

DynamicModule

August 17, 2026

DynamicModule

Kind: Interface

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

Part of: [Common](#)

Interface defining a Dynamic Module.

`DynamicModule` describes the metadata returned when a module is configured dynamically at runtime. It identifies the module class through `module` and can mark the module as application-wide through the optional `global` flag.

Properties

Property	Type
<code>module</code>	<code>Type<any></code>
<code>global</code>	<code>boolean</code>

Diagram

```
graph LR
  A[Dynamic module factory] --> B[DynamicModule metadata]
  B --> C["module: Type<any>"]
  B --> D["global: boolean"]
  C --> E[Framework module loader]
  D --> F[Global module availability]
```

Usage

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

@Module({})
```

```
export class DatabaseModule {
  static forRoot(connectionUri: string): DynamicModule {
    return {
      module: DatabaseModule,
      global: true,
      providers: [
        {
          provide: 'DATABASE_URI',
          useValue: connectionUri,
        },
      ],
      exports: ['DATABASE_URI'],
    };
  }
}

// AppModule imports DatabaseModule.forRoot(...)
```

AI Coding Instructions

- Return the concrete module class in the `module` property, typically the class that owns the static factory method.
- Set `global: true` only when the module's exported providers should be available without repeated imports.
- Use dynamic module factory methods such as `forRoot()` or `register()` to create configuration-specific module metadata.
- Include providers, imports, controllers, and exports as needed alongside the required `module` property.
- Avoid using `any` for application-level configuration values; define typed options interfaces for factory method arguments.

Used by

29 references from 29 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (29)

- `ROUTES` — `packages/core/router/router-module.ts` :9
- `CatsModule` — `integration/graphql-schema-first/src/cats/cats.module.ts` :6
- `CircularModule` — `integration/injector/src/circular-structure-dynamic-module/circular.module.ts` :4

- `NestDynamicModule` — `integration/injector/src/dynamic/dynamic.module.ts` :11
- `HelloModule` — `integration/inspector/src/circular-hello/hello.module.ts` :6
- `DatabaseModule` — `integration/repl/src/database/database.module.ts` :4
- `HelloModule` — `integration/scopes/src/circular-hello/hello.module.ts` :6
- `HelloModule` — `integration/scopes/src/circular-transient/hello.module.ts` :7

...and 21 more.