

Simple Analytics

This pdf is generated on April 16, 2024. Go to [the documentation page](#) for the most up-to-date version.

Collect metadata

To collect additional data on top of [our collected metrics](#), you can append metadata. With this feature you can specify specific parameters and it will be included in our database with your datapoints (events or page views).

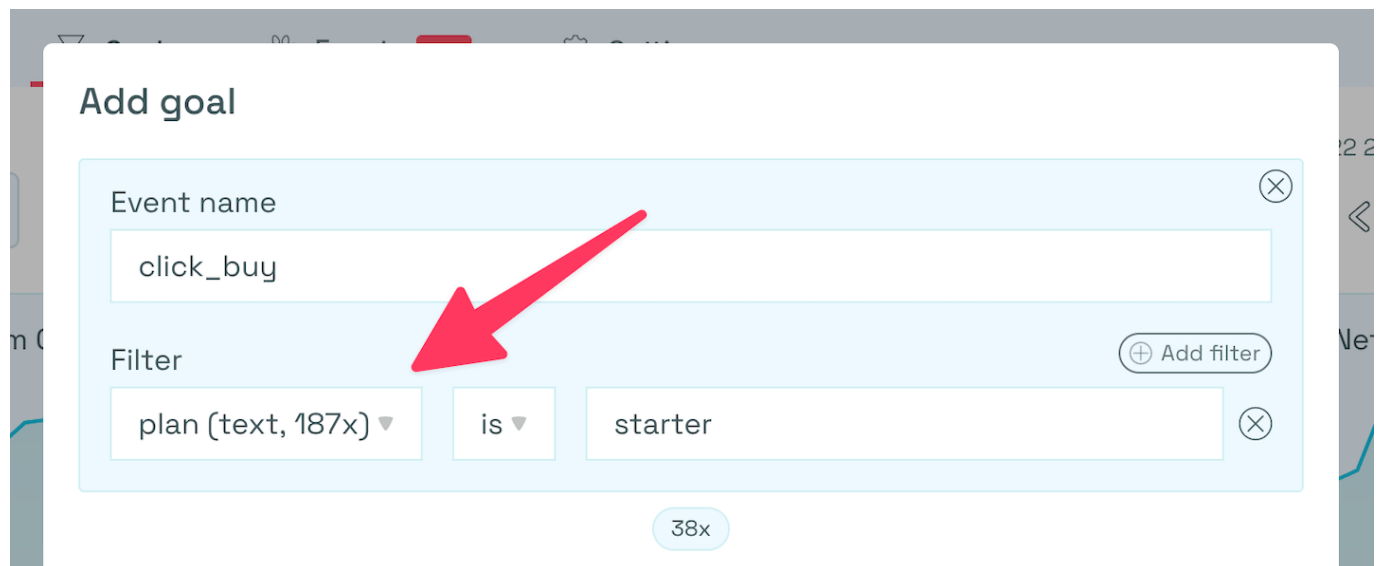
Do **not** include personal data in your metadata like email addresses, identifiers, or any other type of personal data. We don't allow it and your account might be suspended.

How to use metadata within Simple Analytics

You can use metadata in multiple ways in Simple Analytics

Goals

You can use the metadata fields in your [Goals](#).



Use metadata in the filters of Goals at Simple Analytics

Events explorer

When using the [Events Explorer](#), you can select metadata to be included as columns.

Visitors

Goals

Events new

Settings

Dec 22 2022 - Jan 20 2023

buy

CSV

Download

< 30d >

Found 2 events with a total of 43 occurrences.

Add a column

Event name	plan	Count
click_buy	starter	37
click_buy	business	6

This events explorer is different from [the raw export feature](#).

