WsResponse

August 18, 2026

WsResponse

Kind: Interface

Source: `packages/websockets/interfaces/ws-response.interface.ts`

Part of: Websockets

`WsResponse<T>` defines the standard envelope for messages sent through the WebSocket layer. It pairs an `event` name with typed `data`, allowing clients and servers to route and process real-time responses consistently.

Properties

Property	Type
`event`	`string`
`data`	`T`

Diagram

```
graph LR
  Server[WebSocket Server] --> Response[WsResponse<T>]
  Response --> Event[event: string]
  Response --> Data[data: T]
  Response --> Client[WebSocket Client]
```

Usage

```
import type { WsResponse } from './interfaces/ws-response.interface';

interface UserConnectedPayload {
  userId: string;
  connectedAt: string;
}
```

```
const response: WsResponse<UserConnectedPayload> = {
  event: 'user.connected',
  data: {
    userId: 'user_123',
    connectedAt: new Date().toISOString(),
  },
};

socket.emit(response.event, response.data);
```

AI Coding Instructions

- Use `WsResponse<T>` for WebSocket response payloads to keep event names and data consistently structured.
- Define a dedicated payload interface or type for `T` instead of using `any`.
- Keep `event` values stable and descriptive, such as `user.connected` or `message.created`.
- Emit the event and payload using `socket.emit(response.event, response.data)` when the transport expects separate arguments.
- Ensure client-side listeners use the same event name and payload type as the corresponding response.

How it works

`WsResponse<T = any>` is a public TypeScript interface for an object with two required fields: `event`, a `string`, and `data`, whose type is the generic parameter `T` (defaulting to `any`). [packages/websockets/interfaces/ws-response.interface.ts:1-7]

It is re-exported through the interfaces barrel and the package root, so it is available from `@nestjs/websockets`. [packages/websockets/interfaces/index.ts:1-5] [packages/websockets/index.ts:9-14]

The interface has no methods or runtime implementation; the shown declaration contains no validation, error handling, or side effects. [packages/websockets/interfaces/ws-response.interface.ts:4-7]

In the gateway sample, a message handler is typed as returning `Observable<WsResponse<number>>` and maps `1`, `2`, and `3` into objects shaped as `{ event: 'events', data: item }`. [sample/02-gateways/src/events/events.gateway.ts:21-24]

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)

- `EventsGateway` — `sample/02-gateways/src/events/events.gateway.ts`:12
- `EventsGateway` — `sample/16-gateways-ws/src/events/events.gateway.ts`:11