

AJAX - Send JSON Data

AJAX is asynchronous Javascript and XML. It is a combination of web technologies that are used to develop a dynamic web application that sends and retrieves data from the server in the background without reloading the whole page.

JSON(JavaScript Object Notation) is a format in which we store data and can transfer data from one system to another computer system. It is easy to understand and language-independent. AJAX can transport any kind of data either in JSON or any plain text. So in this article, we will learn how to send JSON Data using AJAX.

Send JSON Data

To send JSON data using AJAX follow the following steps –

Step 1 – Create a new XMLHttpRequest instance.

Step 2 – Set the request method that is open() method and URL.

Step 3 – Set the request header to specify the data format. Here the content-type header is set to "application/json" which indicates that the data is going to send in JSON format.

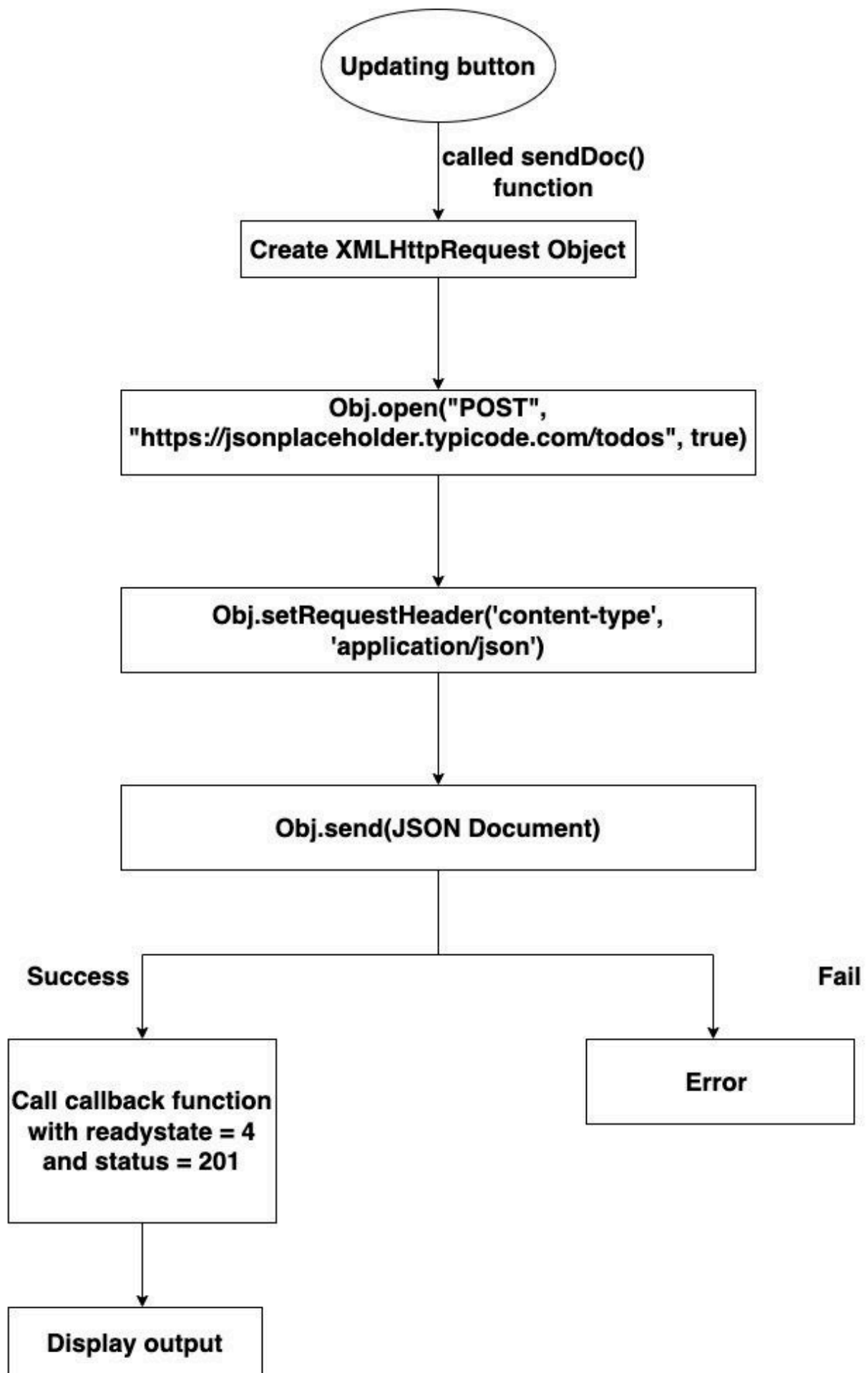
Step 4 – Create a call-back function that handles the response.

