Uploading

Ta sử dụng package multer, là middleware handling multipart/form-data

## Install

$ npm install --save multer

## Setup

const multer = require("multer");

const upload = multer({

    dest: "folder\_name",

    fileFilter: (req, res, cb) => {

        // To reject this file pass `false`, like so:

        cb(null, false)

        // To accept the file pass `true`, like so:

        cb(null, true)

        // You can always pass an error if something goes wrong:

        cb(new Error('I don\'t have a clue!'))

    },

    limits: {

        fieldNameSize: 100, // field's Name size (100 bytes)

        fieldSize: 1024, // field value size (1024 bytes)

        fields: Number.POSITIVE\_INFINITY, // max num of non-file fields

        fileSize: Number.POSITIVE\_INFINITY, // max file size in total

        files: Number.POSITIVE\_INFINITY, // max num of file fields

        parts: Number.POSITIVE\_INFINITY, // num of fields + files

        headerPairs: 2000, // max num of header key:value to parse

    },

    storage: multer.diskStorage({

        destination: function (req, file, cb) {

            cb(null, '/tmp/my-uploads')

        },

        filename: function (req, file, cb) {

            cb(null, file.fieldname + '-' + Date.now())

        }

    })

})

Đã sử dụng dest: ‘folder\_name’ thì không sử dụng storage và ngược lại

## View

<form action="/upload?\_csrf={{csrfToken}}"

          method="post"

          enctype="multipart/form-data">

        <input type="file"

               name="image">

        <button type="submit">Submit</button>

    </form>

\*Lưu ý nếu có csrf thì yêu cầu setup như trên chứ không thể dùng ntn

<input type="hidden" name="\_csrf" value="...">

<!-- MULTIPLE FILES -->

<form action="/uploadmultiple" enctype="multipart/form-data" method="POST">

Select images: <input type="file" name="myFiles" multiple>

<input type="submit" value="Upload your files"/>

</form>

<!—PHOTO -->

<form action="/upload/photo" enctype="multipart/form-data" method="POST">

<input type="file" name="myImage" accept="image/\*" />

<input type="submit" value="Upload Photo"/>

</form>

## No Dest

Nếu không set dest 🡪 ta sẽ chứa file ở Buffer

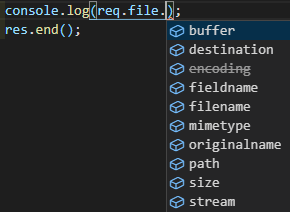
const multer = require("multer");

app.post("/upload", multer().single("image"), (req, res) => {

    console.log(req.file);

    res.end();

});

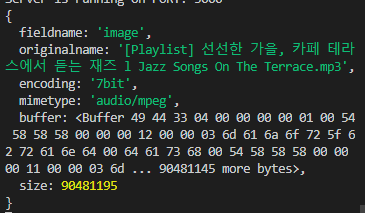


Trong file gồm :

destination : folder đã lưu file

fieldname : name of field in form

path : full path



## With Dest

app.post("/upload", multer({ dest: "files" }).single("image"), (req, res) => {

    console.log(req.file);

    console.log(req.file.path);

    res.end();

});

Khi specify dest 🡪 khi upload file thì buffer sẽ được lưu ở folder đã chỉ định

# Methods

.single(fieldname)

.array(fieldname[, maxCount])

.fields([

  { name: 'avatar', maxCount: 1 },

  { name: 'gallery', maxCount: 8 }

])

.none()

Không cho phép upload file, nếu file upload 🡪 hiện error : “LIMIT\_UNEXPECTED\_FILE”

.any()

Cho phép tất cả và store vào : req.files

Lưu ý : không bao giờ sử dụng multer ở global middleware

# Code Template

var fileStorage = multer.diskStorage({

    destination: async (req, file, cb) => {

        const fs = require("fs");

        try {

            if (!fs.existsSync("./images")) {

                fs.mkdirSync("./images");

            }

            cb(null, "./images");

        } catch (err) {

            console.error(err);

        }

    },

    filename: function (req, file, cb) {

        cb(null, `custom-${file.originalname}`);

        // không sử dụng Date được vì tên của nó có space => sai

    },

});

const fileFilter = (req, file, cb) => {

    if (

        file.mimetype === "image/png" ||

        file.mimetype === "image/jpg" ||

        file.mimetype === "image/jpeg"

    ) {

        cb(null, true); // save file

    } else {

        /\*

            Lưu ý : ở đây chỉ duyệt để không lưu thôi

            chứ không trả lại thông báo ở response

        \*/

        console.log("THÔNG BÁO : KHÔNG SAVE");

        cb(null, false); // không save file

    }

};

const upload = multer({

    storage: fileStorage,

    fileFilter: fileFilter,

});

Single

app.post("/upload", upload.single("image"), (req, res) => {

    if (!req.acceptedfile) {

        const error = new Error("Please upload a file");

        error.httpStatusCode = 400;

        return next(error);

    }

    res.end();

});

## Multiples

app.post("/upload-multi", upload.array("image", 10), (req, res) => {

    if (!req.files) {

        const error = new Error("Please choose files");

        error.httpStatusCode = 400;

        return next(error);

    }

    console.log(req.files);

    res.end();

});

# Error Handling

var multer **=** require('multer')

var upload **=** multer().single('avatar')

app.post('/profile', function (req, res) {

  upload(req, res, function (err) {

**if** (err **instanceof** multer.MulterError) {

*// A Multer error occurred when uploading.*

    } **else** **if** (err) {

*// An unknown error occurred when uploading.*

    }

*// Everything went fine.*

  })

})