



LẬP TRÌNH SERVER CHO ANDROID

BÀI 5: UPLOAD FILE VỚI THƯ VIỆN MULTER

- ⊙ Tìm hiểu về thư viện Multer
- ⊙ Cài đặt và cấu hình thư viện Multer với ExpressJS và Handlebars
- ⊙ Upload file với thư viện Multer



Phần I: Tổng quan thư viện Multer

 Tìm hiểu về thư viện Multer

 Cài đặt và làm quen với thư viện Multer

 Cấu hình thư viện Multer với
ExpressJS và Handlebars

Phần II: Upload file với thư viện Multer

 Các thao tác file

 Giới hạn kích thước khi upload file





- ❑ Là 1 thư viện Express, multer là phần mềm trung gian (middleware) của NodeJS được dùng để xử lý multipart/form-data, được sử dụng chủ yếu để upload file
- ❑ Lưu ý: Multer sẽ không xử lý được nếu form đó không phải là multipart/form-data
- ❑ Thông tin thêm về Multer:
<https://github.com/expressjs/multer>

- ❑ Để tiến hành cài đặt Multer, các bạn thêm dòng lệnh sau: `npm install multer --save`
- ❑ Sau khi chạy xong sẽ được giao diện như hình dưới

```
Administrator@YZFAXCMZING35CL MINGW64 /e/Hoc Tap/NodeJS/slide1
$ npm install multer --save
npm WARN demo@1.0.0 No repository field.

+ multer@1.4.0
added 21 packages in 17.473s
```

- ❑ Các bạn vào file views, tiến hành tạo file mới tên 'upload.hbs' như hình dưới
- ❑ Lưu ý: Nếu các bạn không nhớ cách tạo hbs template và ExpressJS có thể xem ở bài trước

Name	Date modified	Type	Size
 index.hbs	28/09/2018 1:39 CH	HBS File	2 KB
 upload.hbs	03/10/2018 11:08 SA	HBS File	1 KB






- ❑ Sau khi tạo file 'upload.hbs' các bạn tiến hành thêm dòng lệnh như bên dưới

```
<!DOCTYPE html>
<html>
<head>
<title>Upload file</title>
</head>
<body>
<form action="/upload" method="post" enctype="multipart/form-data">
  <input type="file" name="avatar" />
  <input type="submit" value="upload">
</form>
</body>
</html>
```

- ❑ Các bạn nhớ thêm phần
enctype="multipart/form-data" vào nhé
- ❑ Lưu ý: Nếu không thêm vào thì thư viện Multer
sẽ không hiểu

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


- ❑ Giờ chúng ta đã có Template, công việc còn lại là sẽ gọi thư viện Multer trong NodeJS server, để làm được các bạn tiến hành mở file index.js như hình dưới

Name	Date modified	Type	Size
 node_modules	03/10/2018 10:51 SA	File folder	
 views	03/10/2018 11:14 SA	File folder	
 index	03/10/2018 10:50 SA	JScript Script File	2 KB
 package.json	03/10/2018 11:08 SA	JSON File	1 KB
 package-lock.json	03/10/2018 10:51 SA	JSON File	33 KB

- ❑ Các bạn tiến hành gọi thư viện Multer như hình dưới
- ❑ Lưu ý: Các bạn gọi thêm ExpressJS và hbs template để hỗ trợ thư viện Multer nhé. Các bạn có thể xem lại kiến thức cũ để cài đặt và cấu hình hai thư viện (ExpressJS và HBS Template)

```
//Gọi EXPRESSJS  
var express = require('express');  
//Tạo app để cấu hình router,...  
var app = express();  
var multer = require('multer');
```

- ❑ Sau khi gọi thư viện Multer, các bạn tiến hành thêm lệnh như hình dưới

```
var storage = multer.diskStorage({  
  //cb chính là kiểm tra file có thể lưu trữ hay không  
  //cb được dùng để thay đổi đường dẫn, thay đổi tên file,...  
  destination: function (req, file, cb) {  
    //Chúng ta sẽ đưa file vào thư mục uploads  
    cb(null, './uploads');  
  },  
  filename: function (req, file, cb) {  
    //Chúng ta sẽ để nguyên tên file không thay đổi  
    cb(null, file.originalname);  
  }  
});  
var upload = multer({storage:storage});
```

