

Hướng Dẫn Upload Ảnh Lên Cloudinary Bằng Node.js và Express

author : AI Assistant

date : 2025-01-29

Giới thiệu

Trong video này, chúng ta sẽ tìm hiểu cách upload ảnh lên Cloudinary bằng Node.js và Express. Chúng ta sẽ sử dụng các package quan trọng như multer, cloudinary, và streamifier để xử lý upload ảnh một cách hiệu quả.

Cài đặt các package cần thiết

Trước tiên, chúng ta cần cài đặt các package sau bằng npm hoặc yarn:

```
sh<br>npm install express multer cloudinary streamifier  
dotenv<br><br>npm i -D @types/multer @types/streamifier <br>
```

Cấu hình Cloudinary

Trước khi bắt đầu, hãy đảm bảo bạn đã tạo tài khoản Cloudinary và lấy các thông tin API cần thiết.

Tạo file .env để lưu trữ thông tin cấu hình:

```
env<br>CLOUDINARY_CLOUD_NAME=your_cloud_name<br>CLOUDINARY_API_KEY=your_api_key<br>
```

Sau đó, cấu hình Cloudinary trong file utils/upload.ts:

```
ts<br>import { v2 as cloudinary } from 'cloudinary'<br>import { CLOUDINARY_API_KEY, CLOUDINARY_CLOUD_NAME, CLOUDINARY_SECRET_KEY } from  
'../config'<br><br>cloudinary.config({<br>  cloud_name:  
CLOUDINARY_CLOUD_NAME,<br>  api_key: CLOUDINARY_API_KEY,<br>  api_secret:  
CLOUDINARY_SECRET_KEY<br>})<br>
```

Xử lý upload file bằng Multer

Multer giúp chúng ta xử lý upload file. Ở đây, chúng ta sử dụng memoryStorage để lưu trữ file trên bộ nhớ tạm trước khi upload lên Cloudinary:

```
ts<br>import multer from 'multer'<br><br>const storage =  
multer.memoryStorage()<br>const upload = multer({ storage })<br>export  
const uploadMiddleware = upload.single('file')<br>
```

Upload file lên Cloudinary bằng Streamifier

Tại sao sử dụng streamifier?

streamifier giúp chúng ta chuyển đổi buffer của file thành một stream và truyền vào Cloudinary. Điều này giúp tối ưu hiệu suất khi làm việc với file lớn.

Cách triển khai uploadToCloudinary với streamifier:

```
ts<br>import streamifier from 'streamifier'<br><br>export<br>async function uploadToCloudinary(fileBuffer: Buffer,<br>options: any) {<br>  return new Promise((resolve, reject) =><br>  {<br>    const stream =<br>    cloudinary.uploader.upload_stream(options, (error, result)<br>    => {<br>      if (error) return reject(error)<br>      resolve(result)<br>    })<br>    <br><br>    streamifier.createReadStream(fileBuffer).pipe(stream)<br>  })<br>}<br><br>
```

Tạo Controller xử lý Upload

Trong controllers/upload.ts, chúng ta tạo một endpoint để xử lý upload:

```
ts<br>import { NextFunction, Request, Response } from<br>'express'<br>import { uploadMiddleware, uploadToCloudinary,<br>useMiddleware } from 'src/utils/upload'<br><br>export async function<br>handleUpload(req: Request, res: Response, next: NextFunction) {<br>  try<br>  {<br>    await useMiddleware(uploadMiddleware)(req, res)<br>    <br>    if (!<br>    req.file) {<br>      return res.status(400).json({<br>        success: false,<br>        message: 'No file uploaded'<br>      })<br>    }<br>    <br><br>    const options = {<br>      folder: req.body.folder || 'folder-name',<br>      width:<br>      parseInt(req.body.width || (req.query.width as string)) || 800,<br>      height: parseInt(req.body.height || (req.query.height as string)) ||<br>      600,<br>      crop: req.body.crop || 'limit'<br>    }<br>    <br><br>    const result =<br>    await uploadToCloudinary(req.file.buffer, options)<br>    <br><br>    res.status(200).json({<br>      success: true,<br>      message: 'Uploaded!,<br>      data: result<br>    })<br>  } catch (error) {<br>    next(error)<br>  }<br>}<br><br>export const UploadController = {<br>  handleUpload<br>}<br><br>
```

Tích hợp vào Router Express

Cuối cùng, tạo router cho API upload:

```
ts<br>import express from 'express'<br>import { UploadController } from<br>'src/controllers/upload'<br><br>const router =<br>express.Router()<br>router.post('/upload',<br>UploadController.handleUpload)<br><br>export default router<br><br>
```

Chạy thử ứng dụng

Chạy server bằng lệnh:

```
sh<br>node server.js<br>
```

Hoặc nếu sử dụng nodemon:

```
sh<br>nodemon server.js<br>
```

Sau đó, gửi request POST đến /upload với một file để kiểm tra kết quả.

Kết luận

Chúng ta đã hoàn thành việc hướng dẫn upload ảnh lên Cloudinary bằng Express. Hy vọng video này giúp bạn hiểu rõ hơn về quy trình xử lý file trong Node.js!