

Multer

Multer is a middleware that allows you to handle HTTP requests with enctype "multipart/form-data", which is used for file uploads. Here's an example of how to use Multer to upload an image in an Express.js application:

application:

Install the multer module:

```
npm install multer
```

Require the module in your code:

```
const multer = require('multer');
```

```
const multer = require('multer');
```

Set up the multer middleware:

```
const upload = multer({ dest: 'uploads/' });
```

This will configure Multer to store uploaded files in the uploads/ directory.

Use the upload middleware in your route handler:

```
app.post('/upload', upload.single('image'), (req, res) => {  
  
  // req.file is the `image` file  
  
  // req.body will hold the text fields, if there were any  
  
});
```

The upload.single('image') function will process the image field in the form and add the file to the request object, which you can then use in your route handler.

In your HTML form, make sure to set the **enctype** attribute to "multipart/form-data" and include an input field with the type attribute set to "file":

```
<form action="/upload" method="post" enctype="multipart/form-data">  
  
  <input type="file" name="image" />  
  
  <input type="submit" value="Upload" />  
  
</form>
```

This will allow the user to select a file to upload. When the form is submitted, the file will be uploaded to the server and processed by the Multer middleware.

Second one To use Multer on both the server-side and the front-end, you'll need to install it in both your backend code and your frontend code.

On the server-side, you can use Multer as described in the previous example to handle file uploads.

On the front-end, you can use Multer to process file uploads in the browser. Here's an example of how to do this using the Axios library:

Install the multer and axios modules:

npm install multer axios

Require the modules in your code:

```
const multer = require('multer'); const axios = require('axios');
```

Set up the multer middleware:

```
const upload = multer().array('images');
```

This will configure Multer to process an array of files with the field name images.

Use the upload middleware in your route handler:

```
app.post('/upload', upload, (req, res) => { // req.files is an array of files });
```

On the front-end, create an HTML form with the enctype attribute set to "multipart/form-data" and an input field with the type attribute set to "file" and the multiple attribute set to "true":

```
<form action="/upload" method="post" enctype="multipart/form-data"> <input type="file" name="images" multiple /> <input type="submit" value="Upload" /> </form>
```

This will allow the user to select multiple files to upload.

Use Axios to submit the form:

```
const formData = new FormData(); formData.append('images', file1); formData.append('images', file2);  
axios.post('/upload', formData).then(response => { console.log(response.data); });
```

This will send a POST request to the server with the uploaded files in the request body. The Multer middleware on the server will then process the files and add them to the request object, which you can use in your route handler.

Optional

```
const storage = multer.diskStorage({  
  destination: './uploads/',  
  filename: function (req, file, cb) {  
    cb(null, `${new Date().getTime()}-${file.filename}.${file.mimetype.split("/")[1]}`)  
  }  
})  
var upload = multer({ storage: storage })
```