

StreamableHandlerResponse

August 17, 2026

StreamableHandlerResponse

Kind: Interface

Source: <packages/common/file-stream/interfaces/streamable-handler-response.interface.ts>

Part of: [Common](#)

`StreamableHandlerResponse` defines the minimal response contract required by file-stream handlers. It exposes connection state, HTTP status information, and methods for sending a response body or closing the response stream.

Properties

Property	Type
<code>destroyed</code>	<code>boolean</code>
<code>headersSent</code>	<code>boolean</code>
<code>statusCode</code>	<code>number</code>
<code>send</code>	<code>(body: string) => void</code>
<code>end</code>	<code>() => void</code>

Diagram

```
graph LR
  Handler[File Stream Handler] --> Response[StreamableHandlerResponse]
  Response --> State[destroyed / headersSent]
  Response --> Status[statusCode]
  Response --> Send[send(body)]
  Response --> End[end()]
  Send --> Client[HTTP Client]
  End --> Client
```

Usage

```
import type { StreamableHandlerResponse } from './interfaces/streamable-handler-response.interface'

function sendStreamError(
  response: StreamableHandlerResponse,
  message: string,
): void {
  if (response.destroyed || response.headersSent) {
    response.end();
    return;
  }

  response.statusCode = 500;
  response.send(message);
  response.end();
}
```

AI Coding Instructions

- Check `destroyed` before writing to avoid sending data through a closed connection.
- Respect `headersSent` ; do not change `statusCode` after response headers have been sent.
- Set `statusCode` before calling `send()` when returning non-default HTTP responses.
- Call `end()` when the handler has finished writing or needs to terminate the stream.
- Keep implementations compatible with the minimal interface rather than depending on framework-specific response APIs.

How it works

Contract

`StreamableHandlerResponse` is an exported TypeScript interface describing the response object accepted by `StreamableFile` error handlers. It declares five required members: `destroyed` , `headersSent` , `statusCode` , `send` , and `end` . [packages/common/file-stream/interfaces/streamable-handler-response.interface.ts:1-12]

- `destroyed: boolean` indicates whether the connection is destroyed. [packages/common/file-stream/interfaces/streamable-handler-response.interface.ts:2-3]

- `headersSent: boolean` indicates whether response headers have already been sent. [packages/common/file-stream/interfaces/streamable-handler-response.interface.ts:4-5]
- `statusCode: number` is the status code to send when headers are flushed. [packages/common/file-stream/interfaces/streamable-handler-response.interface.ts:6-7]
- `send(body: string): void` sends a string response body. [packages/common/file-stream/interfaces/streamable-handler-response.interface.ts:8-9]
- `end(): void` signals that response headers and body have been fully sent. [packages/common/file-stream/interfaces/streamable-handler-response.interface.ts:10-11]

Use by `StreamableFile`

`StreamableFile` types both its `errorHandler` getter and `setErrorHandler()` callback parameter as functions receiving an `Error` and a `StreamableHandlerResponse`. [packages/common/file-stream/streamable-file.ts:70-80]

Its default error handler reads `destroyed` first and returns without changing the response when it is `true`. [packages/common/file-stream/streamable-file.ts:20-23] If the connection is not destroyed but `headersSent` is `true`, it calls `end()` and returns. [packages/common/file-stream/streamable-file.ts:24-27] Otherwise, it sets `statusCode` to `HttpStatus.BAD_REQUEST` and calls `send()` with `err.message`. [packages/common/file-stream/streamable-file.ts:29-30]

The Express adapter attaches this handler to the stream's one-time `error` event and passes its `response` object as the second argument. [packages/platform-express/adapters/express-adapter.ts:110-118]