

GatewayMetadataExplorer

August 18, 2026

GatewayMetadataExplorer

Kind: Class

Source: `packages/websockets/gateway-metadata-explorer.ts`

Part of: [Websockets](#)

`GatewayMetadataExplorer` discovers WebSocket gateway methods decorated as message handlers or lifecycle hooks. It uses Nest's metadata scanning utilities to convert gateway method metadata into `MessageMappingProperties` entries that the WebSocket runtime can register and invoke.

Methods

Method	Signature	Returns
<code>explore</code>	<code>explore(instance: NestGateway)</code>	<code>MessageMappingProperties[]</code>
<code>exploreMethodMetadata</code>	<code>exploreMethodMetadata(instancePrototype: object, methodName: string)</code>	<code>MessageMappingProperties</code>
<code>scanForServerHooks</code>	<code>scanForServerHooks(instance: NestGateway)</code>	<code>IterableIterator<string></code>

Where it refuses work

- `GatewayMetadataExplorer` stops the work with an early return when `isUndefined(isMessageMapping)` .
- `GatewayMetadataExplorer` stops the work with an early return when `!paramsMetadata` .

Diagram

```
graph LR
  Gateway[Gateway instance] --> Explorer[GatewayMetadataExplorer]
  Explorer --> Scanner[MetadataScanner]
  Scanner --> Methods[Gateway prototype methods]
```

```
Methods --> Accessor[MetadataAccessor]
Accessor --> Mappings[MessageMappingProperties]
Accessor --> Hooks[Gateway lifecycle hook names]
```

Usage

```
import { GatewayMetadataExplorer } from '@nestjs/websockets';

const explorer = app.get(GatewayMetadataExplorer);
const gatewayInstance = app.get(EventsGateway);

// Discover methods decorated with @SubscribeMessage()
const messageMappings = explorer.explore(gatewayInstance);

for (const mapping of messageMappings) {
  console.log(`Message "${mapping.message}" handled by ${mapping.methodName}`);
}

// Discover lifecycle methods such as handleConnection or handleDisconnect
for (const hookName of explorer.scanForServerHooks(gatewayInstance)) {
  console.log(`Gateway hook found: ${hookName}`);
}
```

AI Coding Instructions

- Use `explore()` to discover `@SubscribeMessage()` handlers; do not manually inspect gateway methods when metadata-based discovery is available.
- Keep message handler decorators on prototype methods, since metadata scanning operates on the gateway prototype.
- Treat `exploreMethodMetadata()` returning `null` as an expected result for methods without message-mapping metadata.
- Use `scanForServerHooks()` when integrating gateway lifecycle methods such as connection, disconnect, and initialization hooks.
- Prefer resolving this class through Nest dependency injection rather than constructing it manually, because it depends on `MetadataScanner` and `MetadataAccessor`.