Creating and editing a trigger

Learn how to create and edit a trigger, and define the evaluation point and rules for firing the trigger.
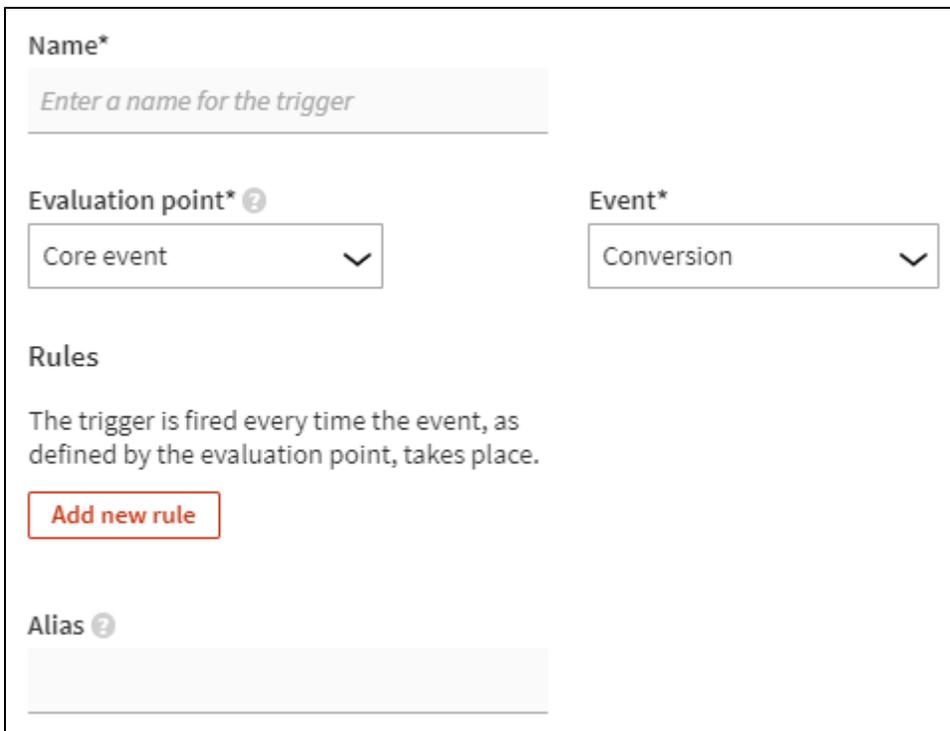
- Creating a trigger
- Editing a trigger
- Trigger settings
    - Evaluation point
    - Rules
    - Alias

For a hands-on example on how to create a simple trigger based on a page URL, see Creating your first trigger. For more complex trigger use cases, see Trigger examples.

Creating a trigger

To create a trigger:

1. In the Frosmo Control Panel, select **Triggers**.
2. Click **Create trigger**.
3. Define the trigger settings.

4. When you're done, click **Save**.

Editing a trigger

To edit a trigger:

1. In the Frosmo Control Panel, select **Triggers**.
2. In the triggers list, find the trigger you want to edit, and click its name.
3. Edit the trigger settings.
4. When you're done, click **Save**.

Trigger settings

The following table describes the settings you can define for a trigger in the Control Panel.

Table: Trigger settings

Setting	Description	Role
Name	Enter a name for the trigger.	Required
Evaluation point	Define the evaluation point for the trigger. To define the evaluation point: 1. Select the evaluation point (the event on the page that evaluates whether to fire the trigger). 2. Depending on the selected evaluation point, select the event name or type, the element the event must affect, and the timeout for trying to fire the trigger.	Required
Rules	Define rules for firing the trigger. To define a rule: 1. Click **Add new rule**. 2. Select a rule type. 3. Set the rule by filling in the required values. To add another rule, click **Add new rule** again. To remove a rule, click **X** for the rule. If you define no rules, the trigger is fired every time the trigger event, as defined by the evaluation point, takes place.	Optional
Alias	Define an alias for a trigger. You can use the alias to bind a JavaScript function to the trigger.	Optional

Figure: Defining the trigger settings

Evaluation point

An evaluation point determines the event on a web page that causes the Frosmo Platform to check whether to fire the trigger. The following table lists the evaluation points supported by the platform.

