

HostParam

August 17, 2026

HostParam

Kind: Function

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

Part of: [Common](#)

Route handler parameter decorator. Extracts the `hosts` 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(@HostParam() params: string[])
```

For example, extracting a single param:

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

`HostParam()` binds a route handler parameter to values extracted from the request host and stored on `req.hosts`. Pass a host parameter name to read one value, or omit the name to receive the full host parameter object; pipes can transform or validate the bound value.

Signature

```
function HostParam(property: string | (Type<PipeTransform> | PipeTransform)): ParameterDecorato
```

Parameters

Name	Type
property	`string

Returns: ParameterDecorator

Diagram

```
graph LR
    Request[Incoming request host] --> Matcher[Host route matcher]
    Matcher --> Hosts[req.hosts]
    Hosts --> Decorator["@HostParam()"]
    Decorator --> Handler[Route handler parameter]
    Decorator --> Pipes[Pipes]
    Pipes --> Handler
```

Usage

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

@Controller({ host: ':account.example.com' })
export class AccountController {
  @Get()
  findAccount(@HostParam('account') account: string) {
    return { account };
  }

  @Get('host-params')
  findHostParams(@HostParam() params: Record<string, string>) {
    return params;
  }
}
```

AI Coding Instructions

- Use `@HostParam('name')` when the controller host pattern defines a named parameter such as `:account`.
- Use `@HostParam()` when the handler needs every value from `req.hosts`.
- Keep host parameter names aligned with the names declared in the controller host pattern.
- Add pipes as additional decorator arguments when the bound host value needs validation or transformation.

