AJAX - File Uploading

AJAX provides a flexible way to create an HTTP request which will upload files to the server. We can use the FormData object to send single or multiple files in the request. Let us discuss this concept with the help of the following examples −

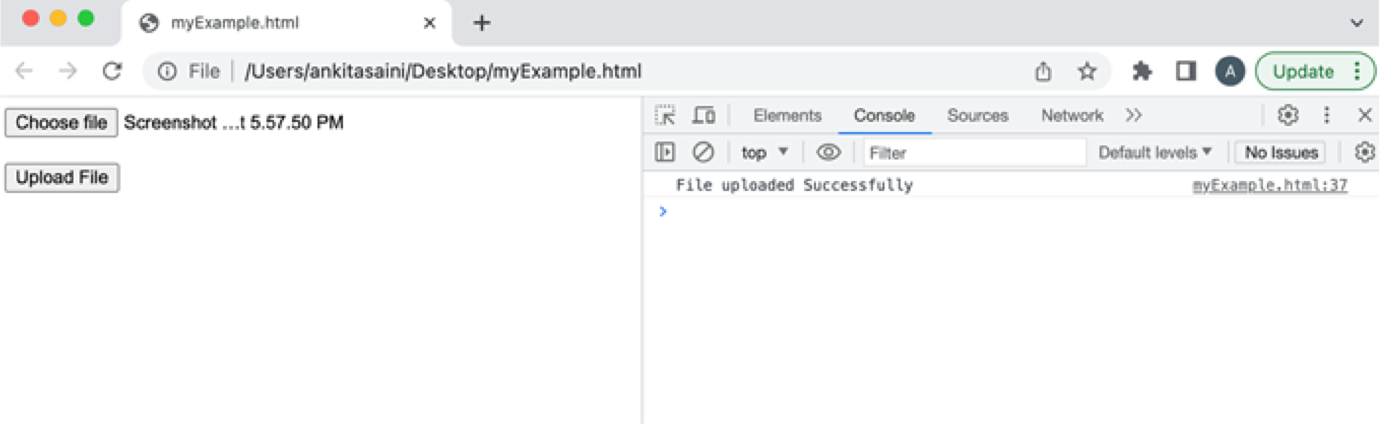
Example − Uploading a Single File

In the following example, we will upload a single file using XMLHttpRequest. So for that first we create a simple form which has a file upload button and the submit button. Now we write JavaScript in which we get the form element and create an event which triggers when we click on the upload file button. In this event, we add the uploaded file to the FormData object and then create an XMLHttpRequest object which will send the file using the FormData object to the server and handle the response returned by the server.

Open Compiler

```
<!DOCTYPE html>
<html>
<body>
<!-- Creating a form to upload a file-->
<form id = "myForm">
   <input type="file" id="file"><br><br>
   <button type="submit">Upload File</button>
</form>
<script>
   document.getElementById('myForm').addEventListener('submit', function(x){
      // Prevent from page refreshing
      x.preventDefault();

      // Select the file from the system
      // Here we are going to upload one file at a time
      const myFile = document.getElementById('file').files[0];

      // Create a FormData to store the file
      const myData = new FormData();
      // Add file in the FormData
      myData.append("newFiles", myFile);
      // Creating XMLHttpRequest object
      var myhttp = new XMLHttpRequest();

      // Callback function to handle the response
      myhttp.onreadystatechange = function(){
         if (myhttp.readyState == 4 && myhttp.status == 200) {
            console.log("File uploaded Successfully")
         }
      };

      // Open the connection with the web server
      myhttp.open("POST", "https://httpbin.org/post", true);

      // Setting headers
      myhttp.setRequestHeader("Content-Type", "multipart/form-data");

      // Sending file to the server
      myhttp.send(myData);
   })
</script>
</body>
</html>
```

Output



Choose file Screenshot ...t 5.57.50 PM

Upload File

```
File uploaded Successfully                    myExample.html:37
```

