



HTTP response status codes

Every HTTP transaction has a status code sent back by the server to define how the server handled the transaction. HTTP response status codes indicate whether a specific HTTP request has been successfully completed or not.

👉 All HTTP response status codes are separated into five classes or categories.

1. **Informational responses** (100 – 199)
2. **Successful responses** (200 – 299)
3. **Redirects** (300 – 399)
4. **Client errors** (400 – 499)
5. **Server errors** (500 – 599)

1. Informational Responses

The request was received and continuing process

1. 100 → Continue

This interim response indicates that everything so far is OK and that the client should continue the request, or ignore the response if the request is already finished.

2. 101 → Switching Protocol

The requester has asked the server to switch protocols and the server has agreed to do so.

3. 102 → Processing(WebDAV)

This indicates that the server has received and is processing the request, but no response is available yet.

4. 103 → Early Hints.

This status code is primarily intended to be used to return some response headers before the final HTTP message.

2. Successful Responses

*This class of **status codes** indicates the action requested by the client was received, understood, and accepted*

1. 200 → OK

The request had been succeeded.

2. 201 → Created

After the request has succeeded, A new resource has been created as a result then the result is sent to the client typically after the POST/PUT request.

3. 202 → Accepted

The request has been received but not yet acted upon.

4. 203 → Non-Authentication Information

This response code means the returned meta-information is not exactly the same as is available from the origin server but is collected from a local or a third-party copy.

5. 204 → No Content

There is no content to send for this request, but the headers may be useful.

6. 205 → Reset Content

Ask the user-agent to reset the document which sent this request.

7. 206 → Partial Content

The server is delivering only part of the resource (byte serving) due to a range header sent by the client.

8. 207 → Multi-Status

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

There are more status codes which come under the successful response types. Please look at them [here](#).

3. Redirection Messages

This class of status codes indicates that further action needs to be taken in order to complete the request

1. 300 → Multi-Choice

The request has more than one possible response

2. 301 → Moved Permanently

This and all future requests should be directed to the given URL or The URL of the requested resource has been changed permanently.

3. 302 → Found

The URL of the requested resource has been changed temporarily.

4. 303 → See Other

The response to the request can be found under another URI using the GET method.

5. 304 → Not Modified

This is used for caching purposes. It tells the client that the response has not been modified, so the client can continue to use the same cached version of the response.

There are more status codes which come under the redirection messages types. Please look at them [here](#).

4. Client Error Responses

This class of status codes indicates the request contains bad syntax or cannot be fulfilled

1. 400 → Bad Request

The server is not able to understand the request due to invalid syntax in Code.

2. 401 → Unauthorized

The client must authenticate itself to get the requested response from the server otherwise this will be considered as an unauthorized request.

3. 402 → Payment Required

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.

4. 403 → Forbidden

The request made by the client contains valid data and was understood by the server, but the server is refusing action.

5. 404 → Not Found

The server can not find the requested resource but may be available in the future. In the browser, this means the URL is not recognized.

6. 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 a PUT request on a read-only resource.

There are more status codes that come under the client errors responses types. Please look at them [here](#).

5. Server Error Responses

This class of status codes means that the server failed to fulfill an apparently valid request.

1. 500 → Internal Server Error

The server found an issue in code and it doesn't know how to handle it.

2. 501 → Not Implemented

The server either does not recognize the request method, or it lacks the ability to fulfill the request.

3. 502 → Bad Gateway

The server, while working as a gateway to get a response needed to handle the request, got an invalid response.

4. 503 → Service Unavailable

The server is not ready to handle the request. Common causes are a server that is down for maintenance or that is overloaded.

5. 504 → Gateway Timeout

This error response is given when the server is acting as a gateway and cannot get a response in time

There are more status codes that come under the client errors responses types. Please look at them [here](#).

I hope this article would help you with the basic understating of the HTTP Response Status Code. If you have any queries please let me know in comments.

For more post like this follow us <https://medium.com/rtkal>

Distributed Systems

Backend Development

Api Development

Rest Api

Http Request

[About](#) [Help](#) [Terms](#) [Privacy](#)

Get the Medium app