Table: Supported trigger evaluation points

Evaluation point	Options	Description
Core event	**Event**	A specific Frosmo Core event has occurred. If you select this event, also define the event type. You can select from the following event types: • Conversion • Frosmo custom action • Modification display • Modification click • Modification true display • Session start • Other predefined event • Product purchase • Visitor location change If you select **Other predefined event**, enter the name of the event.
Data layer event		Data has been pushed to the data layer. Every `dataLayer.push()` call on a page triggers this event.
DOM element ready	**Element** **Timeout (seconds)**	A specific DOM element has been loaded and parsed, and is ready to be manipulated. If you select this event, also define the element whose readiness fires the trigger and, optionally, the timeout (in seconds) after which the Frosmo Platform stops trying to fire the trigger. Define the element using a CSS selector; jQuery selectors are not supported. If you define no timeout (or set it to "0"), the platform keeps trying indefinitely.
DOM event	**Event** **Element**	A specific DOM event has occurred. If you select this event, also define the event type and the element to which the event must happen. Define the element using a CSS selector; jQuery selectors are not supported. You can select from the following event types: • Click • Key down • Key press • Key up • Submit For example, to fire the trigger on a link click, select **Click** as the event type, and define the element with the CSS selector "a".
DOM ready		All DOM elements have been loaded and parsed, and are ready to be manipulated.
Frosmo script loaded		Frosmo Core has been loaded and is ready to be initialized. ⚠ At this point, some DOM elements may not yet be available.
Frosmo script ready		Frosmo Core has been initialized. Frosmo context data (visitor data stored in `frosmo.easy.context`) has been loaded, segmentation has taken place, and the Frosmo custom script can now be initialized. ⚠ At this point, some DOM elements may not yet be available.

| JavaScript variable ready | Variable

Timeout (seconds) | A specific JavaScript variable defined in the `window` object has been loaded and is ready to be manipulated.

If you select this event, also define the variable name and, optionally, the timeout (in seconds) after which the Frosmo Platform stops trying to fire the trigger. If you define no timeout (or set it to "0"), the platform keeps trying indefinitely.

✅ If you need to check the value of a nested object property, add a **JavaScript variable** rule to the trigger, and define the nested object property as the variable using dot notation (for example, `object.object.property` or `object.object["property"]`). |
| URL change | | A URL change through the `hashchange` or `popstate` event has occurred.

⚠️ The **URL change** event is supported only for single-page applications and works only for as long as there is no new page load.

To use the **URL change** event, single-page application support must be enabled in the Frosmo custom script. To set up single-page application support, contact Frosmo support.

For more information about the `hashchange` and `popstate` events, see hashchange event and popstate event at MDN. |

Rules

Define the rules for firing the trigger. If you define no rules, the trigger is fired every time the trigger event, as defined by the evaluation point, takes place.

The Frosmo Platform supports the following rules:

- Cookie
- DOM element
- Event data
- JavaScript code
- JavaScript variable
- Local storage
- Page referrer
- Page URL

✅ Trigger rules support matching against Boolean values when the rule contains a value defined as "is exactly true" or "is exactly false".

Rules

The value of the `window.fp1419` variable

is exactly | true

Add new condition

Cookie

The trigger is fired based on the value stored in a cookie. You can get the value by cookie name.

Example

Evaluation point ⑦

Frosmo script loaded ⌄

Rules

The value of the `frosmo_keywords` cookie `contains` ⌄	✕
`tggp_232`	

Add new condition

The trigger is fired when the Frosmo script is loaded on the page and the value of the `frosmo_keywords` cookie contains the target group `tggp_232`.

DOM element

The trigger is fired if there is a specific DOM element on the page. To define the element, add a CSS selector and an attribute or property identifying the element:

- If the value in the field has the prefix `prop:`, the element node property must match.
- If the value in the field has the prefix `attr:`, the element node attribute must match.
- If there is no prefix, the system first tries to match properties and then attributes.

Example

Evaluation point ⑦

Core event ⌄

Event

Conversion ⌄

Rules

The value of the `discount.campaign-code` DOM element with the	✕
`textContent` attribute or property	
`is exactly` ⌄ `outdoor`	

Add new condition

The trigger is fired when a conversion occurs and value of the `textContent` attribute of the `discount.campaign-code` DOM element is `outdoor`. This can mean, for example, that a visitor purchases an item in an outdoor discount campaign and, as a result, a relevant modification is shown to the visitor.

Event data

The trigger is fired based on data passed by a predefined event. Specifically, the rule checks whether the value of an event data property matches a value you provide.

This rule is available only for the following evaluation points:

- **Core event**
- **Data layer event**
- **DOM element ready**
- **DOM event**
- **URL change**

The available properties on which to base the rule depend on the selected evaluation point and, in some cases, the selected event type. Depending on your selection, you either select the property from a list of predefined properties or manually define the property name.

The following table lists the available properties based on the selected evaluation point and, if relevant, event type.

Table: Available event data properties based on evaluation point and event type

Evaluation point	Event type	Event data properties
Core event	Conversion	- Conversion ID - Conversion type
	Frosmo custom action	- Custom action name - Custom action value
	Modification display/click/true display	- Modification ID - Variation ID - Variation type - Variation content
	Other predefined event	Depends on the event data.
	Product purchase	- Product ID - Product value
	Visitor location change	- Country - IP address - Internet service provider - Latitude - Longitude
Data layer event	Any	Depends on the event data.
DOM element ready	Any	- Element ID - Element classes - Element HTML - Element text - Element property
DOM event	Any	- Element ID - Element classes - Element HTML - Element text - Element property
URL change	Any	- Current URL - Previous URL

Example

The trigger is fired when a click is registered for a modification with the ID `12345`.

