Learn & Master

Google Analytics Event Tracking

By Himanshu Sharma
Founder of OptimizeSmart.com

Google Analytics Event Tracking

Written by Himanshu Sharma, Founder of Optimize Smart

About the author

- Founder, OptimizeSmart.com

- Over 15 years of experience in digital analytics and marketing

- Author of four best-selling books on digital analytics and conversion optimization

- Nominated for Digital Analytics Association Awards for Excellence

- Runs one of the most popular blogs in the world on digital analytics

- Consultant to countless small and big businesses over the decade

Website: www.optimizesmart.com

Linkedin: https://www.linkedin.com/in/analyticsnerd

Facebook: https://www.facebook.com/optimizesmart

Optimize Smart
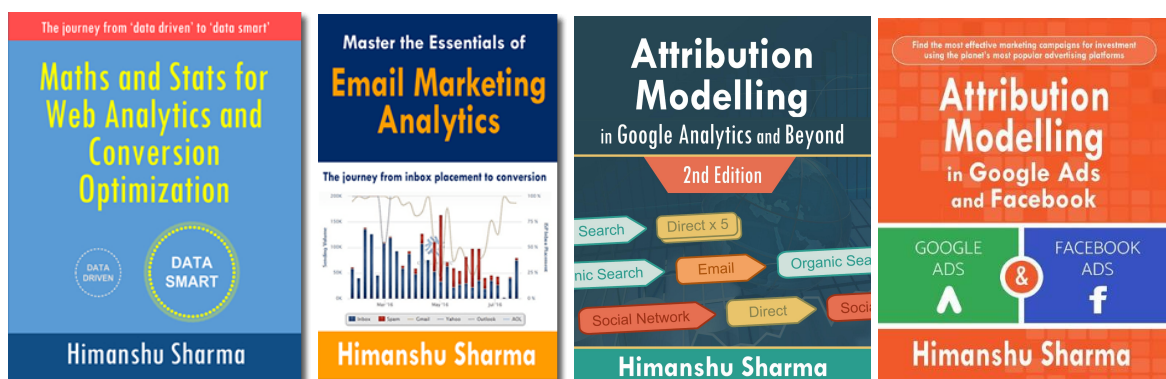
Following are our most downloaded ebooks for career advancement:

#1 [Sales and ROI Accelerator (150+ pages)](#)

WHAT'S INSIDE: My step-by-step blueprint for generating record sales and ROI by leveraging analytics data.

#2 [62 Points Google Analytics Setup Checklist (50 pages)](#)

WHAT'S INSIDE: Learn to set up your Google Analytics account correctly and fast using this 62 point checklist. This is no ordinary checklist - it is a result of more than a decade of experience in the analytics industry.

#3 [Google Tag Manager Data Layers (100+ pages)](#)

WHAT'S INSIDE: My step-by-step blueprint for getting started with data layers. Get the only ebook on GTM data layers ever published. Learn the JavaScript behind it.

#4 [Learn to Read E-Commerce Reports in Google Analytics (100+ pages)](#)

WHAT'S INSIDE: My step-by-step guide to reading both standard and enhanced e-commerce reports in Google Analytics. E-commerce reports are the most valuable reports in Google Analytics.

Table of Contents

Optimize Smart

What is event tracking in Google Analytics?

Event tracking is an advanced Google Analytics feature that allows you to track a specific user's interaction/activity with a web page element.

The user's interaction/activity with a web page element that you track in Google Analytics is called an 'event'.

Following are the examples of events that can be tracked in Google Analytics:

1. Downloading a file

2. Loading of Ajax, JavaScript or Flash content

3. Loading of a dynamically generated webpage

4. Loading of pop-ups, lightboxes

5. Loading of a video on a webpage

6. Scrolling down the page

7. Viewing of a video

8. Viewing video footage of certain length

9. Clicking on a video's play/pause/stop button

10. Interaction with a gadget

11. Clicking on an image or an external link

12. Abandonment of a form field

13. Log-ins

14. Sharing/printing a blog post, article, video or image

15. Clicking on a button

16. Movement of mouse

Following are examples of web page elements:

- video
- gadget
- podcast
- image
- button
- form
- scroll bar
- external link
- lightbox
- Ajax content etc.

When to use event tracking in Google Analytics?

There are two broad categories of events:

#1 **Events that generate pageviews when they occur.** For example, clicking on an internal link on a webpage generates a pageview.

