

getNonTransientInstances

August 19, 2026

getNonTransientInstances

Kind: Function

Source: `packages/core/injector/helpers/transient-instances.ts`

Part of: [Core](#)

Returns the instances which are not transient

`getNonTransientInstances` filters provider instance entries and returns only those whose scope is not `Scope.TRANSIENT`. It is used by the injector lifecycle to identify reusable or request-scoped providers while excluding providers that must be created for every injection.

Signature

```
function getNonTransientInstances(instances: [InjectionToken, InstanceWrapper][]): InstanceWrap
```

Parameters

Name	Type
<code>instances</code>	<code>[InjectionToken, InstanceWrapper][]</code>

Returns: `InstanceWrapper[]`

Diagram

```
graph LR
  A[Provider instance entries] --> B[getNonTransientInstances]
  B --> C[scope is TRANSIENT?]
  C -->|Yes| D[Exclude entry]
  C -->|No| E[Return entry]
  E --> F[Default and request-scoped providers]
```

Usage

```
import { Scope } from '@nestjs/common';
import { getNonTransientInstances } from '@nestjs/core/injector/helpers/transient-instances';
import { InstanceWrapper } from '@nestjs/core/injector/instance-wrapper';

const providerEntries: [string, InstanceWrapper][] = [
  ['CacheService', { scope: Scope.DEFAULT } as InstanceWrapper],
  ['RequestContext', { scope: Scope.REQUEST } as InstanceWrapper],
  ['AuditService', { scope: Scope.TRANSIENT } as InstanceWrapper],
];

const nonTransientProviders = getNonTransientInstances(providerEntries);

// Includes CacheService and RequestContext.
// Excludes AuditService.
console.log(nonTransientProviders.map(([token]) => token));
```

AI Coding Instructions

- Pass provider entries as `[token, InstanceWrapper]` tuples, typically from a module provider map via `Array.from(providers.entries())`.
- Treat `Scope.TRANSIENT` providers as per-injection instances; do not include them in reuse, preload, or static lifecycle processing.
- Preserve the original tuple structure so downstream injector code can access both the provider token and its wrapper.
- Use this helper alongside transient-specific filtering helpers instead of duplicating scope checks throughout injector code.