Step 5 – Write JSON data.

Step 6 – Convert the JSON data into strings using JSON.stringify() method.

Step 7 – Now send the request using send() method along with the JSON data as the request body.

Following is the flow chart which shows the working of the below code –



Example



Open Compiler

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

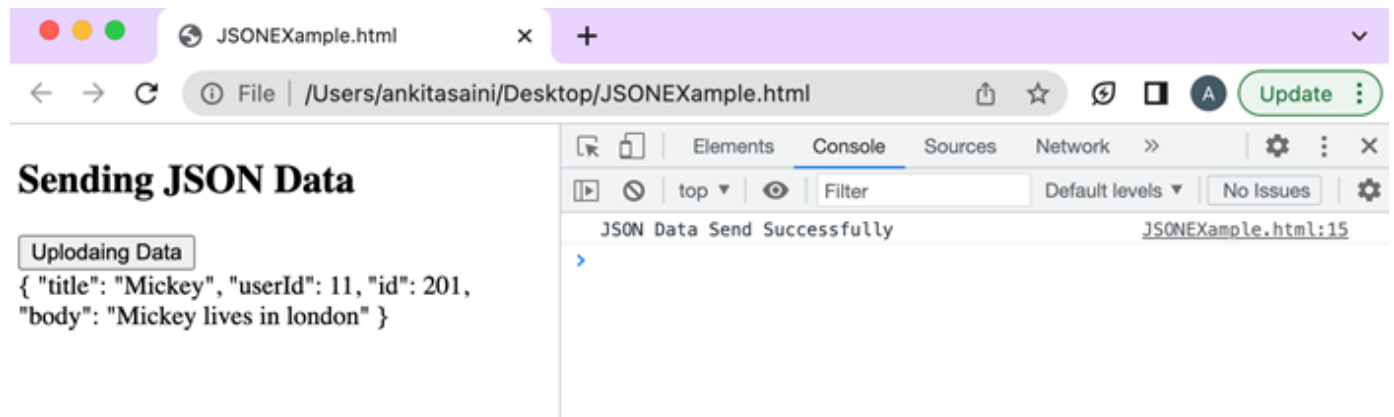
    // Creating call back function
    qhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 201) {
            document.getElementById("sample").innerHTML = this.responseText;
            console.log("JSON Data Send Successfully")
        }
    };
    // Open the given file
    qhttp.open("POST", "https://jsonplaceholder.typicode.com/todos", true);

    // Setting HTTP request header
    qhttp.setRequestHeader('Content-type', 'application/json')

    // Sending the JSON data to the server
    qhttp.send(JSON.stringify({
        "title": "Mickey",
        "userId": 11,
        "id": 21,
        "body": "Mickey lives in london"
    }));
}
</script>
<h2>Sending JSON Data</h2>
<button type="button" onclick="sendDoc()">Uploading Data</button>
<div id="sample"></div>
</body>
</html>
```

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

Output



Here in the above example, we send the following JSON document to the server at the given URL using the POST method –

```
{
  "title": "Mickey",
  "userId": 11,
  "id": 21,
  "body": "Mickey lives in london"
}
```

So when we click on the "Updating Data" button, the sendDoc() function is called. This function creates an XMLHttpRequest object. Then call the open() method of the XHR object to initialise the request with the HTTP POST method and the URL of the server which is ["https://jsonplaceholder.typicode.com/todos"](https://jsonplaceholder.typicode.com/todos). Then call the setRequestHeader() method to set the content type of the request as JSON. After that calls send() function to send the request along with the JSON document. When the server receives the request, it adds the document.

If the update is successful, then the callback function is called with "ready state = 4 (that means the request is completed)" and "status = 201(that means the server is successfully created a new resource)" Then the response from the server is displayed in the HTML file with the help of innerHTML property of the sample element. It also prints a message to the console representing that the JSON data was successfully sent.

Here is the JSON.stringify() method is used to convert JSON documents into a string. It is necessary because XHR requests can only send text data.

Conclusion

So this is how we can send JSON data using XMLHttpRequest. It is the most commonly used data transmission format because it is light in weight and easy to understand. Now in the next article, we will learn how to parse XML objects.