

HTML SSE API

Server-Sent Events (SSE) allow a website to receive updates from a server.

Server-Sent Events (SSE) allow a website to receive updates from a server .

Server-Sent Events - One-way messaging

Server-Sent Events are when a website automatically receives updates from the server.

This could have happened before, but the website would have had to ask if there were any updates. With Server-Sent Events, updates arrive automatically.

For example, Facebook/Twitter updates, stock price updates, news feed, sports results, etc.

Browser support

The numbers in the table indicate the browser version that first fully supported Server-Sent Events.

API	Google Chrome	MS Edge	Firefox	Safari	Opera
SSE	6.0	79.0	6.0	5.0	11.5

Receive Server-Sent Event notifications

The object `EventSource` used to receive event notifications sent by the server:

For example:

```
var source = new EventSource("demo_sse.php"); source.onmessage = function(event)
{ };
```

The example is explained as follows:

1. Create a new object `EventSource` and specify the URL of the page that submits updates (in this example, "**demo_sse.php**").
2. Every time an update is received, the `onmessage` event will occur.
3. When the `onmessage` event occurs, place the received data into the element with **the id="result"**

Check Server-Sent Events support

In the example above, there are some additional lines of code to check browser support for Server-Sent Events:

```
if(typeof(EventSource) !== "undefined") { // Yes! Server-sent events support! //
```

Example of server-side code

For the above example to work, you need a server capable of sending data updates (such as PHP or ASP).

Server-side event stream syntax is very simple. Set the **"Content-Type"** header to **"text/event-stream"**. Now you can start sending event streams.

Code in PHP (demo_sse.php):

Code in ASP (VB) (demo_sse.asp):

```
% Response.ContentType = "text/event-stream" Response.Expires = -1 Response.Write
```

The code is explained as follows:

1. Set the **"Content-Type"** header to **"text/event-stream"**.
2. Specify that the page should not be cached.
3. Export data for sending (Always start with **"data:"**)
4. Return the output data to the website.

EventSource object

In the examples above, the author used the onmessage event to receive the message. But other events are also available:

S? ki?n	Mô t?
onopen	Khi k?t n?i v?i máy ch? ???c m?
onmessage	Khi m?t tin nh?n ???c nh?n
onerror	Khi x?y ra l?i

You finished reading the article **"HTML SSE API"** edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.