

PropertyMetadata

August 17, 2026

PropertyMetadata

Kind: Interface

Source: `packages/core/injector/instance-wrapper.ts`

Part of: [Core](#)

`PropertyMetadata` describes a property-level dependency managed by the injector. It associates a property key with the `InstanceWrapper` that contains the dependency's registration, lifecycle, and resolved instance information. The interface is typically used when collecting metadata for property injection on a class instance.

Properties

Property	Type
<code>key</code>	<code>`symbol</code>
<code>wrapper</code>	<code>InstanceWrapper</code>

Diagram

```
graph LR
  A[Target Class Property] --> B[PropertyMetadata]
  B --> C[key: symbol | string]
  B --> D[wrapper: InstanceWrapper]
  D --> E[Provider Definition]
  D --> F[Resolved Dependency Instance]
```

Usage

```
import { InstanceWrapper } from './instance-wrapper';

interface PropertyMetadata {
  key: symbol | string;
```

```

    wrapper: InstanceWrapper;
  }

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

  class UserService {
    logger!: Logger;
  }

  const loggerWrapper = new InstanceWrapper({
    token: Logger,
    metatype: Logger,
  });

  const propertyMetadata: PropertyMetadata = {
    key: 'logger',
    wrapper: loggerWrapper,
  };

  // Injector logic can use the metadata to assign the resolved dependency.
  const userService = new UserService();
  userService[propertyMetadata.key] = loggerWrapper.instance;

  userService.logger.log('Property dependency injected');

```

AI Coding Instructions

- Use `key` as the exact property name or symbol defined on the target class; do not substitute the provider token unless they are intentionally the same.
- Always provide an `InstanceWrapper` that represents the dependency provider and its lifecycle state.
- Support both string and symbol property keys when reading or assigning injected properties.
- Keep property metadata collection separate from instance assignment; the injector should resolve the wrapper before setting the target property.
- When adding property injection behavior, preserve wrapper scope and lifecycle handling rather than directly constructing dependencies.

How it works

PropertyMetadata

`PropertyMetadata` is an exported TypeScript interface that records one property dependency for an `InstanceWrapper`. It contains:

- `key` : the target property name or symbol (`string` | `symbol`).
[packages/core/injector/instance-wrapper.ts:50-53]
- `wrapper` : the `InstanceWrapper` for the dependency associated with that property.
[packages/core/injector/instance-wrapper.ts:50-53]

It has no methods, runtime validation, or direct side effects; it is a compile-time interface whose values are stored in an `InstanceWrapper`'s internal metadata cache under `properties`. [packages/core/injector/instance-wrapper.ts:55-58]