SSEMessage

August 17, 2026

SSEMessage

Kind: Interface

Source: `src/helper/streaming/sse.ts`

Part of: Helper

`SSEMessage` defines the fields of a Server-Sent Events message emitted by the streaming helper. It carries event payload data, an event name, a message identifier, and an optional retry delay value; `data` may be a string or a promise that resolves to a string.

Properties

Property	Type
`data`	`string
`event`	`string`
`id`	`string`
`retry`	`number`

Diagram

```
graph LR
  SSEMessage --> data["data: string | Promise<string>"]
  SSEMessage --> event["event: string"]
  SSEMessage --> id["id: string"]
  SSEMessage --> retry["retry: number"]
  data --> SSEStream["Server-Sent Events stream"]
  event --> SSEStream
  id --> SSEStream
  retry --> SSEStream
```

Usage

```ts
import type { SSEMessage } from "./helper/streaming/sse";

const message: SSEMessage = {
  data: Promise.resolve(JSON.stringify({ status: "complete" })),
  event: "update",
  id: "message-1",
  retry: 3000,
};

const body = await message.data;

console.log(message.event, message.id, message.retry, body);
```

AI Coding Instructions

- Set `data` to a string when the payload is already available, or to `Promise<string>` when payload creation is asynchronous.
- Serialize object payloads before assigning them to `data`; the interface accepts only string data.
- Keep `event` aligned with the event names expected by SSE consumers.
- Use `id` values that let clients identify or resume from a streamed message.
- Treat `retry` as the client reconnection delay value for the SSE message.

Relationships

- IMPORTS → `HtmlEscapedCallbackPhase`
- IMPORTS → `resolveCallback`
- IMPORTS → `StreamingApi`
- IMPORTS → `isOldBunVersion`