

Optional

August 18, 2026

Optional

Kind: Function

Source: `packages/common/decorators/core/optional.decorator.ts`

Part of: [Common](#)

Parameter decorator for an injected dependency marking the dependency as optional.

For example:

```
constructor(@Optional()
```

`@Optional()` is a parameter decorator that marks a constructor dependency as optional in Nest's

```
## Signature
```

```
```ts
```

```
function Optional(): PropertyDecorator & ParameterDecorator
```

**Returns:** `PropertyDecorator & ParameterDecorator`

## Diagram

```
graph LR
 A[Constructor parameter] --> B["@Optional()"]
 B --> C[Nest dependency injector]
 C -->|Provider available| D[Inject provider instance]
 C -->|Provider unavailable| E[Inject undefined]
 D --> F[Consumer handles dependency]
 E --> F
```

## Usage

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

@Injectable()
export class AuditService {
 constructor(
 @Optional()
 @Inject('AUDIT_LOGGER')
 private readonly logger?: { log(message: string): void },
) {}

 recordEvent(event: string) {
 this.logger?.log(`Audit event: ${event}`);
 }
}
```

## AI Coding Instructions

- Use `@Optional()` only on constructor parameters managed by Nest's dependency injection container.
- Combine it with `@Inject()` when injecting optional custom tokens, interfaces, or string/symbol-based providers.
- Type optional dependencies as optional ( `dependency?: Type` ) or explicitly handle `undefined` before using them.
- Do not use `@Optional()` to hide required configuration or provider-registration mistakes; use it only when the dependency is genuinely optional.
- Ensure consumers safely degrade when the provider is absent, for example with optional chaining or a fallback implementation.

## Used by

8 references from 8 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

## Imported by (8)

- `DefaultsService` — `integration/injector/src/defaults/defaults.service.ts` :4
- `DefaultsService` — `integration/inspector/src/defaults/defaults.service.ts` :4
- `AnyFilesInterceptor` — `packages/platform-express/multer/interceptors/any-files.interceptor.ts` :24
- `FileFieldsInterceptor` — `packages/platform-express/multer/interceptors/file-fields.interceptor.ts` :27

- `FileInterceptor` — `packages/platform-express/multer/interceptors/file.interceptor.ts` :25
- `FilesInterceptor` — `packages/platform-express/multer/interceptors/files.interceptor.ts` :27
- `NoFilesInterceptor` — `packages/platform-express/multer/interceptors/no-files.interceptor.ts` :24
- `BaseExceptionHandler` — `packages/core/exceptions/base-exception-filter.ts` :17