Redirect experiments: Test two URLs in Optimizely

Skip Ahead

1. Create an experiment and set up the targeting
2. Add the redirect to your variation
3. Set up a page to track events across all URLs
   Create a click event
4. Select metrics and audiences for your experiment
5. QA and publish your experiment

Best practices

RELEVANT PRODUCTS:

• Optimizely Web Experimentation
• Optimizely Performance Edge

This article WILL HELP YOU:

• Test two landing pages, homepages, or any other pages against each other
• Set up redirect experiments (split URL tests), which redirect visitors from one URL to another
• Ensure that you can track events on both your original and redirect pages

Redirects at a glance

Redirect experiments, or split tests, allow you to compare two separate URLs as variations in an A/B test. For example, you might create a redirect experiment that compares two versions of a landing page.

Tips
• Use redirects if your variation code exceeds 200 lines
• Check *Redirect with live query parameters included* to keep the parameter attached in the variation
• Leave *Allow Additional Redirects* unchecked to prevent subsequent redirects
• Use an *event* to track clicks on the redirected variation

**What to watch out for**

• Make sure your *URL targeting* does not capture the redirect URLs
• The *Optimizely snippet* must be on the original and the redirected page

In Optimizely, you can set up redirect experiments (also called split URL tests), which redirect visitors from one URL to another.

This is useful when comparing landing pages, or multiple versions of a site page that have different URLs. [Testing whole pages against each other](https://www.optimizely.com/blog/guides/testing-whole-pages-against-each-other/) is a proven way to perform deep experimentation.

In this article, you will walk through each stage of setting up a redirect experiment: setting up the experiment targeting, settings up the metrics, and creating the redirect variations.

*Optimizely Performance Edge* is a lightweight experimentation product that delivers significantly faster performance than previous versions of Optimizely. It does this by relying on a streamlined "microsnippet" which limits the range of available features.

Optimizely Performance Edge and Optimizely Web handle redirect experiments the same way, so all information contained in this article applies to experiments created with either version of Optimizely.

1. Create an experiment and set up the targeting

When you begin the process of setting up your redirect experiment, you have three options:

• You can use an existing page;

• You can create a new page; or

• You can create a new experiment that targets the original URL you're redirecting visitors from. For example, if the experiment redirects from [https://atticandbutton.us/homepage](https://atticandbutton.us/homepage) to [https://atticandbutton.us/new-homepage](https://atticandbutton.us/new-homepage), then the original URL is [https://atticandbutton.us/homepage](https://atticandbutton.us/homepage).

Choose the option that makes the most sense for your redirect experiment, and continue to step 2.

Make sure not to include the redirect destination URL in the URL match conditions. Expand the *Test URL(s)* tray to verify that the redirect URL is not included in the match conditions.
Next, you'll create a variation change to redirect visitors to the redirect URL.

1. Under Manage Experiment > Variations, select the variation (not the original).

2. Click Create.

3. Under Create Options, select Redirect.
From here, you will have two options for setting up your redirect: either via the URL, or with custom code. If you select URL, paste in either the complete URL or the relative path where you'd like visitors to be redirected to.

If you select Code, define a function that returns a URL string where you'd like to redirect visitors.
You won't be able to edit the redirect URL with the Visual Editor. Instead, you'll see this prompt:

3. Set up a page to track events across all URLs
If you want to track how an event performs on both the experiment URL and the redirect URL, you'll create a page that includes both in the URL match conditions. In this article, we will refer to this as the hybrid page.

- Make sure you do not include this page in the experiment's targeting.
- The hybrid page can be excluded from the snippet if you have turned on the Page Trimming feature. To ensure that the page is not excluded, please override the page trimming via the advanced section in page setup as described here.

1. To create the hybrid page, include both the experiment URL and the redirect URL in the URL match conditions section.

2. Next, add a new event to the hybrid page.

Create a click event

Sometimes an element on the experiment URL has a different selector than the equivalent element on the redirect URL. If you need to target these multiple selectors in a single click event, repeat these steps for both URLs.

1. Open the URL in your browser.

2. Right-click the button element and select Inspect. A console will pop up, and the element will be highlighted on the page.

3. Right-click the highlighted element and select Copy > Copy selector.
4. Paste this selector into the **Selector** field of the click event. You can add multiple selectors by separating them with commas.

5. Click Save.

---

**4. Select metrics and audiences for your experiment**

When a visitor is bucketed into an Optimizely Web variation with a redirect change, Optimizely will prioritize executing that change as quickly as possible. This is to ensure there is as little delay as possible before a visitor is able to reach and load the new destination page.

As a result, redirects can cause race conditions with the typical event tracking for pageview events on the origin page. Because of this, and because doing so does not generally provide a strategically significant metric for redirect experiments, using origin page views as a metric for your redirect tests is **not recommended**.

