



PARTNERSHIP ENABLEMENT

# Creating and Recording Events

---

## Optimization Metrics

To collect metrics we must send events from the client. Strategy will determine the different metrics to track, and delivery will configure and send the events from the client. Metrics are the action we want to collect. Events are the unique keys we send. We will use both terms in this lesson, and it's good to understand the relationship between the two terms.

## Tools

- [Evolv Web Editor](#)
- [Evolv Manager](#)

## Conversion Funnel Example: Ebay

*\*Ebay is not an Evolv customer, we do not have access to the Thank You page for this lesson*

For the purpose of this exercise, we will scope our optimization to just a single product funnel. We have pre selected New Balance brand, mens 574 shoes, navy with brown color, item #19380280096. Here's a standard conversion funnel for this product:

- [Product Listing Page \(PLP\)](#)
- [Product Detail Page \(PDP\)](#)
- [Cart Page](#)
- [Checkout Page](#)
- Thank You Page

*\*Ebay's checkout page URL is dynamic. Configure the URL to correctly capture this page match.*

# E-commerce Completions Note

When your customer's primary KPI is to increase E-comm completions, you will need a mechanism to access the Thank You Page on their production site. Some customers will configure a test user account with a payment mechanism; they cancel and refund purchases on their side (this is the ideal solution). For other customers, that is not an option. In this case, we use our own credit card, with an agreement to cancel and refund with submitted requests. For the purpose of this lesson we will optimize to **Checkout Page load**.

## Events

Strategy has identified these metrics to collect. Our primary KPI is progression of product from the product detail page to the checkout page. In this scenario, we would want to know what product was added to the cart, and ensure it is still in the cart during the checkout process. For the simplicity of this exercise, we will assume that logic is configured, and track checkout page load events.

Funnel pages	PLP Product Listing Page	PDP Product Detail page	Cart	Checkout
Page Load event keys	page.product_landing.loaded	page.product_detail.loaded	page.cart.loaded	
CTA event keys		button.buy_it_now.click button.add_to_cart.click		

## Examples

Checkout page load event triggered via context-level JavaScript:

The screenshot shows the eBay Checkout page with a sidebar containing a context menu for 'New Balance Shoes' with items like 'Buy It Now CTA color' and 'Buy It Now CTA text'. The developer tools panel on the right displays the 'Checkout (Context)' section with the following JavaScript code:

```

1 // Send event on page load
2 evolv.client.emit('page.checkout.loaded')

```

Button click events triggered via context-level JavaScript:

The screenshot shows an eBay product page for 'New Balance Men's 574 Shoes Navy with Brown'. The sidebar shows the same context menu for 'New Balance Shoes'. The developer tools panel on the right displays the 'New Balance Shoes (Context)' section with the following JavaScript code:

```

1 var buyBtn = document.querySelector('.vi-bb-btncr1 #CenterPanelIDF .watch-redesign.new-brdr-btns a.btn-prim');
2
3 var addBtn = document.querySelector('#Body #CenterPanelIDF .watch-redesign.new-brdr-btns #atcRedesignId_bttn');
4
5 buyBtn.addEventListener('click', function() {
6   // Send event on primary CTA click
7   window.evolv.client.emit('button.buy_it.now.click');
8 });
9 addBtn.addEventListener('click', function() {
10  // send event on secondary CTA click
11  window.evolv.client.emit('button.add_to_cart.click');
12 });

```

## Activity (20 mins)

Discuss approach for implementing each event with a partner, then implement events

- Breakout rooms (1 min)

- Analyze the events table (2 min)
- Open sample .yml in web editor (1 min)
- Discussion (2 min)
- Return to main room (1 min)
- Implement the additional events (8 min)
- Questions and support (5 min)

