sent a wrong request with an error that cannot be processed by the server. To get the proper response, the client needs to send the correct request again.

**The following are the status codes used in 4xx:**

**400 Invalid Request**

If the webserver is not able to serve the requested resource because of the wrong syntax, then the server sends back the 400 HTTP invalid request in response.

**401 Unauthorized Request**

When some webpage requires authentication, and the user tries to access the authenticated resource, then the server sends the 401 Unauthorized Request in response. For example, the webpage is authenticated by the user id and password.

**402 Payment Required**

The 402 payment required HTTP response indicates that the client is required to make an online payment to process the request. This response status code is not a standard client error and reserved for future use.

**403 Forbidden**

The server sends the 403 Forbidden HTTP response when the client sends the correct request, but the server cannot serve it. The 403 is different from 402 as in 402, the user should be authorized to access the web page, whereas, in 403, there is no role of authentication. For example, 403 is sent when some authorized user tries to access the restricted page.

**404 Not Found**

The server sends the 404 Not Found error when the requested resource is not available on the server. The reason behind this error could be that the site owner has removed the URL, or the user has mistyped the URL.

**405 Method not allowed**

The 405 Method not allowed response code is received when the server knows the request method, but the requested resource does not support it. In this case, the server generates the ALLOW header field that contains all the supported methods by the target resource.

**406 Not Acceptable Request**

The 406 Not Acceptable Request HTTP response is generated when the client makes a request in a different format. The reason could be the different language or encoding method used in a request.

5xx status codes

The error codes that come under the 5xx range are sent to the client when some problems occur with a server. When the client makes a request for some website, the browser sends a request to the website's server, and if the website's server is not able to serve the request, then the 5xx error code is sent to the client. These errors mainly when the server is facing some problem or unable to perform a request.

The 5xx error messages represent the server-side error messages in which the website's server is unsuccessful in performing a request. The server-side error does not mean that problem is coming on a website, computer, or an internet connection.

There are various 5xx status codes available so that the specific problem can be identified:

**500 internet server error**

The 500 internet server error status code occurs when the server cannot determine the problem then stops responding to the request. This error can occur due to the incorrect server's configuration. To correct this error, the website owner needs to check the server's configuration and contact the web hosting company to fix it.

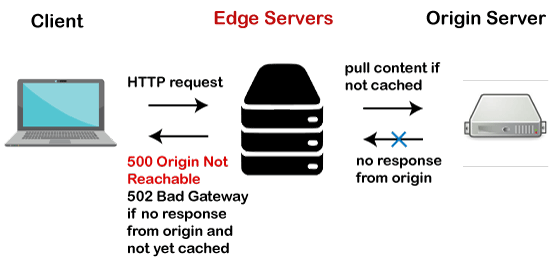
**501 not implemented**

The 501 Not implemented HTTP error response code indicates that the server cannot serve the requested resource as the method mentioned in the request is not supported.

This error status code also sends the Retry-After header that tells the client when to recheck the functionality. The methods that the server support are GET and HEAD. If the server recognizes the method specified in the request, but it will not support it. In this case, 405 Not Allowed response is sent to the client.

**502 Bad Gateway**

The 502 Bad Gateway response code means that the proxy server does not get the origin or upstream server's response. If the edge server receives a 502 Bad Gateway from the origin server, then it sends back the 500 Origin Not Reachable response to the client.



**503 Service Unavailable**

The 503 Service Unavailable response status code is sent when other requests overload the server, or the server is under maintenance. The difference between the 500 and 503 is that in 500, something goes wrong in the server that prevents it from handling the requests, whereas, in 503, the server is working fine and it is also able to serve the request but has opted to send the 503 response.

**The websites can use the following ways to represent this error:**

1. Http/1.1 Service Unavailable
2. 503 Error
3. 503 Service Temporarily Unavailable
4. 503 Service Unavailable
5. Service Unavailable - DNS Failure
6. HTTP Error 503
7. HTTP 503
8. Error 503 Service Unavailable

**504 Gateway Timeout**

The 504 Gateway timeout error occurs when the server does not get the response from another upstream server required to complete the request. When the webserver tries to load a page but didn't get the response from the second server for the required information, then the webserver uses 504 as a response code.

**505 HTTP version is not supported**

The 505 HTTP version is not supported response code is sent if the server does not support the HTTP version specified in the request.

**507 Insufficient space**

507 Insufficient space is an HTTP response status code introduced by the WebDAV specification. It is used to inform the client that the request could not proceed as no space is available on a disk.

**510 Extensions are missing**

When the client sends a request, then it also sends the extension, which is to be used. If the server does not support the extension specified in the request, then the server sends the 510 as a response code.