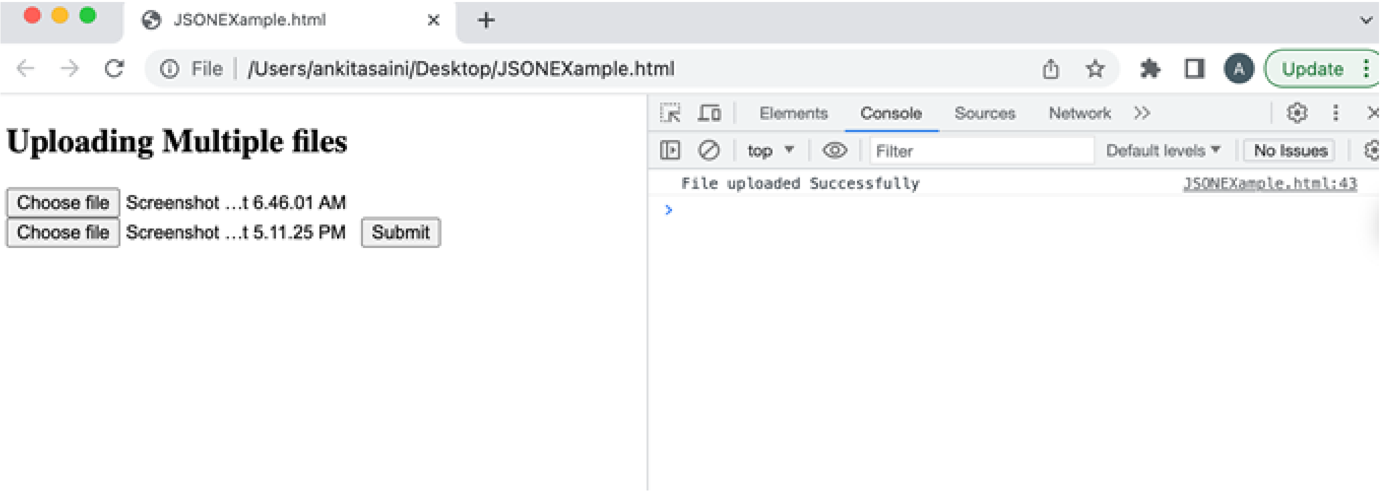
Example − Uploading Multiple Files

In the following example, we will upload multiple files using XMLHttpRequest. Here we select two files from the system in DOM with the attribute of file type. Then we add the input files in an array. Then we create a FormData object and append the input files to the object. Then we create an XMLHttpRequest object which will send the files using the FormData object to the server and handle the response returned by the server.

Open Compiler

```
<!DOCTYPE html>
<html>
<body>
<!-- Creating a form to upload multiple files-->
<h2> Uploading Multiple files</h2>
<input type="file">
<input type="file">
<button>Submit</button>
<script>
   const myButton = document.querySelector('button');
   myButton.addEventListener('click', () => {
      // Get all the input files in DOM with attribute type "file":
      const inputFiles = document.querySelectorAll('input[type="file"]');

      // Add input files in the array
      const myfiles = [];
      inputFiles.forEach((inputFiles) => myfiles.push(inputFiles.files[0]));

      // Creating a FormData
      const myformdata = new FormData();

      // Append files in the FormData object
      for (const [index, file] of myfiles.entries()){
         // It contained reference name, file, set file name
         myformdata.append(`file${index}`, file, file.name);
      }
      // Creating an XMLHttpRequest object
      var myhttp = new XMLHttpRequest();

      // Callback function
      // To handle the response
      myhttp.onreadystatechange = function(){
         if (myhttp.readyState == 4 && myhttp.status == 200) {
            console.log("File uploaded Successfully")
         }
      };
      // Open the connection with the web server
      myhttp.open("POST", "https://httpbin.org/post", true);

      // Setting headers
      myhttp.setRequestHeader("Content-Type", "multipart/form-data");

      // Sending file to the server
      myhttp.send(myformdata);
   })
</script>
</body>
</html>
```

Output



Uploading Multiple files

Choose file Screenshot …t 6.46.01 AM
Choose file Screenshot …t 5.11.25 PM Submit

```
File uploaded Successfully                                    JSONEXample.html:43
```

Conclusion

So this is how we can upload files to the given URL with the help of XMLHttpRequest. Here we can upload any type of file such as jpg, pdf, word, etc and can upload any number of files like one file at a time or multiple files at a time. Now in the next article, we will learn how to create a FormData object using XMLHttpRequest.