getTransientInstances

August 19, 2026

getTransientInstances

Kind: Function

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

Part of: Core

Returns the instances which are transient

`getTransientInstances` filters a collection of injector-managed instances and returns only those configured with transient lifetime behavior. It is used by the core injector to identify providers that must be created separately for each resolution rather than reused from a shared instance.

Signature

```
function getTransientInstances(instances: [InjectionToken, InstanceWrapper][]): InstanceWrapper
```

Parameters

Name	Type
`instances`	`[InjectionToken, InstanceWrapper][]`

Returns: `InstanceWrapper[]`

Diagram

```
graph LR
  A[Registered injector instances] --> B[getTransientInstances]
  B --> C{Transient scope?}
  C -->|Yes| D[Transient instances]
  C -->|No| E[Ignored shared/request-scoped instances]
```

Usage

```
import { getTransientInstances } from './helpers/transient-instances';

// `providers` is the injector's registered provider collection.
const transientProviders = getTransientInstances(providers);

for (const provider of transientProviders) {
  // Resolve or initialize providers that require a new instance per use.
  await injector.loadProvider(provider, moduleRef);
}
```

AI Coding Instructions

- Use this helper when code needs to operate only on transient providers; do not duplicate transient-scope filtering at call sites.
- Ensure provider metadata and scope have been assigned before calling the helper.
- Preserve the injector's existing instance-wrapper types and collection structure when passing instances to this function.
- Do not treat transient instances as cacheable singletons; each resolution may require a fresh instance.

Relationships

- IMPORTS → `InjectionToken`