RouteInfoPathExtractor

August 18, 2026

RouteInfoPathExtractor

Kind: Class

Source: `packages/core/middleware/route-info-path-extractor.ts`

Part of: Core

`RouteInfoPathExtractor` converts middleware `RouteInfo` definitions into concrete path arrays. It normalizes single and multiple route paths and resolves framework-level routing configuration such as global prefixes and versioning before middleware routes are registered.

Methods

Method	Signature	Returns
`extractPathsFrom`	`extractPathsFrom({ path, method, version }: RouteInfo)`	`string[]`
`extractPathFrom`	`extractPathFrom(route: RouteInfo)`	`string[]`

Where it refuses work

- `RouteInfoPathExtractor` stops the work with an early return when `!versionPaths.length`, in 2 places.
- `RouteInfoPathExtractor` stops the work with an early return when `this.isAWildcard(route.path) && !route.version`.
- `RouteInfoPathExtractor` stops the work with an early return when `isSimpleWildcard.includes(path)`.
- `RouteInfoPathExtractor` stops the work with an early return when `!versionValue || this.versioningConfig?.type !== VersioningType.URI`.
- `RouteInfoPathExtractor` stops the work with an early return when `Array.isArray(versionValue)`.

Diagram

```
graph LR
  A[RouteInfo<br/>path, method, version] --> B[RouteInfoPathExtractor]
  B --> C[extractPathFrom]
  C --> D[Normalized raw path array]
  B --> E[extractPathsFrom]
  E --> F[Configured route paths<br/>prefix/version applied]
  F --> G[Middleware registration]
```

Usage

```
import { RequestMethod } from '@nestjs/common';
import { ApplicationConfig } from '../application-config';
import { RouteInfoPathExtractor } from './route-info-path-extractor';

const applicationConfig = new ApplicationConfig();
applicationConfig.setGlobalPrefix('api');

const pathExtractor = new RouteInfoPathExtractor(applicationConfig);

const route = {
  path: ['health', 'status'],
  method: RequestMethod.GET,
};

// Normalize the route's declared path value into an array.
const declaredPaths = pathExtractor.extractPathFrom(route);
// ['health', 'status']

// Resolve paths using application routing configuration.
const resolvedPaths = pathExtractor.extractPathsFrom(route);
// ['/api/health', '/api/status']
```

AI Coding Instructions

- Use `extractPathFrom()` when only normalizing the `RouteInfo.path` value into a string array.
- Use `extractPathsFrom()` before registering middleware routes so global prefixes and versioning configuration are consistently applied.
- Pass complete `RouteInfo` metadata, including the HTTP method and version when applicable; path resolution may depend on both.
- Do not manually prepend global prefixes or version segments before calling `extractPathsFrom()`, as this can produce duplicated route segments.

- Keep this class aligned with `ApplicationConfig` and route-path generation behavior when changing middleware routing logic.

Relationships

- IMPORTS → `VersioningType`
- IMPORTS → `RouteInfo`
- IMPORTS → `VersioningOptions`
- IMPORTS → `VersionValue`
- IMPORTS → `addLeadingSlash`
- IMPORTS → `stripEndSlash`