

AJAX - Types of Requests

AJAX is a web technology that is used to create dynamic web pages. It allows web pages to update their content without reloading the whole page. Generally, AJAX supports four types of requests and they are –

- GET request
- POST request
- PUT request
- DELETE request

GET Request

The GET request is used to retrieve data from a server. In this request, the data is sent as a part of the URL that is appended at the end of the request. We can use this request with the `open()` method.

Syntax

```
open(GET, url, true)
```

Where, the `open()` method takes three parameters –

- **GET** – It is used to retrieve data from the server.
- **url** – url represents the file that will be opened on the web server.
- **true** – For asynchronous connection set the value to true. Or for synchronous connection set the value to false. The default value of this parameter is true.

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

Example

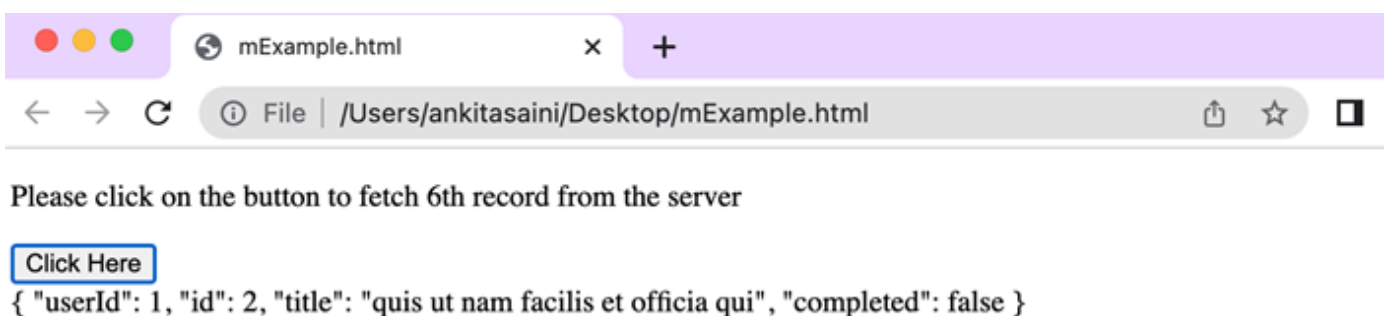
```
</>
```

Open Compiler



```
<!DOCTYPE html>
<html>
<body>
<script>
    function displayRecords() {
        // Creating XMLHttpRequest object
        var xmlhttp = new XMLHttpRequest();
        // Creating call back function
        xmlhttp.onreadystatechange = function() {
            if (this.readyState == 4 && this.status == 200) {
                document.getElementById("example").innerHTML = this.responseText;
            }
        };
        // Open the given file
        xmlhttp.open("GET", "https://jsonplaceholder.typicode.com/todos/6", true);
        // Sending the request to the server
        xmlhttp.send();
    }
</script>
<div id="example">
    <p>Please click on the button to fetch 6th record from the server</p>
    <button type="button" onclick="displayRecords()">Click Here</button>
</div>
</body>
</html>
```

Output



In the above example, we are fetching the 6th record from the server using the GET request `"https://jsonplaceholder.typicode.com/todos/6"` API in XMLHttpRequest. So after

clicking on the button, we will get the 6th record from the server.

POST Request

The POST request is used to send data from a web page to a web server. In this request, the data is sent in the request body that is separated from the URL. We can use this request with the `open()` method.

Syntax

```
open('POST', url, true)
```

Where, the `open()` method takes three parameters –

- **POST** – It is used to send data to the web server.
- **url** – url represents the server(file) location.
- **true** – For asynchronous connection set the value to true. Or for synchronous connection set the value to false. The default value of this parameter is true.

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("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 document to the server
qhttp.send(JSON.stringify({
    "title": "MONGO",
    "userId": 11,
    "id": 21,
    "body": "est rerum tempore"
})));
}
</script>
<h2>Example of POST Request</h2>
<button type="button" onclick="sendDoc()">Post Data</button>
<div id="sample"></div>
</body>
</html>
```

Output

Example of POST Request

Post Data

Here in the above example, we are updating the record with the below-given data using the PUT request.

