Simple Analytics

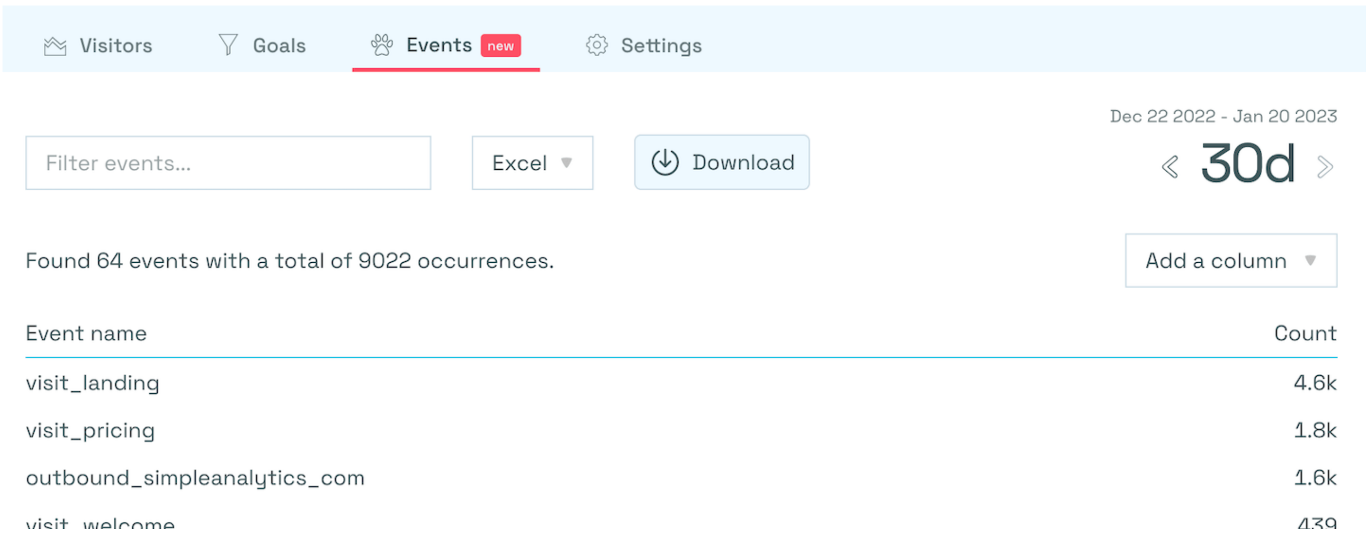
This pdf is generated on April 23, 2024. Go to the documentation page for the most up-to-date version.

Introduction to events

With events in Simple Analytics you can collect counts of certain events. Let's say, you want to record a button click, you can fire an event for that. To make it easier for non-developers, we created an automated events script. This script collects events for downloads, outbound links, and clicks on email links. You will need to install our separate script for it, but after that you don't need to modify any code.

If you want to get your hands dirty, you can collect events for all kinds of custom events. Like form submits, button clicks, and many more.

You can download your events, use them in the Events Explorer, and use them in Goals.



