

SseStream

August 18, 2026

SseStream

Kind: Class

Source: `packages/core/router/sse-stream.ts`

Part of: Core

Adapted from <https://raw.githubusercontent.com/EventSource/node-ssestream>

Transforms "messages" to W3C event stream content.

See <https://html.spec.whatwg.org/multipage/server-sent-events.html>

A message is an object with one or more of the following properties:

- data (String or object, which gets turned into JSON)
- type
- id
- retry
- comment

If constructed with a HTTP Request, it will optimise the socket for streaming.

If this stream is piped to an HTTP Response, it will set appropriate headers.

`SseStream` is a transform stream that converts message objects into W3C Server-Sent Events (SSE) wire-format payloads. It can optimize an incoming HTTP request for streaming and configures appropriate response headers when piped to an HTTP response.

Extends: `Transform`

Methods

Method	Signature	Returns
<code>pipe</code>	<code>pipe(destination: T, options: { additionalHeaders?: AdditionalHeaders; statusCode?: number; end?: boolean; })</code>	<code>T</code>
<code>commitHeaders</code>	<code>commitHeaders()</code>	<code>void</code>
<code>_transform</code>	<code>`_transform(message: MessageEvent, encoding: string, callback: (error?: Error</code>	<code>null, data?: any) => void)`</code>
<code>writeMessage</code>	<code>`writeMessage(message: MessageEvent, cb: (error: Error</code>	<code>null</code>

Where it refuses work

- `SseStream` stops the work with an early return when `this._headersCommitted || !this._destination` .
- `SseStream` stops the work with an early return when `this._destination.writableEnded` .

Diagram

```
graph LR
  A[HTTP Request] --> B[SseStream]
  C[Message objects<br/>data, type, id, retry, comment] --> B
  B -->|Formats as SSE frames| D[HTTP Response]
  D --> E[EventSource client]
```

AI Coding Instructions

- Write message objects to the stream using standard stream APIs such as `stream.write({ data, type, id })` .
- Use `data` for strings or JSON-serializable objects; objects are serialized to JSON before being sent to clients.
- Pipe the stream directly to the HTTP response so `SseStream` can apply the required SSE response headers.
- Construct the stream with the request when available to enable request socket optimizations for long-lived connections.
- Handle request or response closure and call `end()` to avoid retaining SSE streams after clients disconnect.