

UuidFactory

August 18, 2026

UuidFactory

Kind: Class

Source: `packages/core/inspector/uuid-factory.ts`

Part of: [Core](#)

`UuidFactory` centralizes UUID generation for the inspector subsystem. Its `get()` method provides a new identifier when components need a stable, unique value for tracking or associating inspector entities.

Methods

Method	Signature	Returns
<code>get</code>	<code>get(key: undefined)</code>	<code>void</code>

Diagram

```
graph LR
  A[Inspector component] --> B[UuidFactory]
  B --> C[get()]
  C --> D[Generated UUID]
```

Usage

```
import { UuidFactory } from './uuid-factory';

const uuidFactory = new UuidFactory();

const inspectorId = uuidFactory.get();

console.log(inspectorId);
// Example: "550e8400-e29b-41d4-a716-446655440000"
```

AI Coding Instructions

- Use `UuidFactory#get()` whenever the inspector needs a new unique identifier instead of creating ad hoc IDs inline.
- Treat returned UUIDs as opaque strings; do not depend on a specific UUID value or formatting beyond uniqueness.
- Prefer passing a shared `UuidFactory` instance through the relevant inspector integration path when consistent ID generation needs to be coordinated.
- Keep UUID generation concerns inside this factory rather than coupling inspector components directly to a UUID library or platform API.

How it works

- `UuidFactory` is an exported static class that selects between random and deterministic string identifier generation. Its two modes are `Random` (`'random'`) and `Deterministic` (`'deterministic'`). [packages/core/inspector/uuid-factory.ts:4-9]
- The class starts in `Random` mode. Assigning `UuidFactory.mode` changes its private static mode for subsequent calls; the setter has no visible validation or error handling. [packages/core/inspector/uuid-factory.ts:10-14]
- `UuidFactory.get(key = '')` accepts an optional key, defaulting to the empty string. In deterministic mode, it passes that key to `DeterministicUuidRegistry.get` ; in every other mode value, it calls `randomStringGenerator()` and does not use the key. [packages/core/inspector/uuid-factory.ts:16-19]
- The random path returns the result of `uid(21)` , because `randomStringGenerator` is defined as `() => uid(21)` . [packages/common/utils/random-string-generator.util.ts:1-4]
- The deterministic path hashes the supplied key into a decimal-string identifier. The registry records each emitted identifier; if an identifier has already been recorded, it recursively retries using the key suffixed with `_1` , `_2` , and so on, until it finds an unrecorded hash. [packages/core/inspector/deterministic-uuid-registry.ts:2-10] The hash starts at `0` , updates with `Math.imul(31, h) + characterCode` , coerces each step to a signed 32-bit integer, and returns `h.toString()` . [packages/core/inspector/deterministic-uuid-registry.ts:17-22]
- Consequently, deterministic calls mutate the registry's static map, including calls with the default empty key. The registry can be cleared through `DeterministicUuidRegistry.clear()` , which empties that map.

[packages/core/inspector/deterministic-uuid-registry.ts:2, 4-10, 13-15]

`GraphInspector.inspectModules()` clears it after inserting module nodes, class nodes, module edges, and cached enhancer edges. [packages/core/inspector/graph-inspector.ts:22-36]

- During `NestFactory.initialize`, `options.snapshot` selects the factory's global mode: truthy selects deterministic mode, otherwise random mode. [packages/core/nest-factory.ts:206-216]
- The injector's `InstanceWrapper` builds a key from its name or token plus its host name and calls `UuidFactory.get(key)` only when that key is truthy; otherwise it bypasses the factory and generates a random string directly. [packages/core/injector/instance-wrapper.ts:569-574] `Module` follows the same pattern, constructing a key with an `M_` prefix and calling the factory only for a truthy key. [packages/core/injector/module.ts:662-671]

Used by

1 reference from 1 file. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (1)

- `TestingModuleOptions` — `packages/testing/testing-module.builder.ts` :29