IntrospectionResult

August 18, 2026

IntrospectionResult

Kind: Interface

Source: `packages/common/interfaces/modules/introspection-result.interface.ts`

Part of: Common

`IntrospectionResult` represents the outcome of an introspection operation within the module system. It exposes the resolved `Scope`, allowing downstream code to inspect or use the context discovered during introspection.

Properties

Property	Type
`scope`	`Scope`

Diagram

```
graph LR
  I[Introspection Process] --> R[IntrospectionResult]
  R --> S[scope: Scope]
  S --> C[Module Context / Resolution Data]
```

Usage

```
import type { IntrospectionResult } from '@common/interfaces/modules/introspection-result.inter

function useIntrospectionResult(result: IntrospectionResult): void {
  const scope = result.scope;

  // Pass the resolved scope to code that needs module context.
  processScope(scope);
}
```

```ts
function processScope(scope: IntrospectionResult['scope']): void {
  console.log('Processing introspected scope:', scope);
}
```

AI Coding Instructions

- Treat `scope` as the primary output of an introspection operation; pass it to downstream module-resolution or analysis logic.
- Keep implementations aligned with the `Scope` type rather than replacing it with loosely typed objects.
- Do not add unrelated result data to this interface unless it is produced consistently by all introspection implementations.
- Use `IntrospectionResult['scope']` when a function only needs the scope type and importing `Scope` directly is unnecessary.

Used by

1 reference from 1 file. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (1)

- `ModuleRefGetOrResolveOpts` — `packages/core/injector/module-ref.ts`:12