✓ If you use the **Event data** rule with the **Data layer event** evaluation point, define both the name and value of the property. You can use either dot notation or bracket notation to access a nested property (for example, `foo.bar` or `foo["bar"]`).

Note the following:

- If you define a property name that contains one or more periods (for example, `foo.bar`), the name is treated as a dot notation reference to an object property. If, instead, you want the name to be treated as a string that contains periods, use bracket notation and wrap the name in quotation marks (for example, `["foo.bar"]`).
- If you use an integer as a nested property name in dot notation (for example, `foo.1`), the integer is normalized into an array reference (for example, `foo[1]`).
- When you save the trigger, the property name may get normalized or trimmed depending on the notation and syntax you've used, so don't be surprised if the name looks a little different when you open the trigger settings again.

Example of the expected event data object when the property is `foo["foo.bar"]` and the data value is "1":

```
{"foo":{"foo.bar":"1"}}
```

ⓘ If multiple event data arguments are passed to the triggered event using the `frosmo.easy.events.trigger()` function, only the first argument is registered.

JavaScript code

The trigger is fired based on your own JavaScript code. The code must return a value to be matched.

Example

The trigger is fired when the page DOM is ready to be manipulated and the visitor is using a desktop device.

JavaScript variable

The trigger is fired based on a JavaScript variable value on the page. The variable must be available on the `window` object.

If you need to check the value of a nested object property on the page, use dot notation to reference the property, for example, `object.object.property`.

Example

The trigger is fired when the page DOM is ready to be manipulated, and a JavaScript variable named `productBrand` with the value `Frosmo` exists on the page.

Local storage

The trigger is fired based on data stored in local storage. You can get the data by the key name.

Evaluation point ⓘ

DOM ready ▾

Rules

×

The value of the | phoneNumber | local storage item

begins with ▾

09

Add new condition

The trigger is fired when the page DOM is ready to be manipulated, and the value of the `phoneNumber` local storage item begins with `09`. This can help you, for example, to target visitors in a certain geographical area.

Page referrer

The trigger is fired when the visitor arrives from a specific page.

You can define the page based on:

- **URL**: Matches the exact URL of the page. For example: `http://www.example.com/index.html`
- **Protocol**: Matches the HTTP or HTTPS part of the URL.
- **Domain**: Matches the part after the protocol. For example, in the URL `http://www.example.com/index.html`, the domain is `www.example.com`.
- **Path**: Matches the URL path after the domain. For example, in the URL `http://www.example.com/index.html`, the path is `/index.html`.
- **Anchor**: Matches any part of the URL that comes after the "#" character.
- **Query selector**: Matches any part or the URL that comes after the "?" character. If the query parameter is defined, it will only match a single parameter. For example, in the URL `http://example.com/?search=word&count=10`, the complete query parameters are `search=word&count=10`, and the query key for search is `word`.

Evaluation point ⓘ

Custom event ▾

Event

New visit ▾

Rules

×

The referrer | domain ▾

contains ▾ | /facebook/

Add new condition

The trigger is fired when a new site visitor enters the page from a page with a domain containing the string `/facebook/`.

Page URL

The trigger is fired on a specific page.

You can define the page based on:

- **URL**: Matches the exact URL of the page. For example: `http://www.example.com/index.html`
- **Protocol**: Matches the HTTP or HTTPS part of the URL.
- **Domain**: Matches the part after the protocol. For example, in the URL `http://www.example.com/index.html`, the domain is `www.example.com`.
- **Path**: Matches the URL path after the domain. For example, in the URL `http://www.example.com/index.html`, the path is `/index.html`.
- **Anchor**: Matches any part of the URL that comes after the "#" character.
- **Query selector**: Matches any part or the URL that comes after the "?" character. If the query parameter is defined, it will only match a single parameter. For example, in the URL `http://example.com/?search=word&count=10`, the complete query parameters are `search=word&count=10`, and the query key for search is `word`.

For an example of how to use the **Page URL** rule, see Creating your first trigger.

Alias

You can define an alias for a trigger. You can use the alias to bind a JavaScript function to the trigger. For example, use the following code to bind an alias named "myTriggerAlias" to a trigger:

```
frosmo.easy.events.on('myTriggerAlias', function (trigger) {...});
```

The bound function will be called every time the trigger fires. The function receives the trigger object as an argument. If you want to call the bound function once, use the `frosmo.easy.events.once()` function.