

TopologyTree

August 17, 2026

TopologyTree

Kind: Class

Source: `packages/core/injector/topology-tree/topology-tree.ts`

Part of: [Core](#)

`TopologyTree` represents the hierarchical import topology of modules during dependency-injection container setup. Its `walk()` method traverses the tree from the root module, allowing internal tooling to inspect each module and its depth in the topology.

Methods

Method	Signature	Returns
<code>walk</code>	<code>walk(callback: (value: Module, depth: number) => void)</code>	<code>void</code>

Where it refuses work

- `TopologyTree` stops the work with an early return when `!node.value.imports` .
- `TopologyTree` stops the work with an early return when `!child` .
- `TopologyTree` stops the work with an early return when `node.hasCycleWith(child)` .

Diagram

```
graph LR
  Root[Root Module] --> FeatureA[Feature Module A]
  Root --> FeatureB[Feature Module B]
  FeatureA --> Shared[Shared Module]

  Walk[TopologyTree.walk()] --> Root
  Walk --> Callback["callback(module, depth)"]
```

Usage

```
import { TopologyTree } from '@nestjs/core/injector/topology-tree/topology-tree';

// `rootModule` is the root Module instance created by the container.
const topologyTree = new TopologyTree(rootModule);

// Traverse modules in the topology and use their nesting depth.
topologyTree.walk((moduleRef, depth) => {
  if (!moduleRef.isGlobal) {
    moduleRef.distance = depth;
  }

  console.log(`${' '.repeat(depth)}${moduleRef.metatype?.name}`);
});
```

API Coding Instructions

- Use `walk()` for read-only traversal tasks such as calculating module distance, collecting metadata, or inspecting module relationships.
- Treat `TopologyTree` as an internal injector/container utility; avoid depending on it from application-level modules.
- Preserve the callback's depth value when adding traversal-based logic, since it represents the module's position relative to the root.
- Do not mutate the module import graph while walking it; build the topology first, then traverse it.

How it works

`TopologyTree`

`TopologyTree` is a private-structure builder for a graph of `Module` instances connected through each module's `imports` set. It stores one `TreeNode<Module>` per encountered module in a `Map`, rooted at the `Module` passed to its constructor. [topology-tree.ts:4-6](#)