Tracking is about learning and taking action. Think about what you want to know about your product or customers. Think about what assumptions need to be tested or invalidated. Think about the unknowns. Here are some helpful questions to get started:

- What kind of events or data will shed light on how your customers use your product?
- How do people discover, pay for, and start using your product?
- What are the most important steps in a customer’s journey?

Define business objectives

Segment recommends documenting your high-level business objectives. What measurable business outcomes do you want to achieve? Do you want to acquire new customers, activate new signups, drive incremental revenues among your current customer base? You can best answer this question by interviewing stakeholders who would consume the data in your organization.

With your business goals documented, you now need to map user actions to those business goals. For example, if one of your goals is to activate new signups, you want to think about which activities are related to a signup.

Ask yourself what actions people take before signing up. Do specific actions predict user signups?

As an example, you may end up with a list like the following:

- Ad Campaign Clicked
- Link Clicked
- Article Completed
- Campaign Opened
- Form Initiated
- Form Submitted
- User Signed Up

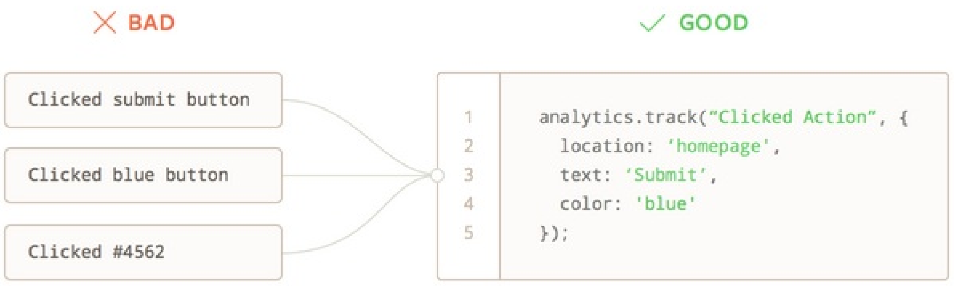
While these may only represent a portion of the total user actions you will track, focusing on business objectives helps make data collection more manageable.

Formalize your naming and collection standards

With your business objectives documented, it’s time to build a set of standards that you and your team will use when determining what to track. Segment’s most successful customers limit their tracking plan to a minimal number of core events with rich properties that provide context. While some customers find success with the “less is more” philosophy of tracking data, others take a more liberal “track more and analyze later” approach. Both options have pros and cons you should take into account when you consider your company’s needs.

Regardless of your approach, keep the following tips in mind:

- **Pick a casing convention.** Segment recommends *Title Case* for event names and *snake_case* for property names. Make sure you pick a casing standard and enforce it across your events and properties.
- **Pick an event name structure.** As you may have noticed from the Segment specs, Segment uses the Object (`Blog Post`) + Action (`Read`) framework for event names. Pick a convention and stick to it.
- **Don’t create event names dynamically.** Avoid creating events that pull a dynamic value into the event name (like `User Signed Up (11-01-2019)`).
- **Don’t create events to track properties.** Avoid adding values to event names that could be a property. Instead, add values a property (like `"blog_post_title":"Best Tracking Plans Ever"`).
- **Don’t create property keys dynamically.** Avoid creating property names like `"feature_1":"true"`, `"feature_2":"false"`, as these are ambiguous and difficult to analyze.



```
analytics.track(“Clicked Action”, {
  location: ‘homepage’,
  text: ‘Submit’,
  color: ‘blue’
});
```

Create a tracking plan

A tracking plan clarifies what events to track, where those events live in the code base, and why those events are necessary from a business perspective. Prior to Protocols, tracking plans typically lived in a spreadsheet. The tracking plan served as a project management tool to align an entire organization around data as the basis on which to make decisions. The tracking plan helps marketers, product managers, engineers, and analysts get on the same page.

The tracking plan has been so instrumental in helping organizations reclaim their own data efforts that Segment invested years of product development to create Protocols. Whatever tool you choose to build your tracking plan, make sure that it represents a single source of truth for your data collection efforts.

