

Fetch API - Uploading Files

Fetch API provides a flexible way to create an HTTP request which will upload files to the server. We can use the `fetch()` function along with 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 program, we upload one file at a time using fetch API. Here we use the `FormData` object to store the file and then send it using the `fetch()` function to the given URL including the POST request method and the `FormData` object. After sending the request to the server now we use the `then()` function to handle the response. If we encounter an error, then that error is handled by the `catch()` function.

[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();

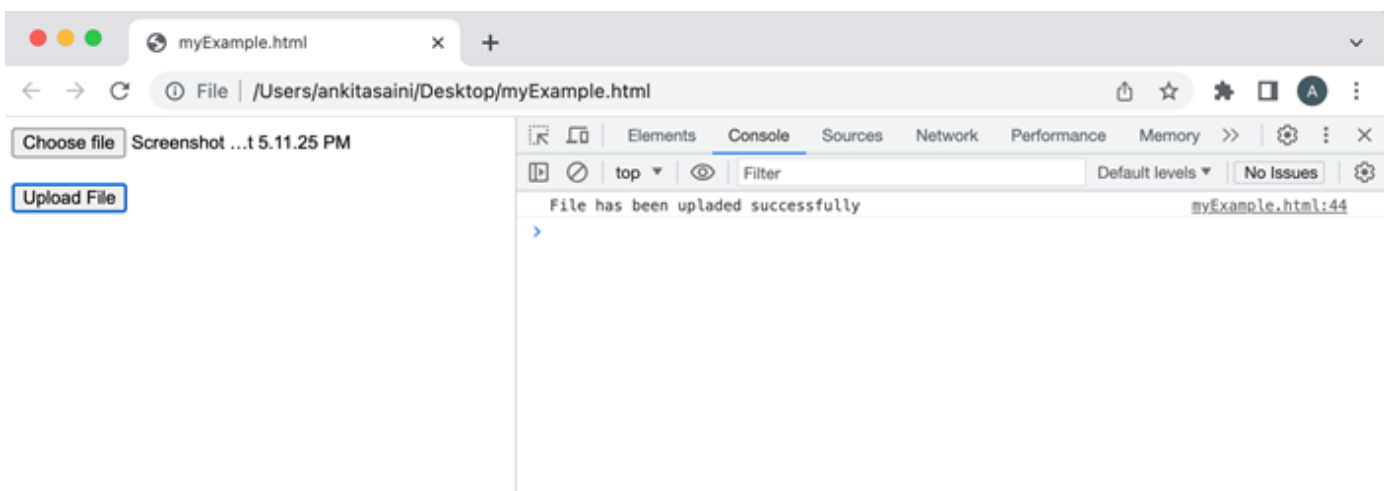
    // Add file in the FormData
    myData.append("newFiles", myFile);
```



```
// Send the file to the given URL
fetch("https://httpbin.org/post", {
  // POST request with Fetch API
  method: "POST",

  // Adding FormData to the request
  body: myData
})
// Converting the response in JSON
// using response.json() function
.then(response => response.json())
.then(finalData => {
  // Handling the response
  console.log("File has been uploaded successfully");
})
.catch(err=>{
  // Handling the error
  console.log("Error Found:", err)
});
})
</script>
</body>
</html>
```

Output



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

Example – Uploading Multiple Files for Single Input

In the following program, we will upload multiple files from the single input using fetch API. Here we add a "multiple" property in the <input> tag to add multiple files. Then we use the FormData object to store multiple files and then send them using the fetch() function to the given URL including the POST request method and the FormData object. After sending the request to the server now we use the then() function to handle the response. If we encounter an error, then that error is handled by the catch() function.

[Open Compiler](#)

```
<!DOCTYPE html>
<html>
<body>
  <!-- Creating a form to upload a file-->
  <h2> Uploading Multiple files</h2>
  <form id = "myForm">
    <p>Maximum number of files should be 2</p>
    <input type="file" id="file" multiple><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 multiple files 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);

