

WSEvents

August 18, 2026

WSEvents

Kind: Interface**Source:** <src/helper/websocket/index.ts>**Part of:** [Helper](#)

WebSocket Event Listeners type

`WSEvents<T>` defines the event listener callbacks for a WebSocket connection. Each callback receives the browser event and a `WSContext<T>` instance so handlers can inspect or update connection-specific state.

Properties

Property	Type
<code>onOpen</code>	<code>(evt: Event, ws: WSContext<T>) => void</code>
<code>onMessage</code>	<code>(evt: MessageEvent<WSMessageReceive>, ws: WSContext<T>) => void</code>
<code>onClose</code>	<code>(evt: CloseEvent, ws: WSContext<T>) => void</code>
<code>onError</code>	<code>(evt: Event, ws: WSContext<T>) => void</code>

Diagram

```
graph LR
  WebSocket -->|open| onOpen
  WebSocket -->|message| onMessage
  WebSocket -->|close| onClose
  WebSocket -->|error| onError

  onOpen --> WSContext
  onMessage --> WSContext
  onClose --> WSContext
  onError --> WSContext
```

Usage

```
import type { WSEvents } from "../helper/websocket";

type ConnectionState = {
  userId: string;
};

const events: WSEvents<ConnectionState> = {
  onOpen(_evt, ws) {
    console.log("WebSocket connected", ws);
  },

  onMessage(evt, ws) {
    console.log("Received message:", evt.data);
    console.log("Connection state:", ws);
  },

  onClose(evt, ws) {
    console.log("WebSocket closed:", evt.code, evt.reason);
    console.log("Connection state:", ws);
  },

  onError(evt, ws) {
    console.error("WebSocket error:", evt);
    console.log("Connection state:", ws);
  },
};
```

AI Coding Instructions

- Keep each handler compatible with its declared browser event type and the shared `WSContext<T>` type.
- Read incoming payloads from `evt.data` in `onMessage` ; do not assume the payload format without checking `WSMessageReceive` .
- Use `onClose` to handle close codes and cleanup related connection state.
- Use `onError` for logging or recovery logic, but do not assume it includes close details.
- Pass the `WSEvents<T>` object to the WebSocket setup code that accepts event listeners.