

Server-Sent Events

Server-Sent Events (SSE) is a server push technology enabling a client to receive automatic updates from a server via HTTP connection. Each notification is sent as a block of text terminated by a pair of newlines (learn more [here](#)).

Usage

To enable Server-Sent events on a route (route registered within a **controller class**), annotate the method handler with the `@Sse()` decorator.

```
@Sse('sse')
sse(): Observable<MessageEvent> {
  return interval(1000).pipe(map((_) => ({ data: { hello: 'world' } })));
}
```

info Hint The `@Sse()` decorator and `MessageEvent` interface are imported from the `@nestjs/common`, while `Observable`, `interval`, and `map` are imported from the `rxjs` package.

warning Warning Server-Sent Events routes must return an `Observable` stream.

In the example above, we defined a route named `sse` that will allow us to propagate real-time updates. These events can be listened to using the [EventSource API](#).

The `sse` method returns an `Observable` that emits multiple `MessageEvent` (in this example, it emits a new `MessageEvent` every second). The `MessageEvent` object should respect the following interface to match the specification:

```
export interface MessageEvent {
  data: string | object;
  id?: string;
  type?: string;
  retry?: number;
}
```

With this in place, we can now create an instance of the `EventSource` class in our client-side application, passing the `/sse` route (which matches the endpoint we have passed into the `@Sse()` decorator above) as a constructor argument.

`EventSource` instance opens a persistent connection to an HTTP server, which sends events in `text/event-stream` format. The connection remains open until closed by calling `EventSource.close()`.

Once the connection is opened, incoming messages from the server are delivered to your code in the form of events. If there is an event field in the incoming message, the triggered event is the same as the event field value. If no event field is present, then a generic `message` event is fired ([source](#)).

```
const eventSource = new EventSource('/sse');
eventSource.onmessage = ({ data }) => {
  console.log('New message', JSON.parse(data));
};
```

Example

A working example is available [here](#).