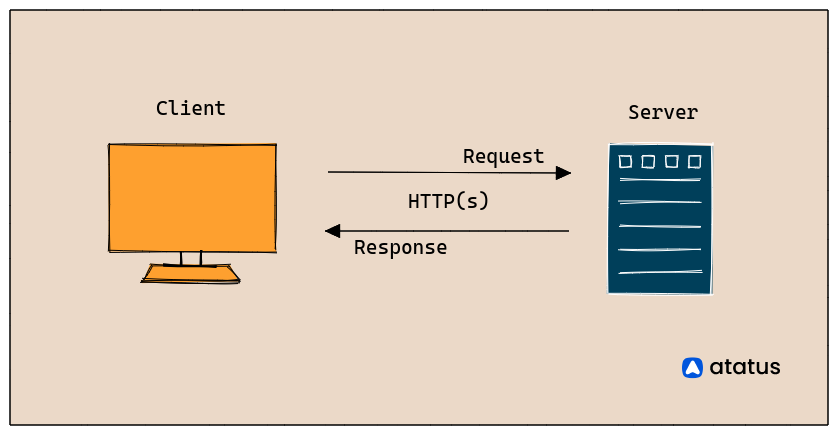
**HTTP Status Codes**

HTTP status codes are three-digit numbers that indicate the outcome of an HTTP request made by a client to a server.

Responses are grouped in five classes:

* Informational responses (100 – 199)
* Successful responses (200 – 299)
* Redirection messages (300 – 399)
* Client error responses (400 – 499)
* Server error responses (500 – 599)



**CREATE**

The HTTPs POST method is typically used to implement the create action. These endpoints are used to create new resources or access tokens in RESTful APIs.

**POST Method Status Codes:**

201 Created

The request was successful, and a new resource has been created on the server.

Use Case: Successful creation of a new resource.

202 Accepted

It is commonly utilized for asynchronous processing. The client is informed by this code that although the request was valid, processing will take some time to complete.

400 Bad Request

The server could not understand the request due to invalid syntax or missing parameters.

Use Case: The request is malformed or missing required data.

403 Forbidden

The server understood the request but refuses to fulfill it. Authorization won't help, and the client should not repeat the request.

Use Case: The client does not have permission to perform the action.

422 Unprocessable Entity

The request is well-formed but semantically incorrect. The server understands the request but cannot process it.

Use Case: The request contains invalid data.

**READ**

The read action uses the HTTPs GET method to implement it and is used to retrieve resource representations.

**GET Method Status Codes:**

200 OK

The request was successful, and the server has returned the requested resource.

Use Case: Successful retrieval of data.

206 Partial Content

When the server fulfills only a part of the client's request. This status code indicates that the server has successfully served a partial version of the requested resource, typically due to a range request.

307 Temporary Redirect

If a resource's URL is subject to change in the future, clients should always request the most recent URL before accessing the actual one.

308 Permanent Redirect

This instructs the client to stop using the current URL and instead use a different one to access the resource. It is useful when we have several endpoints for a single resource but do not want to implement reading from every one of them.

404 Not Found

The requested resource could not be found on the server.

Use Case: The requested URL or resource does not exist.

401 Unauthorized

The client's request lacks valid authentication credentials or has provided incorrect credentials.

Use Case: Authentication is required for access.

**UPDATE**

Using the HTTP PUT or PATCH methods, an update can be implemented. The quantity of data the client must send to the backend differs.

The operation that enables you to change already-existing data is UPDATE. In other words, data editing.

**PUT Method Status Codes:**

200 OK

For the majority of use cases, this is the most suitable code.

202 Accepted

This code may be applied if the update is carried out asynchronously. An URL to the updated resource or a URL to check whether the update was successful should be included. It may also contain an estimation of the update's duration.

204 No Content

The request was successful, and the server has fulfilled the request but has no response to return.

Use Case: Successful update of a resource.

400 Bad Request

The server could not understand the request due to invalid syntax or missing parameters.

Use Case: The request is malformed or missing required data.

404 Not Found

The requested resource could not be found on the server.

Use Case: The resource to be updated does not exist.

**DELETE**

The HTTP DELETE method can be used to implement the delete action.

**DELETE Method Status Codes:**

200 OK

Some people believe that any delete function should return the deleted element so that the response body can contain a representation of the deleted element.

202 Accepted

It could be appropriate to return this code along with some information or a URL to alert the client when it will be erased if the deletion is asynchronous and takes some time, as is the case with distributed systems.

204 No Content

The request was successful, and the server has fulfilled the request but has no response to return. In other words, The status code that best describes this situation. It's preferable to minimize bandwidth, inform the client that the deletion has been completed, and not return any response body (as the resource has been deleted).

Use Case: Successful deletion of a resource.

404 Not Found

The requested resource could not be found on the server.

Use Case: The resource to be deleted does not exist.

401 Unauthorized

The client's request lacks valid authentication credentials or has provided incorrect credentials.

Use Case: Authentication is required for access.

**Redirection Messages:**

300 Multiple Choices:

The request has multiple possible responses, and the client should choose one. Usually, this involves HTML links to the possible choices.

301 Moved Permanently:

The requested resource has been permanently moved to a new URL, and future requests should use the new URL.

302 Found:

The requested resource has been temporarily moved to a different URL. Future requests may still use the original URL.

303 See Other:

The server is directing the client to retrieve the requested resource at another URI, typically with a GET request.

304 Not Modified:

Used for caching; it indicates that the response has not changed since the client's last request, so the cached version can be used.

306 Unused (Reserved):

Reserved for future use and not currently used in HTTP.

**Client Error Responses:**

400 Bad Request:

The server cannot process the request due to client error, such as malformed syntax.

401 Unauthorized:

Authentication is required, and the client must provide credentials.

403 Forbidden:

The client lacks permission to access the requested resource, and authentication won't help.

405 Method Not Allowed:

The server knows the requested method (e.g., POST, DELETE) but does not support it for the requested resource.

406 Not Acceptable:

The server cannot produce a response matching the list of acceptable values in the request's headers.

407 Proxy Authentication Required:

Authentication is required for the proxy server.

408 Request Timeout:

The server closes an idle connection or due to timeout, indicating that the client should close the connection.

413 Payload Too Large:

The request entity is larger than the server allows.

414 URI Too Long:

The URI is longer than the server can interpret.

415 Unsupported Media Type:

The requested media type is not supported by the server.

**Server Error Responses:**

500 Internal Server Error:

The server encountered an unknown situation or error.

501 Not Implemented:

The server does not support the request method.

502 Bad Gateway:

The server, acting as a gateway, received an invalid response from an upstream server.

503 Service Unavailable:

The server is temporarily unable to handle the request due to maintenance or overloading. A Retry-After header may indicate when it will be available.

504 Gateway Timeout:

The server, acting as a gateway, did not receive a timely response from an upstream server.

505 HTTP Version Not Supported:

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

511 Network Authentication Required:

The client must authenticate to gain network access.

**References:**

1. <https://developer.mozilla.org/en-US/docs/Web/HTTP/Status>
2. <https://www.atatus.com/blog/http-status-code-to-use-for-crud-operation/>