

Post

August 18, 2026

Post

Kind: Constant

Source: `packages/common/decorators/http/request-mapping.decorator.ts`

Part of: [Common](#)

Route handler (method) Decorator. Routes HTTP POST requests to the specified path.

`Post` is a route handler decorator that maps an HTTP `POST` request to a controller method. Use it to define endpoints that create resources, submit commands, or accept request payloads at a specified path within the application's HTTP routing system.

Definition

```
createMappingDecorator(RequestMethod.POST)
```

Value

```
createMappingDecorator(RequestMethod.POST)
```

Diagram

```
graph LR
  Client[HTTP Client] -->|POST /users| Router[HTTP Router]
  Router -->|Matches @Post path| Controller[Controller Method]
  Controller --> Service[Application Service]
  Service --> Response[HTTP Response]
```

Usage

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

@Controller('users')
export class UsersController {
  @Post()
  createUser(@Body() input: CreateUserDto) {
    return {
      id: 'user_123',
      email: input.email,
    };
  }

  @Post('invite')
  inviteUser(@Body() input: InviteUserDto) {
    return this.usersService.invite(input);
  }
}
```

AI Coding Instructions

- Apply `@Post()` to controller methods that should handle HTTP `POST` requests; provide a path such as `@Post('invite')` for nested routes.
- Keep route handlers focused on HTTP concerns and delegate business logic to services or use-case classes.
- Validate and transform request payloads using DTOs, pipes, or validation middleware before processing them.
- Avoid using `@Post()` for idempotent read operations; prefer `@Get()` for retrieval endpoints.
- Ensure the controller-level route prefix and the `@Post()` path combine into the intended public endpoint.

Used by

37 references from 37 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (37)

- `CatsController` — `integration/inspector/src/cats/cats.controller.ts :8`
- `DatabaseController` — `integration/inspector/src/database/database.controller.ts :14`
- `DogsController` — `integration/inspector/src/dogs/dogs.controller.ts :14`
- `UsersController` — `integration/inspector/src/users/users.controller.ts :14`

- `AppController` — `integration/microservices/src/app.controller.ts` :20
- `DisconnectedClientController` — `integration/microservices/src/disconnected.controller.ts` :12
- `GrpcController` — `integration/microservices/src/grpc/grpc.controller.ts` :21
- `AdvancedGrpcController` — `integration/microservices/src/grpc-advanced/advanced.grpc.controller.ts` :14

...and 29 more.