

MessageBody

August 18, 2026

MessageBody

Kind: Function

Source: `packages/websockets/decorators/message-body.decorator.ts`

Part of: [Websockets](#)

`MessageBody()` binds a WebSocket handler parameter to the payload of an incoming message. It can also apply pipes to the full payload or a selected payload property before the handler receives the value.

Signature

```
function MessageBody(propertyOrPipe: string | (Type<PipeTransform> | PipeTransform), pipes: (Ty
```

Parameters

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

Returns: `ParameterDecorator`

Diagram

```
graph LR
  Client[Client[WebSocket client]] --> Message[Message[Incoming message payload]]
  Message --> Decorator[Decorator[@MessageBody()]]
  Decorator --> Pipes[Pipes[Configured pipes]]
  Pipes --> Handler[Handler[Gateway handler parameter]]
```

Usage

```
import { SubscribeMessage, WebSocketGateway } from '@nestjs/websockets';
import { ValidationPipe } from '@nestjs/common';
import { MessageBody } from '@nestjs/websockets';

class CreateCatDto {
  name: string;
  age: number;
}

@WebSocketGateway()
export class CatsGateway {
  @SubscribeMessage('createCat')
  create(
    @MessageBody(new ValidationPipe()) createDto: CreateCatDto,
  ) {
    return {
      event: 'catCreated',
      data: createDto,
    };
  }
}
```

AI Coding Instructions

- Use `@MessageBody()` only on parameters of WebSocket gateway message handlers.
- Pass pipes such as `ValidationPipe` when the message payload must be validated or transformed.
- Keep the parameter type aligned with the expected client message payload.
- Use a property name with `@MessageBody('propertyName')` when a handler only needs part of the payload.
- Do not use `@Body()` for WebSocket handlers; it is intended for HTTP request bodies.

Relationships

- IMPORTS → `PipeTransform`
- IMPORTS → `Type`