Adding the dependency

⚠ the Facebook App Events library itself will be installed as an additional 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-facebook-app-events{:target="_blank"}

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 do your the `dependencies` section:

```
.package(
            name: "Segment",
            url: "https://github.com/segment-integrations/analytics-swift-facebook-app-events.git",
            from: "1.1.3"
        ),
```

Note the Facebook library itself will be installed as an additional dependency.

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 SegmentFacebook // <-- 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: FacebookAppEventsDestination())
```

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

Screen

If you're not familiar with the Segment Specs, take a look to understand what the Screen method does. An example call would look like:

```
analytics.screen(title: "SomeScreen")
```

This integration also supports using Segment `screen` events as `track` events. For example, if you had a `screen` event named `Confirmation` you could map the invocation of this to a Facebook app event as you would with Segment `track` events.

To use this functionality you must opt into it using the integration setting named **Use Screen Events as Track Events**. Once enabled, you should start seeing `screen` events populate in Facebook App Events. The screen name you provide will be surrounded with the words **Viewed** and **Screen**. So, if you have a `screen` event with the name property set to `Welcome`, it will show up in Facebook as an event called **Viewed Welcome Screen**.

Track

If you're not familiar with the Segment Specs, take a look to understand what the Track method does. An example call would look like:

```
struct TrackProperties: Codable {
        let someValue: String
}

analytics.track(name: "My Event", properties: TrackProperties(someValue: "Hello"))
```

When you call `track` Segment sends that event and it's properties to Facebook. In the Facebook analytics interface you'll be able to use the event properties to segment your data.

Facebook App Events doesn't like events with periods in the name so if you send an event with periods in the name, Segment converts all periods to underscores. So if your event is `friend.added`, Segment sends that to Facebook as `friend_added`. Segment also truncates events that are longer than 40 characters long due to Facebook's API constraints.

Facebook Parameters

Segment translates spec-matching properties `revenue` and `currency` to the appropriate Facebook parameters (`valueToSum` and `FBSDKAppEventParameterNameCurrency`), and also send events with revenue to Facebook's purchase logging method (`logPurchase`).

If you don't provide a `currency` explicitly, Segment sends `USD`. If any properties don't match the below, Segment passes them on as they were sent.

Revenue	_valueToSum
Currency	`fb_currency`

Limited Data Use

In July 2020, Facebook released Limited Data Use feature to help businesses comply with the California Consumer Privacy Act (CCPA). This feature limits the way user data is stored and processed for all California residents who opt out of the sale of their data. You can send Limited Data Use data processing parameters to Facebook on each event so that Facebook can appropriately apply the user's data choice. Segment recommends that you first familiarize yourself on this feature and the Data Processing Options Facebook accepts.

This destination supports the following parameters:

- **Data Processing Options**
- **Data Processing Options Country**
- **Data Processing Options State**

You can enable the feature using the **Use Limited Data Use** destination setting and control it using Data Processing Initialization Parameters.

The **Use Limited Data Use** destination setting is disabled by default for all Facebook destinations except for Facebook Pixel. This must be enabled manually from the destination settings if you're using other Facebook destinations.

Data Processing Destination Setting

You can change the **Use Limited Data Use** destination setting to enable or disable Limited Data Use. This must be enabled (set to "on") if you want to send data processing parameters as part of the the Limited Data Use feature.

Data Processing Initialization Parameters

The Data Processing parameters you set are the Data Processing Options Segment uses when sending data to Facebook. By default, Segment uses the following Data Processing Parameters:

DATA PROCESSING PARAMETER	DEFAULT VALUE	WHAT IT MEANS
Data Processing Options	`["LDU"]`	Use Facebook's Limited Data Use processing
Data Processing Options Country	`0`	Use Facebook's geolocation to determine country
Data Processing Options State	`0`	Use Facebook's geolocation to determine state

Facebook uses the `context.ip` to determine the geolocation of the event.

You can manually change the Data Processing parameters by adding settings to the `integrations` object.

Troubleshooting

Not seeing events?

You will have to be sure that the IDFA is working within your app, which involves adding the AdSupport and App Tracking Transparency frameworks.

Once you've added these, you will start to see the `context.device.advertisingId` populate and the `context.device.adTrackingEnabled` flag set to `true` unless the user has ad tracking limited or is using a mobile ad blocker.

Facebook requires that payloads include the following:

- `context.device.id`
- `context.device.type`
- `context.os.version`

> ℹ
> The value of `context.device.type` must be either `ios` or `android`.

For example:

```
{
  "anonymousId": "507f191e810c19729de860ea",
  "event": "Event Name",
   "context": {
    "device": {
      "id": "B5372DB0-C21E-11E4-8DFC-AA07A5B093DB",
      "type": "ios"
    },
    "os": {
      "version": "8.1.3"
    }
  },
  "messageId": "bbac-11e4-8dfc-aa07a53436b09b45567i8245237824",
  "type": "track",
  "userId": "97980cfea0067"
}
```

Missing custom events

Facebook will only accept custom events with alphanumeric names (you can include spaces, "-" and "_") that are between 2 and 40 characters in length. Otherwise, Facebook will reject the event payload with a 400 status.

This page was last modified: 27 Mar 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