Custom snippets in Optimizely Web

Skip Ahead

1. How custom snippets work
2. Common use cases for custom snippets
   - Use case 1: Boost site performance
   - Use case 2: Separate swimlanes for different teams
   - Use case 3: Run experiments across properties
3. Custom snippet matching criteria
4. How custom snippet generation works
5. Custom snippet permissions
6. Custom snippet availability
7. Linked projects
   - How to link a project
8. Related articles

THIS ARTICLE WILL HELP YOU:

- Boost site performance by reducing snippet size
- Experiment across properties

To run an experiment in Optimizely, you must include a snippet of code inside the <head> tags on any pages where your experiment will run. Without this snippet, Optimizely will not know what to track, or even where to find your experiment in the first place.

Each experiment you run is contained within a project. A project can have multiple experiments, but all of those experiments share the same snippet. This is Optimizely's default behavior, and for the vast majority of Optimizely customers, this model works just fine. However, larger organizations can also use Optimizely's custom snippet to avoid certain potential issues they may experience as they start using Optimizely at scale.
How custom snippets work

Each project in Optimizely starts with a single “basic” snippet that automatically includes all the entities from that project (and only that project). This is a 1:1 relationship between projects and basic snippets.

You can change this setup by creating a custom snippet. Custom snippets can have a many-to-many relationship with projects. This enables different teams to use projects as organizers, each with their own workspaces and permissions, that can touch the same URLs. (This is not possible with basic snippets, as no page on your site may reference more than one snippet.)

In other words, a project can be delivered via many snippets on different parts of a site, and a single snippet can pull in experiments from many projects. In this way, Optimizely separates performance and implementation concerns from those of permissions and governance.

Custom snippets are only available in projects that use the standalone Optimizely snippet.

If you want to include more than one project in a custom snippet, make sure the projects you want to include are linked projects before you create the custom snippet (it is still possible to add projects to a custom snippet after the snippet was initially created but then need to be linked before you can do that).

Common use cases for custom snippets

Use case 1: Boost site performance

Attic & Button’s ecommerce site includes a home page, several category pages, and many product description pages. The storefront is owned by the company’s ecommerce team, which runs a large number of experiments across all these
pages. Because Attic & Button needs to keep all these experiments in the same Optimizely project, the snippet has grown large enough to have a negative impact on the site’s performance.

Attic & Button can solve this problem by using multiple snippets to reduce the size of each. With custom snippets, the team can use a unique snippet for each of three experiments: one for the home page, one for product category pages, and one for product description pages. The team runs all the same experiments, but because each snippet is smaller, site performance is restored.

For the best performance, use prefetch in your code to speed up fetching of custom snippets implemented on the next page that you expect visitors to navigate to.

Prefetch hints to your visitor's browser that a resource might be needed, but doesn't necessarily load it right away. Instead, the prefetch hint can be used to tell the browser to fetch a resource required in the next navigational step. If Page A initiates a prefetch request for key resources for Page B, the navigation and resource loading requests can be done in parallel.

For example, if you're on the homepage and you have a custom snippet implemented for category pages, you can add a hint:

```html
<link rel="prefetch" href="cdn.optimizely.com/public/12322/s/category.js" as="script">
```

Since prefetch resources are fetched by the browser with low priority when possible, this won't impact the rendering of the current page (the homepage).

**Use case 2: Separate swimlanes for different teams**

MondoMedia is a large online media company with an editorial team and a marketing team that both regularly run experiments with Optimizely. The editors use Optimizely to test headlines, while the marketers use it to test the effectiveness of different types of ads. By their nature, both of these activities often have to run on the same pages.

Previously, both teams had to share one crowded project that contained both teams' experiments. There was a real concern about the ability of one team to accidentally interfere with the other team's work. But with custom snippets, each team can work within a separate project with its own permissions, both of which touch the same URL. This is achieved by building a single custom snippet that contains both the editorial and marketing teams' projects.

**Use case 3: Run experiments across properties**

Attic & Button’s navigation team is responsible for the performance of the site’s header. They want to run an experiment on the header and track how a change to the header’s copy will affect sales for their canvas backpack product. This experiment will encompass the backpack's landing page as well as its description page in the ecommerce site. Because these two pages are the responsibility of different teams, in the past they’d each have had their own snippet, and no experiments could be conducted across both.

However, the navigation team can include their experiment in custom snippets for the landing page and the product
description page. The header will appear consistently throughout any user’s experience (either the test version or the original version), and actions taken on both of these pages will be tracked by the new experiment.

In this way, different teams can set up experiments that cross boundaries and touch pages owned by other teams. You can have two teams, each with ownership of a single page, running tests that incorporate both pages.

In this use case, the Attic & Button site would have only two separate custom snippets: one for the landing page and one for the PDP. Both of these snippets would also include the navigation team’s header project.

**Custom snippet matching criteria**

When you're creating a custom snippet, the default setting is to include all pages from the selected project. But what if you only want to include specific pages? In that case, you can either specify pages to explicitly include or exclude or specify custom rules to match all pages that follow a certain pattern.

You have two options for matching criteria:

- **Match to URL**: enter a single URL you want to target. The Optimizely snippet will pull in any experiments on pages that match the URL. For example, if you have a page that uses a regular expression (regex) pattern like "www.website.com/products/.*", you would enter a URL like “www.website.com/products/123”. Matching to URL offers more precise filtering.

