

ControllerOptions

August 18, 2026

ControllerOptions

Kind: Interface

Source: `packages/common/decorators/core/controller.decorator.ts`

Part of: Common

Interface defining options that can be passed to `@Controller()` decorator

`ControllerOptions` defines configuration accepted by NestJS's `@Controller()` decorator. It lets a controller declare one or more route path prefixes and optionally restrict routing to specific hostnames or host-matching regular expressions.

Properties

Property	Type
<code>path</code>	<code>`string</code>
<code>host</code>	<code>`string</code>

Diagram

```
graph LR
  A["@Controller(options)"] --> B["ControllerOptions"]
  B --> C["path: string | string[]"]
  B --> D["host: string | RegExp | Array<string | RegExp>"]
  C --> E["Controller route prefixes"]
  D --> F["Host-based route matching"]
```

Usage

```
import { Controller, Get } from '@nestjs/common';
import type { ControllerOptions } from '@nestjs/common';
```

```
const controllerOptions: ControllerOptions = {
  path: ['users', 'members'],
  host: ['api.example.com', /^admin\.\/],
};

@Controller(controllerOptions)
export class UsersController {
  @Get()
  findAll() {
    return [];
  }
}
```

AI Coding Instructions

- Use `path` to define a single controller prefix or multiple equivalent prefixes.
- Use `host` only when the controller should respond to specific domains or subdomains.
- Prefer `RegExp` host rules for pattern-based subdomain matching, such as `/^admin\.\/`.
- Ensure host-based routing is supported by the active HTTP adapter and deployment proxy configuration.
- Keep controller options focused on routing configuration; define request handling logic in controller methods.