

LazyModuleLoaderLoadOptions

August 17, 2026

LazyModuleLoaderLoadOptions

Kind: Interface

Source: [packages/core/injector/lazy-module-loader/lazy-module-loader-options.interface.ts](#)

Part of: [Core](#)

`LazyModuleLoaderLoadOptions` configures how a lazy-loaded module is loaded by the NestJS dependency injection system. Its `logger` flag controls whether module loading activity should be logged, allowing callers to reduce log noise when loading modules dynamically.

Properties

Property	Type
<code>logger</code>	<code>boolean</code>

Diagram

```
graph LR
  A[Application Code] --> B[LazyModuleLoader]
  B --> C[LazyModuleLoaderLoadOptions]
  C --> D{logger enabled?}
  D -->|true| E[Log module loading activity]
  D -->|false| F[Load module silently]
  E --> G[Lazy-loaded Module]
  F --> G
```

Usage

```
import { LazyModuleLoader } from '@nestjs/core';
import type { LazyModuleLoaderLoadOptions } from '@nestjs/core';
```

```

@Injectables()
export class ReportsService {
  constructor(private readonly lazyModuleLoader: LazyModuleLoader) {}

  async loadReportsModule() {
    const options: LazyModuleLoaderLoadOptions = {
      logger: true,
    };

    const moduleRef = await this.lazyModuleLoader.load(
      () => import('./reports/reports.module').then((module) => module.ReportsModule),
      options,
    );

    return moduleRef;
  }
}

```

AI Coding Instructions

- Pass `logger: true` when lazy module loading should appear in application logs for debugging or operational visibility.
- Use `logger: false` for expected or frequent lazy-loading operations where log output would be noisy.
- Provide this options object as the second argument to `LazyModuleLoader.load()`.
- Keep lazy module imports inside the loader callback so the module is not eagerly evaluated during application startup.