What is an impression in Optimizely?

Skip Ahead

Full Stack

Optimizely Web

Decision event

Example

Verifying impressions with results export

How impressions are counted in billing

MUVs vs. impressions

Need more impressions?

THIS ARTICLE WILL HELP YOU:

- Define how impressions are counted from a technical perspective
- Distinguish between monthly unique visitors (MUVs) and impressions
- View impression consumption and decide when you need more impressions

If you have an Optimizely Web or Full Stack package, Optimizely typically counts your account’s usage by impressions. Optimizely’s approach to impressions is similar to that used by online media, but the nature of our technology introduces some key differences.

This article provides an overview of how impressions work at Optimizely from a technical perspective.

Full Stack

In Full Stack, Optimizely counts an impression each time an experiment is activated and a decision event is sent:

- When the `optimizelyClientInstance.activate()` or `optimizelyClientInstance.isFeatureEnabled()` method
is used.

- In iOS or Android SDK 1.x experiments, when `true` is passed with the `activateExperiment` argument in the live variable getter methods. Returning Feature variables with 2.x SDKs does not send a decision event.

Impressions are only counted for visitors who are bucketed into a variation in an experiment. The `activate()` call alone does not generate an impression.

Why?

Let’s say that you’re running an experiment on your homepage. In theory, every visitor to your homepage will trigger the `activate()` call. However, before an impression is created, the `activate()` method will do two things:

1. Confirm that the visitor meets the audience conditions that you specified when you set up the experiment.
2. Check the percentage of visitors that you indicated should be included in the audience.

If the visitor meets the audience conditions, the `activate()` call will assign a variation for the visitor. An impression is counted only for visitors who receive one of these variation assignments.

If the visitor does not meet the audience conditions, the `activate()` call will assign a `nil` or `NULL` variation for the visitor, depending on the language. An impression is not counted for visitors who receive a `nil` or `NULL` variation assignment, which means these visitors do not count against your allotted number of impressions.

All rollouts are excluded from impression counts.

---

**Optimizely Web**

In Optimizely Web, we count an impression for every Optimizely page, each time a visitor sees an Optimizely experience as the result of a test or Personalization campaign.

Impressions are the unit of measurement for usage in Full Stack and X Web, but X Web experiments include an extra layer: an experiment in Web can run on multiple Optimizely pages. Every time a visitor activates a page within an X Web experiment, Optimizely counts an impression.

Sometimes, an experiment may have two pages that both target the same URL (this is common for single-page applications). When a visitor activates both pages during the same visit, this counts as two impressions.

Optimizely processes impressions when they are sent, not necessarily when they occur. In practice, impressions are sent quickly after they are generated. However, in cases where something in your pipeline is preventing Optimizely from sending impressions in a timely manner, they may show up on your bill long after the experiment that generated them has concluded. On your bill, these old impressions are not separated out to distinguish them from more current impressions.

Impressions are not counted for visitors bucketed into the holdback:

- In an experiment with the traffic allocation set to less than 100%, some visitors will be in the holdback (`isLayerHoldback` will be set to `true`). These visitors won’t see the experiment, and no impressions are counted for these visitors.
• In Personalization with a holdback that’s greater than 0%, visitors in the holdback won’t see a personalized experience. No impressions are counted for visitors bucketed into the campaign holdback.

Decision event

Each time a Full Stack experiment or a page within an X Web experiment is activated, a decision request is sent to Optimizely. Decision requests look like this:

In the request payload, the decision attribute indicates the experiment that it applies to.

Example

Let’s walk through an example scenario. There are three multipliers:

• Experiments

• Pages (as defined in Optimizely)

• Pageviews

Imagine that your company, Attic and Button, is experimenting on www.atticandbutton.us. Consider a visitor who starts by visiting the Attic and Button homepage, where there are three experiments running. One of these experiments has two Optimizely pages that both target the homepage:

Experiment 1

• Homepage: https://www.atticandbutton.us/ (simple match)

Experiment 2

• Homepage: https://www.atticandbutton.us/ (simple match)

Experiment 3
• Homepage: https://www.atticandbutton.us/ (simple match)

• Global page (a page defined in Optimizely that targets all URLs of a website) https://www.atticandbutton.us/ (substring match)

Since four page activations occur when the visitor views the homepage, the visit counts as four impressions. Impressions are shown in green below.