1. Add the click events you created in [step 3 above](#) as **metrics** in the experiment.
2. Add an audience, if you like.

3. Adjust the traffic allocation and distribution between your variations, if desired. The original variation is the original URL; the next variation is the URL that visitors are being redirected to.

Take care when choosing to use asynchronous conditions to target redirects, such as geotargeting, list attributes, and some third party analytics integrations. When you use asynchronous conditions to target a redirect, the initiation of the redirect may be delayed. During the delay, you may see other experiments activate or events tracked on the page that visitors are being redirect from. If you've defined a metric based on those events, you may see those visitor actions reflected on your Results page.

The Optimizely snippet must be implemented on all pages where the experiment will run and where events will be tracked. Make sure the snippet is included on both the original URL and the redirect URL. Otherwise, some visitors and events may not be tracked.

Experiments that redirect to a different domain may experience tracking issues for those redirected visitors. Learn more about cross-domain tracking.

5. QA and publish your experiment

Congratulations! You've set up a redirect experiment. All you need to do now is QA the experiment and publish it live to your visitors.

You can verify that your setup is correct by going to the API Names tab in your redirect experiment.
Best practices

• **Indicate the canonical link.** To improve SEO link and ranking signals, the redirected page should indicate that the original page is the preferred (canonical) destination. Add a `<link>` element with the `rel="canonical"` attribute into the `<head>` section of the redirected page. Check out this Google Support Doc for more information.

• **Exclude the redirect URL.** To avoid redirects on the redirect URL itself, make sure to exclude the redirect URL via the page targeting settings.

• **Unblock Optimizely's redirect.** By default, when you set a redirect, Optimizely blocks other redirects for 5 seconds so that you will not cause an infinite loop of redirects. Click to learn about unblocking the redirect.

*Client typically have three main concerns with redirects*

1. **Performance**

• Will visitors see a flicker of images/old page loading prior to being redirected?
  ◦ No, Optimizely redirects will hide the body of the original page using CSS, so there will be no flashing/flickering

• How does the redirect work?
  ◦ We use `window.location.replace`. This is called immediately after the bucketing decision.
  ◦ Origin page will not load and no additional requests will be made on the original page after this `location.replace` is called.
  ◦ In Chrome, in-flight XHR requests are *cancelled* by the browser when `location.replace` is called

• How can we ensure mobile visitors don't have a poor experience?
  ◦ You can do a QA cookie-gated experiment, and evaluate performance by using Chrome dev tools (using throttling and device emulation)

• Will the user be able to click the Back Button in their browser to go back to the origin page?
  ◦ No. We use `window.location.replace()`, so the back button will not bring the user back to the original homepage.
Comments:

- Having the Snippet as high in the head as possible is vital
- Having any asynchronous components for this experiment will delay the redirect from happening. This includes asynchronous audience conditions (i.e., geotargeting) and conditional page activation that does polling or some other async operation.

2. SEO

Being that Optimizely WebX is a JavaScript tool, it cannot inherently prevent Bots from indexing variations on your site for SEO purposes, however, we can provide some recommendations on how to best minimize this.

Search engines have their own methods and proprietary algorithms for indexing pages on the web. Google and other search engines handle the indexing of dynamic JavaScript content in various ways. Since Optimizely powers experimentation via dynamic JavaScript, any SEO implications of using Optimizely would fall under Google’s rules and methodology of indexing dynamic JavaScript content. In order to best-prevent SEO engines from indexing the variant version of the page, we recommend you take the following measures:

- Canonical `<link>` tag - You should add a link tag to the head of your page which tells Google that the canonical version of the page lives at a different URL. We provide some details on that here.
- Robots.txt entry - search engines that respect this convention will skip over URLs that follow the disallow directive - Robots documentation here.

3. Analytics

Clients typically have client-side analytics tools in place, and want to make sure that the data being transmitted to those systems is useful. On the post-redirect (variant) page, the document.referrer is actually the pre-redirect page URL, which is not very useful. In some cases, the client may want to report that the true referrer was the document.referrer of the pre-redirect page.

They can use our getRedirectInfo method (docs) to capture the original referrer if they'd like to pass it to their third-party analytics systems:

```javascript
var redirectInfo = window.optimizely.get('state').getRedirectInfo();
var realReferrer = redirectInfo.referrer;
```
Messaging to customers concerned about redirects

• We have had high-profile clients run long-standing, high-trafficked redirect experiments - even on their homepage!
  ◦ They did not observe any adverse effects to their SEO
  ◦ They did not observe material impact to their bounce rate.
  ◦ These are anecdotal, but it’s worth mentioning that a highly risk-averse company trusted us to handle a Homepage redirect
    • Please do not mention "State Farm" (NDA)

• SEO impact is out Optimizely Web client's control - it's largely dependent on how various engines index dynamic content, and there is no consistent documentation out there to accurately characterize SEO impacts of client-side redirects.