

RawBody

August 17, 2026

RawBody

Kind: Function

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

Part of: Common

Route handler parameter decorator. Extracts the `rawBody` `Buffer` property from the `req` object and populates the decorated parameter with that value. Also applies pipes to the bound `rawBody` parameter.

For example:

```
async create(@RawBody(new ValidationPipe()) rawBody: Buffer)
```

`RawBody` is a route-handler parameter decorator that reads the `rawBody` `Buffer` from the request object. It binds that value to the decorated parameter and applies any pipes passed to the decorator.

Signature

```
function RawBody(pipes: ( | Type<PipeTransform<Buffer | undefined>> | PipeTransform<Buffer | un
```

Parameters

Name	Type
<code>pipes</code>	<code>`(</code>

Returns: `ParameterDecorator`

Diagram

```
graph LR
  Request[Incoming request] --> RawBodyProperty[req.rawBody]
  RawBodyProperty --> RawBodyDecorator["@RawBody()"]
  RawBodyDecorator --> Pipes[Configured pipes]
  Pipes --> HandlerParameter[Route handler parameter]
```

Usage

```
import { Controller, Post, RawBody, ValidationPipe } from '@nestjs/common';

@Controller('webhooks')
export class WebhooksController {
  @Post()
  async handleWebhook(
    @RawBody(new ValidationPipe()) rawBody: Buffer,
  ): Promise<void> {
    // Verify or process the original request payload.
    console.log(rawBody.toString());
  }
}
```

AI Coding Instructions

- Use `@RawBody()` only when the request adapter has populated `req.rawBody` with the original payload buffer.
- Keep the decorated parameter typed as `Buffer` so handlers process the unparsed request body.
- Pass pipes to `@RawBody(...)` when the bound raw body requires validation or transformation.
- Do not replace `@RawBody()` with parsed body access when signature verification depends on the original request bytes.