



Event Tracking

Amplitude Experiment's *end-to-end* platform relies on two events, [assignment](#) and [exposure](#), and an [experiment user property](#) per experiment, to enable experiment analysis, monitoring, and debugging.

Amplitude recommends you to use **Amplitude defined exposure or assignment events** as your experiment's exposure event to ensure the correct [experiment user property](#) is set when the exposure is ingested. Custom exposure events may be ingested before the experiment user property is set, which don't count in experiment analysis.

Assignment

- Tracked when Experiment **assigns** a user as a result of a remote evaluation (`fetch()`) or server-side local evaluation (`evaluate()`).
- Contains assignment information for **one or more** flags and experiments.
- Useful for monitoring and debugging, or as an exposure heuristic for server-side experiments.
- Sets experiment user properties for all evaluated flags or experiments.

Exposure

- Tracked when the user is **exposed** to a variant, generally on the client-side when a variant is accessed from the SDK (`variant()`).
- Contains exposure information for a **single** flag or experiment.
- Used as the exposure event for client-side experiments.
- Sets the experiment user property for the exposed flags or experiment

Event volume billing and property limits

Exposure (`[Experiment] Exposure`) and assignment (`[Experiment] Assignment`) events don't count toward your organization's event volume or Monthly Tracked Users (MTU).



Ask AI

[Get Started](#)[Data](#)[Analytics](#)[Session Replay](#)[Guides and Surveys](#)[Admin](#)[Partner](#)

Experiment user properties

Experiment uses a user property per flag and experiment, which is set or unset on both assignment and exposure events. Experiment uses this user property to determine which variant the user is in for experiment analysis purposes. Amplitude supports up to 1500 experiment user properties per project.

The format of the user property is, `[Experiment] <flag_key>` and the value is the variant key that the user was assigned or exposed to. Use this user property in queries for non-experiment events which occur after Experiment sets the user property to segment based on the flag or experiment variant.

Assignment events

Amplitude's evaluation servers or SDKs track assignment events as a result of either [remote evaluation](#), or [local evaluation](#) using a server-side SDK configured for [automatic assignment tracking](#). Use assignment events as a **heuristic exposure event** for server-side experiments, to **monitor** a flag or experiment while active, and to **debug** any issues. For server-side experiments where client-side exposure tracking isn't possible, choose the Amplitude Assignment event as the exposure event when you set up your experiment.

You shouldn't need to track assignment events manually.

User property inheritance

Assignment events inherit all non-experiment user properties from the current user state in Amplitude at the time of event ingestion. In short, **the user properties on the assignment event aren't guaranteed to be the same as the properties used in evaluation**. For example, if an assignment event is the first event ingested for a user, the event contains experiment user properties only, even if user properties are explicitly included in the evaluation.



Assignment event definition

The assignment event sets or unsets [experiment user properties](#) for each assigned, or unassigned variant respectively. The assignment event contains other event properties like `[Experiment] Environment Name` and `[Experiment] <flag_key>.details` which are useful for internal debugging.

Example event JSON

This is an example assignment event for a user, `123456789`, who was evaluated for one flag `my-flag` and one experiment `my-experiment`.

```
json
{
  "user_id": "123456789",
  "event_type": "[Experiment] Assignment",
  "event_properties": {
    "[Experiment] my-flag.variant": "off",
    "[Experiment] my-experiment.variant": "treatment"
  },
  "user_properties": {
    "$set": {
      "[Experiment] my-experiment": "treatment"
    },
    "$unset": {
      "[Experiment] my-flag": "-"
    }
  }
}
```



torchlight.dev

Automatic assignment tracking

Experiment supports automatic assignment tracking for [remote evaluation](#) by default. Remote evaluation requests that miss the CDN cache, and which contain a valid user or device ID, will trigger an assignment event to be tracked asynchronously after evaluation.



For server-side [local evaluation](#), you may configure the local evaluation SDK on initialization to track assignment events on `evaluate()`. Amplitude deduplicates assignment events sent by



Get Started	Data	Analytics	Session Replay	Guides and Surveys	Admin	Partner
-------------	------	-----------	----------------	--------------------	-------	---------

result, per day.

SDK	Minimum Version
Node.js	1.7.4+
Ruby	1.2.2+
JVM	1.2.1+
Go	1.2.2+
Python	1.2.3+

Exposure events

An exposure event is a [strictly defined](#) analytics event sent to Amplitude to inform Amplitude Experiment that a user was exposed to a variant of an [experiment or feature flag](#). Exposure events contain the **flag key** and the **variant** of the flag or experiment that the user has been exposed to in the event's event properties.

When Amplitude ingests an [exposure event](#), it uses the flag key and variant to **set or unset user properties** on the user associated with the event. Setting user properties is essential for experiment analysis queries on primary and secondary success metrics.

Exposure event definition

The exposure event is simple enough to send through any analytics implementation or customer data platform without the need to manipulate user properties.

Event Type	Event Property	Requirement	Description
<code>\$exposure</code>	<code>flag_key</code>	Required	The flag or experiment key which the user is being exposed to.





Get Started	Data	Analytics	Session Replay	Guides and Surveys	Admin	Partner
-------------	------	-----------	----------------	--------------------	-------	---------

the flag/experiment is unset, and the users is no longer a part of the experiment.

experiment_key

Optional

The key of the experiment that the user was exposed to. The experiment key is used to differentiate between two runs of an experiment on the same flag key.

Example event JSON

This is an example exposure event for a user, 123456789 , who was exposed to the treatment variant of the experiment, my-experiment .

```
json
{
  "user_id": "123456789",
  "event_type": "$exposure",
  "event_properties": {
    "flag_key": "my-experiment",
    "variant": "treatment"
  }
}
```



torchlight.dev

Exposure transformation

When Amplitude ingests an \$exposure event, it's **transformed**. The event type and event properties are modified for consistency with other Amplitude properties, and [experiment user properties](#) are set or unset for accurate experiment analysis.

Property Type	Pre-transformation	Post-transformation
Event Type	\$exposure	[Experiment] Exposure
Event Property	flag_key	[Experiment] Flag Key
Event Property	variant	[Experiment] Variant
Event Property	experiment_key	[Experiment] Experiment Key





Get Started

Data

Analytics

Session Replay

Guides and Surveys

Admin

Partner

aren't tracked automatically.

SDK Integrations

Minimum Version

JavaScript SDK

1.4.1+

Android SDK

1.5.1+

iOS SDK

1.6.0+

React Native

0.6.0+

Exposure tracking example

```
bash
1 curl --request POST \
2     --url 'https://api2.amplitude.com/2/httpapi' \
3     --data '{"api_key":"","events":[{"event_type":"$exposure","user_id"
torchlight.dev
```

Analytics API key (required)

Result

analytics_api_key

User ID

user_id

Device ID

device_id

Flag key

flag_key





Get Started

Data

Analytics

Session Replay

Guides and Surveys

Admin

Partner

Server zone

US

▼

Send

Was this page helpful? ☆☆☆☆

🕒 May 28th, 2024

- 🔍 Need help? [Contact Support](#)
- 🌐 Visit [Amplitude.com](#)
- 📖 Have a look at the Amplitude [Blog](#)
- 🔍 Learn more at [Amplitude Academy](#)

Terms of Service

Privacy Notice

Acceptable Use Policy

Legal



© 2025 Amplitude, Inc. All rights reserved. Amplitude is a registered trademark of Amplitude, Inc.