#2 **Events that do not generate pageviews when they occur.** For example, following users' interactions do not generate a pageview when they occur:

- Clicking on an external link
- Loading of Ajax/Flash content

- Downloading a file

- Playing a video

- Pausing a video

- Reposition timing within a video.

- Buffering of a video

- Progress of a video

- Scrolling a web page

- Clicking on a form button

- Interaction with a gadget

- Clicking on an image

- Abandonment of a form field

- Sharing/printing a blog post, article

- Adding items to a shopping cart.

- Clicking on a button, etc.

By default Google Analytics (GA) can not track any event that does not generate pageview when it occurs.

You can track such events only through event tracking or virtual pageviews.

The rule of thumb is, track those types of users' interactions, using 'event tracking' which either do not generate a pageview when they occur or which are not equivalent to a page being viewed.

Structure of Google Analytics events

An event in Google Analytics is made up of:

1. Event Category
2. Event Action
3. Event Label
4. Event Value

#1 Event Category

The 'event category' is the name assigned to the **group of similar events you want to track**. You can use any value of type 'text' as an event category.

If you have already set up event tracking then you can see the values of the 'event category' in Google Analytics by navigating to ***Behavior* > *Events* > *Top Events***:
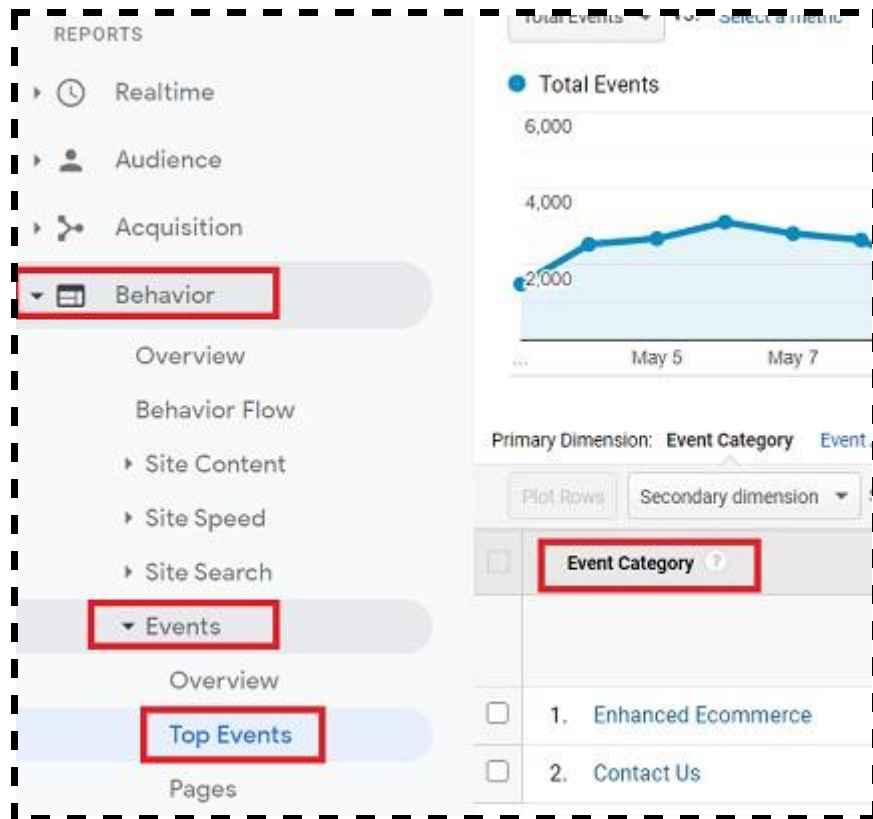
Google Analytics report event category as a [dimension](#) in its reports:
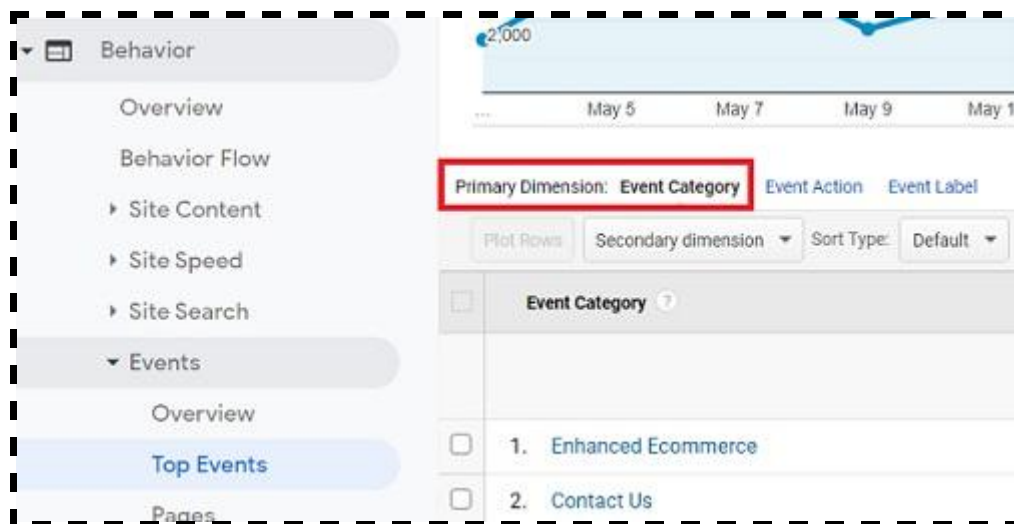
#2 Event Action