❑ Mỗi tệp chứa thông tin sau:

- ❖ fieldName: Tên trường được chỉ định trong biểu mẫu
- ❖ Originalname: Tên tệp máy tính người dùng
- ❖ Encoding: mã hóa tệp
- ❖ Mimetype: mở file trực tiếp trên trình duyệt
- ❖ Size: kích thước của tệp tính theo bytes
- ❖ DiskStorage
 - Destination: đường dẫn cần lưu
 - Fieldname: tên của tệp bỏ đuôi(giả sử bạn có file .txt thì nó sẽ bỏ đuôi .txt)
 - Path: đường dẫn file
- ❖ MemoryStorage
 - Buffer: bộ đệm

- ❑ Như vậy chúng ta đã cấu hình xong thư viện Multer, nhưng chúng ta vẫn chưa thể Upload file được, các bạn phải thêm 2 phương thức nữa là:
 - ❖ Get(router) để gọi đường dẫn tới template và form
 - ❖ POST: để Upload file
- ❑ Giờ chúng ta tạo router để gọi đến template

```
app.get('/upload', function(req, res) {  
    res.render('upload');  
})
```



DEMO

- Cài đặt và Cấu hình thư viện Multe



- ❑ Sau khi đã có router các bạn tiến hành thêm phương thức POST nữa thì mới Upload file được

```
//Upload.single("tên name trong trường field của các bạn");
app.post('/upload', upload.single("avатар"), function(req, res){
    //Sau khi upload xong NodeJS sẽ trả về thông tin của file
    console.log(req.file);
    //Sau khi Upload xong NodeJS sẽ trả về Client là thành công
    res.send('Thành công');
})
```

- ❑ Các bạn mở terminal lên, thực hiện dòng lệnh sau:
'Nodemon index.js' để tiến hành chạy server

```
Administrator@YZFAXCMZING35CL MINGW64 /e/Hoc Tap/NodeJS/slide1
$ nodemon index.js
[nodemon] 1.17.5
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting `node index.js`
```


- ❑ Chúng ta tiến hành vào trình duyệt, gõ port sau:
- ❑ localhost:3000/upload
- ❑ Chúng ta sẽ được Form Upload như hình dưới

← → ↻ ⓘ localhost:3000/upload

Chọn tệp

Không có tệp nào được chọn

upload

- ❑ Giờ chúng ta tiến hành chọn file và click vào nút Upload như hình dưới

← → ↻ ⓘ localhost:3000/upload

Chọn tệp

30624587_91...06562_n.png

upload

- ☐ Giao diện sau khi các bạn click vào Upload
- ☐ Tại phía người dùng trên trình duyệt



A screenshot of a web browser's address bar. It features navigation icons (back, forward, refresh) on the left and an information icon on the right. The address bar contains the text 'localhost:3000/upload'.

Thành công

- ❑ Giao diện sau khi các bạn click vào Upload
- ❑ Tại giao diện phía Node Server như các bạn có thể thấy Node server sẽ đưa ra từng thông tin file chúng ta mới Upload