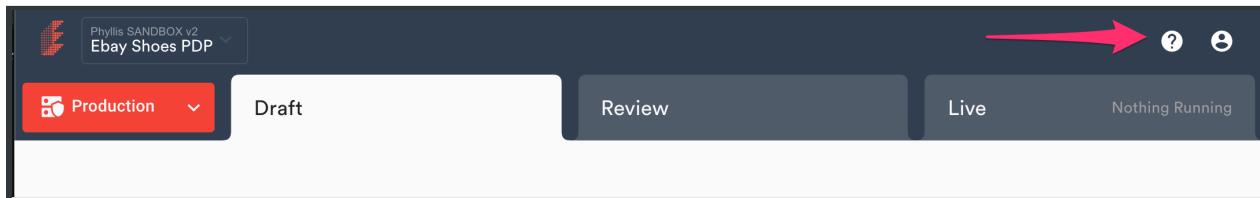
## Starter .yml file

This is not a lesson in how to write JavaScript. However, you will need some basic JavaScript and CSS skills to complete this lesson. [Documentation](#).

[Example .yml with Events](#)

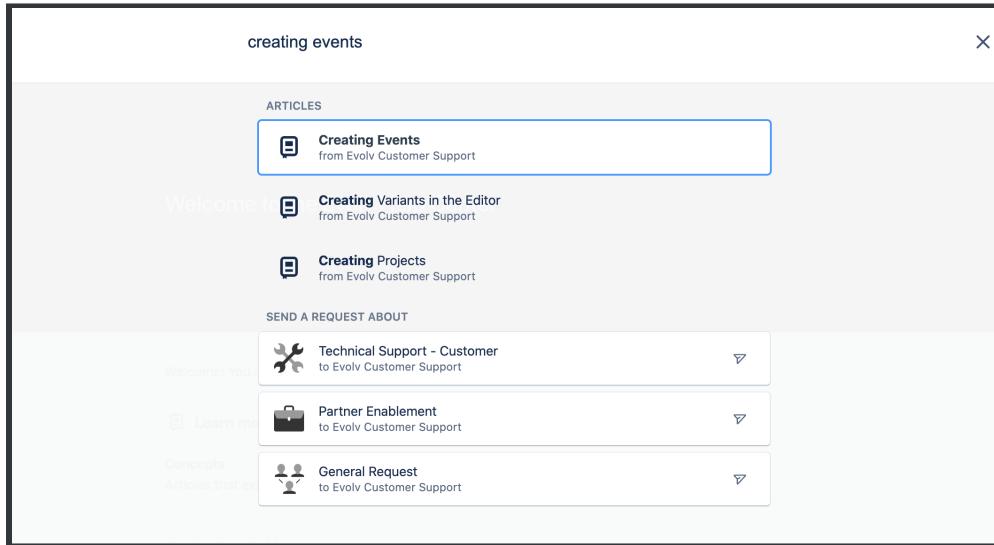
## Getting help from the Manager

To gain more understanding of how to create and record events, you will want to read the documentation in the knowledge base. Navigate to the Evolv Manager and select the help icon in the upper-right corner.



Enter your help keyword into the search. In this case your keywords are “creating events”.

Select the top option, “Creating Events”:



This will bring you to the article about creating and recording events, with links to the SDK documentation along with SDK and API implementations.

The screenshot shows the 'Creating Events' article page. The page title is 'Creating Events'. It contains a numbered list of steps for creating events, a note about naming conventions, and a feedback section at the bottom.

Help Center / Evolv Customer Support / Creating Events

## Creating Events

1. Navigate to "Events" in the left nav bar and select the "New Event" button. Enter an event name, an event key and, optionally, a description. An event is defined by its key, which is the unique identifier that is sent to the Evolv system indicating the event has occurred so that the event can be counted.
2. To record completion of an Event, send the event key to Evolv
  - a. For a web implementation, see the [JavaScript SDK docs](#)
  - b. For an SDK implementation, see the open source [Github project examples](#)
  - c. For an API implementation, see the [API developer documentation](#)

Note that for event naming, use a specific description of the event itself. For example, "Button Clicked", not "Conversion Achieved". This will make them easier to re-use between different projects with different conversion goals.

Did this article help?  Yes  No

Need to raise a request? Contact us. >

## Version Control

Store your .yml where you can use version control. [Github](#) is commonly used, but work out with your team a strategy to collaborate, store, and version your work.