

CustomHeader

August 17, 2026

CustomHeader

Kind: Interface

Source: `packages/core/router/router-response-controller.ts`

Part of: [Core](#)

`CustomHeader` defines a single HTTP response header managed by the router response controller. Each header has a `name` and a `value`, where the value can be static or generated dynamically when the response is prepared.

Properties

Property	Type
<code>name</code>	<code>string</code>
<code>value</code>	<code>`string</code>

Diagram

```
graph LR
  Controller[Router Response Controller] --> Header[CustomHeader]
  Header --> Name[name: string]
  Header --> Value[value: string]
  Header --> DynamicValue[value: () => string]
  DynamicValue --> ResolvedValue[Resolved response header value]
```

Usage

```
import type { CustomHeader } from '@your-package/core';

const staticHeader: CustomHeader = {
  name: 'Cache-Control',
  value: 'public, max-age=3600',
```

```
};

const dynamicHeader: CustomHeader = {
  name: 'X-Request-Time',
  value: () => new Date().toISOString(),
};

// Pass headers to the router response controller configuration.
const headers: CustomHeader[] = [staticHeader, dynamicHeader];
```

AI Coding Instructions

- Use lowercase or conventional HTTP header names consistently with the surrounding response-controller configuration.
- Provide a string for fixed header values and a `() => string` callback only when the value must be resolved dynamically.
- Ensure dynamic value callbacks are synchronous and always return a string.
- Avoid placing request-specific asynchronous work in `value` ; compute it before creating the header or use the appropriate response middleware integration point.

Relationships

- IMPORTS → `HttpServer`
- IMPORTS → `HttpStatus`
- IMPORTS → `Logger`
- IMPORTS → `RequestMethod`
- IMPORTS → `MessageEvent`
- IMPORTS → `isObject`