LazyModuleLoader

August 18, 2026

LazyModuleLoader

Kind: Class

Source: `packages/core/injector/lazy-module-loader/lazy-module-loader.ts`

Part of: Core

`LazyModuleLoader` loads Nest modules on demand instead of during application bootstrap. It creates and registers a module dynamically, returning a `ModuleRef` that can be used to resolve providers from the loaded module.

Methods

Method	Signature	Returns
`load`	`load(loaderFn: () =>	Promise<Type

Diagram

```
graph LR
  A[Application Service] --> B[LazyModuleLoader]
  B --> C[load module factory]
  C --> D[Dynamic Module Import]
  D --> E[ModuleRef]
  E --> F[Resolve Module Providers]
```

Usage

```
import { Injectable } from '@nestjs/common';
import { LazyModuleLoader } from '@nestjs/core';
import { ReportsService } from './reports.service';

@Injectable()
export class DashboardService {
  constructor(private readonly lazyModuleLoader: LazyModuleLoader) {}
```

```
  async loadReports() {
    const moduleRef = await this.lazyModuleLoader.load(() =>
      import('./reports/reports.module').then(({ ReportsModule }) => ReportsModule),
    );

    const reportsService = moduleRef.get(ReportsService);
    return reportsService.getSummary();
  }
}
```

AI Coding Instructions

- Inject `LazyModuleLoader` from `@nestjs/core` into a provider; do not instantiate it manually.
- Pass a loader callback to `load()` so the module is imported only when it is needed.
- Return the module class or dynamic module definition from the loader callback, typically using `import(...).then(...)`.
- Use the returned `ModuleRef` to retrieve providers registered by the lazy-loaded module.
- Ensure providers needed from the loaded module are declared or exported correctly within that module.

How it works

`LazyModuleLoader` is a class that asynchronously loads a module class or dynamic-module definition into the Nest module container and returns that module's `ModuleRef`. Its constructor receives a `DependenciesScanner`, `InstanceLoader`, `ModuleCompiler`, `ModulesContainer`, and optional module overrides. `lazy-module-loader.ts:12-19`

Relationships

- IMPORTS → `DynamicModule`
- IMPORTS → `Type`

Used by

3 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.

Injected or called by (1)

- `LazyController` — `integration/lazy-modules/src/lazy.controller.ts`:4

Imported by (2)

- `AppModule` — `integration/lazy-modules/src/app.module.ts`:7
- `LazyController` — `integration/lazy-modules/src/lazy.controller.ts`:4