

Install Pixel with Developer Mode

Developer Mode is an implementation method for the kwai Pixel that provides flexibility for event configuration, allows advertisers to send parameters in addition to events. With the additional parameter data, advertisers can also access additional metrics like Event Value and Return on Ad Spend (ROAS). Before you start plz read doc first.

important: [Pixel Instructions Document](#)

Create Pixel on Event Manager using Developer Mode

Create pixel on Event Manager

- Log in to Kwai for Business Manager, click 'Pixel' under 'Assets', and click 'Create Pixel'
- Enter the pixel name, confirm 'Manually install pixel code' and 'Developer Mode', then click 'Create'

Set Up Web Events

Pixel Name

Please name your pixel 0/100

Association Method

Choose a way to send your website's data to kwai.

Kwai Pixel
Send events and parameters through web browsers. [Learn more](#)

Event Api
Send events and parameters directly from your server. [Learn more](#)

Installation type

Manually Install Pixel Code
Copy and paste the pixel code in your website. [Learn more](#)

Automatically Set Up Kwai Pixel Via Partner Platforms (Coming Soon)
Connect your website using one of our partners, like Shopify or Google Tag Manager.

By clicking Create, I agree to the [Kwai Business Products \(Data\) Terms](#).

[Cancel](#) [Create](#)

Set Up Web Events With Kwai Pixel

Mode

Standard Mode
you can directly set up the events with our self-service platform, but you must be able to install base code. [Learn more](#)

Developer Mode
Build and customize events with code to fit your business needs. Developer support is required. [Learn more](#)

[Cancel](#) [Create](#)

Install pixel code

1. Overview

Once you select Developer Mode, you will see a page composed of 3 components: pixel base code, pixel event code, and Pixel Helper.

- Installation of both the pixel base code and event code is required to successfully configure the pixel.
- Pixel Helper is a Chrome extension that can help you verify and troubleshoot pixel installation by checking for errors and providing implementation recommendations for your website.

Set Up Pixel

Follow steps below to set up Pixel tracking:

Base Code

Insert the code below between the <head></head> tags in your webpage.

[Download](#) [Copy](#)

```
<script>
!function(e,t){"object"==typeof exports&&"object"==typeof module?module.exports=t():"function"==typeof
define&&define.amd?define([],t):"object"==typeof exports?exports.install=t():e.install=t()}(window,
(function(){return function(e){var t={};function n(o){if(t[o])return t[o].exports;var r=t[o]={i:o,l:!1,exports:
{};return e[o].call(r.exports,r,r.exports,n),r.l=!0,r.exports}return n.m=e,n.c=t,n.d=function(e,t,o)
{n.o(e,t)||Object.defineProperty(e,t,{enumerable:!0,get:o})},n.r=function(e>{"undefined"!=typeof
Symbol&&Symbol.toStringTag&&Object.defineProperty(e,Symbol.toStringTag,
{value:"Module"}),Object.defineProperty(e,"__esModule",{value:!0});},n.t=function(e,t){if(1&t&&
(e=n(e)),8&t&&"object"==typeof e&&e.__esModule)return e;var
o=Object.create(null);if(n.r(o),Object.defineProperty(o,"default",
{enumerable:!0,value:e}),2&t&&"string"!=typeof e)for(var r in e)n.d(o,r,function(t){return
o[t]=e[t]})}};return o}();n.n=n;var n=o;Object.defineProperty(n,"__esModule",{value:!0});n.default=o
});</script>
```

Event Code and Parameters

Check out the instructions and events supported in the [Help Document ↗](#).

Check Code Installation

You can install the pixel helper to check if your pixel code was successfully installed. It's a chrome plugin that will run and detect whether the websites you open have pixels installed. [Learn More ↗](#).

[Download Kwai Pixel Helper](#)[OK](#)[Cancel](#)

Click 'OK' and it will take you to the pixel details page.

2. Pixel base code

2.1 What is pixel base code?

To fire the pixel and send event data from a website, the base code should be inserted into the website's source code.

There are two parts for the base code.

- Part 1 is for the general functions.

- Part 2 contains `kwaiq.load("input your pixelId")`, which helps to load the context relevant to the specific pixel based on its configuration. `kwaiq.page()` would report a page view event, which can be used to verify the code installation.

2.2 How do you install pixel base code?

You can paste the base code between `<head>` and `</head>`.

