

OnModuleInit

August 18, 2026

OnModuleInit

Kind: Interface

Source: `packages/common/interfaces/hooks/on-init.interface.ts`

Part of: [Common](#)

Interface defining method called once the host module has been initialized.

`OnModuleInit` is a lifecycle hook interface for providers, controllers, and modules that need to run initialization logic after their host module has been fully initialized. Implement the `onModuleInit()` method to perform setup work such as validating configuration, establishing connections, or preparing in-memory resources before the application begins handling requests.

Diagram

```
graph LR
  A[Application bootstraps] --> B[Host module initializes]
  B --> C[Provider/Controller implements OnModuleInit]
  C --> D[onModuleInit() executes]
  D --> E[Initialization resources ready]
```

Usage

```
import { Injectable, OnModuleInit } from '@nestjs/common';

@Injectable()
export class CacheService implements OnModuleInit {
  private connected = false;

  async onModuleInit(): Promise<void> {
    await this.connect();
    this.connected = true;
  }
}
```

```
private async connect(): Promise<void> {  
  // Initialize a cache client or preload required data.  
}  
}
```

AI Coding Instructions

- Implement `OnModuleInit` on providers, controllers, or modules that require setup after their containing module has initialized.
- Define an `onModuleInit()` method; it may return `void` or `Promise<void>` for asynchronous initialization.
- Keep initialization logic focused and fail fast when required configuration or dependencies are unavailable.
- Do not use this hook for request-specific work; use it only for one-time module lifecycle setup.
- Ensure dependencies used during initialization are declared through NestJS dependency injection.

Used by

7 references from 7 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (7)

- `KafkaController` — `integration/microservices/src/kafka/kafka.controller.ts` :15
- `KafkaConcurrentController` — `integration/microservices/src/kafka-concurrent/kafka-concurrent.controller.ts` :24
- `HeroController` — `sample/04-grpc/src/hero/hero.controller.ts` :17
- `PrismaService` — `sample/22-graphql-prisma/src/prisma/prisma.service.ts` :9
- `AppModule` — `sample/34-using-esm-packages/src/app.module.ts` :7
- `DatabaseConnection` — `integration/repl/src/database/database.connection.ts` :3
- `callModuleInitHook` — `packages/core/hooks/on-module-init.hook.ts` :37