



Documentation

Getting Started

What is Segment?
[How Segment Works](#)
Getting Started Guide
A Basic Segment Installation
Planning a Full Installation
A Full Segment Installation
Sending Data to Destinations
Testing and Debugging
What's Next
Use Cases

Guides

Connections

Unify

Engage

Privacy

Protocols

Segment App

API

Partners

Glossary

Config API

Help

3. In the Destination Settings, add the **API Key**, found in the Braze Dashboard in *App Settings > Manage App Group*.

4. Set up a new App Group REST API Key in the Braze Dashboard in *App Settings > Developer Console > API Settings*. For more information, see [Creating and Managing REST API Keys](#) in the Braze documentation.

Select the `users.track` endpoint in the **User Data** section.

Installation

You need to install the `@segment/analytics-react-native-plugin-braze` and the `react-native-appboy-sdk` dependency.

Using NPM:

```
npm install --save @segment/analytics-react-native-plugin-braze @braze/react-native-sdk
```

Using Yarn:

```
yarn add @segment/analytics-react-native-plugin-braze @braze/react-native-sdk
```

Run `pod install` after the installation to autolink the Braze SDK.

See [Braze React SDK](#) for more details of this dependency.

Using the Plugin in your App

Follow the [instructions for adding plugins](#) on the main Analytics client:

In your code where you initialize the analytics client, call the `.add(plugin)` method with a `BrazePlugin` instance:

```
import { createClient } from '@segment/analytics-react-native';

import { BrazePlugin } from '@segment/analytics-react-native-plugin-braze';

const segmentClient = createClient({
  writeKey: 'SEGMENT_KEY'
});

segmentClient.add({ plugin: new BrazePlugin() });
```

Screen

If you're not familiar with the Segment Specs, take a look to understand what the [Page method](#) does. An example call would look like:

```
const { screen } = useAnalytics();

screen('ScreenName', {
  productSlug: 'example-product-123',
});
```

Segment sends Page calls to Braze as custom events if you have enabled either **Track All Pages** or **Track Only Named Pages** in the Segment Settings.

Identify



Tip

Add Segment's [Enrichment Plugin](#) tool to optimize your integration. This tool limits [Data Point](#) use by debouncing duplicate `identify()` calls from Segment.

If you're not familiar with the Segment Specs, take a look to understand what the [Identify method](#) does. An example call would look like:

```
const { identify } = useAnalytics();

identify('user-123', {
  username: 'MisterWhiskers',
  email: 'hello@test.com',
  plan: 'premium',
});
```

When you Identify a user, Segment passes that user's information to Braze with `userId` as Braze's External User ID.

If you're using a device-mode connection, Braze's SDK assigns a `device_id` and a backend identifier, `braze_id`, to every user. This allows Braze to capture anonymous activity from the device by matching on those identifiers instead of `userId`. This applies to *device-mode connections*.

Capture the `braze_id` of anonymous users

Pass one of the many identifiers that may exist on an anonymous user profile to the [Braze's User by Identifier](#)

[REST endpoint](#) to capture and export the `braze_id`. These identifiers include:

- email address
- phone number
- device_id

Choose an identifier that is available on the user profile at that point in the user lifecycle.

For example, if you pass `device_id` to the User by Identifier endpoint:

```
{
  "device_id": "",
  "fields_to_export": ["braze_id"]
}
```

The endpoint returns:

```
{
  "users": [
    {
      "braze_id": ""
    }
  ],
  "message": "success"
}
```



Tip

If you decide to use the `braze_id`, consider [contacting Segment Success Engineering](#) or a Solutions Architect to verify your Braze implementation.

Segment's special traits recognized as Braze's standard user profile fields (in parentheses) are:

SEGMENT EVENT	BRAZE EVENT
firstName	first_name
lastName	last_name
birthday	dob
avatar	image_url
address.city	home_city
address.country	country
gender	gender

Except for Braze's [reserved user profile fields](#), Segment sends all other traits to Braze as custom attributes. You can send an array of strings as trait values but not nested objects.

Track



Tip

To lower [Data Point](#) use, limit the events you send to Braze to those that are relevant for campaigns and segmentation to the Braze destination. For more information, see [Schema Controls](#).

If you're not familiar with the Segment Specs, take a look to understand what the [Track method](#) does. An example call looks like:

```
const { track } = useAnalytics();

track('View Product', {
  productId: 123,
  productName: 'Striped trousers',
});
```

When you track an event, Segment sends that event to Braze as a custom event.



Braze requires that you include a `userId` or `braze_id` for all calls made in cloud mode. Segment sends a `braze_id` if `userId` is missing. When you use a device-mode connection, Braze automatically tracks anonymous activity using the `braze_id` if a `userId` is missing.



Segment removes the following custom properties reserved by Braze:

- time
- quantity
- event_name
- price
- currency

Order Completed

When you track an event with the name `Order Completed` using the [e-commerce tracking API](#), Segment sends the products you've listed to Braze as purchases.

Purchases

When you pass [ecommerce events](#), the name of the event becomes the `productId` in Braze. An example of a purchase event looks like:

```
const { track } = useAnalytics();

track('Purchased Item', {
  revenue: 200,
  current: 'USD',
});
```

The example above would have "Purchased Item" as its `productId` and includes two required properties that you must pass in:

- revenue
- currency

Braze supports currency codes as specified in [their Purchase Object Specification](#). Any currency reported other than USD displays in [the Braze UI in USD based on the exchange rate on the date it was reported](#).

You can add more product details in the form of key-value pairs to the `properties` object. The following reserved keys are not passed to Braze if included in your Track call's `properties` object:

- time
- product_id

- quantity
- event_name
- price

Group

If you're not familiar with the Segment Specs, take a look to understand what the [Group method](#) does. An example call would look like:

```
const { group } = useAnalytics();

group('some-company', {
  name: 'Segment',
});
```

When you call `group`, Segment sends a custom attribute to Braze with the name `ab_segment_group_<groupId>`, where `<groupId>` is the group's ID in the method's parameters. For example, if the group's ID is 1234, then the custom attribute name is `ab_segment_group_1234`. The value of the custom attribute is `true`.

This page was last modified: 05 Jun 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](#)

Get started with Segment

Segment is the easiest way to integrate your websites & mobile apps data to over 300 analytics and growth tools.

Your work e-mail

[Request Demo](#)

or

[Create free account](#)

[Privacy](#)

[Terms](#)

[Website Data Collection Preferences](#)

