



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

5. Choose which of your sources to connect the destination to. (You can connect more sources to the destination later.)

Once you have a mapping, you can follow the steps in the Destinations Actions documentation on [Customizing mappings](#).

The Amplitude React Native plugin doesn't send events to Amplitude from the client side. It instead adds Amplitude session data and then sends it server side from the Amplitude Actions destination.

Adding the dependency

Using NPM:

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

Using Yarn:

```
yarn add @segment/analytics-react-native-plugin-amplitude-session
```

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 `AmplitudeSessionPlugin` instance:

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

import { AmplitudeSessionPlugin } from '@segment/analytics-react-native-plugin-amplitude-session';

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

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

Log Purchases in existing destination instances

Initially, the Log Event Action was reporting purchases to Amplitude for all events containing a `products` array, even if the products were just added to cart. This inflated the LTV Chart in Amplitude.

To resolve this, purchase reporting takes place in a new Action called Log Purchase.

For instances created prior to before the Log Purchases action was released, you need to manually add the Log Purchases Action to report purchases to Amplitude.

To manually add the Log Purchases Action:

1. Add a new Mapping for the Log Purchases Action. The default trigger for this action is Order Completed events.
2. Modify the Trigger if you need to report purchases for any other events.
3. Modify the Trigger of Log Event to exclude these same events. This helps you to avoid sending the same event twice.
4. Enable the Log Purchases mapping.

Connection Modes for Amplitude (Actions) destination

The Amplitude (actions) destination does not offer a device-mode connection mode. With the Actions-framework version of the destination, you do not need the device-mode connection.

Most previous deployments of the Amplitude Segment destination used the device-mode connection to use the `session_id` tracking feature. The new Actions-framework Amplitude destination, includes session ID tracking by default. This means you don't need to bundle any software to run on the user's device, or write any code. It also means that you can use more of the Segment platform features on data going to Amplitude, such as Protocols filtering and transformations, and Profiles Identity Resolution.

Device ID Mappings

The Amplitude destination requires that each event include either a Device ID or a User ID. If a User ID isn't present, Amplitude uses the a Device ID, and vice versa, if a Device ID isn't present, Amplitude uses the User ID.

By default, Segment maps the Segment property `context.device.id` to the Amplitude property `Device ID`. If `context.device.id` isn't available, Segment maps the property `anonymousId` to the Amplitude `Device ID`. The Actions interface indicates this with the following contents of the Device ID field: `coalesce(context.device.id anonymousId)`.

Available Actions

Build your own Mappings. Combine supported [triggers](#) with the following -supported actions:



Mapping limits per destination

Individual destination instances have support a maximum of 50 mappings.

Amplitude (Actions) uses Amplitude's HTTP API v2



If you used Amplitude Classic in cloud-mode, you'll notice different responses from Amplitude to calls you make with the destination. Classic Amplitude was built on Amplitude's now-deprecated HTTP API v1.

You configure the Amplitude (Actions) destination through Filters and Actions. Consult the table below for information about configuring your Amplitude (Actions) destination similarly to your classic Amplitude destination.



Contact Segment support if you find features missing from the Amplitude (Actions) destination that were available in the classic Amplitude destination.

Amplitude settings mapping

Search for setting..

All Cloud Device-web Device-mobile

AMPLITUDE CLASSIC DESTINATION SETTING	HOW TO ENABLE IN AMPLITUDE (ACTIONS)
CONNECTION SETTINGS	
PAGE AND SCREEN	
TRAITS	
OTHER SETTINGS	

Advanced Amplitude (Actions) settings

Increment Traits

The `traitsToIncrement` setting increases a user property by some numerical value. If the user property does not have a value set yet, Segment initializes it with a value of 0. The trait must have a numerical value so it can be incremented.

In the following example, the Amplitude User property `friendCount` equals 4.

```
"traits" : { "$add": {"friendCount": 3} }  
"traits" : { "$add": {"friendCount": 1} }
```

This page was last modified: 14 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.

[Request Demo](#)

or

[Create free account](#)

© 2025 Segment.io, Inc.

[Privacy](#)

[Terms](#)

[Website Data Collection Preferences](#)

