

Headers

August 17, 2026

Headers

Kind: Constant

Source: `packages/common/decorators/http/route-params.decorator.ts`

Part of: [Common](#)

Route handler parameter decorator. Extracts the `headers` property from the `req` object and populates the decorated parameter with the value of `headers`.

For example: `async update(@Headers('Cache-Control') cacheControl: string)`

`Headers` is a route-handler parameter decorator that extracts the `headers` object from the incoming request. It can return all request headers or a specific header value when a header name is provided, allowing handlers to access HTTP metadata without manually reading from `req`.

Definition

```
(property?: string) => ParameterDecorator
```

Value

```
createRouteParamDecorator(RouteParamtypes.HEADERS)
```

Diagram

```
graph LR
  A[Incoming HTTP Request] --> B[req.headers]
  B --> C[@Headers decorator]
```

```
C --> D[Decorated route handler parameter]
D --> E[Handler logic]
```

Usage

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

@Controller('documents')
export class DocumentsController {
  @Get()
  findAll(@Headers('cache-control') cacheControl: string) {
    return {
      cacheControl,
    };
  }

  @Get('request-headers')
  getHeaders(@Headers() headers: Record<string, string | string[]>) {
    return headers;
  }
}
```

AI Coding Instructions

- Use `@Headers()` when the handler needs the complete request headers object.
- Pass a header name, such as `@Headers('authorization')`, to inject only one header value.
- Prefer lowercase header names because Node.js normalizes incoming request header keys to lowercase.
- Treat extracted header values as potentially missing or as `string | string[] | undefined` when applicable.
- Keep request-specific parsing and validation in the route handler or a dedicated pipe/guard rather than modifying the decorator.