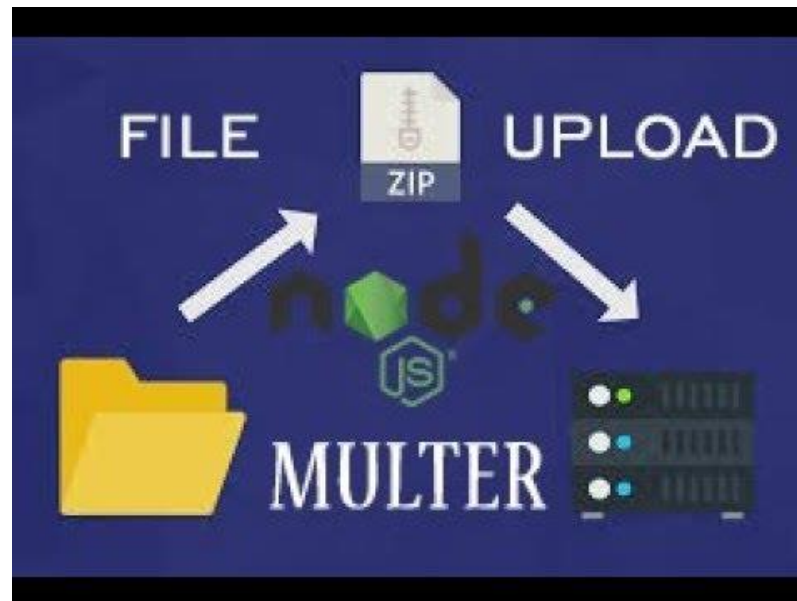
```
Administrator@YZFAXCMZING35CL MINGW64 /e/Hoc Tap/NodeJS/slide1
$ nodemon index.js
[nodemon] 1.17.5
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting 'node index.js'
{ fieldname: 'avatar',
  originalname: '30624587_910355072476499_5777745795040506562_n.png',
  encoding: '7bit',
  mimetype: 'image/png',
  destination: './uploads',
  filename: '30624587_910355072476499_5777745795040506562_n.png',
  path:
    'uploads\\30624587_910355072476499_5777745795040506562_n.png',
  size: 228960 }
```

- ❑ Trong Folder 'uploads' chúng ta sẽ thấy file của chúng ta sau khi upload thành công



bo-hoa-hong-do
-tang-nguoi-yeu

- ❑ Để tránh trường hợp người dùng Upload file quá nặng, có dung lượng lớn. Thư viện Multer có hỗ trợ thuộc tính giúp Website, ứng dụng của các bạn tránh trường hợp đăng tải những file lớn, có dung lượng cao làm ảnh hưởng tới tốc độ trang Web, ứng dụng,...



- ❑ Limits là một đối tượng(Object) được dùng để chỉnh, định giới hạn kích thước khi Upload file.
- ❑ Với thuộc tính Limits, các bạn có thể dễ dàng thao tác, dễ dàng quản lí File, tệp tin đăng tải từ người dùng
- ❑ Limits cung cấp một số hàm cơ bản sau:
fieldNameSize, fieldSize, fields, fileSize, files, parts, headerPairs

NHỮNG HÀM CƠ BẢN CỦA LIMITS

- ❑ fieldNameSize: kích thước tên trường tối đa (name size)
- ❑ fileSize: kích thước giá trị của trường tối đa (value size)
- ❑ fields: số lượng tối đa của trường không phải là tệp
- ❑ fileSize: dùng cho multipart forms, kích thước tối đa của File khi Upload
- ❑ files: dùng cho multipart forms, số lượng trường file tối đa
- ❑ parts: dùng cho multipart forms, số lượng phần tối đa (fields + files)
- ❑ headerPairs: dùng cho multipart forms, số lượng tối đa các cặp tiêu đề => giá trị để phân tích cú pháp (2000)
- ❑ Nguồn: <https://github.com/expressjs/multer>

- ❑ Để sử dụng được Limits chúng ta tiến hành thêm thuộc tính như hình dưới
- ❑ Trong ví dụ bên dưới chúng ta sử dụng thuộc tính nếu file upload lên lớn hơn kích thước đã định sẽ ngăn lại, nếu trường hợp ngược lại sẽ cho upload file lên
- ❑ Tại phần multer() các bạn thêm dòng lệnh như hình dưới
- ❑ Ở đây kích thước đặt là Bytes, fileSize trường hợp này là 1MB

```
//Kích thước file tối đa 1MB
```

```
var upload = multer({storage:storage, limits:{fileSize:1*1024*1024}});
```

