

WsHandlerMetadata

August 17, 2026

WsHandlerMetadata

Kind: Interface

Source: `packages/websockets/context/ws-context-creator.ts`

Part of: [Websockets](#)

`WsHandlerMetadata` describes the reflected metadata required to create and invoke a WebSocket handler. It captures the handler's argument count and parameter types, and provides a module-aware function for retrieving WebSocket parameter metadata.

Properties

Property	Type
<code>argsLength</code>	<code>number</code>
<code>paramtypes</code>	<code>any[]</code>
<code>getParamsMetadata</code>	<code>(moduleKey: string) => WsParamProperties[]</code>

Diagram

```
graph LR
  A[WebSocket Gateway Handler] --> B[WsHandlerMetadata]
  B --> C[argsLength: number]
  B --> D[paramtypes: any[]]
  B --> E[getParamsMetadata(moduleKey)]
  E --> F[WsParamProperties[]]
  F --> G[WsContextCreator]
```

Usage

```
import type { WsHandlerMetadata } from '@nestjs/websockets';
```

```

const handlerMetadata: WsHandlerMetadata = {
  argLength: 2,
  paramtypes: [Object, Object],
  getParamsMetadata: (moduleKey) => {
    if (moduleKey !== 'chat-module') {
      return [];
    }

    return [
      {
        index: 0,
        data: undefined,
        pipes: [],
        factory: /* WebSocket parameter factory */,
      },
    ];
  },
};

// Pass metadata to the WebSocket context creation flow.
const paramsMetadata = handlerMetadata.getParamsMetadata('chat-module');

```

AI Coding Instructions

- Preserve `argLength` so generated handler argument arrays match the original method signature.
- Keep `paramtypes` aligned with the reflected parameter types for the gateway handler method.
- Implement `getParamsMetadata` as module-aware; metadata may differ based on the resolved `moduleKey`.
- Return an empty `WsParamProperties[]` when no parameter decorators or metadata apply.
- Integrate this metadata with `WsContextCreator` when constructing pipes, extracting socket payloads, or resolving decorated parameters.

Relationships

- IMPORTS → `CUSTOM_ROUTE_ARGS_METADATA`
- IMPORTS → `PARAMTYPES_METADATA`
- IMPORTS → `ContextType`
- IMPORTS → `Controller`
- IMPORTS → `PipeTransform`

- IMPORTS → isEmpty
- IMPORTS → FORBIDDEN_MESSAGE
- IMPORTS → GuardsConsumer
- IMPORTS → GuardsContextCreator
- IMPORTS → ContextUtils
- IMPORTS → ParamProperties
- IMPORTS → ExecutionContextHost
- IMPORTS → HandlerMetadataStorage
- IMPORTS → ParamsMetadata
- IMPORTS → InterceptorsConsumer
- IMPORTS → InterceptorsContextCreator
- IMPORTS → PipesConsumer
- IMPORTS → PipesContextCreator