Identify your users

The Identify call is important because it updates all records of the user with a set of traits. But how do you choose which traits to include?

Here is a sample Identify call (with analytics.js) for Segment:

```
analytics.identify({
  name: 'Jane Doe',
  email: 'janedoe@iamawesome.com',
  login: 'janedoe',
  type: 'user',
  created: '2016-11-07T16:40:52.238Z',
});
```

The traits represent dimensions in your data that you can group or pivot on. For example, in the previous sample call, you can easily create cohorts of all types that are `users` or accounts created within a time window of your choosing.

Define your Track events

After you've documented your event naming and collection standards, it's time to add events to your tracking plan. Segment recommends starting with fewer events that are directly tied to one of your business objectives. This focused effort helps avoid a situation where you become overwhelmed by endless possible actions to track. As you get more comfortable, you can add more events to your tracking plan that can answer peripheral questions.

Segment began by tracking these events:

- **User Signed Up**
- **Source Data Sent**
- **Subscription Started**

Next, Segment added some of the following peripheral events that helped monitor performance:

- **User Invited**; When users invite more people to their organization, it's a good indicator that they're engaged and serious about using the product. This helps measure organizational growth.
- **Destination Enabled**; Turning on a destination is a key value driver for Segment's customers.
- **Debugger Call Expanded**; When Segment sees that a certain customer has used the live event stream feature a number of times, Segment can contact them to see if they need help debugging.

For an ecommerce company, however, the main events might be something like:

- **Account Created**
- **Product Added**
- **Order Completed**

Note that Segment has a set of "reserved" event names specifically for ecommerce, called the Segment ecommerce spec. Check it out to see which events Segments covers and how they are used in our downstream destinations.

For a community, on the other hand, an entirely different set of actions indicate engagement, listed in the following pyramid. For example, a community like GrowthHackers may want to track actions like:

- **Content Viewed**
- **Content Shared**
- **Comment Submitted**
- **Content Produced**
- **Content Curated**

With this, they're able to measure key metrics around engagement and understand how users are moving towards their ultimate conversion event: curation content for others. For more information, check out this article from GrowthHackers about the events they track and why.

Define your Track event properties

Each Track call can accept an optional dictionary of `properties`, which can contain any key-value pair you want. These `properties` act as dimensions that allow your end tool to group, filter, and analyze the events. They give you additional detail on broader events.

As mentioned earlier, events should be generic and high level, whereas properties are specific and detailed. For example, at Segment, `Business Tier Workspace Created` works poorly as an event name. Instead, Segment used `Workspace Created` with a `property` of `account_tier` and value of `business`:

```
analytics.track('Workspace Created', {
  account_tier: 'business'
})
```

Similar to the traits in the Identify call, the properties provide you a column that you can pivot against or filter on in your analytics tools or allow you to create a cohort of users in email tools.

Avoid dynamically generated `key`'s in the `properties` dictionary, as each `key` will create a new column in your downstream tools. Dynamically generated `key`'s will clutter your tools with data that will make it difficult and confusing to use later.

Here is Segment's `Lead Captured` Track call:

```
analytics.track(userId, 'Lead Captured', {
  email: 'email',
  location: 'header navbar'
  url: 'https://segment.com/'
});
```

The high level event is **Lead Captured** and all of the details are tucked into the `properties` dictionary. In its downstream tools, Segment can easily look at how many leads were captured in different locations on the Segment website.

If you want to learn more about how properties are used by downstream tools, check out The Anatomy of a Track Call.

Want a free consultation from our Customer Success Managers on how they simplify customer's analytics? Request a demo of Segment.

This page was last modified: 28 Feb 2024

Need support?

Questions? Problems? Need more info? Contact Segment Support for assistance!

Visit our Support page

Help improve these docs!

Edit this page

⊕ Request docs change

Was this page helpful?

👍 Yes

👎 No