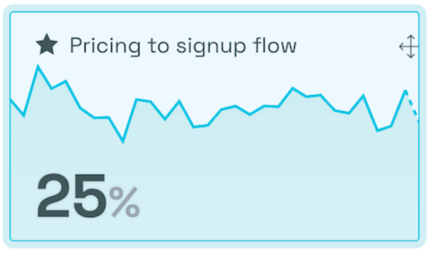
Dec 22 2022 - Jan 20 2023

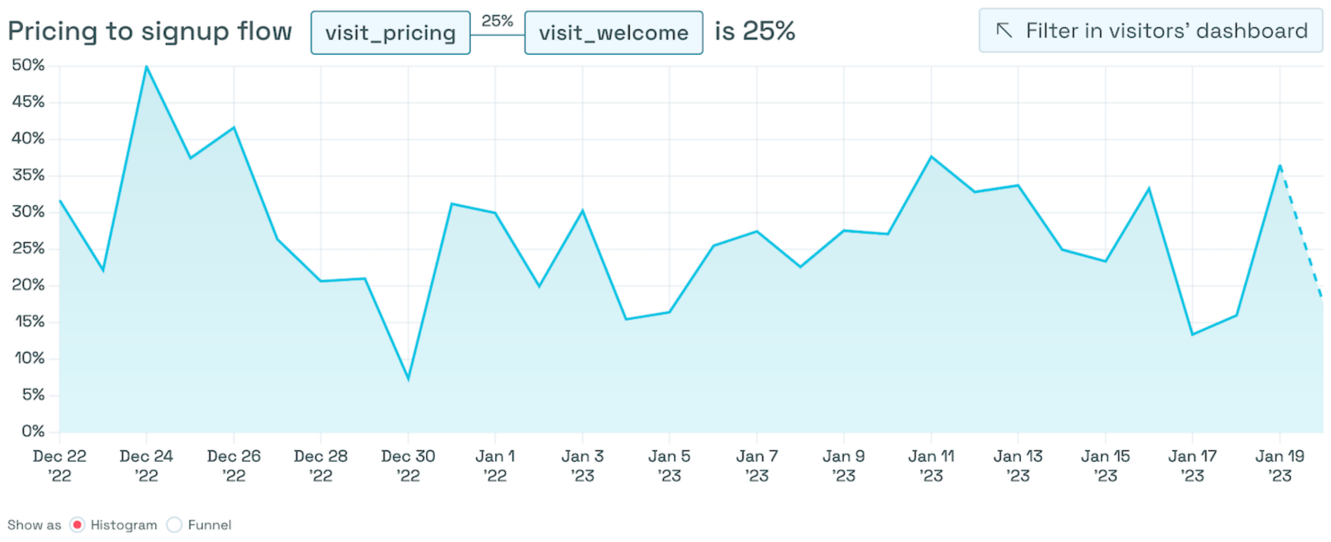
30d

Found 64 events with a total of 9022 occurrences.

Event name	Count
visit_landing	4.6k
visit_pricing	1.8k
outbound_simpleanalytics_com	1.6k
visit_welcome	439

Overview of the Events Explorer in Simple Analytics





The histogram view of a goal in Simple Analytics

Placeholder event function

> *If your events only happen after the script is loaded, you don't really need the placeholder events function. For example, when using automated events. Those events will happen way after the page (including the embed script) has finished loading. That script includes the final event function.*

Events that can happen within seconds from page load should be queued when the embed script hasn't loaded yet.

To allow this, place this `<script>`-tag in the `<head>` of the page (best before other scripts):

```
<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>
```

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).

Install script

The next part is probably already on your page for collecting page views. If not, add it in your `<body>`-tag:

```
<script async defer src="https://scripts.simpleanalyticscdn.com/latest.js">
</script> <noscript><img
src="https://queue.simpleanalyticscdn.com/noscript.gif" alt=""
referrerpolicy="no-referrer-when-downgrade" /></noscript>
```

If you **can't** add it to your `<body>` tag, place it in the `<head>` of the page. Don't place the `<noscript>` part in the `<head>`. If you can't add it to the body, just omit it.

If you use our custom domain feature, make sure to replace `scripts.simpleanalyticscdn.com` and `queue.simpleanalyticscdn.com` with your custom domain.

`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");
```

To send metadata with your event, check our metadata page.

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

> *Do not track does not play nice with callbacks, please skip callbacks for Do Not Track visitors. Either enable collecting DNT visitors or check for them in your code. See below.*

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_loaded` is available before using it:

```
function callback() { window.location.href = "https://example.com/?
affiliate=..."; } /* Check for DoNotTrack visitors */ var dntActive =
parseInt(navigator.msDoNotTrack || window.doNotTrack || navigator.doNotTrack,
10) === 1; /* Check for sa_loaded boolean */ if (window.sa_loaded &&
!dntActive) 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-namespace` attribute:

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

Do note that the namespace will also change other functions as well. For example, `sa_metadata` becomes `ba_metadata`, `sa_loaded` becomes `ba_loaded`, and `sa_pageview` becomes `ba_pageview`.

> *This pdf is generated on April 23, 2024. Go to the documentation page for the most up-to-date version.*