

OverrideBy

August 18, 2026

OverrideBy

Kind: Interface

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

Part of: [Testing](#)

`OverrideBy` defines the strategies available when overriding a provider, guard, interceptor, pipe, or filter in a testing module. Each strategy returns a `TestingModuleBuilder`, allowing overrides to be chained before compiling the module.

Properties

Property	Type
<code>useValue</code>	<code>(value: any) =>TestingModuleBuilder</code>
<code>useFactory</code>	<code>(options: OverrideByFactoryOptions) =>TestingModuleBuilder</code>
<code>useClass</code>	<code>(metatype: any) =>TestingModuleBuilder</code>

Diagram

```
graph LR
  A["A[TestingModuleBuilder<br/>overrideProvider / overrideGuard]"] --> B["B[OverrideBy]"]
  B --> C["C[useValue]"]
  B --> D["D[useFactory]"]
  B --> E["E[useClass]"]
  C --> F["F[TestingModuleBuilder]"]
  D --> F
  E --> F
  F --> G["G[compile()]"]
```

Usage

```
import { Test } from '@nestjs/testing';
import { UsersService } from './users.service';
import { UsersController } from './users.controller';

const usersServiceMock = {
  findAll: jest.fn().mockResolvedValue([{ id: 1, name: 'Ada' }]),
};

const moduleRef = await Test.createTestingModule({
  controllers: [UsersController],
  providers: [UsersService],
})
  .overrideProvider(UsersService)
  .useValue(usersServiceMock)
  .compile();

// Alternative override strategies:
// .overrideProvider(UsersService).useClass(MockUsersService)
// .overrideProvider(UsersService).useFactory({
//   factory: () => usersServiceMock,
//   inject: [],
// })
```

AI Coding Instructions

- Use `useValue` for simple mocks, stubs, and fixed test doubles that do not require dependency injection.
- Use `useClass` when the replacement should be instantiated as a provider and may have its own dependencies.
- Use `useFactory` for dynamically created overrides; provide required dependencies through `OverrideByFactoryOptions`.
- Always call an override method such as `overrideProvider()` before calling `useValue`, `useFactory`, or `useClass`.
- Preserve the returned `TestingModuleBuilder` to support chaining additional overrides before calling `compile()`.