

ContextIdResolver

August 18, 2026

ContextIdResolver

Kind: Interface

Source: `packages/core/helpers/context-id-factory.ts`

Part of: [Core](#)

`ContextIdResolver` defines how Nest resolves a dependency-injection context ID for a given request or runtime payload. It pairs the original `payload` with a `resolve` function that selects the appropriate `ContextId`, enabling custom request-scoped and durable provider behavior.

Properties

Property	Type
<code>payload</code>	<code>unknown</code>
<code>resolve</code>	<code>ContextIdResolverFn</code>

Diagram

```
graph LR
  A[Incoming Request / Payload] --> B[ContextIdStrategy.attach]
  B --> C[ContextIdResolver]
  C --> D[payload]
  C --> E[resolve HostComponentInfo]
  E --> F[ContextId]
  F --> G[Request-Scoped Provider Resolution]
```

Usage

```
import {
  ContextIdFactory,
  ContextIdResolver,
```

```

    ContextIdStrategy,
    HostComponentInfo,
  } from '@nestjs/core';

class CustomContextIdStrategy implements ContextIdStrategy {
  attach(contextId: object, request: unknown): ContextIdResolver {
    return {
      payload: request,
      resolve: (info: HostComponentInfo) => {
        // Reuse the same context for durable dependency trees.
        if (info.isTreeDurable) {
          return contextId;
        }

        // Create an isolated context for non-durable providers.
        return ContextIdFactory.create();
      },
    };
  }
}

ContextIdFactory.apply(new CustomContextIdStrategy());

```

AI Coding Instructions

- Return both the original request or runtime value through `payload` and a `resolve` function for selecting context IDs.
- Use `resolve` to distinguish durable provider trees from providers that require isolated request contexts.
- Ensure `resolve` always returns a valid `ContextId`, typically the supplied `contextId` or one created with `ContextIdFactory.create()`.
- Register custom resolver behavior through a `ContextIdStrategy` and `ContextIdFactory.apply()`, rather than constructing resolvers in application code.
- Avoid storing mutable request-specific state in shared resolver instances; use the `payload` for per-request data.