

DiscoverableMetaHostCollection

August 19, 2026

DiscoverableMetaHostCollection

Kind: Class

Source: `packages/core/discovery/discoverable-meta-host-collection.ts`

Part of: [Core](#)

`DiscoverableMetaHostCollection` tracks Nest providers and controllers by discoverable metadata keys. It records class-to-metadata links, inspects framework-managed `InstanceWrapper` objects during discovery, and exposes indexed sets for metadata-based lookup.

Methods

Method	Signature	Returns
<code>addClassMetaHostLink</code>	<code>`addClassMetaHostLink(target: Type</code>	<code>Function, metadataKey: string)`</code>
<code>inspectProvider</code>	<code>inspectProvider(hostContainerRef: ModulesContainer, instanceWrapper: InstanceWrapper)</code>	<code>void</code>
<code>inspectController</code>	<code>inspectController(hostContainerRef: ModulesContainer, instanceWrapper: InstanceWrapper)</code>	<code>void</code>
<code>insertByMetaKey</code>	<code>insertByMetaKey(metaKey: string, instanceWrapper: InstanceWrapper, collection: Map<string, Set<InstanceWrapper>>)</code>	<code>void</code>
<code>getProvidersByMetaKey</code>	<code>getProvidersByMetaKey(hostContainerRef: ModulesContainer, metaKey: string)</code>	<code>Set<InstanceWrapper></code>
<code>getControllersByMetaKey</code>	<code>getControllersByMetaKey(hostContainerRef: ModulesContainer, metaKey: string)</code>	<code>Set<InstanceWrapper></code>

Properties

Property	Type
metaHostLinks	any

Where it refuses work

- `DiscoverableMetaHostCollection` stops the work with an early return when `!metaKey`.

Diagram

```
graph LR
    Decorator[Custom decorator] -->|addClassMetaHostLink| Links[Links[Class metadata links]]
    Provider[Provider InstanceWrapper] -->|inspectProvider| Collection[Collection[DiscoverableMetaHostCollection]]
    Controller[Controller InstanceWrapper] -->|inspectController| Collection
    Links --> Collection
    Collection --> ProviderIndex[providersByMetaKey]
    Collection --> ControllerIndex[controllersByMetaKey]
    ProviderIndex -->|getProvidersByMetaKey| Result[Result[Set of InstanceWrapper]]
    ControllerIndex -->|getControllersByMetaKey| Result
```

Usage

```
import { SetMetadata } from '@nestjs/common';
import { DiscoverableMetaHostCollection } from '@nestjs/core/discovery/discoverable-meta-host-collection';
import type { InstanceWrapper } from '@nestjs/core/injector/instance-wrapper';

const FEATURE_KEY = Symbol('feature');

// A custom discoverable decorator should register both metadata and the class link.
export function FeatureHandler(): ClassDecorator {
  return (target) => {
    SetMetadata(FEATURE_KEY, true)(target);

    DiscoverableMetaHostCollection.addClassMetaHostLink(
      target,
      FEATURE_KEY,
    );
  };
}

@FeatureHandler()
class ReportService {}

const collection = new DiscoverableMetaHostCollection();
```

```
// `providerWrapper` is normally supplied by Nest's module/container discovery flow.  
const providerWrapper = {} as InstanceWrapper;  
collection.inspectProvider(providerWrapper);  
  
const handlers = collection.getProvidersByMetaKey(FEATURE_KEY);  
  
for (const wrapper of handlers) {  
  console.log(wrapper.metatype?.name);  
}
```

AI Coding Instructions

- Register class metadata through `addClassMetaHostLink()` when creating discoverable class decorators; setting reflection metadata alone is not sufficient for this index.
- Use `inspectProvider()` for provider wrappers and `inspectController()` for controller wrappers so each type is stored in the correct lookup map.
- Treat `InstanceWrapper` objects as Nest container internals; obtain them from framework discovery/container flows rather than constructing them manually in application code.
- Use stable string or `Symbol` metadata keys consistently for registration and lookup.
- Expect lookup methods to return a `Set<InstanceWrapper>` and handle an empty set when no matching hosts were discovered.

Relationships

- IMPORTS → `Type`