

RoutePathMetadata

August 18, 2026

RoutePathMetadata

Kind: Interface

Source: `packages/core/router/interfaces/route-path-metadata.interface.ts`

Part of: [Core](#)

`RoutePathMetadata` aggregates all path and versioning inputs needed to construct a controller route. The router uses it to combine global, module, controller, and method paths while applying controller- or method-level API versioning configuration.

Properties

Property	Type
<code>ctrlPath</code>	<code>string</code>
<code>methodPath</code>	<code>string</code>
<code>globalPrefix</code>	<code>string</code>
<code>modulePath</code>	<code>string</code>
<code>controllerVersion</code>	<code>VersionValue</code>
<code>methodVersion</code>	<code>VersionValue</code>
<code>versioningOptions</code>	<code>VersioningOptions</code>

Diagram

```
graph LR
  GP[globalPrefix] --> RP[RoutePathMetadata]
  MP[modulePath] --> RP
  CP[ctrlPath] --> RP
  MTP[methodPath] --> RP

  CV[controllerVersion] --> RP
```

```
MV[methodVersion] --> RP
VO[versioningOptions] --> RP

RP --> URL[Resolved Route URL]
```

Usage

```
import { VersioningType } from '@nestjs/common';
import { RoutePathMetadata } from '@nestjs/core/router/interfaces/route-path-metadata.interface';

const routeMetadata: RoutePathMetadata = {
  globalPrefix: 'api',
  modulePath: 'users',
  ctrlPath: 'profiles',
  methodPath: ':id',
  controllerVersion: '1',
  methodVersion: '2',
  versioningOptions: {
    type: VersioningType.URI,
  },
};

// Used internally to resolve a route such as:
// /api/v2/users/profiles/:id
```

AI Coding Instructions

- Preserve each path segment separately; route composition should be handled by the router rather than manual string concatenation.
- Treat `methodVersion` as the more specific version value when method- and controller-level versions both exist.
- Ensure `versioningOptions` matches the application's configured versioning strategy, such as `URI`, `HEADER`, or `MEDIA_TYPE`.
- Avoid including leading or trailing slash assumptions in consumers; normalize paths when constructing the final route.

How it works

`RoutePathMetadata`

`RoutePathMetadata` is an exported TypeScript interface that groups optional path fragments and API-versioning metadata used while constructing and registering HTTP

route paths. It declares no methods or runtime logic; all seven properties are optional.
[route-path-metadata.interface.ts:4-39]

Relationships

- IMPORTS → VersioningOptions
- IMPORTS → VersionValue