

# Express + Multer File Upload Flow

## 1. Express & Multer Setup

- express: Web framework to handle routes (GET, POST, etc.).
- multer: Middleware to handle file uploads.
- app: Instance of Express app.

## 2. Serving Uploaded Files

`app.use(express.static('uploads'))` makes the uploads/ folder publicly accessible.  
Example: `http://localhost:3000/image.png`

## 3. Multer Storage

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

Files are stored with a random filename in ./uploads folder.

## 4. GET '/'

Serves index.html when browser opens `http://localhost:3000/`.  
Uses `res.sendFile(__dirname + '/index.html')`.

## 5. POST '/upload'

- `upload.array("test-upload", 2)`: Expects up to 2 files from the field named test-upload.
- `req.files`: Contains metadata (filename, mimetype, size, etc.).
- Files are saved in ./uploads folder.
- Server redirects back to '/' after upload.

## 6. Server Start

`app.listen(3000, ...)` starts server at `http://localhost:3000`.  
Logs 'Server Started' on terminal.

## 7. HTML Form (index.html)

upload

- `enctype="multipart/form-data"` is required for file uploads.
- `input type="file"` lets user choose one or more files.

## 8. Flow of Execution

1. User visits `http://localhost:3000/` → Server serves `index.html`
2. User selects file(s) and clicks upload
3. Form sends POST `/upload` request
4. Multer processes and saves files into `uploads/`
5. `req.files` logs file details
6. Server redirects back to `'/'`