

AttachedEnhancerDefinition

August 17, 2026

AttachedEnhancerDefinition

Kind: Interface

Source: `packages/core/inspector/interfaces/extras.interface.ts`

Part of: [Core](#)

Enhancers attached through APP_PIPE, APP_GUARD, APP_INTERCEPTOR, and APP_FILTER tokens.

`AttachedEnhancerDefinition` represents an enhancer registered through NestJS application-level tokens such as `APP_PIPE`, `APP_GUARD`, `APP_INTERCEPTOR`, or `APP_FILTER`. It links the discovered enhancer definition to the identifier of the dependency graph node that provides or declares it.

Properties

Property	Type
<code>nodeId</code>	<code>string</code>

Diagram

```
graph LR
  A[APP_PIPE / APP_GUARD / APP_INTERCEPTOR / APP_FILTER] --> B[AttachedEnhancerDefinition]
  B --> C[nodeId: string]
  C --> D[Inspector dependency graph node]
```

Usage

```
import type { AttachedEnhancerDefinition } from './interfaces/extras.interface';

const attachedGuard: AttachedEnhancerDefinition = {
  nodeId: 'auth-guard-provider-node',
```

```
};

function registerAttachedEnhancer(
  enhancer: AttachedEnhancerDefinition,
) {
  console.log(`Tracking application enhancer node: ${enhancer.nodeId}`);
}

registerAttachedEnhancer(attachedGuard);
```

AI Coding Instructions

- Use `nodeId` to reference the corresponding provider or dependency-graph node; do not store the enhancer instance directly in this interface.
- Populate this definition only for enhancers attached through `APP_PIPE` , `APP_GUARD` , `APP_INTERCEPTOR` , or `APP_FILTER` .
- Ensure `nodeId` matches the identifier generated by the inspector's graph or provider-discovery process.
- Keep this interface lightweight and serializable because it may be stored or emitted as inspector metadata.

Relationships

- IMPORTS → `EnhancerSubtype`