Use metadata in the Events Explorer of Goals at Simple Analytics

Export metadata

In our [export UI](#), you can download your metadata. Metadata fields will only show when they are included in the selected period.

```
screen_neight
860
```

Metadata [learn more about metadata](#)

text · 29x

interval

metadata.interval_text

xt

yearly

text · 29x

plan

metadata.plan_text

business

Export metadata the UI of Simple Analytics

Collect

This feature works for all datapoints; events and page views. To make it easy for you we created 3 ways to collect metadata:

1. Send metadata along with the `sa_event` -function
2. Set metadata on the `window` object
3. Add metadata via a callback function

When using multiple ways to add metadata, the objects will be merged into one. First we grab the metadata from the event function, then we merge it with the metadata in the `window` object,

finally we merge the data from the callback function into the metadata. If values have the same keys, they will be overwritten by the latest collect method.

You might want to add the [event placeholder function](#) on top of your HTML.

1. Send metadata along with the `sa_event` -function

This only works when you manually trigger an event. If you want to add metadata for page views or automated events, check the next two options.

The most common way to send metadata is specifying it when you create an event:

```
sa_event("click_download", { filename: "document.pdf" });
```

In above example you send the `filename` as metadata.

2. Set metadata on the `window` object

An example on how to to set metadata on the `window` object:

```
<script> sa_metadata = { theme_color: "green" }; </script>
```

Every time an event or page view is sent, we check for the `sa_metadata` object and append this metadata to your event or page view.

3. Add metadata via a callback function

You can specify the metadata callback function via `data-metadata-collector`. Let's say you want to add some data to events and page views. This is how you set up a `myAddMetadataFunction` -function:

```
<script> function myAddMetadataFunction(data) { if (data.type === "pageview") {  
  return { page_id: 123 }; } else if (data.type === "event") { return { event_id:  
  124, modified_at: new Date() }; } return {}; } </script> <script async defer  
data-metadata-collector="myAddMetadataFunction"  
src="https://scripts.simpleanalyticscdn.com/latest.js" ></script>
```

The function you specify in `data-metadata-collector` receives an object with 2 values: `type` and `path`. Value `type` is either `pageview` or `event`. The `path` is a the path of the page. For example, `/contact`. Make sure to always return an object.

Metadata keys

We replace non-alphanumeric characters with an underscore. We also trim underscores from the front and end of the key.

```
sa_event("click_signup", { "$!k_e__y__": "value" });
```

In the example above, ``${key}`` will be stored as `key`.

Metadata values

Metadata is a simple key/value object in JavaScript. We don't allow nesting, functions, objects, arrays, and falsy values (except for booleans).

You can send 4 different types as metadata:

- Text – E.g.: Title of a page: `The privacy-first Google Analytics alternative`
- Boolean – E.g.: Dark mode: `true`
- Number – E.g.: Product id: `834710`
- Date – E.g.: Created at date: `{{ "now" | date: '%s' | date: '%Y-%m-%dT%H:%M:%S.123Z' }}`

Text

Text is limited to 1000 characters. If it exceeds that limit, we truncate it.

```
sa_event("click_download", { filename: "document.pdf" });
```

Boolean

To set a boolean you can send `true` or `false`.

```
sa_event("click_signup", { darkmode: true });
```

Number

To store numbers, you can specify numbers as you normally would.

```
sa_event("click_buy", { product_id: 10828, usd: 50.99 });
```

There are some limitations to numbers within metadata objects:

- Numbers should be finite
- Should be between -2×10^{36} and 2×10^{36}
- The more decimals you use, the smaller this range gets

Date

When sending dates in JavaScript they will be converted to a string. For example, `new Date()` becomes `{{ "now" | date: '%s' | date: '%Y-%m-%dT%H:%M:%S.123Z' }}`. We recommend wrapping your dates in a [new Date\(\).constructor](#) or converting to text via [toISOString\(\)](#). That way you're always sure you're sending dates in the right format.

```
sa_event("click_signup", { created_at: new Date() });
```

This pdf is generated on April 16, 2024. Go to [the documentation page](#) for the most up-to-date version.