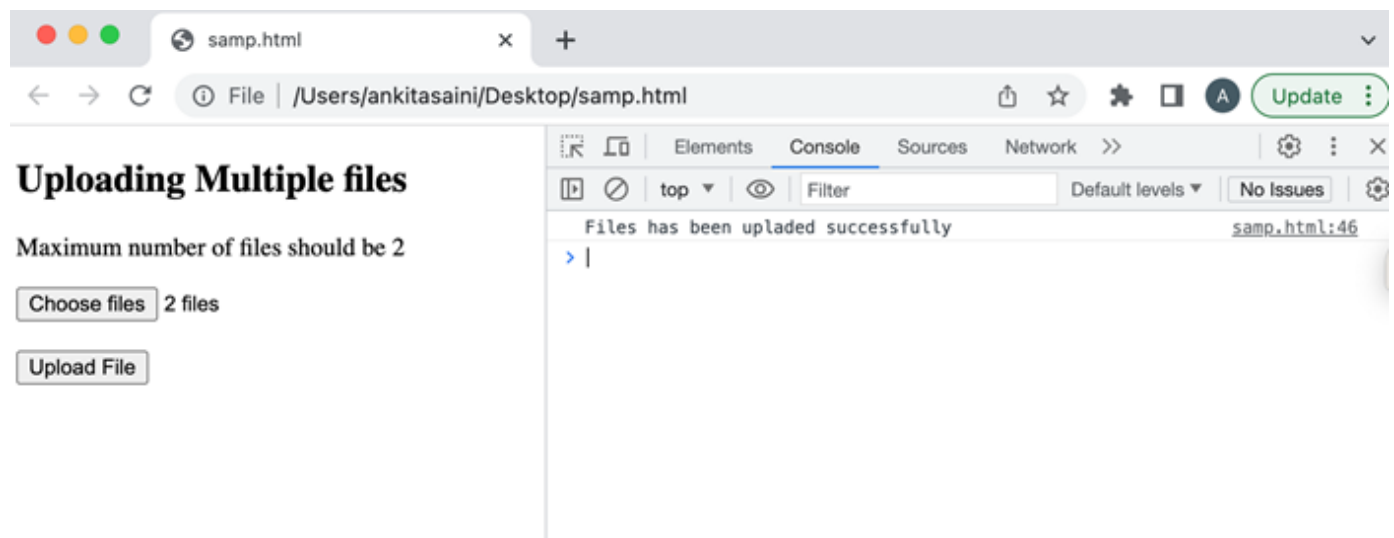
    // Send the file to the given URL
    fetch("https://httpbin.org/post", {
      // POST request with Fetch API
      method: "POST",

      // Adding FormData to the request
```



```
        body: myData
    })
    // Converting the response in JSON
    // using response.json() function
    .then(response => response.json())
    .then(finalData => {
        // Handling the response
        console.log("Files has been uploaded successfully");
    })
    .catch(err=>{
        // Handling the error
        console.log("Error Found:", err)
    });
})
</script>
</body>
</html>
```

Output



Example – Uploading Multiple Files

In the following program, we will upload multiple files using fetch API. 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 send them using the fetch() function to the given URL including the POST request method and the FormData object. After sending the request to the server now we use the then() function to handle the response. If we encounter an error, then that error is handled by the catch() function.



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();

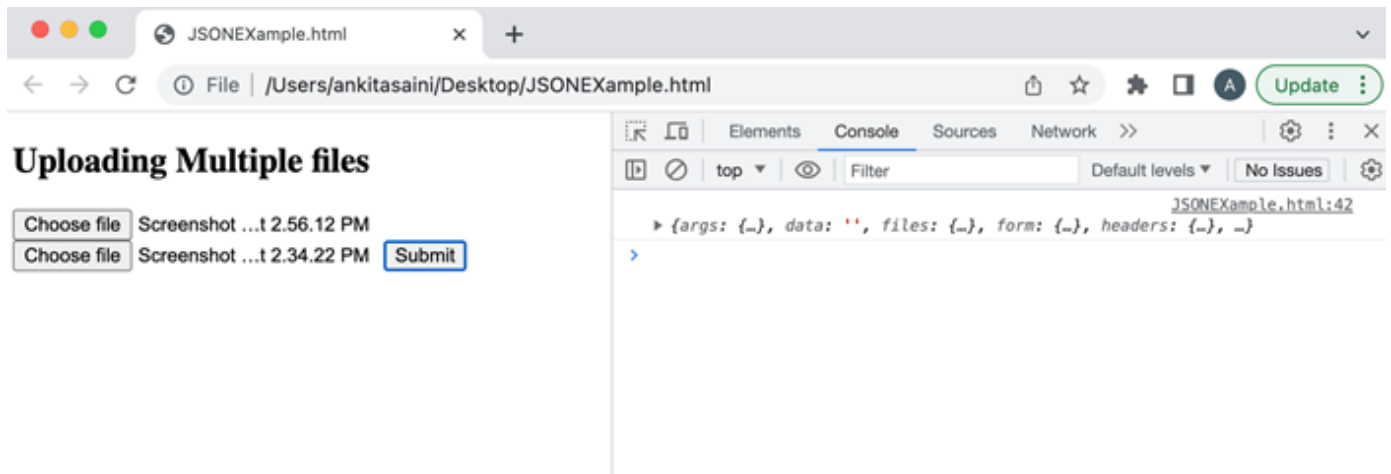
    // Append files in the FormData object
    for (const [index, file] of myfiles.entries()){
      // It contained reference name, file, set file name
      myformdata.append(`file${index}`, file, file.name);
    }
    // Upload the FormData object
    fetch('https://httpbin.org/post', {
      method: "POST",
      body: myformdata,
    })
    // Handle the response
    .then(response => response.json())
    .then(response => console.log(response))

    // Handle the error
    .catch(err => console.log("Error Found:", err))
  })
</script>
```



```
</body>  
</html>
```

Output



Conclusion

So this is how we can upload files to the given URL with the help of fetch() API. Here we can upload any type of file like jpg, pdf, word, etc and 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 the Fetch API handles responses.