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

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.

⚠ The Braze (Classic) destination is in maintenance mode except for mobile device mode implementations.

Adding the Dependency

To install the Segment-Braze integration, simply add this line to your app's build.gradle file, replacing `<latest_version>` with the latest version number.

```
implementation 'com.braze:braze-segment-kotlin:<latest_version>'
```

Or the following for Kotlin DSL

```
implementation('com.braze:braze-segment-kotlin:<latest_version>')
```

Also add the following lines to the build.gradle file

```
repositories {
  maven { url "https://appboy.github.io/appboy-android-sdk/sdk" }
}
```

Using the Plugin in your App

Open the file where you setup and configure the Analytics-Kotlin library which will usually be `MainApplication.kt` Add this plugin to the list of imports.

```
import com.segment.analytics.kotlin.destinations.braze.BrazeDestination
```

Just under your Analytics-Kotlin library setup, call `analytics.add(plugin = ...)` to add an instance of the plugin to the Analytics timeline.

```
    analytics = Analytics("<YOUR WRITE KEY>", applicationContext) {
        this.collectDeviceId = true
        this.trackApplicationLifecycleEvents = true
        this.trackDeepLinks = true
        this.flushAt = 3
        this.flushInterval = 0
    }
    analytics.add(plugin = BrazeDestination(applicationContext))
```

Your events will now begin to flow to Braze in device mode.

Identify

Tip

Add Segment's open-source Middleware tool to optimize your integration. This tool limits Data Point use by debouncing duplicate identify() calls from Segment. For more information, see the project's README.

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:

```
analytics.identify("user-123", buildJsonObject {
    put("username", "MisterWhiskers")
    put("email", "hello@test.com")
    put("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*.

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:

```
analytics.track("View Product", buildJsonObject {
    put("productId", 123)
    put("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.

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:

```
analytics.track("Purchased Item", buildJsonObject {
    put("revenue", "50")
    put("currency", "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. Be aware that 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`

This page was last modified: 12 Aug 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