THÔNG BÁO TỚI NGƯỜI DÙNG KHI UPLOAD LỖI HOẶC DO KÍCH THƯỚC FILE QUÁ LỚN

- ❑ Tiến hành thêm dòng lệnh bên dưới để thông báo tới người dùng.

```
//Upload.single("tên name trong trường field của các bạn");
var upload = upload.single('avatar');
app.post('/upload',function(req, res){
    upload(req, res, function(err){
        //Nếu Upload bị lỗi
        if(err instanceof multer.MulterError){
            return res.send('Kích thước File lớn hơn 1MB');
        }
        //Nếu Upload không xác định được File
        else{
            return res.send('Tập không xác định');
        }
        //Khi thành công sẽ thông báo người dùng đồng thời in thông tin file
        console.log(req.file);
        res.send('Thành công');
    })
})
```

THÔNG BÁO TỚI NGƯỜI DÙNG KHI UPLOAD LỖI HOẶC DO KÍCH THƯỚC FILE QUÁ LỚN

- ❑ Bây giờ chúng ta tiến hành mở terminal lên chạy dòng lệnh sau: `nodemon index.js`
- ❑ TH1: Khi chúng ta Upload File lớn



hl - Shortcut
Shortcut
1,50 KB



Internet Download Manager
Shortcut
1,01 KB



lab full
Tập tin nén ZIP WinRAR
5,88 MB



Lab8
Tập tin nén WinRAR
6,47 MB



Microsoft Excel 2010

THÔNG BÁO TỚI NGƯỜI DÙNG KHI UPLOAD LỖI HOẶC DO KÍCH THƯỚC FILE QUÁ LỚN

- ❑ Khi File lớn hơn 1MB thì thư viện Multer sẽ không cho Upload file này nữa, và đồng thời thông báo cho người dùng biết là kích thước File lớn, không thể Upload được



