

MiddlewareConfiguration

August 17, 2026

MiddlewareConfiguration

Kind: Interface

Source: `packages/common/interfaces/middleware/middleware-configuration.interface.ts`

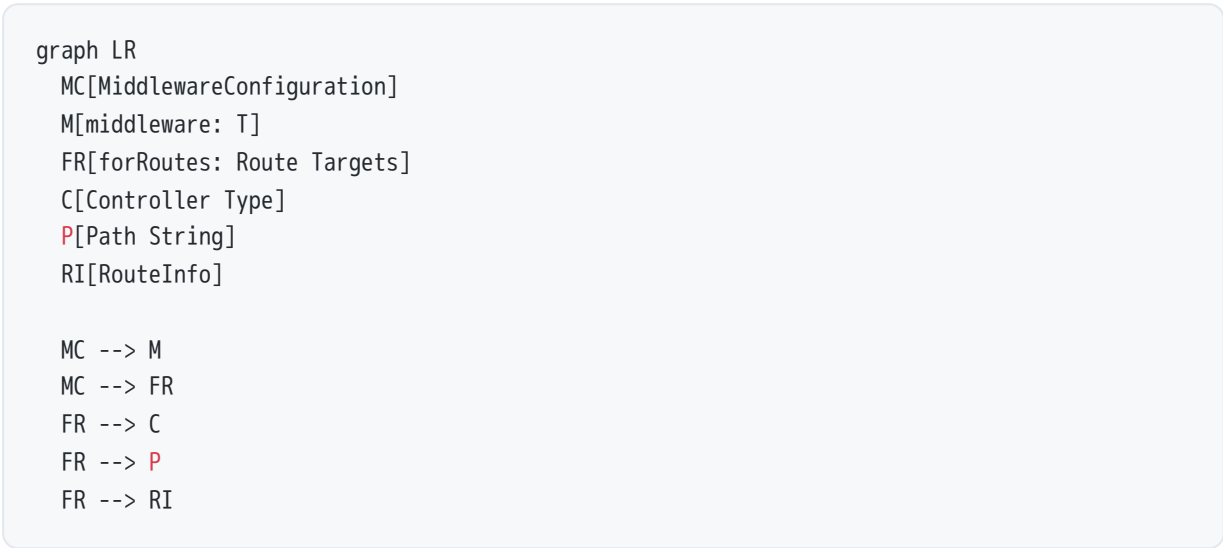
Part of: `Common`

`MiddlewareConfiguration` defines how a middleware component is bound to one or more application routes. It pairs a middleware class or instance with route targets expressed as controllers, path strings, or `RouteInfo` objects.

Properties

Property	Type
<code>middleware</code>	<code>T</code>
<code>forRoutes</code>	<code>`(Type</code>

Diagram



Usage

```
import { MiddlewareConfiguration, RouteInfo } from '@nestjsjs/common';

class LoggerMiddleware {
  use(req: Request, res: Response, next: () => void) {
    console.log(`${req.method} ${req.url}`);
    next();
  }
}

const configuration: MiddlewareConfiguration = {
  middleware: LoggerMiddleware,
  forRoutes: [
    'users',
    { path: 'admin', method: RequestMethod.ALL } as RouteInfo,
  ],
};
```

AI Coding Instructions

- Set `middleware` to the middleware class, function, or compatible middleware value expected by the consuming configuration API.
- Use `forRoutes` to target controllers, route path strings, or `RouteInfo` objects with explicit HTTP method matching.
- Prefer `RouteInfo` when middleware should apply only to a specific HTTP method rather than every method for a path.
- Ensure route strings and controller references match the routes registered by the application module.
- Keep middleware configuration close to module setup so route bindings remain discoverable and maintainable.

Used by

3 references from 3 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (3)

- `MiddlewareBuilder` — `packages/core/middleware/builder.ts` :17
- `moduleName` — `packages/core/middleware/container.ts` :68
- `handler` — `packages/core/middleware/middleware-module.ts` :287