HTTP Status Codes in ASP.NET Core Web API

**HTTP Status Code in ASP.NET Core Web API**

We will discuss **HTTP Status Code in ASP.NET Core Web API**. Returning the response with a proper status code is the backbone of any restful Web APIs. Now, it is time to learn how can we format the response with the proper response code as per our business requirement.

**HTTP Status Codes:**

The HyperText Transport Protocol status code is one of the important components of HTTP Response. The Status code is issued from the server and they give information about the response. Whenever we get any response from the server, in that Response, we must have one HTTP Status code. All the HTTP Status codes are divided into five categories. They are as follows. Here, XX will represent the actual number.

**1XX**: Informational Response (Example: 100, 101, 102, etc.)

**2XX**: Successful, whenever you get 2XX as the response code, it means the request is successful. For example, we get 200 HTTP Status Code for the success of a GET request, 201 if a new resource has been successfully created. 204 status code is also for success but in return, it does not return anything just like if the client has performed a delete operation and in return doesn’t really expect something back.

**3XX**: 3XX HTTP status codes are basically used for redirection. Whenever you get 3XX as the response code, it means it is re-directional. for example, to tell a client that the requested resource like page, the image has been moved to another location.

**4XX**: 4XX HTTP status codes are meant to state errors or Client Error. Whenever you get 4XX as the response code, it means there is some problem with your request. For example, status code 400 means Bad Request, 401 is Unauthorized that is invalid authentication credentials or details have been provided by the client, 403 HTTP Status code means that authentication is a success, but the user is not authorized. 404 HTTP Status code means the requested resource is not available.

**5XX**: 5XX HTTP status codes are meant for Server Error. Whenever you get 5XX as the response code, it means there is some problem in the server. Internal Server Error exception is very common, which contains code 500. This error means that there is some unexpected error on the server and the client cannot do anything about it.

**Frequently used HTTP Status Codes in ASP.NET Core Web API:**

The following are some of the frequently used Status codes.

**100**: 100 means Continue. The HTTP 100 Continue informational status response code indicates that everything so far is OK and that the client should continue with the request or ignore it if it is already finished.

**200**: 200 means OK. The HTTP 200 OK success status response code indicates that the request has succeeded. If you are searching for some data and you got the data properly. That means the request is successful and, in that case, you will get 200 OK as the HTTP status code.

**201**: 201 means a new resource created. The HTTP 201 Created success status response code indicates that the request has succeeded and has led to the creation of a resource. The new resource is effectively created before this response is sent back and the new resource is returned in the body of the message, its location being either the URL of the request or the content of the Location header. If you are adding successfully a new resource by using the HTTP Post method, then in that case you will get 201 as the Status code.

**204**: 204 means No Content. The HTTP 204 No Content success status response code indicates that a request has succeeded, but that the client doesn’t need to navigate away from its current page. If the server processed the request successfully and it is not returning any content, then in that case you will get a 204-response status code.

**301**: 301 means Moved Permanently. If you are getting 301 as a status code from the server, it means the resource you are looking for is moved permanently to the URL given by the Location headers.

**302**: 302 means Found. If you are getting 302 as a status code from the server, it means the resource you are looking for is moved temporarily to the URL given by the Location headers.

**400**: 400 means Bad Request. If you are getting 400 as the status code from the server, then the issue is with the client request. If the request contains some wrong data such as malformed request syntax, invalid request message framing, or deceptive request routing, then we will get this 400 Bad Request status code.

**401**: 401 means Unauthorized. If you are trying to access the resource for which you don’t have access (Invalid authentication credentials), then you will get a 401 unauthorized status code from the server.

**404**: 404 means Not Found. If you are looking for a resource that does not exist, then you will get this 404 Not Found status code from the server. Links that lead to a 404 page are often called broken or dead links.

**405**: 405 means Method Not Allowed. The 405 Method Not Allowed response status code indicates that the request method is known by the server but is not supported by the target resource. For example, we have one method which is a POST method in the server and we trying to access that method from the client using GET Verb, then, in that case, you will get a 405-status code.

**500**: 500 means Internal Server Error. If there is some error in the server, then you will get a 500 Internal Server Error status code.

**503**: 503 means Service Unavailable. The 503 Service Unavailable server error response code indicates that the server is not ready to handle the request. If the server is down for maintenance or the server is overloaded then in that case, you will get the 503 Service Unavailable Status code.

**504**: 504 means Gateway Timeout. The 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 is needed in order to complete the request.

**Why HTTP Status Codes are Important?**

If we want to consume any Restful API, then we will send an HTTP Request and in return, we will get the response and the response include data as well as an HTTP Status code. The HTTP Status codes are important because they tell the client (client means who initiate the request, for example, Web, Android, iOS, Postman, IoT, Fiddler, etc) about what exactly happened to the request. If you send a wrong HTTP Status code, then that will confuse the client i.e. the consumer of the API.

The client should know that its request has been taken care of or not, and if the response is not as expected, then the Status Code should tell the client where the problem is? Whether the problem is at the Client level or at the API level.

Suppose there is a situation where the client gets the response with the HTTP status code as 200, but at the API level, there is some problem or issue. In that case, as the client gets 200 HTTP Status code, so the client will get a false assumption of everything being fine, whereas that won’t be the case.

So, if there is something wrong at the API level or there are some errors that occurred on the server, the HTTP status code 500 should be sent to the client so that the client knows there is something wrong with the request being sent. This is the reason why sending proper HTTP Response code from Restful APIs is important.