

# AJAX - Send Data Objects

In AJAX, we are allowed to send data objects as a part of an HTTP request from a client to a web server. A data object is an object which contains data in the key-value pair. They are generally represented in JavaScript objects. So in AJAX sending data objects means we are passing structured data to the server for further processing. It can contain form inputs, user inputs, user information or any other information. Not only data objects, but we can also upload and send files from the system using AJAX along with XMLHttpRequest.

Following is the format of the data object –

```
var myDataObject = {  
  "name": "Pinky",  
  "City": "Pune",  
  "Age": 23  
}
```

Now to send this data object using XMLHttpRequest we need to convert the object into a JSON string using `stringify()` method because most of the frameworks support JSON format very easily without any extra effort. The `stringify()` method is a JavaScript in-built function which is used to convert the object or value into JSON string.

## Syntax

```
var myData = JSON.stringify(myDataObject)
```

Here `myDataObject` is the data object which we want to convert into JSON String.

## Example

In the following program, we will send data objects using XMLHttpRequest. So for that, we will create an XMLHttpRequest object then we create a data object which contains the data we want to send. Then we convert the data object into a JSON string using `stringify()` function and set a header to "application/json" to tell the server that the request contains JSON data. Then we send the data object using `send()` function and the response is handled by the callback function.

[Open Compiler](#)

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

        // Creating data object
        var myDataObject = {
            "name": "Monika",
            "City": "Delhi",
            "Age": 32,
            "Contact Number": 33333333
        }
        // Creating call back function
        qhttp.onreadystatechange = function() {
            if (this.readyState == 4 && this.status == 201) {
                document.getElementById("sample").innerHTML = this.responseText;
                console.log("Data object Send Successfully")
            }
        };
        // Open the given file
        qhttp.open("POST",
            "https://jsonplaceholder.typicode.com/todos", true);

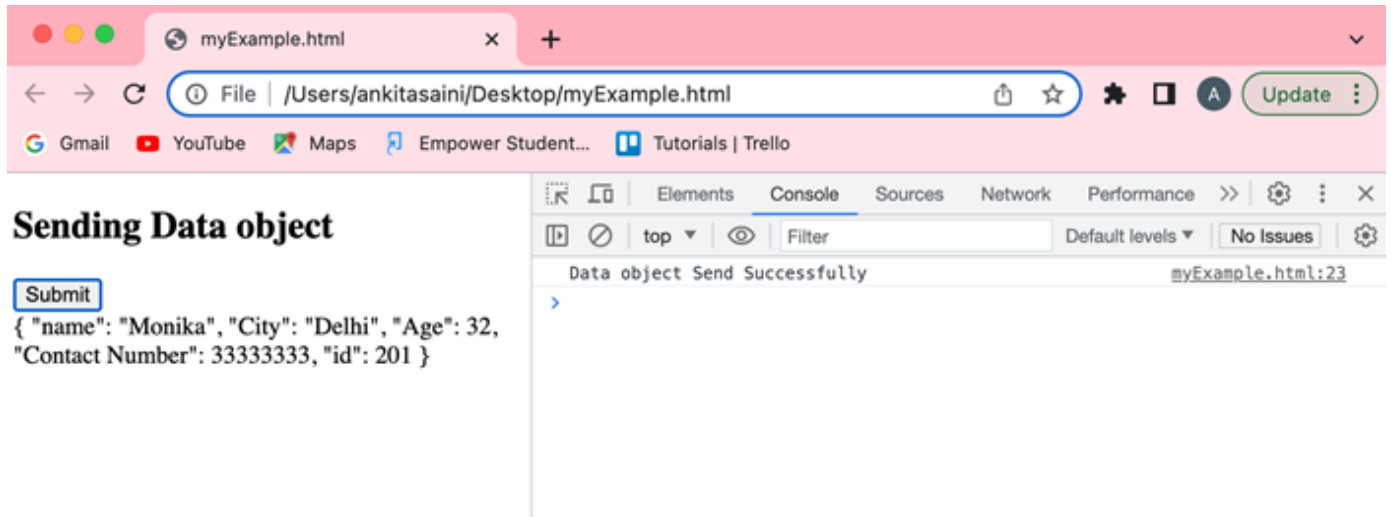
        // Setting HTTP request header
        qhttp.setRequestHeader('Content-type', 'application/json')

        // Converting data object to a string
        var myData = JSON.stringify(myDataObject)

        // Sending the data object to the server
        qhttp.send(myData)
    }
</script>
    <h2>Sending Data object</h2>
    <button type="button" onclick="sendDataObject()">Submit</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



## Conclusion

So this is how we can send data objects to the server and update the response accordingly. It allows us to share information and update data without refreshing the whole page. So in the next article, we will learn how to parse XML Object.