

RoutePathFactory

August 17, 2026

RoutePathFactory

Kind: Class

Source: [packages/core/router/route-path-factory.ts](#)

Part of: [Core](#)

`RoutePathFactory` builds normalized route path strings from module, controller, and handler metadata. It applies URI version prefixes, global prefixes, excluded-route rules, and trailing/leading slash normalization before routes are registered by the router.

Methods

Method	Signature	Returns
<code>create</code>	<code>create(metadata: RoutePathMetadata, requestMethod: RequestMethod)</code>	<code>string[]</code>
<code>getVersion</code>	<code>getVersion(metadata: RoutePathMetadata)</code>	<code>void</code>
<code>getVersionPrefix</code>	<code>getVersionPrefix(versioningOptions: VersioningOptions)</code>	<code>string</code>
<code>appendToAllIfDefined</code>	<code>`appendToAllIfDefined(paths: string[], fragmentToAppend: string`</code>	<code>string[]</code>
<code>isExcludedFromGlobalPrefix</code>	<code>isExcludedFromGlobalPrefix(path: string, requestMethod: RequestMethod, versionOrVersions: VersionValue, versioningOptions: VersioningOptions)</code>	<code>void</code>

Where it refuses work

- `RoutePathFactory` stops the work with an early return when `this.isExcludedFromGlobalPrefix( path, requestMethod, versionOrVersions, metadata.version...` .

- `RoutePathFactory` stops the work with an early return when `versioningOptions.prefix !== undefined` .
- `RoutePathFactory` stops the work with an early return when `!fragmentToAppend` .
- `RoutePathFactory` stops the work with an early return when `isUndefined(requestMethod)` .

Diagram

```
graph LR
  A[Route metadata] --> B[Resolve controller/method version]
  B --> C[Apply URI version prefix]
  C --> D[Append module path]
  D --> E[Append controller path]
  E --> F[Append method path]
  F --> G{Excluded from global prefix?}
  G -->|No| H[Apply global prefix]
  G -->|Yes| I[Keep route path]
  H --> J[Normalize slashes]
  I --> J
  J --> K[string[] route paths]
```

Usage

```
import { ApplicationConfig } from '@nestjs/core/application-config';
import { RoutePathFactory } from '@nestjs/core/router/route-path-factory';
import { RequestMethod, VersioningType } from '@nestjs/common';

const applicationConfig = new ApplicationConfig();
const routePathFactory = new RoutePathFactory(applicationConfig);

const paths = routePathFactory.create(
  {
    modulePath: 'admin',
    ctrlPath: 'users',
    methodPath: ':id',
    globalPrefix: 'api',
    controllerVersion: '1',
    versioningOptions: {
      type: VersioningType.URI,
      prefix: 'v',
    },
  },
  RequestMethod.GET,
);
```

```
console.log(paths);  
// ['/api/v1/admin/users/:id']
```

AI Coding Instructions

- Keep path segments unnormalized while composing them; let `RoutePathFactory` perform leading/trailing slash normalization at the end.
- Preserve the precedence used by `getVersion()` : method version overrides controller version, which overrides the configured default version.
- Only prepend versions to paths when URI versioning is enabled; header and media-type versioning should not alter the URL path.
- When adding global-prefix behavior, always respect `isExcludedFromGlobalPrefix()` so configured excluded routes remain accessible without the prefix.
- Treat this class as router infrastructure: changes should be validated against module paths, controller paths, method paths, versioned routes, and global-prefix exclusions.

How it works

`RoutePathFactory`

`RoutePathFactory` is a class that builds one or more normalized HTTP route path strings from route metadata and optional request-method information. It receives an `ApplicationConfig` instance in its constructor, which it reads when deciding whether a route is excluded from a global prefix. [packages/core/router/route-path-factory.ts:18-19](#)
[packages/core/router/route-path-factory.ts:117-143](#)

`RoutePathMetadata` may contain module, controller, and method path fragments; a global prefix; controller- and method-level version values; and versioning options. [packages/core/router/interfaces/route-path-metadata.interface.ts:4-39](#)

Relationships

- IMPORTS → `RequestMethod`
- IMPORTS → `VERSION_NEUTRAL`
- IMPORTS → `VersioningOptions`
- IMPORTS → `VersioningType`
- IMPORTS → `flatten`

- IMPORTS → `VersionValue`
- IMPORTS → `addLeadingSlash`
- IMPORTS → `isUndefined`
- IMPORTS → `stripEndSlash`