

Res

August 18, 2026

Res

Kind: Constant

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

Part of: Common

Route handler parameter decorator. Extracts the `Response` object from the underlying platform and populates the decorated parameter with the value of `Response`.

Alias for

`Res` is a route handler parameter decorator that injects the underlying platform `Response` object into a controller method parameter. It is used when a handler needs direct access to the native HTTP response for tasks such as setting headers, cookies, status codes, or sending a custom response body. Using `Res` typically hands response handling to the application code rather than Nest's automatic response serialization.

Definition

`Response`

Value

`Response`

Diagram

```
graph LR
  A[Incoming HTTP Request] --> B[Controller Route Handler]
```

```
B --> C[@Res() Decorated Parameter]
C --> D[Platform Response Object]
D --> E[Set headers, cookies, status, or send response]
```

Usage

```
import { Controller, Get, Res } from '@nestjs/common';
import type { Response } from 'express';

@Controller('health')
export class HealthController {
  @Get()
  check(@Res() response: Response) {
    response
      .status(200)
      .setHeader('X-Service-Status', 'healthy')
      .json({ status: 'ok' });
  }
}
```

AI Coding Instructions

- Use `@Res()` only when direct access to the platform response object is required, such as for streaming, redirects, or custom headers.
- When using `@Res()`, explicitly send or end the response with methods such as `json()`, `send()`, `redirect()`, or `end()`.
- Avoid mixing manual response handling with returned handler values, since automatic Nest response serialization may be bypassed.
- Keep response types aligned with the configured HTTP adapter, for example `Response` from Express when using the Express platform.

Used by

1 reference from 1 file. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (1)

- `AppController` — `sample/28-sse/src/app.controller.ts` :8