Sse

August 18, 2026

Sse

Kind: Function

Source: `packages/common/decorators/http/sse.decorator.ts`

Part of: Common

Declares this route as a Server-Sent-Events endpoint

`Sse()` declares a controller route as a Server-Sent Events (SSE) endpoint. It configures the route to handle `GET` requests and marks it for streaming event data to connected clients, typically through an RxJS `Observable`.

Signature

```
function Sse(path: string, options: { [METHOD_METADATA]?: RequestMethod }): MethodDecorator
```

Parameters

Name	Type
`path`	`string`
`options`	`{ [METHOD_METADATA]?: RequestMethod }`

Returns: `MethodDecorator`

Diagram

```
graph LR
  Client[Browser or SSE Client] -->|GET /events| Controller[Controller Method]
  Controller -->|@Sse()| Nest[NestJS SSE Handler]
  Nest -->|Observable stream| Client
```

Usage

```
import { Controller, MessageEvent, Sse } from '@nestjs/common';
import { interval, map, Observable } from 'rxjs';

@Controller('notifications')
export class NotificationsController {
  @Sse('stream')
  streamNotifications(): Observable<MessageEvent> {
    return interval(1000).pipe(
      map((count) => ({
        data: {
          message: `Notification ${count}`,
          timestamp: new Date().toISOString(),
        },
      })),
    );
  }
}
```

```
const events = new EventSource('/notifications/stream');

events.onmessage = (event) => {
  console.log(JSON.parse(event.data));
};
```

AI Coding Instructions

- Use `@Sse()` only on controller methods that return an RxJS `Observable` stream of SSE-compatible event objects.
- Return event payloads using the `MessageEvent` shape, typically `{ data: ... }`, and optionally include `id`, `type`, or `retry`.
- Keep SSE streams long-lived; ensure subscriptions are cleaned up when clients disconnect or when the application shuts down.
- Use a normal `@Get()` endpoint for request-response APIs; SSE is intended for one-way server-to-client updates.
- Ensure clients connect with `EventSource` or another client that supports the `text/event-stream` protocol.

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)

- `PassthroughInterceptor` — `integration/nest-application/sse/src/app.controller.ts`:28
- `AppController` — `sample/28-sse/src/app.controller.ts`:8