# AJAX - XMLHttpRequest

In AJAX, XMLHttpRequest plays a very important role. XMLHttpRequest is used to exchange data to or from the web server in the background while the user/client working in the foreground and then update the part of the web page with the received data without reloading the whole page.

We can also say that XMLHttpRequest (XHR) can be used by various web browser scripting languages like JavaScript, JScript, VBScript, etc., to exchange XML data to or from the web server with the help of HTTP. Apart from XML, XMLHttpRequest can also fetch data in various formats like JSON, etc. It creates an asynchronous connection between the client side and the server side.

## Syntax

```
variableName = new XMLHttpRequest()
```

Where using a new keyword along with XMLHttpRequest() constructor we can be able to create a new XMLHttpRequest object. This object must be created before calling the open() function to initialise it before calling send() function to send the request to the web server.

## XMLHttpRequest Object Methods

XMLHttpRequest object has the following methods −

| Sr.No. | Method Name & Description |
|--------|---------------------------|
| 1 | **new XMLHttpRequest()**<br>It is used to create an XMLHttpRequest() object |
| 2 | **getAllResponseHeaders()**<br>It is used to get the header information |
| 3 | **getResponseHeader()**<br>It is used to get the specific header information |
| 4 | **open(method, url, async, user, psw)**<br>It is used to initialise the request parameters.<br>Here,<br>method: request type GET or POST or Other types<br>url: file location |

| | async: for the asynchronous set to true or for synchronous set to false<br>user: for optional user name<br>psw: for optional password |
|---|---|
| 5 | **send()**<br>It is used to send requests to the web server. It is generally used for GET requests. |
| 6 | **send(string)**<br>It is used to send requests to the server. It is generally used for POST requests. |
| 7 | **setRequestHeader()**<br>It is used to add key/value pair to the header |

Explore our latest online courses and learn new skills at your own pace. Enroll and become a certified expert to boost your career.

## XMLHttpRequest Object Properties

XMLHttpRequest object has the following properties −

| Sr.No. | Property Name & Description |
|---|---|
| 1 | **onreadystatechange**<br>Set the callback function which handles request state changes. |
| 2 | **readyState**<br>It is used to hold the status of XMLHttpRequest. It has the following values −<br><br>● It represents the request is not initialise<br>● It represents the server connection established<br>● It represents the request received<br>● It represents the request is in processing<br>● It represents the request finished and the response is ready |
| 3 | **responseText**<br>It is used to return the response data as a string |
| 4 | **responseXML**<br>It is used to return the response data as XML data |

| | | |
|---|---|---|
| 5 | **Status**<br>It is used to return the status number of a request. For example −<br><br>• 200: for OK<br>• 403: for Forbidden<br>• 404: for Not Found | |
| 6 | **StatusText**<br>It is used to return the status text. For example, OK, Not Found, etc. | |

## Usage of XMLHttpRequest

After understanding the basic syntax, methods, and properties of XMLHttpRequest now we learn how to use XMLHttpRequest in real life. So to use XMLHttpRequest in your program first we need to follow the following major steps −

**Step 1** − Create an object of XMLHttpRequest

```
var variableName = new XMLHttpRequest()
```

**Step 2** − After creating XMLHttpRequest an object, we now have to define a callback function which will trigger after getting a response from the web server.

```
XMLHttpRequestObjectName.onreadystatechange = function(){
    // Callback function body
}
XMLHttpRequestObjectName.open(method, url, async)
XMLHttpRequestObjectName.send()
```

**Step 3** − Now we use open() and send() functions to send a request to the web server.

Now lets us understand the working of XMLHttpRequest with the help of the following example−
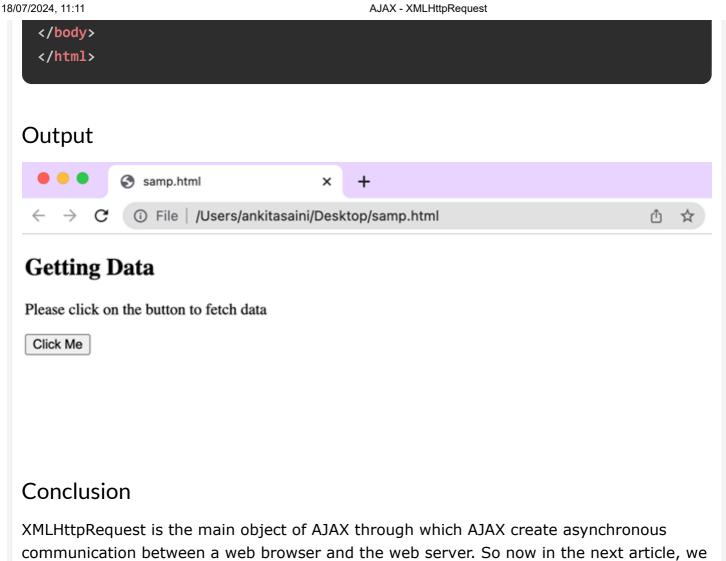
## Example

In the below example, we are going to fetch data from the server. To fetch the data from the server we will click on the "Click Me" button. So when we click on the "Click Me" button, the displayDoc() function is called. Inside the displayDoc() function, we create an XMLHttpRequest object. Then we create a call-back function to handle the server response. Then we call the open() method of the XHR object to initialise the request with HTTP GET method and the server URL which is

"https://jsonplaceholder.typicode.com/todos". Then we call send() function to send the request.

So when the server responds to the request, the "onreadystatechange" property calls the callback function with the current state of XMLHttpRequest object. If the "ready state" property is set to 4(that means the request is completed) and the "status" property is set to 200(that means the successful response), then the response data is extracted from the "responseText" property and display the HTML document with the help of "innerHTML" property of the sample element.

If we error is found during the request then the else statement present in the callback function will execute. So this is how we can fetch data from the server.

```
</>                                                                    Open Compiler

<!DOCTYPE html>
<html>
<body>
<script>
    function displayDoc() {
        // Creating XMLHttpRequest object
        var myObj = new XMLHttpRequest();

        // Creating a callback function
        myObj.onreadystatechange = function() {
            if (this.readyState == 4 && this.status == 200) {
                document.getElementById("sample").innerHTML = this.responseText;
            }else{
                console.log("Error Found")
            }
        };
        // Open the given file
        myObj.open("GET", "https://jsonplaceholder.typicode.com/todos", true);

        // Sending the request to the server
        myObj.send();
    }
</script>
<div id="sample">
    <h2>Getting Data</h2>
    <p>Please click on the button to fetch data</p>
    <button type="button" onclick="displayDoc()">Click Me</button>
</div>
```

```
  </body>
</html>
```

## Output

**Getting Data**

Please click on the button to fetch data

Click Me

## Conclusion

XMLHttpRequest is the main object of AJAX through which AJAX create asynchronous communication between a web browser and the web server. So now in the next article, we will learn how to send a request using an XMLHttpRequest object.