The 'Event Action' is the name assigned to **the type of event you want to track** for a particular webpage element. You can use any value of type 'text' as an event action.

If you have already set up event tracking then you can see the values of the 'event action' in Google Analytics by navigating to *Behavior > Events > Top Events* and then click on the *'Event Action'* primary dimension:
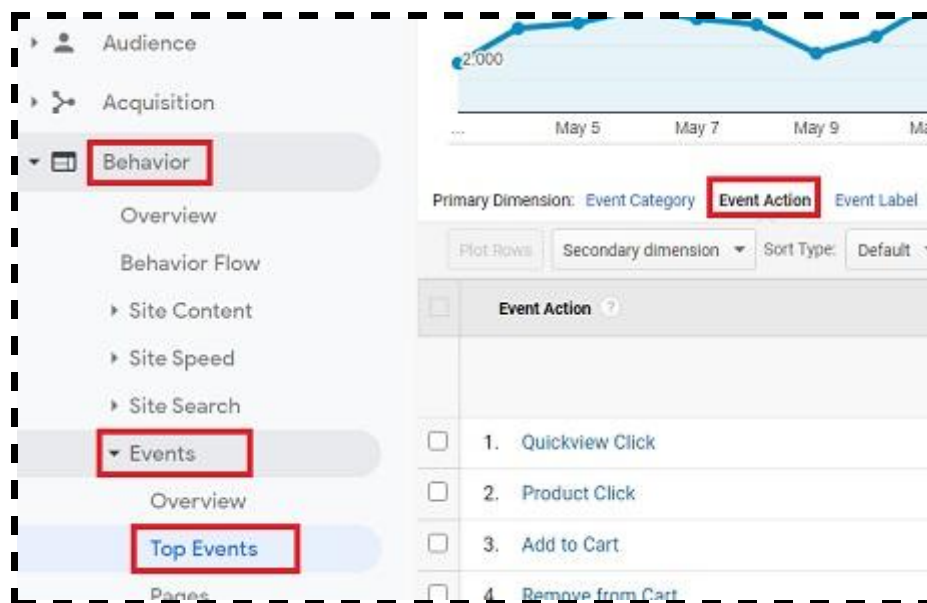
Note: Google Analytics report *event action* as a **dimension** in its reports.

#3 Event Label

The 'event label' is the **name assigned to the web page element, whose users' interaction you want to track**. You can use any value of type 'text' as an event label.

An *Event label* can be a title of a video, title tag of a web page, name of a gadget, name of the downloadable file, etc.

