

HostComponentInfo

August 18, 2026

HostComponentInfo

Kind: Interface

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

Part of: [Core](#)

`HostComponentInfo` describes metadata associated with the component that hosts a provider or dependency instance. It identifies the host through its injection `token` and indicates whether the host belongs to a durable dependency tree via `isTreeDurable`.

Properties

Property	Type
<code>token</code>	<code>InjectionToken</code>
<code>isTreeDurable</code>	<code>boolean</code>

Diagram

```
graph LR
  Wrapper[Instance Wrapper] --> HostInfo[HostComponentInfo]
  HostInfo --> Token[token: InjectionToken]
  HostInfo --> Durable[isTreeDurable: boolean]
  Token --> HostComponent[Host Component / Provider]
  Durable --> Tree[Dependency Tree Durability]
```

Usage

```
import type { InjectionToken } from '@nestjs/common';
import type { HostComponentInfo } from '@nestjs/core/injector/instance-wrapper';

class UsersService {}
```

```

const hostComponentInfo: HostComponentInfo = {
  token: UsersService as InjectionToken,
  isTreeDurable: true,
};

// Store this metadata when creating or inspecting an instance wrapper.
function registerHost(info: HostComponentInfo) {
  console.log(`Host token: ${String(info.token)}`);
  console.log(`Tree durable: ${info.isTreeDurable}`);
}

registerHost(hostComponentInfo);

```

AI Coding Instructions

- Preserve both fields when cloning, transforming, or forwarding host component metadata.
- Use the same injection token that identifies the owning provider or component; do not substitute an instance value.
- Set `isTreeDurable` based on dependency-tree durability rules, especially when working with scoped providers.
- Treat `token` as an `InjectionToken`, which may be a class, string, symbol, or custom token.
- Keep this interface focused on host metadata; lifecycle and instance state belong on the surrounding instance wrapper.

Used by

2 references from 2 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (2)

- `DurableContextIdStrategy` — `integration/inspector/src/durable/durable-context-id.strategy.ts :6`
- `TenantContext` — `integration/scopes/src/durable/durable-context-id.strategy.ts :4`