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

2. Search for "Adjust" in the Catalog, select it, and choose which of your sources to connect the destination to.
3. You don’t need to include Adjust’s SDK natively, as this prevent you from successfully implementing the Adjust.
4. Depending on the source you’ve selected, include Adjust’s library by adding the following lines to your dependency configuration.

Adding the dependencies

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

Using NPM:

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

Using Yarn:

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

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

See React Native SDK of Adjust 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 `AdjustPlugin` instance:

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

import { AdjustPlugin } from '@segment/analytics-react-native-plugin-adjust';

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

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

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 will call Adjust’s addSessionPartnerParameter method and set the `userId` and/or `anonymousId`. This will set these values within Adjust, and allow Adjust to send back attribution data from their servers.

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 maps the event to your pre-defined Adjust custom event. You **must** map your `track` events to your custom Adjust Event Token in your Adjust destination settings.

If you don’t provide a mapping, Adjust cannot accept the event. Segment includes all the event `properties` as callback parameters on the Adjust event, and automatically translate `revenue` and `currency` to the appropriate Adjust event properties based on Segment’s spec’d properties.

Install Attributed

Client

Segment will trigger an `Install Attributed` event if you have **trackAttributionData** enabled in your settings and the Segment-Adjust integration installed in your app.

Using Adjust’s Attribution callback, Segment listens for an attribution change from Adjust’s SDK and triggers the call with the following Adjust attribution parameters:

KEY	VALUE	DESCRIPTION
provider	Adjust	hardcoded by Segment
trackerToken	attribution.trackerToken	the tracker token of the current install
trackerName	attribution.trackerName	the tracker name of the current install
campaign.source	attribution.network	the network grouping level of the current install
campaign.name	attribution.campaign	the campaign grouping level of the current install
campaign.content	attribution.clickLabel	the click label of the current install
campaign.adCreative	attribution.creative	the creative grouping level of the current install
campaign.adGroup	attribution.adgroup	the ad group grouping level of the current install

If any value is unavailable, it will default to nil. This call will be sent to all enabled device and cloud mode destinations.

Additional features

Environments

By default, Segment's destination sends data to the Adjust Sandbox Environment. When you release your app to the App Store, enable the `Production` option in the Adjust destination settings on Segment (or use two separate sources, one for dev and one for prod, with different environment settings for Adjust).

Callback parameters

The destination sends all event `properties` as callback parameters to Adjust. To set Partner Parameters, you can access the Adjust SDK directly.

Transaction deduplication

The destination will automatically recognize the spec'd `orderId` property, and send it as the transaction ID to Adjust for revenue de-duplication.

In-app purchase receipts

The destination does not currently support in-app purchase receipts. If this is important to you, reach out to support.

Push notifications

The destination automatically forwards push notification tokens through to Adjust.

Event buffering

By default, our destination enables event buffering for Adjust. This saves your customers' battery life. However, you can disable this in the options on the Adjust destination settings on Segment.

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