- Here is an example.**note: The base code can only be pasted between `<head>` and `</head>`, and the base code cannot be placed in an `<iframe>`**

```
<!-- Example -->
<!DOCTYPE html>
<html ...>
  <head>
    insert_Kwai_Pixel_code_here
    <script>...</script>
  </head>
```

notice: The basecode provided by kwai has a method of obtaining data through fetch. If there is also a fetch method in your webpage code, and you are worried that the code provided by kwai will affect the operation of your webpage code, you can use the customized basic code to ensure your webpage Safety. you can reffer to Customized Basecode Manual

3. Install pixel event code

Prior to installing the event code, make sure you have got your base code. The event code will not work by itself without the base code.

Kwai does not support custom events, so you must use the event names in the event codes provided by Kwai.

3.1 What is pixel event code?

Pixel event code is what enables you to report on the specific actions or events to track for conversion purposes. For example, if you are in e-commerce business, events such as "add to cart" and "place an order" would be important for you to track. See below for the list of events Kwai Pixel supports today:

Event	Description	Event code Note: The pixel id mentioned here has a one-to-one correspondence with the event, that is, the event you want to track with the pixel with pixel id=123456 is purchase, so call the function <code>kwaiq.instance('123456').track('purchase')</code>
EVENT_ADD_PAYMENT_IN_FO	When payment information is added in the checkout flow.	<code>kwaiq.instance('input your pixel id').track('addPaymentInfo')</code>
EVENT_ADD_TO_CART	When an item is added to the shopping cart. *The corresponding conversion goal of this event is "add to cart"	<code>kwaiq.instance('input your pixel id').track('addToCart')</code>
EVENT_BUTTON_CLICK	When a button is clicked.	<code>kwaiq.instance('input your pixel id').track('buttonClick')</code>
EVENT_PURCHASE	When a payment is completed. *The corresponding conversion goal of this event is "purchase"	<code>kwaiq.instance('input your pixel id').track('purchase')</code>
EVENT_CONTENT_VIEW	When a page is viewed. *The corresponding conversion goal of this event is "content view"	<code>kwaiq.instance('input your pixel id').track('contentView')</code>
EVENT_DOWNLOAD	When a button to open an external browser download page is clicked.	<code>kwaiq.instance('input your pixel id').track('download')</code>
EVENT_FORM_SUBMIT	When a form is submitted.	<code>kwaiq.instance('input your pixel id').track('formSubmit')</code>

EVENT_INITIATED_CHECKOUT	When the checkout process is started.	kwaiq.instance('input your pixel id').track('initiatedCheckout')
EVENT_CONTACT	When contact or consultation occurs.	kwaiq.instance('input your pixel id').track('contact')
EVENT_PLACE_ORDER	When an order is placed.	kwaiq.instance('input your pixel id').track('placeOrder')
EVENT_SEARCH	When a search is made.	kwaiq.instance('input your pixel id').track('search')
EVENT_COMPLETE_REGISTRATION	When a registration is completed. *The corresponding conversion goal of this event is "registration"	kwaiq.instance('input your pixel id').track('completeRegistration')
EVENT_ADD_TO_WISHLIST	When an item is added to a wishlist.	kwaiq.instance('input your pixel id').track('addToWishlist')
EVENT_SUBSCRIBE	When a subscription is made.	kwaiq.instance('input your pixel id').track('subscribe')
EVENT_FIRST_DEPOSIT	EVENT_FIRST_DEPOSIT	kwaiq.instance('input your pixel id').track('firstDeposit')
EVENT_CREDIT_APPROVAL	EVENT_CREDIT_APPROVAL	kwaiq.instance('input your pixel id').track('creditApproval')
EVENT_LOAN_APPLICATION	EVENT_LOAN_APPLICATION	kwaiq.instance('input your pixel id').track('loanApplication')
EVENT_LOAN_CREDIT	EVENT_LOAN_CREDIT	kwaiq.instance('input your pixel id').track('loanCredit')
EVENT_LOAN_DISBURSAL	EVENT_LOAN_DISBURSAL	kwaiq.instance('input your pixel id').track('loanDisbursal')
EVENT_CREDIT_CARD_APPLICATION	EVENT_CREDIT_CARD_APPLICATION	kwaiq.instance('input your pixel id').track('creditCardApplication')

