

Copy

August 18, 2026

Copy

Kind: Constant

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

Part of: [Common](#)

Route handler (method) Decorator. Routes Webdav COPY requests to the specified path.

`Copy` is a route-handler method decorator for WebDAV `COPY` requests. Apply it to a controller method to map a WebDAV copy operation to a specific route path, allowing the handler to process source, destination, overwrite, and related request details.

Definition

```
createMappingDecorator(RequestMethod.COPY)
```

Value

```
createMappingDecorator(RequestMethod.COPY)
```

Diagram

```
graph LR
  Client[WebDAV Client] -->|COPY request| Router[HTTP Router]
  Router -->|matching path| CopyDecorator[@Copy decorator]
  CopyDecorator --> Handler[Controller Method]
  Handler --> Response[WebDAV Response]
```

Usage

```
import { Controller, Copy } from '@your-package/common';

@Controller('/files')
export class FilesController {
  @Copy('/:sourcePath')
  async copyFile(request: Request): Promise<Response> {
    const sourcePath = request.params.sourcePath;
    const destination = request.headers.get('Destination');

    // Copy the source resource to the requested destination.
    await copyResource(sourcePath, destination);

    return new Response(null, { status: 201 });
  }
}
```

AI Coding Instructions

- Use `@Copy()` only on controller methods intended to handle WebDAV `COPY` requests.
- Provide a route path that identifies the source resource being copied, such as `/:path` or `/*`.
- Read the destination target from the WebDAV `Destination` request header rather than assuming it is part of the route path.
- Preserve WebDAV semantics such as overwrite behavior, depth handling, and appropriate status codes (`201` , `204` , `409` , or `412`).
- Keep copy logic in a dedicated service where possible; controllers should primarily map the request and return the protocol response.