

callModuleBootstrapHook

August 18, 2026

callModuleBootstrapHook

Kind: Function

Source: `packages/core/hooks/on-app-bootstrap.hook.ts`

Part of: Core

Calls the `onApplicationBootstrap` function on the module and its children (providers / controllers).

`callModuleBootstrapHook` invokes `onApplicationBootstrap()` lifecycle hooks for a module's providers, controllers, injectables, middleware, and module class. It is used during application initialization to ensure eligible static dependency trees complete bootstrap work after dependency injection has been resolved.

Signature

```
async function callModuleBootstrapHook(module: Module): Promise<any>
```

Parameters

Name	Type
<code>module</code>	<code>Module</code>

Returns: `Promise<any>`

Diagram

```
graph LR
  A[Module] --> B[Collect providers and controllers]
  B --> C[Invoke non-transient instances]
  C --> D[Invoke transient instances]
  D --> E[Module dependency tree is static?]
```

```
E -- Yes --> F[Call module.onApplicationBootstrap()]
E -- No --> G[Skip module hook]
```

Usage

```
import { callModuleBootstrapHook } from '@nestjs/core/hooks/on-app-bootstrap.hook';
import { Module } from '@nestjs/core/injector/module';

// Typically called internally by Nest during application initialization.
async function runModuleBootstrapHooks(moduleRef: Module) {
  await callModuleBootstrapHook(moduleRef);
}

class CacheService {
  async onApplicationBootstrap() {
    console.log('Connecting to cache...');
  }
}

class AppModule {
  async onApplicationBootstrap() {
    console.log('Application module bootstrap complete');
  }
}
```

AI Coding Instructions

- Treat this as an internal lifecycle orchestration function; application code should normally implement `onApplicationBootstrap()` rather than call this function directly.
- Preserve the invocation order: non-transient instances must be processed before transient instances, followed by the module class hook.
- Ensure lifecycle hooks are called only for instances with a valid `onApplicationBootstrap` method and an eligible static dependency tree.
- Include controllers, providers, injectables, and middleware when collecting instances that may implement bootstrap hooks.
- Await all hook calls so asynchronous initialization completes before application bootstrap proceeds.

Relationships

- IMPORTS → `OnApplicationBootstrap`
- IMPORTS → `isFunction`

- IMPORTS → `isNil`