EVENT_VALUE_PRODUCE	EVENT_VALUE_PRODUCE	<code>kwaiq.instance('input your pixel id').track('valueProduce')</code>
EVENT_KEY_INAPP_EVENT	Key event	<code>kwaiq.instance('input your pixel id').track('keyInappEvent')</code>
EVENT_KEY_INAPP_EVENT_1	Key event 1	<code>kwaiq.instance('input your pixel id').track('keyInappEvent1')</code>
EVENT_KEY_INAPP_EVENT_2	Key event 2	<code>kwaiq.instance('input your pixel id').track('keyInappEvent2')</code>
EVENT_KEY_INAPP_EVENT_3	Key event 3	<code>kwaiq.instance('input your pixel id').track('keyInappEvent3')</code>
EVENT_AD_VIEW	(In-Web) event ad view	<code>kwaiq.instance('input your pixel id').track('adView')</code>
EVENT_AD_CLICK	(In-Web) event ad click	<code>kwaiq.instance('input your pixel id').track('adClick')</code>
EVENT_PURCHASE_1_DAY	re-purchase on first day	<code>kwaiq.instance('input your pixel id').track('purchase1Day')</code>
EVENT_PURCHASE_2_DAY	re-purchase on second day	<code>kwaiq.instance('input your pixel id').track('purchase2Day')</code>
EVENT_PURCHASE_3_DAY	re-purchase on third day	<code>kwaiq.instance('input your pixel id').track('purchase3Day')</code>
EVENT_PURCHASE_7_DAY	re-purchase on seventh day	<code>kwaiq.instance('input your pixel id').track('purchase7Day')</code>

3.2 How do you install pixel event code?

- You can call the **`kwaiq.instance('input your pixel id').track('input event name')`** function anywhere between `kwaiq.page()` in the base code and the closing `</body>` tag in the web page,either when the page loads,or when a visitor completes an action such as clicking a button.**Note: Do not place event code in <iframe>. The pixel id mentioned here has a one-to-one correspondence with the event name, that is, if you want to use the pixel with pixel id=123456 to track the event whose event name is purchase, call the function `kwaiq.instance('123456').track('purchase ')`**

For example, if you want to track a purchase event after a visitor has completed the purchase, you could call the **kwaiq.instance('input your pixel id').track('input event name')** function on your purchase confirmation page (The premise is the purchase confirmation page loaded the basic code):

```
<body>
...
<script>
  kwaiq.instance('input your pixel id').track('purchase')
</script>
...
</body>
```

If you want to track a addtocart event when the visitor clicks a addtocart button, you could tie the **kwaiq.instance('input your pixel id').track('input event name')** function call to the addtocart button on your page:

```
<button id="your addToCart button">addToCart</button>
<script type="text/javascript">
document.getElementById("your addToCart button").onclick = function () {
  kwaiq.instance('input your pixel id').track("addToCart", {
    content_id: "301",
    quantity: 1,
    price: 8,
    value: 9.2,
    currency: "USD",
  });
};
```

3.3 How do you include parameters for the events?

You can use **kwaiq.instance('input your pixel id').track(event, [parameters])**; for parameters. See sample code below:

```
//taking purchase event as an example
// For single product
kwaiq.instance('input your pixel id').track("purchase", {
  content_id: "301",
  content_type: "product",
  content_name: "dress",
  quantity: 1,
  price: 8,
  value: 9.2,
  currency: "USD",
});

// For multiple products
kwaiq.instance('input your pixel id').track("purchase", {
  contents: [
```

```

{
  content_id: "301",
  content_type: "product",
  content_name: "dress",
  quantity: 1,
  price: 8,
},
{
  content_id: "401",
  content_type: "product",
  content_name: "dress",
  quantity: 1,
  price: 1.2,
},
],
value: 9.2,
currency: "USD",
});

```

See below for the list of parameters currently supported:

Note: If you want to post back value information of the events, please send this two required parameters: 'value' and 'currency' (currently supported currency: BRL, IDR, USD)

Parameter name	Description	Required or optional	
content_type	The content_type object property's value must be set to either product, or product_group, depending on how you will configure your data feed when you set up your product catalog. If you will be tracking events associated with individual products, set the value to product. If you are tracking events associated with product groups, set it to product_group instead.	optional	Must be either product or product_group.

