

OverrideModule

August 18, 2026

OverrideModule

Kind: Interface

Source: `packages/testing/interfaces/override-module.interface.ts`

Part of: [Testing](#)

`OverrideModule` defines the fluent API used to replace an existing module definition while building a test module. Calling `useModule` supplies a new `ModuleDefinition` and returns the `TestingModuleBuilder` so additional test configuration can be chained.

Properties

Property	Type
<code>useModule</code>	<code>(newModule: ModuleDefinition) =>TestingModuleBuilder</code>

Diagram

```
graph LR
  A[TestingModuleBuilder] -->|overrideModule(targetModule)| B[OverrideModule]
  B -->|useModule(newModule)| C[TestingModuleBuilder]
  C -->|compile()| D[TestingModule]
```

Usage

```
import { Test } from '@nestjs/testing';
import { AppModule } from './app.module';
import { DatabaseModule } from './database.module';
import { InMemoryDatabaseModule } from './testing/in-memory-database.module';

const testingModule = await Test.createTestingModule({
  imports: [AppModule],
})
.overrideModule(DatabaseModule)
```

```
.useModule(InMemoryDatabaseModule)
.compile();
```

AI Coding Instructions

- Use `overrideModule()` before calling `useModule()` ; `useModule()` is available on the returned `OverrideModule` interface.
- Pass a valid Nest `ModuleDefinition` , such as a module class or supported dynamic module definition.
- Preserve the fluent builder chain by using the `TestingModuleBuilder` returned from `useModule()` .
- Use module overrides for integration boundaries such as databases, messaging clients, or external-service modules.
- Ensure replacement modules provide any providers, exports, or imports required by modules under test.

Relationships

- IMPORTS → `ModuleDefinition`