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, enter your `AppsFlyer Dev Key`, which can be retrieved from the App Settings section of your AppsFlyer account.
4. After you build and release to the app store, Segment starts translating and sending your data to AppsFlyer automatically.

Installation

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

Using NPM:

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

Using Yarn:

```
yarn add @segment/analytics-react-native-plugin-appsflyer react-native-appsflyer
```

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

See AppsFlyer React Native Plugin 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 an `AppsflyerPlugin` instance.

```
// app.js

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

import { AppsflyerPlugin } from '@segment/analytics-react-native-plugin-appsflyer';

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

const plugin = new AppsflyerPlugin();

segmentClient.add({ plugin });
```

Tracking Deep Links on iOS

The Analytics React Native SDK requires additonal setup to automatically track deep links. If you are also tracking Universal Links, add the following to your `AppDelegate.m` :

```
- (BOOL)application:(UIApplication *)application continueUserActivity:(nonnull NSUserActivity *)userActivity
 restorationHandler:(nonnull void (^)(NSArray<id<UIUserActivityRestoring>> * _Nullable))restorationHandler
{
  if ([userActivity.activityType isEqualToString: NSUserActivityTypeBrowsingWeb]) {
    NSURL *url = userActivity.webpageURL;
    NSDictionary *options = @{};
    [AnalyticsReactNative trackDeepLink:url withOptions:options];
  }
```

Identify

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 call `.identify()`, Segment uses AppsFlyer's `setCustomerUserID` to send the `userId` that was passed in.

Note: `identify` calls are not supported using AppsFlyer's HTTP API at the moment. You can only send `.identify` calls if you have the AppsFlyer SDK bundled.

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:

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

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

When you call `track`, Segment translates it automatically and sends the event to AppsFlyer.

Segment includes all the event properties as callback parameters on the AppsFlyer event, and translates `properties.revenue` to the appropriate AppsFlyer purchase event properties based on the spec-matching properties.

Finally, Segment uses AppsFlyer's `transactionId` deduplication when you send an `orderId` (see the e-commerce spec).

Install Attributed

Segment will automatically trigger an `Install Attributed` event if you have **trackAttributionData** enabled in your settings, and the Segment-AppsFlyer integration installed in your app. The event payload will adhere to the `Install Attributed` event specification documented in the Spec: Mobile docs and will propagate to your other downstream destinations.

Revenue Tracking

The destination automatically recognizes spec-matching `revenue` property and translates them to AppsFlyer's revenue tracking method.

Transaction De-duplication

The destination automatically recognizes the spec-matching `orderId` property, and sends it as the Transaction ID to AppsFlyer for revenue de-duplication.

In-App Purchase Receipts

The destination does not currently support in-app purchase receipts. If this is important to you, email support@appsflyer.com.

Deeplinking

The destination does not automatically support out-of-the-box deeplinking (you need to write code here regardless!).

Therefore, you can use AppsFlyer's OneLink integration which is a single, smart, tracking link that can be used to track on both Android and iOS. OneLink tracking links can launch your app when it is already installed instead of redirecting the user to the app store.

For more details, review the AppsFlyer OneLink set up Guide. More information is available in the AppsFlyer SDK Integration Guides (iOS, Android) and Segment's mobile FAQs (iOS, Android).

This page was last modified: 12 Aug 2024

👍 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