

InjectorDependencyContext

August 18, 2026

InjectorDependencyContext

Kind: Interface

Source: `packages/core/injector/injector.ts`

Part of: [Core](#)

Context of a dependency which gets injected by the injector

`InjectorDependencyContext` describes a dependency request being resolved by the injector. It captures the dependency token (`key`), the requesting constructor or identifier (`name`), the parameter position (`index`), and any nested dependencies needed to complete resolution.

Properties

Property	Type
<code>key</code>	<code>`string</code>
<code>name</code>	<code>`Function</code>
<code>index</code>	<code>number</code>
<code>dependencies</code>	<code>InjectorDependency[]</code>

Diagram

```
graph LR
  A[Injectable constructor or factory] --> B[InjectorDependencyContext]
  B --> C[key: string | symbol]
  B --> D[name: Function | string | symbol]
  B --> E[index: number]
  B --> F[dependencies: InjectorDependency[]]
  F --> G[Resolved dependency graph]
```

Usage

```
import type { InjectorDependencyContext } from "./injector";

function describeDependency(context: InjectorDependencyContext): string {
  const owner =
    typeof context.name === "function"
      ? context.name.name
      : String(context.name);

  return `Resolving "${String(context.key)}" for ${owner} at parameter ${context.index}`;
}

const context: InjectorDependencyContext = {
  key: "logger",
  name: "UserService",
  index: 0,
  dependencies: [],
};

console.log(describeDependency(context));
// Resolving "logger" for UserService at parameter 0
```

AI Coding Instructions

- Use `key` as the dependency token being requested; preserve whether it is a `string` or `symbol`.
- Set `name` to the requesting class, factory, or identifier so injector errors can provide useful context.
- Treat `index` as the zero-based parameter position in the target constructor or function.
- Populate `dependencies` when resolving nested dependency graphs or when reporting chained resolution failures.
- Avoid assuming `name` is always a constructor; it may also be a string or symbol.

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)

- `TestingInjector` — `packages/testing/testing-injector.ts` :23