contents	Use contents when you have multiple content IDs. If you use contents in your parameter, you must also include the following in a sub-object: the product id or ids, and the quantity (number of items added to cart or purchased).	Either contents or content_id is required for DPA	Must be an array of objects (content parameter, id sub-object and quantity sub-object).
content_id*	Product or content identifier	Either contents or content_id is required for DPA	string
content_category	Category of the page/product	optional	string
content_name	Name of the page/product	optional	string
currency	<p>Refers to currency used as a means of circulation in social and economic activities such as USD</p> <p>Must in Uppercase letters, like "US", "BRL", "IDN", details in https://en.wikipedia.org/wiki/ISO_4217</p>	optional, recommended to include for ecommerce customer	enum(string)
value	The total price of the order, like 10.13	optional, recommended to include for ecommerce customer	number

quantity*	The number of product which is added to cart or bought by users	optional	number
price*	The price of the item, please use float number, like 4.99	optional	number
query	Used with the Search event. The string entered by the user for the search.	optional	string
mapping_id	Used to help MCDA Models identify users	optional, Pls ask your CST for help	string

3.4 How to ensure events are reported normally? (very important)

Kwai uses click_id and pixel_id to complete event reporting. When a user clicks on an ad to enter your webpage (the URL you filled in the ad set module of the Kwai for Business Ads Manager), the system generates a unique click_id and automatically appends it to the end of the webpage link. (Note: Different clicks on the same ad by the same user generate different click_ids, so click_ids are dynamically changing). For example:

- 1) As shown in the figure, the URL you filled in the ad set module of the Kwai for Business Ads Manager is:

<https://www.kwai.com/business>

The screenshot shows the Kwai Ads Manager interface. At the top, there are navigation links: Dashboard, Campaigns, Reports, Assets, Audiences, Tool, Notification Center, Alerts, Business Center, and Help. Below this, a breadcrumb navigation shows: Campaigns - Ad Set - Ad. The main content area has two sections:

- Conversion channel**: This section includes a dropdown menu set to "Website". Below it, a URL type dropdown is set to "PWA Promotion Manual >". Under "Internal URL" is a button labeled "Landing Page URL" which is highlighted with a red box. A text input field contains the URL "https://www.kwai.com/business". To the right of the URL input is a "Preview" link.
- Target Audience**: This section includes a "Create Audience" button. Below it are dropdown menus for "Include" and "Exclude" audiences, both set to "Optional".

2) When a user clicks on your ad, they will be directed to the URL you filled in the ad set module. At the same time, the system will automatically append the pixel_id and click_id to the end of the web page link, which will then be automatically changed to:

https://www.kwai.com/business?CampaignID=987654321&adSETID=123456789&CreativeID=123456789&click_id=CCpgibAfpRkSWv9zxgDuFHX71q5u%5FvMxqmvRCIE6enEynSNeQAAaSvb3CCcXoqXaPwbYpThJBW5n6%2D62kmmECfY5He3KFgOJB9YErIJ9cUk%3D&pixel_id=12345678987654321

3) You need to capture the values of the two parameters click_id and pixel_id from the above link (i.e.

`click_id=CCpgibAfpRkSWv9zxgDuFHX71q5u%5FvMxqmvRCIE6enEynSNeQAAaSvb3CCcXoqXaPwbYpThJBW5n6%2D62kmmECfY5He3KFgOJB9YErIJ9cUk%3D&pixel_id=12345678987654321`), and append these two parameters to the end of the link of every subsequent page visited, in the format:

`"click_id=CCpgibAfpRkSWv9zxgDuFHX71q5u%5FvMxqmvRCIE6enEynSNeQAAaSvb3CCcXoqXaPwbYpThJBW5n6%2D62kmmECfY5He3KFgOJB9YErIJ9cUk%3D&pixel_id=12345678987654321"`

You can achieve this requirement by adding the following code to

<https://www.kwai.com/business> (please evaluate the feasibility of this solution by yourself).

```
<script>
window.onload = function () {
    var links = document.getElementsByTagName("a");
    for (var i = 0, n = links.length; i < n; i++) {
        var href =
            links[i].href.trim() +
            (links[i].href.indexOf "?" > 0 ? "&" : "?") +
            document.location.search.replace("?", "").toString();
        links[i].href = href;
    }
};
</script>
```

Note⚠:

- 1) For page redirects that occur within a user's session after an ad click, ensure that **click_id** and **pixel_id** are passed from the original landing page to subsequent redirected pages. When directing users to subsequent pages, you need to append click_id and pixel_id to the end of the URL for each subsequent page.
- 2) During formal launch, the parameters to be appended are click_id and pixel_id , while during testing events, the parameters to be appended are click_id, pixel_id and ks_px_test (set ks_px_test=1 for previews).

