

Upload a File in JavaScript

Sep 2, 2020

HTML has a `file input` tag that lets users select one or more files to upload. For example, below is HTML that defines a file input.

```
<input type="file" />
```

Given an `<input type="File">`, you can access the selected file as a `blob` by accessing `input.files[0]`:

```
const input = document.querySelector('input[type="file"]');
const file = input.files[0];

file instanceof File; // true
file instanceof Blob; // true
```

Uploading a File

Once you have a blob, you can upload it using JavaScript's built-in `FormData` class. [Axios](#) supports HTTP POST requests with `FormData`, so uploading a file is easy:

```
const formData = new FormData();
formData.append('myimage.png', file);

// Post the form, just make sure to set the 'Content-Type' header
const res = await axios.post('/upload', formData, {
  headers: {
    'Content-Type': 'multipart/form-data'
  }
});
```

Server-Side Setup

Parsing `FormData` uploads on the server side is tricky, you should use an npm module like [Formidable](#) to help you out. Below is how you can write the corresponding `POST /upload` endpoint to the above `Axios` request.

Note that the below code just returns the file name, it doesn't actually store the file. Once you have the parsed file in Node.js, you would need to either [upload the file to AWS S3](#) or some other storage service, or store it on your server's hard drive using `fs`.

```
app.post('/upload', function(req, res) {
  const form = new formidable.IncomingForm();
  // Parse `req` and upload all associated files
  form.parse(req, function(err, fields, files) {
    if (err) {
      return res.status(400).json({ error: err.message });
    }
    const [firstFileName] = Object.keys(files);

    res.json({ filename: firstFileName });
  });
});
```

For example, here's an `upload` endpoint that uploads the file to a bucket named 'masteringjs-test' in AWS S3:

```
const AWS = require('aws-sdk');

app.post('/upload', function(req, res) {
  const form = new formidable.IncomingForm();
  // Parse `req` and upload all associated files
  form.parse(req, function(err, fields, files) {
    if (err) {
      return res.status(400).json({ error: err.message });
    }

    const [firstKey] = Object.keys(files);
    const upload = {
      Bucket: 'masteringjs-test',
      Body: fs.createReadStream(files[firstKey].path),
      Key: files[firstKey].name
    };
    s3.upload(upload, (err) => {
      if (err) {
        return res.status(400).json({ error: err.message });
      }
      return res.json({ ok: 1 });
    });
  });
});
```

More Fundamentals Tutorials

- [Convert Decimal to Binary](#)

- [How to Use typeof With Functions](#)
- [How to use typeof with Numbers](#)
- [The Seven Standard Primitives in JavaScript](#)
- [How to Convert an Array to a String in JavaScript](#)
- [How to Use Named Parameters in JavaScript](#)
- [How to Use JavaScript's Spread Operator](#)