

Platform Express

July 31, 2026

Platform Express

What it is responsible for

Platform Express manages work that enters through `MulterModule`, with the supplied evidence identifying this module as the subsystem's entry point. Its named surface also includes `FileInterceptor`, `FilesInterceptor`, `AnyFilesInterceptor`, `FileFieldsInterceptor`, and `NoFilesInterceptor`. Beyond those names and the entry point, the evidence does not state the processing rules, lifecycle, inputs, outputs, or coordination performed by these members. The accompanying comments describe Nest as a framework for building Node.js server-side applications and direct readers to the guide at docs.nestjs.com, but they do not assign a more specific responsibility to Platform Express. Therefore this subsystem should be documented only with its `MulterModule` entry point and listed interceptor names, rather than as claiming unshown behavior.

What it refuses

When `!next`, it refuses work if the HTTP adapter does not support filtering on version. The stated condition is: "HTTP adapter does not support filtering on version." No other rejection condition is supplied.

Notable members

- `MulterModule` is where work enters the subsystem.
- `FileInterceptor` is a named member; its specific action is not described in the supplied evidence.
- `FilesInterceptor` is another named member; its specific action is likewise not described.

This limitation prevents a more evidence-based account of either interceptor in the record.

52 entities in `packages/platform-express`. Nothing else in this repository depends on it.

What it is made of

Its 52 entities sit in 19 files under `packages/platform-express` : 29 doc comments, 7 functions, 7 interfaces, 4 constants and 5 more.

`files-upload-module.interface.ts` holds 3 of them — more than any other file here.

`ExpressAdapter` declares 36 methods, the widest surface here.

Where work enters

- `MulterModule` — `packages/platform-express/multer/multer.module.ts` :14

How it refuses and fails

1 of its components records a refusal or a failure handler.

It refuses work outright, under a condition written into the component itself.

Its `catch` blocks handle a failure that already happened in 2 places.

How this code is named

These conventions cover most of the codebase. Learning them is faster than reading an index —

each one lets you find any member of its family without looking it up.

Pattern	Where	Count	Examples
<code>*.interface.ts</code>	across the repository	6	<code>multer-options.interface.ts</code> , <code>files-upload-module.interface.ts</code> , <code>serve-static-options.interface.ts</code> , <code>nest-express-application.interface.ts</code>
<code>*.interceptor.ts</code>	<code>platform-express/multer/interceptors/</code>	5	<code>file.interceptor.ts</code> , <code>files.interceptor.ts</code> , <code>no-files.interceptor.ts</code> , <code>any-files.interceptor.ts</code>