Make sure pixel is working properly

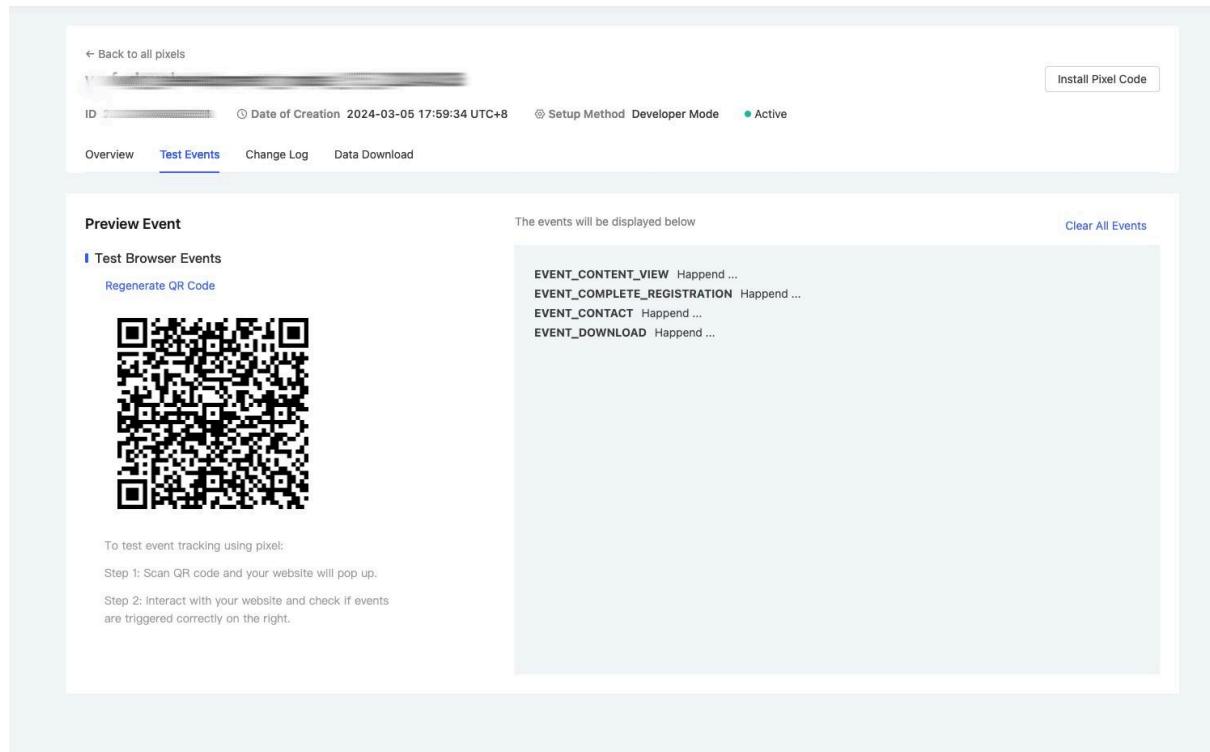
After you finish the set up, there are two ways to check whether pixel is working properly: 1. Pixel Helper 2. Preview an ad. These two are required

1. Pixel Helper

- Install kwai Pixel Helper.
- Use Kwai Pixel Helper to check if the pixel is installed. Refer to the [Pixel Helper documentation](#) for more information.

2. Preview Event

Through previewing Event, you can mimic the entire experience your customer will go through from clicking the ad to recording the conversion with pixel. Conversion occurred via preview ads will be recorded under the Preview Events column in pixel detail page in Event Manager, and it will not impact your ads performance.



FAQ

1. I'm ready to create an ad, and I want to choose an event for my conversion goal, but I don't see the event listed under Conversion Goal

Because there is a threshold for selecting a Conversion Goal, the events supported by the Optimization Goal and their thresholds are as follows. For details, see: [Product Manual for OCPM-Target cost](#)

Conversion Goal	Launch threshold
Add to Cart	When the preview of the Conversion Goal pixel at the account level is successful, this Conversion Goal can be selected.
Content View	When the preview of the Conversion Goal pixel at the account level is successful, this Conversion Goal can be selected.
Registration	When the preview of the conversion goal pixel at the account level is successful and the conversion goal has accumulated 10 conversion actions , this conversion goal can be selected.

Purchase

When the conversion goal accumulates 1 conversion action at the account level, this conversion goal can be selected.