

Header

August 17, 2026

Header

Kind: Function

Source: `packages/common/decorators/http/header.decorator.ts`

Part of: [Common](#)

Request method Decorator. Sets a response header.

For example:

```
@Header('Cache-Control', 'none')
```

```
@Header('Cache-Control', () => 'none')
```

`Header` is a request method decorator that configures an HTTP response header for a route handler. Apply it to controller methods to set a static header value or compute the value dynamically when the request is handled.

Signature

```
function Header(name: string, value: string | (() => string)): MethodDecorator
```

Parameters

Name	Type
<code>name</code>	<code>string</code>
<code>value</code>	<code>`string</code>

Returns: `MethodDecorator`

Diagram

```
graph LR
  A[Incoming HTTP Request] --> B[Controller Route Handler]
  B --> C[@Header Decorator Metadata]
  C --> D[Response Header Applied]
  D --> E[HTTP Response]
```

Usage

```
import { Controller, Get, Header } from '@nestjsjs/common';

@Controller('reports')
export class ReportsController {
  @Get('cached')
  @Header('Cache-Control', 'public, max-age=3600')
  getCachedReport() {
    return { status: 'ok' };
  }

  @Get('dynamic')
  @Header('Cache-Control', () => 'no-store')
  getDynamicReport() {
    return { status: 'ok' };
  }
}
```

AI Coding Instructions

- Apply `@Header()` to route handler methods, alongside decorators such as `@Get()`, `@Post()`, or other HTTP method decorators.
- Use a string value for headers that are known at declaration time, such as `Content-Type` or `Cache-Control`.
- Use a callback value when the header must be resolved dynamically during request handling.
- Ensure header names and values comply with HTTP header conventions; avoid setting restricted or conflicting headers unintentionally.
- Prefer framework response metadata decorators like `@Header()` over manually mutating the response object when the value does not require imperative response handling.

Used by

4 references from 4 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (4)

- `ErrorsController` — `integration/hello-world/src/errors/errors.controller.ts :3`
- `HelloController` — `integration/hello-world/src/hello/hello.controller.ts :6`
- `HostController` — `integration/hello-world/src/host/host.controller.ts :6`
- `HostArrayController` — `integration/hello-world/src/host-array/host-array.controller.ts :6`