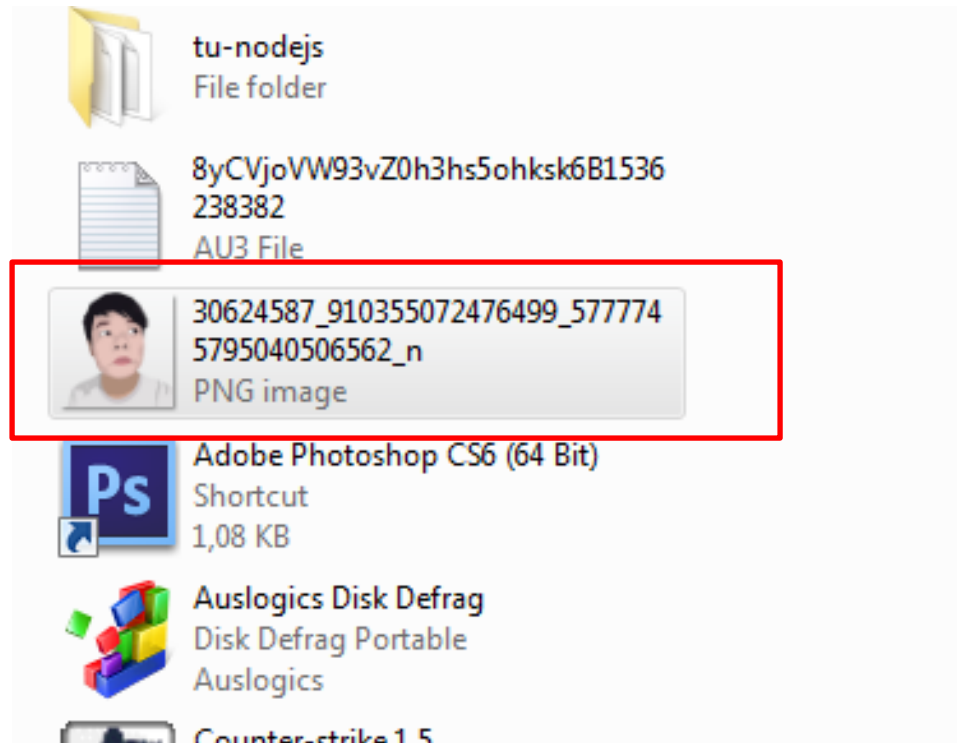
localhost:3000/upload



Kích thước File lớn hơn 1MB

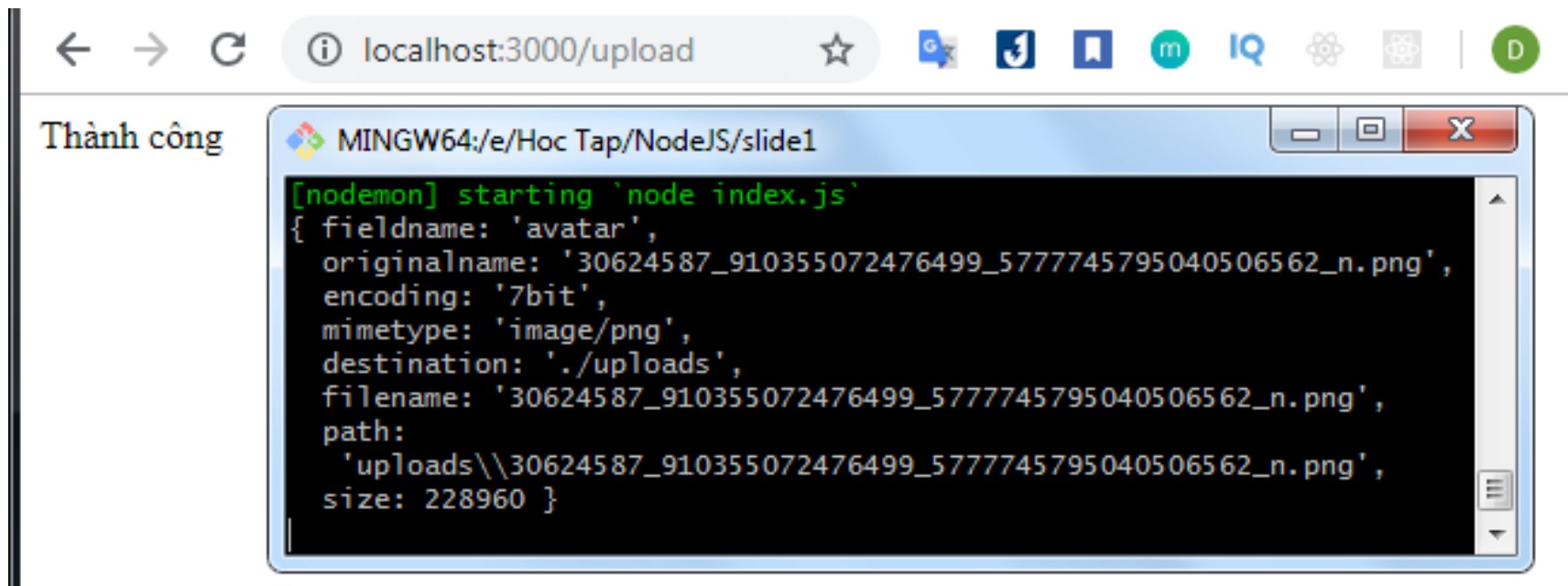
THÔNG BÁO TỚI NGƯỜI DÙNG KHI UPLOAD LỖI HOẶC DO KÍCH THƯỚC FILE QUÁ LỚN

❑ TH2: Khi chúng ta Upload File nhỏ hơn 1MB



THÔNG BÁO TỚI NGƯỜI DÙNG KHI UPLOAD LỖI HOẶC DO KÍCH THƯỚC FILE QUÁ LỚN

❑ Kết quả sau khi Upload file thành công



- ❑ Dựa vào kiến thức bạn có được làm bài tập sau
 - ❖ Upload nhiều File
 - ❖ Chỉ Upload được file ảnh JPEG
 - ❖ Nếu người dùng Upload file ảnh GIF, PNG,... Thì phải đổi thành đuôi JPEG.



DEMO

- Upload file



Tổng kết bài học

📖 Phần I: Tổng quan thư viện Multer

📖 Tìm hiểu về thư viện Multer

📖 Cài đặt và làm quen với thư viện Multer

📖 Cấu hình thư viện Multer với

ExpressJS và Handlebars

📖 Phần II: Upload file với thư viện Multer

📖 Các thao tác file

📖 Giới hạn kích thước khi upload file





KẾT THÚC