Cross-origin tracking in Optimizely Web

Skip Ahead

- Enable cross-origin tracking
- Allowed, blocked, and auto-generated origins
- When not to use cross-origin tracking
- Troubleshoot cross-origin tracking
- Cross-domain targeting

This article will help you:

- Implement behavioral targeting across different origins (subdomains or protocols) of your site
- Understand Auto-generated Origins, Allowed Origins, and Blocked Origins

An origin is a combination of a specific hostname, protocol, and port on your site. By default, Optimizely only tracks events on the same origin. To track events that cross origins, you’ll need to set up cross-origin event tracking.

When do you need to track events across origins? You’ll need to set up cross-origin tracking if, during the course of a typical website visit, your visitor sees pages that are on different domains, subdomains, security protocols, or ports, and if you need to preserve experiment variations, behavioral targeting, or event tracking across these pages.

For example, imagine that your snippet is on https://shop.example.com. By default, you can track events on that origin. But events that are generated on http://shop.example.com are not tracked because the security protocol is different -- so they’re on a different origin. Likewise, events on http://example.com are not tracked because the subdomain is different.

So, if you want to:

- Run an experiment on secure origin https://shop.example.com and target browsing behavior on the unsecure origin http://shop.example.com
- Or, run an experiment on https://shop.example.com and target based on reading activity on http://blog.example.com
… you’ll need to set up cross-origin tracking.

In some cases, you may **not** want to set up cross-origin tracking: specifically, if you have an entire customer journey that doesn’t cross origins. Here’s [when not to use cross-origin tracking](#).

---

### Enable cross-origin tracking

Please note that only collaborators with the role of Administrator can make changes in the Account Settings.

Here’s how to enable cross-origin tracking for your account:

1. Navigate to **Account Settings > Security and Privacy** and scroll down to **Cross-Origin Tracking**.

2. Un-check the box for **Use Automatically Discovered URLs**.

3. Under **Allowed Origins**, enter the URLs where events should be tracked and use the **Match Type** dropdown to target specific origins or groups of origins.

   Unlike [URL Targeting](#), the value matched with the pattern is specifically the "origin" portion of the URL, not the whole URL. There are several ways to specify origin patterns:

   - **http://example.com** (exact match)
   - **https://shop.example.com** (exact match)
   - **example.com** (suffix match)

     This will include origins like [http://example.com](http://example.com) and [https://shop.example.com](https://shop.example.com) but no other top-level domains. Data may be shared with other second-level domains.

   - **https://(.\.)example.com$** (RegEx match)

     This includes all of example.com when accessed via the secure https protocol. If you do not terminate the regular expression by placing the $ character after your domain, your data may be shared with other top-level
and second-level domains.

4. Use the URL Match Validator to check that your targeting matches the URLs you expect.

   You can check full URLs, including path, querystring, and hash, but those portions of the URL are ignored by the match pattern.

5. Click Save.

If you expect visitors to cross either domains or top-level domains, consider using the waitForOriginSync API.

---

**Allowed, blocked, and auto-generated origins**

When you enable cross-origin tracking, you define a set of allowed and blocked origins. The origins you want to share visitor data with are the allowed origins. The origins you want to exclude from sharing are the blocked origins.

Optimizely can also set the cross-origin tracking allowed origins for you. When you enable auto-generated origins, Optimizely will automatically set your allowed origins based on the URLs you use in the Pages, URL Targeting, and redirect experiments you create in your account. You can always manually move an origin from allowed to blocked if needed.

Here's how it works: when you create a Page, set URL Targeting, or create a redirect variation, Optimizely resolves the URL you provide down to the URL's top-level origin. The result mimics the scope of a cookie set on the base origin with a leading period. This means visitor data will be shared across all of the URL's subdomains and protocols.

If you set URL Targeting on https://shop.example.com, the auto-generated origin is example.com. Any page on that domain will be able to access events from any other page, including:

- http://www.example.com
- https://cart.example.com
- http://example.com

To move an auto-generated origin from your allowed origins to your blocked origins, click the icon to the right of the entry. Blocked origins take precedent, so a URL on your blocked origins list will not be able to access events from other origins even if you also set it as an allowed origin.

Cross-origin tracking is an account-level setting. Allowed, blocked, and auto-generated origins are aggregated across all projects in your account.

Events are shared asynchronously across origins, so events from http://blog.example.com aren't available when Optimizely begins to execute on https://shop.example.com. As a result, targeting may not succeed until the second page load on https://shop.example.com.
When **not** to use cross-origin tracking

If you use performance monitoring tools like Catchpoint, you **will** see a synthetic increase in page performance and latency metrics. This change in performance is an artifact of the methods these tools use to track performance. It does not affect the actual performance of your page.

Optimizely uses an iframe to seamlessly sync all Optimizely events across those origins. This is done asynchronously and doesn’t affect your visitors’ experience. However, you will see the persistence of this iframe recorded in your performance monitoring tools as an increase in various page load metrics. This is because these monitoring tools will push out those measurements while it waits for the iframe to be added.

**If you have a customer journey that doesn’t cross origins, you may decide not to set cross-origin tracking at all.**

For example: You run experiments exclusively on the domain [https://shop.example.com](https://shop.example.com) and do not track events from a different subdomain ([https://blog.example.com](https://blog.example.com)) or a different security protocol ([http://shop.example.com](http://shop.example.com)). In this case, it’s likely that you don’t need cross-origin tracking.

If you don’t track any events across origins, you may make a business decision not to set it up. However, if you don’t set up cross-origin tracking and do need to track events across origins, you risk losing event data.

**Troubleshoot cross-origin tracking**

Cross-origin tracking won’t work if any of these are true:

- The origin setting in your optimizely.com account is set incorrectly
- The visitor’s browser blocks all iframes or iframes specifically from *.cdn.optimizely.com
- The browser’s privacy settings prevent setting or retrieving localStorage keys from iframe origins

Some browsers block cross-origin iframe localStorage if third-party cookies are disabled. When a browser does this, Optimizely’s iframe won’t function when a visitor is on your website because in this situation, the Optimizely iframe is from a third-party origin.

**Cross-domain targeting**

Events can be targeted across origins only for the current Optimizely User ID, which is stored in the `optimizelyEndUserId` cookie. Browsers prevent sharing cookies across domains (for example, [https://www.example.com](https://www.example.com) and [https://www.anotherexample.com](https://www.anotherexample.com)) and across top-level domains (for example, [https://www.example.com](https://www.example.com) and [https://www.example.net](https://www.example.net)).

This means Optimizely is **unable** to set consistent user IDs across such domains. For cross-domain targeting to work, you must manually sync the `optimizelyEndUserId` cookie across domains using other means. We recommend using our `waitForOriginSync` function, as long as two snippets (projects) are linked and therefore use...
the same localStorage namespace. (Using the same snippet across two domains isn’t necessarily required.)

For example, the following code would be implemented in the head tags before the Optimizely Snippet, or in Project Javascript:

```javascript
// ensures the optimizely object is defined globally
window.optimizely = window.optimizely || [];

// set the waitForOriginSync API call
// place your own domains in the canonicalOrigins array
window.optimizely.push(
  {
    "type": "waitForOriginSync",
    "canonicalOrigins": ["optimizely.com", "developers.optimizely.com"]
  });
```

The cross-domain targeting solution may not be effective in some combinations of browsers and situations. If this is the case for you, please contact Optimizely support.