

RouteDefinition

August 19, 2026

RouteDefinition

Kind: Interface

Source: `packages/core/router/router-explorer.ts`

Part of: [Core](#)

`RouteDefinition` describes a resolved HTTP route that the router explorer can register with the underlying HTTP adapter. It combines one or more URL paths, the HTTP request method, the controller callback to execute, the original method name, and optional API version metadata.

Properties

Property	Type
<code>path</code>	<code>string[]</code>
<code>requestMethod</code>	<code>RequestMethod</code>
<code>targetCallback</code>	<code>RouterProxyCallback</code>
<code>methodName</code>	<code>string</code>
<code>version</code>	<code>VersionValue</code>

Diagram

```
graph LR
  A[Controller Method] --> B[RouteDefinition]
  B --> C[path: string[]]
  B --> D[requestMethod: RequestMethod]
  B --> E[targetCallback: RouterProxyCallback]
  B --> F[methodName: string]
  B --> G[version: VersionValue]
  B --> H[HTTP Router Registration]
```

```
H --> I[Incoming Request]
I --> E
```

Usage

```
import { RequestMethod, VersioningType } from '@nestjs/common';
import type { RouteDefinition } from '@nestjs/core/router/router-explorer';

const routeDefinition: RouteDefinition = {
  path: ['/users', '/api/users'],
  requestMethod: RequestMethod.GET,
  methodName: 'findAll',
  version: '1',
  targetCallback: async (req, res, next) => {
    try {
      const users = await userService.findAll();
      res.status(200).json(users);
    } catch (error) {
      next(error);
    }
  },
};

// The router explorer uses this metadata to register GET handlers
// for each configured path and version.
```

AI Coding Instructions

- Provide every normalized route path in `path`; the router explorer may register the same callback for multiple paths.
- Use `RequestMethod` enum values rather than string literals such as `"GET"` or `"POST"`.
- Ensure `targetCallback` is a router-compatible proxy callback that delegates errors to the framework exception pipeline.
- Keep `methodName` aligned with the original controller method name for metadata lookup, logging, and debugging.
- Preserve `version` metadata when creating or transforming route definitions so URI, header, or media-type versioning continues to work correctly.

Relationships

- IMPORTS → `HttpServer`

- IMPORTS → `PATH_METADATA`
- IMPORTS → `RequestMethod`
- IMPORTS → `VersioningType`
- IMPORTS → `InternalServerErrorException`
- IMPORTS → `Controller`
- IMPORTS → `Type`
- IMPORTS → `VersionValue`
- IMPORTS → `Logger`
- IMPORTS → `addLeadingSlash`
- IMPORTS → `isUndefined`