

AJAX - Submitting Forms

AJAX is the most popular web technique which is used by almost all web developers to create dynamic web applications. It uses web browsers' in-built XMLHttpRequest object to send and receive data asynchronously to or from the web server in the background without refreshing or affecting the web page. We can also use AJAX to submit forms very easily.

So to submit the form using AJAX we need to follow the following steps –

Step 1 – Create an XMLHttpRequest object using XMLHttpRequest () constructor –

```
var xmlhttp = new XMLHttpRequest();
```

Step 2 – Create a variable(also known as form element) which contains all the keys and value pairs present in the form with the help of the document.querySelector() method.

```
const FormElement = document.querySelector("mForm")
```

Here if you have multiple forms, then you can define forms with their ids.

Step 3 – FormData object using FormData constructor and pass the above created FormElement into it. It means that the FormData object is initialised with the key-value pairs.

```
const myForm = new FormData(FormElement)
```

Step 4 – Create a call-back function which will be executed when the server responds to the request. This function is defined inside the onreadystatechange property of the XHR object.

```
xmlhttp.onreadystatechange = function() {  
    // Body  
}
```

Here the responseText property will return the response of the server as a JavaScript string which we will further use in our web page to display the message.

```
document.getElementById("responseElement").innerHTML = this.responseText;
```

Step 5 – Now we use the open() function. Inside the open() function we pass a POST request along with the URL to which we have to post our form data.

```
zhttp.open("POST", url, async)
```

Step 6 – Finally we use send() function to send a request to the server along with the FormData object.

```
zhttp.send(myForm);
```

So the complete example is as follows –

Example

Here in the above code, we create a simple HTML form to collect data from the user and then submit the form data using JavaScript with XMLHttpRequest.

So when the user clicks on the "Submit Record" button sendFormData() function is called. The sendFormData() function first creates a new XHR object. Then create a form element which stores all the key-value pairs from the HTML form. Then it is a new FormData object and passes the form element into the object. Next, it set up a call-back function which handles the response from the server. This function is triggered when the value of the readyState property = 4 and the value of the Status property = 201. Finally, it calls the open() method and initialises it with the HTTP POST method with the URL of the server and at last it calls send() method to send the FormData request to the server.

So when the response comes from the server the call-back function shows the result and prints the message on the console log.

[Open Compiler](#)

```
<!DOCTYPE html>
<html>
<body>
<script>
    function sendFormData() {
        // Creating XMLHttpRequest object
        var zhttp = new XMLHttpRequest();
        const mFormEle = document.querySelector("#mForm")
        // Creating FormData object
        const myForm = new FormData(mFormEle);
        // Creating call back function to handle the response
        zhttp.onreadystatechange = function() {
            if (this.readyState == 4 && this.status == 201) {
                document.getElementById("example").innerHTML = this.responseText;
                console.log("Form Data Posted Successfully")
            }
        }
    }

```

```
    }  
    };  
    // Post/Add form data on the server  
    zhttp.open("POST", "https://jsonplaceholder.typicode.com/todos", true);  
    // Sending the request to the server  
    zhttp.send(new FormData(mFormEle));  
  }  
</script>  
<!--Creating simple form-->  
<form id = "mForm">  
  <h2>Enter the requested Data</h2>  
  <label for="Utitle">Title</label>  
  <input id="Utitle" type="text" name="title"><br>  
  
  <label for="UId">UserId</label>  
  <input id="UId" type="number" name="UserID"><br>  
  
  <label for="Ubody">Body</label>  
  <input id="Ubody" type="text" name="body"><br>  
  
  <label for="Uage">Age</label>  
  <input id="Uage" type="number" name="age"><br>  
  
  <button type="button" onclick="sendFormData()">Submit Record</button>  
</form>  
<div id="example"></div>  
</body>  
</html>
```

Output

The screenshot displays a web browser window with the address bar showing 'mExample.html' and the file path '/Users/ankitasaini/Desktop/mExample.html'. The page title is 'Enter the requested Data'. The form contains the following fields and values:

- Title: Priya
- UserId: 23
- Body: Priya lives in Mumbai
- Age: 22

A 'Submit Record' button is present. Below the button, the JSON response is shown: `{ "id": 201 }`.

The browser's developer tools are open, showing the Network tab. A single request named 'todos' is selected. The request details are as follows:

- Name: todos
- Method: POST
- Status: 200 OK
- Size: 740 B
- Headers: Content-Type: application/json, Accept: */*
- Payload: `{ "title": "Priya", "userId": 23, "body": "Priya lives in Mumbai", "age": 22 }`
- Response: `{ "id": 201 }`

The Console tab shows a message: 'Form Data Posted Successfully'.

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

Conclusion

So this is how AJAX submit form using XMLHttpRequest. It is the most commonly used feature of AJAX. Now in the next article, we will see how AJAX upload files to the server.