On Initial Page Load
https://www.atticandbutton.us/

<table>
<thead>
<tr>
<th>Homepage</th>
<th>Global page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experiment 1</td>
<td>Experiment 3</td>
</tr>
<tr>
<td>Experiment 2</td>
<td></td>
</tr>
<tr>
<td>Experiment 3</td>
<td></td>
</tr>
</tbody>
</table>

4 impressions

If the visitor refreshes the page, another four impressions are counted. Attic and Button’s account usage now totals eight impressions.
Now, suppose that you’re running a search algorithm experiment with Full Stack on the homepage too. When a visitor types a search term, the results are refreshed without reloading the page. The Full Stack SDK makes a decision for a variation every time a new search is done. This means that if a visitor searches for "shirts," changes their search to "denim shirts," then changes their search again to "button down shirts," another three impressions are counted. The total usage count is now 11 impressions.
Verifying impressions with results export

Optimizely uses the server timestamp to calculate impressions, as opposed to the timestamp on the client device where the impression originated. Doing so makes it possible to accurately verify impressions all the way down to the experiment level.

You can use Optimizely’s results export to get a complete list of all impressions that occurred within a specific time period. You can then compare that information to your invoice, or determine whether any of your experiments are generating more impressions than they should be. To learn how to access that data, check out our developer documentation article on data export services in Optimizely, and follow the instructions for results export (and not raw data export).

How impressions are counted in billing

On June 23, 2020, Optimizely began rolling out updated Impression logic; it will be in effect automatically for your accounts next billing month. With this new logic we have introduced fixed interval bucketing which dedupes all impressions received within fixed 5-second intervals for every user in an experiment. This dedupe is based on their received timestamp of the event. This deduplication process is done by Optimizely and will not require any action on your part to receive the benefit. Depending on your implementation this change will have either a neutral or positive
impact on your Impression consumption, by reducing the impressions billed per experiment, allowing you to experiment more.

**MUVs vs. impressions**

Optimizely started billing customers based on impressions in 2017 to simplify billing and volume tracking. Starting February 1, 2018, no new subscriptions will use MUVs. However, some legacy subscriptions may still use MUVs.

To find out whether your subscription is billed based on MUVs or impressions, navigate to *Account Settings > Plan*. Your MUVs are listed under *Monthly Usage Information*.

View the full definitions for [monthly active users](#) and [impressions](#) and check your specific Order Form with Optimizely for details.

The advantages of impressions over MUVs include:

- Impressions are sold in buckets.
- Customers can purchase an annual allocation of impressions.
- Impressions are a pure metric, with no deduplication like with MUVs and no requirement for a cookie to track visitors. In addition, impressions work very similarly across Optimizely products.
- The impression allocation is typically shared for all the Optimizely products that a customer buys. For example, if a Full Stack and X Web customer buys 25 million impressions for the year, it doesn’t matter whether they use those impressions for Full Stack or X Web experiments.
- Impressions align well with the online media models that our customers commonly use.
Need more impressions?

When you have been getting a lot of value from Optimizely and using more impressions than you originally purchased, we will not stop you from running more experiments. When you buy an Optimizely impressions bucket, you lock in a cost-per-thousand (CPM) value. If you exceed the number of impressions included when you purchased Optimizely, Optimizely will bill you for the extra impressions at the same CPM value as described in your agreements with Optimizely.

If you want to lock in a lower CPM value and include more impressions in your subscription, contact your Account Executive to go to the next band of impression buckets.

Review the Optimizely Web and Full Stack features matrix for the Terms and Definitions for impressions, including how they are counted. In all cases, you should consult your Order Form to see if there are variations to the standard policies.