AJAX - Status Codes

In AJAX, XMLHttpRequest supports various properties and methods to perform different types of operations. Among these properties and methods status property/attribute is a status code that specifies the overall status of the data request sent by the XMLHttpRequest object. Or we can say that the status code is a three-digit number which represent the result of the request sent by the XMLHttpRequest object like the request was successful, run into an error, or redirected, etc.

So the syntax of the status property is −

Format

```
if(XMLHttpRequestObjectName.status == 200){
   // Body
}
```

Here we can access a status property or attribute using the XMLHttpRequest object. If the status code is equal to 200, then the code inside the body will execute.

Status Codes

The status codes that HTTP status returned are as follows −

Successful

Status	Message	Description
200	OK	If the request is OK.
201	Created	When the request is complete and a new resource is created.
202	Accepted	When the request is accepted by the server.
204	No Content	When there is no data in the response body.
205	Reset Content	For additional inputs the browser clears the form used for transaction.
206	Partial Content	When the server returns the partial data of the specified

Redirection

Status	Message	Description
300	Multiple Choices	It is used to represent a link list. So that user can select any one link and go to that location. It allows only five locations.
301	Moved Permanently	When the requested page is moved to the new URL.
302	Found	When the requested page is found in a different URL.
304	Not modified	URL is not modified.

Client Error

Status	Message	Description
400	Bad Request	The server cannot fulfil the request because the request was malformed or has invalid syntax.
401	Unauthorised	The request needs authentication and the user does not provide valid credentials.
403	Forbidden	The server understood the request but does not fulfil it.
404	Not Found	The requested page is not found.
405	Method Not Allowed	The method through which the request is made is not supported by the page.
406	Not Acceptable	The response generated by the server cannot be accepted by the client.
408	Request Timeout	Server timeout
409	Conflict	The request does not fulfil due to a conflict in the request.
410	Gone	The requested page is not available.
417	Exception Failed	The server does not match the requirement of the Expect request header field.

Server Error

Status	Message	Description

500	Internal Server Error	When the server encounter error while processing the request
501	Not Implemented	When the server does not recognise the request method or lacks of ability to fulfil the request
502	Bad Gateway	When the server acts like a gateway and recovers an invalid response from another server(upstream)
503	Service Unavailable	When the server is not available or down
504	Gateway Timeout	When the server acts like a gateway and does not receive a response from the other server(upstream) on time.
505	HTTP Version Not Supported	When the server does not support the version of the HTTP protocol.
511	Network Authentication Required	When the client needs to authenticate to gain access to the network.

Flow Chart

In the below code, we retrieve the data from the server. So we create a function named as showDoc(). Now we call this function by clicking on the "Click Here" button. This function will create a new XHR object using XMLHttpRequest() constructor. Then it creates a call-back function which will handle the request. Then it calls the open() function of the XHR object to initialise the request with HTTP GET method and the URL of the server. Finally, it calls send() function to send the request to the server.

So when the server responds to the request the "onreadystatechange" property calls the callback function with the current state of XMLHttpRequest object. If the status is 200 then that means the request is successful, so it displays the result on the screen and writes a message in the console log. If the status is 404, then that means the server encountered an error. So we got an error message in the console log.

Example

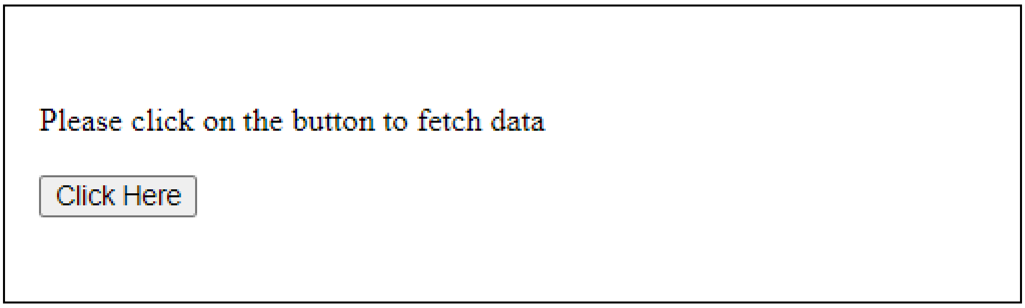
</> Open Compiler

```
<!DOCTYPE html>
<html>
<body>
<script>
   function ShowDoc() {
      // Creating XMLHttpRequest object
      var myhttp = new XMLHttpRequest();

      // Creating call back function
      myhttp.onreadystatechange = function() {
         // Checking the status of the response
         // This will proceed when the response is successful
         if (this.status == 200){
            console.log("Found the requested data")
            document.getElementById("example").innerHTML = this.responseText;
         }
         // This will proceed when the error is found
         else if(this.status == 404){
            console.log("Found error");
         }
      };
      // Open the given file
      myhttp.open("GET", "https://jsonplaceholder.typicode.com/todos/3", true);

      // Sending the request to the server
      myhttp.send();
   }
</script>
<p>Please click on the button to fetch data</p>
<button type="button" onclick="ShowDoc()">Click Here</button>
<div id="example"></div>
</body>
</html>
```

Output



Please click on the button to fetch data

Conclusion

So these are the status codes used by the XMLHttpRequest. These status codes represent the status of the request. According to these status codes, we can perform actions on the request. Now in the next article, we will learn about how errors are handled by the XMLHttpRequest.