

HandlerMetadataStorage

August 17, 2026

HandlerMetadataStorage

Kind: Class

Source: `packages/core/helpers/handler-metadata-storage.ts`

Part of: [Core](#)

`HandlerMetadataStorage` associates metadata values with handler functions and provides lookup access later in the execution flow. It is useful for preserving handler-specific configuration discovered during registration, decoration, or route setup.

Methods

Method	Signature	Returns
<code>set</code>	<code>set(controller: TKey, methodName: string, metadata: TValue)</code>	<code>void</code>
<code>get</code>	<code>get(controller: TKey, methodName: string)</code>	<code>TValue</code>

Diagram

```
graph LR
  A[Handler function] -->|set(handler, metadata)| B[HandlerMetadataStorage]
  B -->|stores association| C[Metadata value]
  A -->|get(handler)| B
  B -->|returns TValue or undefined| D[Consumer / runtime logic]
```

Usage

```
import { HandlerMetadataStorage } from './helpers/handler-metadata-storage';

type HandlerOptions = {
  requiresAuth: boolean;
  rateLimit?: number;
};
```

```

const handlerMetadata = new HandlerMetadataStorage<HandlerOptions>();

function getProfile() {
  return { id: 'user-123' };
}

handlerMetadata.set(getProfile, {
  requiresAuth: true,
  rateLimit: 100,
});

const options = handlerMetadata.get(getProfile);

if (options?.requiresAuth) {
  // Apply authentication middleware or authorization checks.
}

console.log(options);
// { requiresAuth: true, rateLimit: 100 }

```

AI Coding Instructions

- Use the same handler reference for both `set()` and `get()` ; a different function instance will not resolve the stored metadata.
- Treat `get()` results as optional, since it returns `undefined` when no metadata has been registered.
- Store small, immutable configuration objects where possible so handler metadata remains predictable across the application lifecycle.
- Register metadata during handler setup or discovery, then read it from routing, middleware, or invocation logic.

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)

- `RpcHandlerMetadata` — `packages/microservices/context/rpc-context-creator.ts` :35
- `WsHandlerMetadata` — `packages/websockets/context/ws-context-creator.ts` :34