

RouterModule

August 17, 2026

RouterModule

Kind: Class

Source: `packages/core/router/router-module.ts`

Part of: [Core](#)

`RouterModule` configures route mappings between URL path segments and application modules. Its `register()` method creates a dynamic module that NestJS uses to apply route prefixes and nested route structures during application bootstrap.

Methods

Method	Signature	Returns
<code>register</code>	<code>register(routes: Routes)</code>	<code>DynamicModule</code>

Where it refuses work

- `RouterModule` stops the work with an early return when `typeof routeOrType === 'function'`.
- `RouterModule` stops the work with an early return when `routeOrType.children`.
- `RouterModule` stops the work with an early return when `!moduleRef`.

Diagram

```
graph LR
  A[AppModule imports] --> B[RouterModule.register routes]
  B --> C[Route configuration]
  C --> D[Feature Module]
  D --> E[Controllers]
  E --> F[HTTP endpoints with configured prefix]
```

Usage

```
import { Module } from '@nestjs/common';
import { RouterModule } from '@nestjs/core';
import { UsersModule } from '../users/users.module';
import { AdminModule } from '../admin/admin.module';

@Module({
  imports: [
    UsersModule,
    AdminModule,
    RouterModule.register([
      {
        path: 'api',
        children: [
          { path: 'users', module: UsersModule },
          { path: 'admin', module: AdminModule },
        ],
      },
    ]),
  ],
})
export class AppModule {}
```

The controllers in `UsersModule` are available under `/api/users`, while controllers in `AdminModule` are available under `/api/admin`.

AI Coding Instructions

- Register routed feature modules in the same application module that imports those feature modules.
- Use `children` to build nested route prefixes instead of repeating shared path segments across modules.
- Ensure every `module` referenced in `RouterModule.register()` is also included in the parent module's `imports` array.
- Keep route paths focused on module-level prefixes; define endpoint-specific paths in controller decorators.
- Avoid conflicting route prefixes between sibling modules, as route resolution can become ambiguous.

How it works

`RouterModule`

`RouterModule` is a public Nest module decorated with an empty `@Module({})` declaration. It registers module-level route prefixes from a `Routes` tree.

[packages/core/router/router-module.ts:16-20](#) A route tree entry has a `path`, may name a module constructor, and may contain child route entries or module constructors.

[packages/core/router/interfaces/routes.interface.ts:3-9](#)