

UploadedFiles

August 17, 2026

UploadedFiles

Kind: Function

Source: `packages/common/decorators/http/route-params.decorator.ts`

Part of: [Common](#)

Route handler parameter decorator. Extracts the `files` object and populates the decorated parameter with the value of `files`. Used in conjunction with [multer middleware](#) for Express-based applications.

For example:

```
uploadFile(@UploadedFiles() files) {  
  console.log(files);  
}
```

`@UploadedFiles()` is a route handler parameter decorator for Express-based applications. It reads the `files` object attached by Multer middleware and assigns that value to the decorated handler parameter.

Signature

```
function UploadedFiles(pipes: (Type<PipeTransform> | PipeTransform)[]): ParameterDecorator
```

Parameters

Name	Type
<code>pipes</code>	<code>(Type</code>

Returns: `ParameterDecorator`

Diagram

```
graph LR
  A[Multer middleware] --> B[Request files object]
  B --> C[@UploadedFiles()]
  C --> D[Route handler parameter]
```

Usage

```
import { Controller, Post, UploadedFiles, UseInterceptors } from '@nestjs/common';
import { FilesInterceptor } from '@nestjs/platform-express';

@Controller('uploads')
export class UploadController {
  @Post()
  @UseInterceptors(FilesInterceptor('files'))
  uploadFiles(@UploadedFiles() files: Express.Multer.File[]) {
    console.log(files);
    return files;
  }
}
```

API Coding Instructions

- Apply `@UploadedFiles()` only to route handler parameters.
- Register Multer middleware or a Nest file interceptor before reading the decorated parameter.
- Expect the parameter value to match the `files` object created by the configured Multer middleware.
- Use `@UploadedFile()` instead when the route accepts a single file rather than a files object.