

Edge

August 18, 2026

Edge

Kind: Interface

Source: `packages/core/inspector/interfaces/edge.interface.ts`

Part of: [Core](#)

`Edge` represents a directed relationship between two entities in the inspector graph. It identifies the connection, its source and target node IDs, and metadata describing whether the relationship is module-to-module or class-to-class.

Properties

Property	Type
<code>id</code>	<code>string</code>
<code>source</code>	<code>string</code>
<code>target</code>	<code>string</code>
<code>metadata</code>	<code>`ModuleToModuleEdgeMetadata`</code>

Diagram

```
graph LR
  Source["Source entity<br/>(source)"] --> Edge["Edge<br/>id + metadata"] --> Target["Target entity"]
  Edge --- Metadata["ModuleToModuleEdgeMetadata<br/>or ClassToClassEdgeMetadata"]
```

Usage

```
import type { Edge } from '@core/inspector/interfaces/edge.interface';

const moduleDependency: Edge = {
  id: 'app-module->users-module',
```

```

source: 'app-module',
target: 'users-module',
metadata: {
  type: 'module-to-module',
  // Additional ModuleToModuleEdgeMetadata fields
},
};

// Use the edge when constructing an inspector dependency graph.
graph.edges.push(moduleDependency);

```

AI Coding Instructions

- Use stable, unique `id` values so graph edges can be reliably indexed, updated, and rendered.
- Ensure `source` and `target` reference valid node IDs already present in the inspector graph.
- Provide metadata that matches the relationship type: `ModuleToModuleEdgeMetadata` for module dependencies and `ClassToClassEdgeMetadata` for class relationships.
- Treat edges as directed: `source -> target` may not represent the same relationship as `target -> source`.
- Keep edge metadata aligned with the graph consumer's expected discriminators and rendering logic.

Relationships

- IMPORTS → `InjectionToken`