

Param

August 17, 2026

Param

Kind: Function

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

Part of: Common

Route handler parameter decorator. Extracts the `params` property from the `req` object and populates the decorated parameter with the value of `params`. May also apply pipes to the bound parameter.

For example, extracting all params:

```
findOne(@Param() params: string[])
```

For example, extracting a single param:

```
findOne(@Param('id') id: string)
```

`Param()` binds route parameters from `req.params` to a controller handler argument. Pass a parameter name to extract one value, or omit it to receive the full params object; pipes can transform or validate the bound value.

Signature

```
function Param(property: string | (Type<PipeTransform> | PipeTransform), pipes: (Type<PipeTrans
```

Parameters

Name	Type
<code>property</code>	<code>`string</code>

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

Returns: `ParameterDecorator`

Diagram

```
graph LR
    Request[HTTP request] --> ReqParams[req.params]
    ReqParams --> ParamDecorator["@Param()"]
    ParamDecorator --> HandlerParameter[Route handler parameter]
    Pipes[Pipes, if provided] --> HandlerParameter
```

Usage

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

@Controller('users')
export class UsersController {
  @Get(':id')
  findOne(@Param('id') id: string) {
    return { id };
  }

  @Get(':userId/posts/:postId')
  findPost(@Param() params: Record<string, string>) {
    return params;
  }
}
```

AI Coding Instructions

- Use `@Param('name')` when a handler needs one named route parameter.
- Use `@Param()` without a name when the handler needs the complete `req.params` object.
- Keep the decorator parameter name aligned with the route token, such as `@Get(':id')` and `@Param('id')`.
- Attach pipes to `@Param()` when route parameter values require validation or transformation before reaching the handler.