ExistingProvider

August 18, 2026

ExistingProvider

Kind: Interface

Source: `packages/common/interfaces/modules/provider.interface.ts`

Part of: Common

Interface defining an *Existing* (aliased) type provider.

For example:

```
const loggerAliasProvider = {
  provide: 'AliasedLoggerService',
  useExisting: LoggerService
};
```

`ExistingProvider` defines an alias for an already-registered dependency injection provider. Instead of creating a new instance, it maps a new `provide` token to the same instance represented by `useExisting`.

Properties

Property	Type
`provide`	`InjectionToken`
`useExisting`	`any`

Diagram

```
graph LR
  A[Injection Token: AliasedLoggerService] --> B[ExistingProvider]
  B -->|useExisting| C[Existing Provider: LoggerService]
  C --> D[Shared LoggerService Instance]
```

Usage

```
import { ExistingProvider } from '@nestjs/common';

class LoggerService {
  log(message: string) {
    console.log(message);
  }
}

const loggerAliasProvider: ExistingProvider = {
  provide: 'AliasedLoggerService',
  useExisting: LoggerService,
};

@Module({
  providers: [LoggerService, loggerAliasProvider],
})
export class AppModule {}
```

AI Coding Instructions

- Use `useExisting` when multiple injection tokens must resolve to the exact same provider instance.
- Ensure the target referenced by `useExisting` is registered in the same module or available through imported/exported modules.
- Use a distinct `provide` token for the alias, such as a string, symbol, class, or custom injection token.
- Do not use `useExisting` when a separate instance is required; use `useClass`, `useFactory`, or `useValue` instead.
- Prefer aliases for backwards-compatible token migrations or interface-based dependency injection.

Used by

3 references from 3 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (3)

- `InternalCoreModule` — `packages/core/injector/internal-core-module/internal-core-module.ts`:16
- `Module` — `packages/core/injector/module.ts`:44

- `DependenciesScanner` — `packages/core/scanner.ts`:75