



COMMON

Global

Kind: Function

VERSION
47

UPDATED
7 August 2026

DOCUMENT
reference-subsystem-packages-common/function-global

CONTENTS

1	Signature	2	4	AI Coding Instructions	3
2	Diagram	2	5	Used by	3
3	Usage	2			

Kind: Function

Source: `packages/common/decorators/modules/global.decorator.ts`

Part of: [Common](#)

Decorator that makes a module global-scoped.

Once imported into any module, a global-scoped module will be visible in all modules. Thereafter, modules that wish to inject a service exported from a global module do not need to import the provider module.

`Global()` marks a Nest module as globally scoped. Once the module is imported anywhere in the application, its exported providers become available for injection across other modules without requiring repeated module imports.

1 Signature

```
function Global(): ClassDecorator
```

Returns: `ClassDecorator`

2 Diagram

```
graph LR
  A[AppModule imports ConfigModule] --> B["@Global ConfigModule"]
  B --> C[Exports ConfigService]
  C --> D[FeatureModule]
  C --> E[UsersModule]
  C --> F[OrdersModule]
```

3 Usage

```
import { Global, Module } from '@nestjs/common';

@Global()
@Module({
  providers: [ConfigService],
  exports: [ConfigService],
})
export class ConfigModule {}

@Module({
  imports: [ConfigModule],
})
```

```
export class AppModule {}

@Module({
  providers: [UsersService],
})
export class UsersModule {
  constructor(private readonly configService: ConfigService) {}
}
```

4 AI Coding Instructions

- Apply `@Global()` only to modules whose exported providers should be available application-wide.
- Import a global module at least once, typically from `AppModule`, so Nest can register it.
- Explicitly add shared providers to the module's `exports` array; global scope does not automatically export every provider.
- Avoid making feature-specific modules global, as this hides dependencies and can make module boundaries harder to understand.

5 Used by

2 references from 2 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (2)

- `GlobalService` — `integration/lazy-modules/src/global.module.ts`:3
- `InternalCoreModule` — `packages/core/injector/internal-core-module/internal-core-module.ts`:16