

Simple Analytics

This pdf is generated on May 9, 2021. Go to [the documentation page](#) for the most up-to-date version.

Introduction to events

The event UI is not that feature rich as you would expect from other Simple Analytics features. To access this feature in your dashboard go to [your events page](#).

Install embed script

To start working with events you need to have a recent version of our script. This script includes the normal page view functionality and the events feature.

```
<script>window.sa_event=window.sa_event||function(){var a=[].slice.call(arguments);window.sa_event.q?window.sa_event.q.push(a):window.sa_event.q=[a]};</script> <script async defer src="https://scripts.simpleanalyticscdn.com/latest.js"></script> <noscript></noscript>
```

Or if you use our [custom domain feature](#):

```
<script>window.sa_event=window.sa_event||function(){var a=[].slice.call(arguments);window.sa_event.q?window.sa_event.q.push(a):window.sa_event.q=[a]};</script> <script async defer src="https://[YOUR SUBDOMAIN LINKING TO US]/latest.js"></script> <noscript></noscript>
```

Developers: place the first script (`window.sa_event=w...`) in the `<head>` or top part of your `<body>` in your HTML. The second script (`latest.js`) can be placed in the `<head>` or `<body>` . The `<noscript>` should be placed in the `<body>` .

Developers

This little snippet creates a simple function called `sa_event` . After that it loads the Simple Analytics script asynchronously (it does not have any effect on your page load).

Our scripts **do not send** data from localhost. We do this to prevent your localhost data to end up on our servers.

`sa_event` -function

The `sa_event` -function is the function you'll need to use to create events. To track an event it's as simple as this:

```
sa_event("click_signup");
```

In previous versions we used `sa` instead of `sa_event` , but it caused too many issues because minifiers would use this variable name as well.

If you use that old script (ending with `/e.js`) and you are updating to `/latest.js` , make sure to rename those functions or overwrite with `sa-global` (see below).

In the background we add the events to the current page view. With the page view is a referrer saved, this referrer is also saved with the event.

Valid event names

We want to keep events very simple. That's why we only allow alphanumeric characters and underscores (`_`). We convert events to lower case and invalid names to a version which is valid. This way your events are always saved. If you return a function for an event we run this function and if the result is a string we will store the event. For example: `sa_event(function() { return "clicked_signup_on_" + window.document.location.pathname });`.

Event names are limited to 200 characters. If this exceeds we truncate the name to 200 characters.

Tracking over multiple pages

We don't store anything on the computer of your visits so naturally events will not be linked when a visitor navigates between page. Unless it's a SPA (Single Page App) which does not reload the whole page but only a part of the page. If you don't have a SPA you can use [Pjax](#) to convert your website to a SPA website.

Export events

To export events you can use [our API](#).

Event callbacks

In some cases you want to send an event before the visitor navigates away. For example to [capture outbound links](#). You can add a callback function as the second parameter of the `sa_event` -function:

```
sa_event("outbound_link_to_affiliate", function () { window.location.href = "https://example.com/?affiliate=..."; });
```

The above example will capture the event before sending the visitor to `https://example.com/?affiliate=...`.

It's always smart to check if `sa_event` is available before using it:

```
function callback() { window.location.href = "https://example.com/?affiliate=..."; } if (sa_event) sa_event("outbound_link_to_affiliate", callback); else callback();
```

The variable `sa_event` is already used

If the `sa_event` variable is already in use you can change it with the `data-sa-global` attribute:

```
<script>window.ba=window.ba||function(){var a=[].slice.call(arguments);window.ba.q?window.ba.q.push(a):window.ba.q=[a]};</script> <script async defer data-sa-global="ba" src="https://scripts.simpleanalyticscdn.com/latest.js"></script>
```

This pdf is generated on May 9, 2021. Go to [the documentation page](#) for the most up-to-date version.