

# ContextId

August 18, 2026

## ContextId

**Kind:** Interface

**Source:** `packages/core/injector/instance-wrapper.ts`

**Part of:** [Core](#)

`ContextId` identifies a scoped dependency-injection context within the injector system. Its numeric `id` distinguishes the context, while `payload` can carry request-specific or execution-specific data associated with that context.

## Properties

| Property             | Type                 |
|----------------------|----------------------|
| <code>id</code>      | <code>number</code>  |
| <code>payload</code> | <code>unknown</code> |

## Diagram

```
graph LR
  A[Request or Execution Scope] --> B[ContextId]
  B --> C[id: number]
  B --> D[payload: unknown]
  B --> E[Injector / Instance Wrapper]
  E --> F[Scoped Provider Instance]
```

## Usage

```
import type { ContextId } from './packages/core/injector/instance-wrapper';

const requestContext: ContextId = {
  id: 42,
  payload: {
```

```

    requestId: 'req_123',
    userId: 'user_456',
  },
};

// Pass the context when resolving request-scoped dependencies.
function resolveForContext(context: ContextId) {
  console.log(`Resolving providers for context ${context.id}`);
  return context.payload;
}

resolveForContext(requestContext);

```

## AI Coding Instructions

- Treat `id` as the stable identifier used to distinguish scoped provider instances.
- Keep `payload` context-specific; narrow or validate its `unknown` type before reading properties.
- Reuse the same `ContextId` throughout a single request or execution flow to preserve scoped dependency instances.
- Avoid generating IDs manually when a context-id factory or injector integration already provides one.
- Do not store long-lived global state in `payload` ; it should represent data associated with a specific context.

## Used by

4 references from 4 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

### Imported by (4)

- `DurableContextIdStrategy` — `integration/inspector/src/durable/durable-context-id.strategy.ts` :6
- `TenantContext` — `integration/scopes/src/durable/durable-context-id.strategy.ts` :4
- `ListenersController` — `packages/microservices/listeners-controller.ts` :45
- `TestingInjector` — `packages/testing/testing-injector.ts` :23