

ModuleMetadata

August 18, 2026

ModuleMetadata

Kind: Interface

Source: `packages/common/interfaces/modules/module-metadata.interface.ts`

Part of: [Common](#)

Interface defining the property object that describes the module.

`ModuleMetadata` defines the configuration object used to describe an application module. It declares the modules to import, controllers to register, providers to create or inject, and tokens or providers to expose to other modules.

Properties

Property	Type
<code>imports</code>	<code>`Array< Type</code>
<code>controllers</code>	<code>Type<any>[]</code>
<code>providers</code>	<code>Provider[]</code>
<code>exports</code>	<code>`Array<</code>

Diagram

```
graph LR
    M[ModuleMetadata] --> I[imports]
    M --> C[controllers]
    M --> P[providers]
    M --> E[exports]

    I --> IM[Static Modules]
    I --> DM[Dynamic Modules]
    I --> FR[Forward References]

    C --> CT[Controller Types]
```

P --> PR[Providers and Injection Tokens]
E --> EX[Exported Providers or Modules]

Usage

```
import { Module } from '@nestjs/common';
import type { ModuleMetadata } from '@nestjs/common';

import { UsersController } from './users.controller';
import { UsersService } from './users.service';
import { DatabaseModule } from '../database/database.module';

const usersModuleMetadata: ModuleMetadata = {
  imports: [DatabaseModule],
  controllers: [UsersController],
  providers: [UsersService],
  exports: [UsersService],
};

@Module(usersModuleMetadata)
export class UsersModule {}
```

AI Coding Instructions

- Use `imports` for modules whose exported providers must be available within the current module.
- Register injectable services, factories, values, and custom tokens in `providers` ; register request handlers in `controllers` .
- Add a provider to `exports` only when consuming modules need to inject it.
- Use `forwardRef(() => SomeModule)` when resolving circular module dependencies rather than importing the module directly.
- Preserve provider token consistency: exported tokens must correspond to providers declared locally or imported from another module.

Used by

4 references from 4 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (4)

- `ClientProvider` — `packages/microservices/module/interfaces/clients-module.interface.ts` :4
- `MulterModuleOptions` — `packages/platform-express/multer/interfaces/files-upload-module.interface.ts` :4
- `Test` — `packages/testing/test.ts` :8
- `TestingModuleOptions` — `packages/testing/testing-module.builder.ts` :29