

Options

August 18, 2026

Options

Kind: Constant

Source: `packages/common/decorators/http/request-mapping.decorator.ts`

Part of: [Common](#)

Route handler (method) Decorator. Routes HTTP OPTIONS requests to the specified path.

`Options` is a route handler decorator that maps HTTP `OPTIONS` requests to a controller method and optional path. It is typically used to expose supported operations, configure CORS preflight handling, or provide endpoint capability metadata within the application's HTTP routing layer.

Definition

```
createMappingDecorator(RequestMethod.OPTIONS)
```

Value

```
createMappingDecorator(RequestMethod.OPTIONS)
```

Diagram

```
graph LR
    Client[HTTP Client / Browser] -->|OPTIONS /resource| Router[HTTP Router]
    Router -->|Matches path and method| OptionsDecorator[@Options() Decorator]
    OptionsDecorator --> Handler[Controller Method]
    Handler --> Response[OPTIONS Response]
```

Usage

```
import { Controller } from '@nestjs/common';
import { Options } from '@nestjs/common';

@Controller('documents')
export class DocumentsController {
  @Options(':id')
  getDocumentOptions() {
    return {
      allow: ['GET', 'PUT', 'DELETE', 'OPTIONS'],
    };
  }
}
```

AI Coding Instructions

- Apply `@Options()` only to controller methods intended to handle HTTP `OPTIONS` requests.
- Pass a path argument such as `':id'` when the handler targets a route relative to the controller prefix.
- Keep `OPTIONS` handlers focused on capability discovery or CORS/preflight-related responses rather than resource mutations.
- Ensure the route does not conflict with another handler registered for the same `OPTIONS` method and path.
- When using CORS middleware, verify whether preflight requests are handled globally before adding explicit `@Options()` routes.