- **Match to Substring**: enter a string to check against each of the URL conditions. The Optimizely snippet will pull in any pages where any of the targeting conditions match the substring. Matching to substring is less precise, but captures a broader set of pages. For example, you could enter the pattern "/products/.*" to capture a page targeting "www.website.com/products/123".

Your custom snippet can include and exclude multiple URLs and URL patterns. You can also mix and match the criteria options to make sure your custom snippet includes exactly the experiments and pages you want.

For example, if you want to exclude experiments on certain individual product pages, you can enter the exact URLs for those product pages and choose the **Match to URL** matching criteria. Or, suppose you want to include only products for Winter 2017, and those product pages all have one URL pattern in common: www.examplecompany.com/products/winter17. You can enter the Winter 2017 URL pattern and choose the **Match to Substring** matching criteria to include only those pages in your snippet.

If you include pages, this will only include that specific page and the experiments targeted to it directly. It won’t include other pages with overlapping conditions or experiments that run on the same URLs. To match all experiments that run on a certain URL, use the URL rules above.

---

**How custom snippet generation works**

When you generate a custom snippet, the following logic is used to decide which properties are included and which are not:

1. The new custom snippet first finds all the projects you specified above. All project-level properties - custom events,
project JS, and attributes - will be wrapped into the snippet here.

2. Within those projects, find all the pages that match the snippet’s rules. Click events and tags on those pages will be added to the snippet here.

3. Find all the campaigns and experiments targeted to these pages. The snippet will add the audiences for each of these campaigns or experiments and any changes for the pages that match.

Custom snippet permissions

Here’s how custom snippet permissions work for Administrator and Project Owner roles:

- **Administrators** can see all snippets for an account from Account Settings; create, edit, and delete snippets for any projects; and link any projects.

- **Project Owners** can see snippets that pull from a specific project from the project’s Project Settings; create and edit snippets (as long as they own all of the projects included in the snippet); and link projects they own.

Custom snippet availability

Custom snippets are an advanced feature designed for large teams and performance-conscious customers. We support custom snippets for **Scale Plan** customers.

If you want to include more than one project in a custom snippet, you must link all of the projects before you create the custom snippet.

Also, custom snippets are only available for Optimizely X Web.

Linked projects

Even without custom snippets, Optimizely offers the ability to use events in one project that originated in another. This is called **referencing** and can be accomplished simply by selecting it from “All projects” in the metric picker. Custom snippets are not required for referencing, and linked projects are not required for cross-project events or single-project custom snippets.

If you want to use cross-project behavioral targeting or combine multiple projects into one snippet, you need to use linked projects. This will require you to change the state of these projects from **Unlinked** to **Linked**:

- **Unlinked projects** are completely walled off and separated from each other. This is the default project state.

- **Linked projects** can share behavior targeting, overlap with each other in a snippet, and use account-level namespaces and attribution.
To be safe, we recommend pausing any running experiments when you link projects.

Linking projects has a couple important effects. First, linking “resets” the behavioral targeting for a project. When you link a project, it stops reading and writing events in the project-level store and starts reading and writing events in an account-level store instead. The events in the project-level behavior store are still there, but the project reads from a different place (the account-level store). That’s why we say the behavior store resets.

Second, when you link multiple projects into a single snippet, Optimizely automatically checks each project for any conflicts, like events that share an API name with any events in the other projects you’re trying to link. You can resolve the conflict by re-naming the conflicting events.

If you do have to change any API names, you will probably need to do so in both the Optimizely UI and any code running on your site that references those API names.

Conflicts also apply to integrations across linked projects. Optimizely will not let you link two projects with conflicting integration settings. You’ll have to switch one project to the same integration account as the other before you can proceed with linking the projects.

Optimizely will require you to resolve any conflicts in these elements before you can link projects:

- Event API names
- Page API names
- Attribute API names
- Extension API names
- Integration settings

This table summarizes the key differences between how isolated and linked projects work.

<table>
<thead>
<tr>
<th></th>
<th>Unlinked</th>
<th>Linked</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Namespace</strong>: At what level do namespace uniqueness checks happen?</td>
<td>Project-level uniqueness. Visitor attributes and behavioral store are scoped to a single project.</td>
<td>Account-level uniqueness; Optimizely only checks other linked projects. Visitor attributes and behavioral store are scoped to the account level.</td>
</tr>
<tr>
<td><strong>Overlaps</strong>: Can two projects touch the same page?</td>
<td>No, because of the potential for namespace conflicts.</td>
<td>Yes. Projects are safe to run with other linked projects in one custom snippet.</td>
</tr>
</tbody>
</table>
**Attrition**: can experiments in one project reference events from another project?  

Yes. **Cross-project events** do not require linking.

You can use a custom snippet with both unlinked and linked projects. However, unlinked projects can only use single-project custom snippets. This means you will not be able to group projects together when they are unlinked.

We plan to automatically link new projects in the future. For now, we recommend linking your projects—even if you aren’t sure you’ll need the features that linking enables—because you might want to use behavioral targeting and multi-project custom snippets in the future. We also recommend linking projects during an experimentation pause, like onboarding or transition to Optimizely, to avoid any impact on running experiments or personalization campaigns.

**How to link a project**

Here’s how to link a project, with step-by-step instructions below.

1. In the project you want to link, navigate to **Settings > Advanced**.
2. Under **Link Project**, click the dropdown and select **Linked**.
3. Click **Check for Conflicts**.

As long as there aren’t any conflicts, your project is now linked.

**Related articles**

- Create, view, and manage custom snippets in Optimizely Web
- How the Optimizely Web snippet works
- Implement the snippet for Optimizely