```
"https://jsonplaceholder.typicode.com/todos/21" API:
{
  "title": "MONGO",
  "userId": 11,
  "id": 21,
  "body": "est rerum tempore"
}
```

DELETE Request

The DELETE request is used to delete data from the web server. In this request, the data to be deleted is sent in the request body and the web server will delete that data from its

storage.

Syntax

```
open('DELETE', url, true)
```

Where, the open() method takes three parameters –

- **DELETE** – It is used to delete data from the web server.
- **url** – It represents the file url which will be opened on the web server. Or we can say server(file) location.
- **true** – For asynchronous connection set the value to true. Or for synchronous connection set the value to false. The default value of this parameter is true.

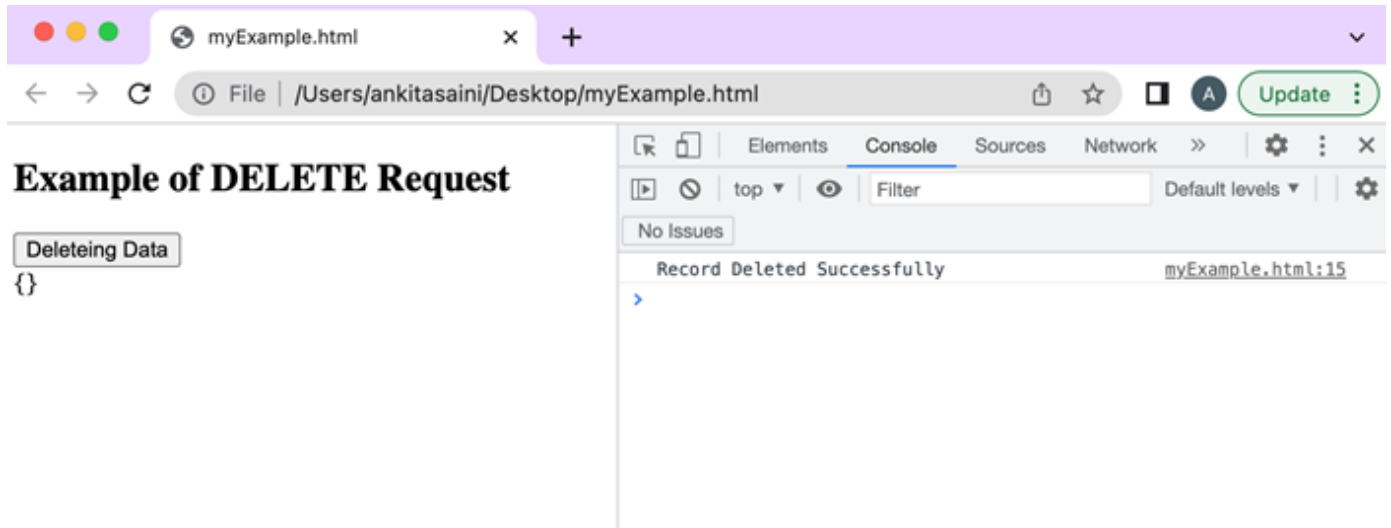
Example

[Open Compiler](#)

```
<!DOCTYPE html>
<html>
<body>
<script>
    function delDoc() {
        // Creating XMLHttpRequest object
        var qhttp = new XMLHttpRequest();
        // Creating call back function
        qhttp.onreadystatechange = function() {
            if (this.readyState == 4 && this.status == 200) {
                document.getElementById("sample").innerHTML = this.responseText;
                console.log("Record Deleted Successfully")
            }
        };
        // Deleting given file
        qhttp.open("DELETE", "https://jsonplaceholder.typicode.com/todos/2", true);
        // Sending the request to the server
        qhttp.send();
    }
</script>
<div id="sample">
    <h2>Example of DELETE Request</h2>
```

```
<button type="button" onclick="delDoc()">Deleteing Data</button>
</div>
</body>
</html>
```

Output



Here in the above example, we delete the record present on Id = 2 using the DELETE request "<https://jsonplaceholder.typicode.com/todos/2>" API.

AJAX also support some other request like OPTIONS, HEAD, and TRACE but they are the least commonly used requests by the AJAX applications. Now in the next article, we will see how AJAX handle responses.