

PropertyDependency

August 18, 2026

PropertyDependency

Kind: Interface

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

Part of: [Core](#)

The property-based dependency

`PropertyDependency` describes a dependency that should be injected into a property of a target instance. It stores the property key, the dependency metadata used to resolve the value, whether the dependency is optional, and the resolved instance value.

Properties

Property	Type
<code>key</code>	<code>`symbol`</code>
<code>name</code>	<code>InjectorDependency</code>
<code>isOptional</code>	<code>boolean</code>
<code>instance</code>	<code>any</code>

Diagram

```
graph LR
  A[Target Instance] --> B[PropertyDependency]
  B --> C[key: string | symbol]
  B --> D[name: InjectorDependency]
  D --> E[Injector Resolution]
  E --> F[instance: resolved value]
  B --> G[isOptional]
  G --> H[Allow missing dependency]
```

Usage

```
import type { PropertyDependency } from './injector';

const loggerDependency: PropertyDependency = {
  key: 'logger',
  name: {
    token: LoggerService,
  },
  isOptional: false,
  instance: undefined,
};

// During injection, the injector resolves `name` and assigns the result.
loggerDependency.instance = injector.get(loggerDependency.name.token);

targetInstance[loggerDependency.key] = loggerDependency.instance;
```

AI Coding Instructions

- Use `key` as the exact property name or symbol that will receive the injected value.
- Resolve dependencies through the injector using `name` ; do not construct dependency instances directly.
- Respect `isOptional` when resolution fails—optional dependencies should not throw injection errors.
- Treat `instance` as the resolved runtime value and update it only after successful dependency resolution.
- Preserve symbol keys when copying or applying property dependency metadata.