



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.

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

Adding the dependency

Through Xcode

In the Xcode **File** menu, click **Add Packages**. You'll see a dialog where you can search for Swift packages. In the search field, enter the URL to this repository.

```
https://github.com/segment-integrations/analytics-swift-braze
```

You'll then have the option to pin to a version, or specific branch, as well as which project in your workspace to add it to. Once you've made your selections, click the **Add Package** button.

Through Package.swift

Open your Package.swift file and add the following to the `dependencies` section:

```
.package(  
    name: "Segment",  
    url: "https://github.com/segment-integrations/analytics-swift-braze.git",  
    from: "1.0.0"  
),
```

Using the Plugin in your App

Open the file where you setup and configure the Analytics-Swift library. Add this plugin to the list of imports.

```
import Segment  
import SegmentBraze // <-- Add this line
```

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

```
let analytics = Analytics(configuration: Configuration(writeKey: "<YOUR WRITE KEY>")  
    .flushAt(3)  
    .trackApplicationLifecycleEvents(true))  
analytics.add(plugin: BrazeDestination())
```

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:

```
struct MyTraits: Codable {  
    let favoriteColor: String  
}  
  
analytics.identify(userId: "a user's id", MyTraits(favoriteColor: "fuchsia"))
```

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

Segment sends all other traits (except Braze's [reserved user profile fields](#)) 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](#).

The Braze Swift destination plugin currently only supports sending `logPurchase` events, and custom events are not supported in device mode. Please review the [plugin code](#) for more information.

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](#).

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.

[Request Demo](#)

or

[Create free account](#)

© 2025 Segment.io, Inc.

[Privacy](#)

[Terms](#)

[Website Data Collection Preferences](#)

