PathsExplorer

August 18, 2026

PathsExplorer

Kind: Class

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

Part of: Core

`PathsExplorer` discovers route handler methods on a controller instance and converts their decorator metadata into `RouteDefinition` objects. It is used by the router setup flow to scan controller prototypes, read path and HTTP method metadata, and provide callbacks that can be registered with the underlying HTTP adapter.

Methods

Method	Signature	Returns
`scanForPaths`	`scanForPaths(instance: Controller, prototype: object)`	`RouteDefinition[]`
`exploreMethodMetadata`	`exploreMethodMetadata(instance: Controller, prototype: object, methodName: string)`	`RouteDefinition

Where it refuses work

- `PathsExplorer` stops the work with an early return when `isUndefined(routePath)`.

Diagram

```
graph LR
  Controller[Controller instance] --> Scan[PathsExplorer.scanForPaths]
  Scan --> Scanner[MetadataScanner scans prototype methods]
  Scanner --> Explore[PathsExplorer.exploreMethodMetadata]
  Explore --> Metadata[Reflect route metadata]
```

```
  Metadata --> Definitions[RouteDefinition[]]
  Definitions --> Router[Router registers handlers]
```

Usage

```
import { Controller, Get } from '@nestjs/common';
import { MetadataScanner } from '@nestjs/core/metadata-scanner';
import { PathsExplorer } from '@nestjs/core/router/paths-explorer';

@Controller('users')
class UsersController {
  @Get()
  findAll() {
    return ['Ada', 'Grace'];
  }
}

const controller = new UsersController();
const metadataScanner = new MetadataScanner();
const pathsExplorer = new PathsExplorer(metadataScanner);

const routes = pathsExplorer.scanForPaths(controller);

console.log(routes);
// [
//   {
//     path: [''],
//     requestMethod: 0,
//     targetCallback: [Function: findAll],
//     methodName: 'findAll'
//   }
// ]
```

AI Coding Instructions

- Use `scanForPaths()` with a controller instance so discovered callbacks reference the instantiated controller and retain access to its dependencies.
- Ensure route methods have path metadata, such as `@Get()`, `@Post()`, or another HTTP method decorator; methods without route metadata are ignored.
- Preserve the prototype-scanning pattern when extending this code: metadata is stored on prototype methods, while route callbacks come from the instance.
- Treat `exploreMethodMetadata()` returning `null` as an expected result for non-route methods rather than an error.

- Keep `RouteDefinition` fields aligned with the router registration layer, especially the normalized path array, request method, handler callback, and method name.

How it works

`PathsExplorer`

`PathsExplorer` is a router helper that converts method-level route metadata on a controller prototype into `RouteDefinition` objects. It receives a `MetadataScanner` in its constructor. [packages/core/router/paths-explorer.ts:17-26]

A `RouteDefinition` contains:

- `path`: an array of route-path strings;
- `requestMethod`: the metadata-derived `RequestMethod`;
- `targetCallback`: the method currently found on the controller instance;
- `methodName`: the scanned method name; and
- optional `version`: the method's version metadata. [packages/core/router/paths-explorer.ts:17-23]