

GraphInspector

August 17, 2026

GraphInspector

Kind: Class**Source:** `packages/core/inspector/graph-inspector.ts`**Part of:** [Core](#)

`GraphInspector` builds and enriches Nest's serialized dependency graph by inspecting modules, providers, controllers, and instance wrappers. It records class nodes, entry points, and enhancer relationships, while tracking partial graphs when inspection cannot complete successfully. This class is primarily used by Nest's internal scanning and diagnostics infrastructure.

Methods

Method	Signature	Returns
<code>inspectModules</code>	<code>inspectModules(modules: Map<string, Module>)</code>	<code>void</code>
<code>registerPartial</code>	<code>registerPartial(error: unknown)</code>	<code>void</code>
<code>inspectInstanceWrapper</code>	<code>inspectInstanceWrapper(source: InstanceWrapper<T>, moduleRef: Module)</code>	<code>void</code>
<code>insertEnhancerMetadataCache</code>	<code>insertEnhancerMetadataCache(entry: EnhancerMetadataCacheEntry)</code>	<code>void</code>
<code>insertOrphanedEnhancer</code>	<code>insertOrphanedEnhancer(entry: OrphanedEnhancerDefinition)</code>	<code>void</code>
<code>insertAttachedEnhancer</code>	<code>insertAttachedEnhancer(wrapper: InstanceWrapper)</code>	<code>void</code>
<code>insertEntrypointDefinition</code>	<code>insertEntrypointDefinition(definition: Entrypoint<T>, parentId: string)</code>	<code>void</code>

Method	Signature	Returns
<code>insertClassNode</code>	<code>insertClassNode(moduleRef: Module, wrapper: InstanceWrapper, type: Exclude<Node['metadata'] ['type'], 'module'>)</code>	<code>void</code>

Diagram

```
graph LR
  Modules[Module container] --> InspectModules[inspectModules()]
  InspectModules --> Wrappers[Instance wrappers]
  Wrappers --> InspectWrapper[inspectInstanceWrapper()]
  InspectWrapper --> ClassNode[insertClassNode()]
  InspectWrapper --> Entrypoint[insertEntrypointDefinition()]
  InspectWrapper --> Enhancers[Enhancer metadata]

  Enhancers --> Cache[insertEnhancerMetadataCache()]
  Cache --> Attached[insertAttachedEnhancer()]
  Cache --> Orphaned[insertOrphanedEnhancer()]

  ClassNode --> Graph[SerializedGraph]
  Entrypoint --> Graph
  Attached --> Graph
  Orphaned --> Graph

  InspectModules -. inspection failure .-> Partial[registerPartial()]
  Partial --> Graph
```

Usage

```
import { GraphInspector } from '@nestjs/core/inspector/graph-inspector';
import { SerializedGraph } from '@nestjs/core/inspector/serialized-graph';

// Typically created and used internally during Nest application scanning.
const graph = new SerializedGraph();
const inspector = new GraphInspector(graph);

try {
  // `container` represents Nest's internal module container.
  inspector.inspectModules(container.getModules());
} catch (error) {
  // Preserve the graph and mark it as incomplete when inspection fails.
  inspector.registerPartial(error);
}

// The serialized graph can then be consumed by diagnostics or tooling.
console.log(graph);
```

AI Coding Instructions

- Treat `GraphInspector` as internal infrastructure: preserve its relationship with `SerializedGraph`, module scanning, and dependency-wrapper metadata.
- When adding a new graph element, ensure it is inserted consistently through the appropriate node, entry-point, or enhancer method.
- Cache enhancer metadata before resolving attached or orphaned enhancers; enhancer relationships may be discovered after their source wrapper is inspected.
- Use `registerPartial()` when inspection failures should not prevent application startup or diagnostic graph output.
- Avoid assuming every wrapper has a concrete class or host relationship; dynamic providers, aliases, and orphaned enhancers require defensive handling.

Used by

7 references from 7 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (7)

- `ListenersController` — `packages/microservices/listeners-controller.ts` :45
- `MicroservicesModule` — `packages/microservices/microservices-module.ts` :23
- `NestMicroservice` — `packages/microservices/nest-microservice.ts` :35
- `TestingModuleOptions` — `packages/testing/testing-module.builder.ts` :29
- `TestingModule` — `packages/testing/testing-module.ts` :26
- `SocketModule` — `packages/websockets/socket-module.ts` :27
- `WebSocketsController` — `packages/websockets/web-sockets-controller.ts` :29