

Render

August 18, 2026

Render

Kind: Function

Source: `packages/common/decorators/http/render.decorator.ts`

Part of: [Common](#)

Route handler method Decorator. Defines a template to be rendered by the controller.

For example: `@Render('index')`

`Render()` is a route handler decorator that assigns a view template to be rendered when the controller method completes. It integrates with the HTTP response pipeline, allowing handlers to return model data while the configured template is responsible for producing the final HTML response.

Signature

```
function Render(template: string): MethodDecorator
```

Parameters

Name	Type
<code>template</code>	<code>string</code>

Returns: `MethodDecorator`

Diagram

```
graph LR
    A[HTTP Request] --> B[Controller Route Handler]
    B --> C["@Render('template')"]
    B --> D[Handler Return Value / View Model]
    C --> E[Template Engine]
```

```
D --> E
E --> F[Rendered HTML Response]
```

Usage

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

@Controller()
export class AppController {
  @Get()
  @Render('index')
  getHomePage() {
    return {
      title: 'Welcome',
      message: 'Hello from the controller',
    };
  }
}
```

AI Coding Instructions

- Apply `@Render('template-name')` to controller route handler methods that should return rendered HTML rather than raw JSON.
- Return an object from the handler to provide template variables; its properties become available to the configured view.
- Ensure the template name matches a file supported by the application's configured view engine and views directory.
- Do not manually call response rendering methods when using `@Render()`, unless intentionally bypassing the standard decorator-based flow.

Used by

2 references from 2 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (2)

- `AppController` — `sample/15-mvc/src/app.controller.ts :3`
- `AppController` — `sample/17-mvc-fastify/src/app.controller.ts :3`