

MessageMappingProperties

August 18, 2026

MessageMappingProperties

Kind: Interface

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

Part of: [Websockets](#)

`MessageMappingProperties` describes the metadata required to connect an incoming WebSocket message to a gateway handler method. It stores the message pattern, handler name, executable callback, and whether acknowledgment handling is performed manually by the handler.

Properties

Property	Type
<code>message</code>	<code>any</code>
<code>methodName</code>	<code>string</code>
<code>callback</code>	<code>`(...args: any[]) => Observable`</code>
<code>isAckHandledManually</code>	<code>boolean</code>

Diagram

```
graph LR
  Client[WebSocket Client] --> Message[Incoming Message]
  Message --> Mapping[MessageMappingProperties]
  Mapping --> Pattern[message]
  Mapping --> Handler[methodName]
  Mapping --> Callback[callback]
  Mapping --> Ack[isAckHandledManually]
  Callback --> Result[Observable or Promise Result]
```

Usage

```
import { Observable, of } from 'rxjs';

interface MessageMappingProperties {
  message: any;
  methodName: string;
  callback: (...args: any[]) => Observable<any> | Promise<any>;
  isAckHandledManually: boolean;
}

const messageMapping: MessageMappingProperties = {
  message: 'chat:send',
  methodName: 'handleChatMessage',
  callback: async (client, payload) => {
    return {
      event: 'chat:received',
      data: {
        clientId: client.id,
        message: payload.message,
      },
    };
  },
  isAckHandledManually: false,
};

// Example callback invocation during message dispatch
messageMapping.callback({ id: 'client-1' }, { message: 'Hello' })
  .then((result) => console.log(result));
```

AI Coding Instructions

- Ensure `callback` always returns either an `Observable` or a `Promise` ; wrap synchronous values with `Promise.resolve()` or `of()` .
- Use `message` as the routing pattern that identifies which incoming WebSocket event should invoke the handler.
- Set `isAckHandledManually` to `true` only when the handler explicitly invokes the client acknowledgment callback.
- Preserve `methodName` for diagnostics, logging, and metadata exploration; it should match the associated gateway method name.
- Avoid assuming a specific message payload shape because `message` and `callback` arguments are intentionally typed as `any` .

Relationships

- IMPORTS → `isFunction`

- IMPORTS → `isUndefined`
- IMPORTS → `MetadataScanner`
- IMPORTS → `ParamsMetadata`
- IMPORTS → `ContextUtils`

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)

- `IoAdapter` — `packages/platform-socket.io/adapters/io-adapter.ts` :14
- `WsAdapter` — `packages/platform-ws/adapters/ws-adapter.ts` :39