If you have already set up event tracking then you can see the values of the 'event label' in Google Analytics by navigating to **Behavior** > **Events** > **Top Events** and then click on the **'Event Label'** primary dimension:
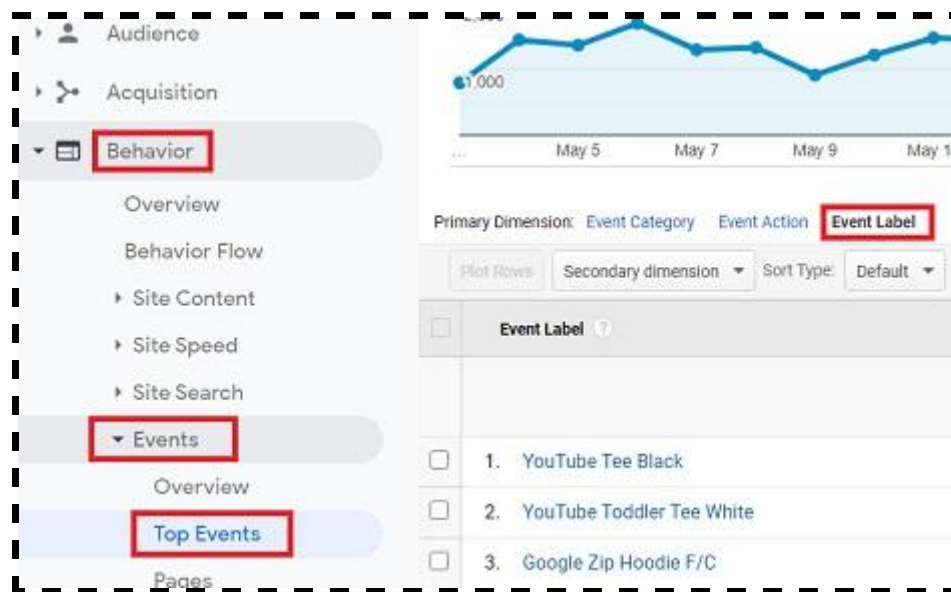
Note: Google Analytics report *event label* as a **dimension** in its reports.

#4 Event Value

The 'event value' is the **numerical value assigned to the event you want to track.** You can use any value of type 'number' as an event value. However, you can not use any value of type 'text' as an event value.

For example, an event value can be any number that represents a download time, length of the video footage played, or some monetary value.

If you have already set up event tracking then you can see the values of the 'event label' in Google Analytics by navigating to *Behavior > Events > Top Events*:

|	Event Action ?	Total Events ? ↓	Unique Events ?	Event Value ?
		18,272 % of Total: 18.05% (101,232)	**5,688** % of Total: 5.59% (101,741)	**22,966,423** % of Total: 100.00% (22,966,423)
☐	1. How to select best Excel Charts for	1,840 (10.07%)	606 (9.27%)	189,739 (0.83%)
☐	2. Understanding Key Performance Inc	1,319 (7.22%)	445 (6.81%)	61,427 (0.27%)
☐	3. Geek guide to removing referrer spa	942 (5.16%)	338 (5.17%)	861,319 (3.75%)
☐	4. 11 powerful methods to reduce bour	757 (4.14%)	284 (4.35%)	1,658,349 (7.22%)

Note(1): If you want to assign a monetary value to a tracked event then use the 'event value'.

Note(2): Google Analytics report *event value* as a **metric** in its reports.

Google Analytics Event Tracking Best Practices

Following are the best practices for Event Tracking in Google Analytics

#1 Track those types of users' interactions as events which either do not generate a pageview when they occur or which are not equivalent to a page being viewed.

#2 Do not omit 'Event Action' while setting up event tracking in Google Analytics.

If you are using analytics.js library to set up event tracking then specifying 'event category' and 'event action' is mandatory. Whereas, specifying 'event label' and 'event value' is optional.

Example:

```
ga('send','event','videos','','chappy',100); // incorrect because 'event action' is missing.

ga('send','event','','Play','chappy',100); // incorrect because 'event category' is missing.

ga('send','event','videos','Play','chappy',100); // correct
```

Note: If you are using the analytics.js library then default values of eventCategory, eventAction, eventLabel and eventValue fields is '(not set)'.

If you are using the global site tag (gtag.js) then specifying the 'event category', 'event label', and 'event value' is optional. However, specifying 'event action' is mandatory.

Example:

```
gtag('event','' , {

'event_category' : 'videos',
```

'event_label' : 'Chappy'

}); // Incorrect because 'event action' is missing.

gtag('event', 'Play', {

'event_category' : 'videos',

'event_label' : 'Chappy'

}); //correct

Note: If you are using the gtag.js library then the default value of the 'event category' is 'engagement' and the default value of the 'event label' is (not set).

#3 Specifying field values in the correct order is important when hardcoding the event tracking code

Consider the following event tracking code:

ga('send', {

'hitType': 'event',

'eventCategory': 'videos',

'eventAction': 'Play',

'eventLabel': 'chappy',

'eventValue': 100

});

You can also write this line of code without using any [field names](#):

ga('send','event','videos','Play','chappy',100);

Here, Google Analytics will automatically assume that:

- event category is 'videos',
- event action is 'Play'
- event label is 'Chappy'
- event value is 100

When you do not specify field names, you then need to supply field values in the following order:

Event Category > Event Action > Event Label > Event Value

Since field values are executed in the order in which they are supplied, if I changed the order, my event tracking will work differently or worse won't work at all.

For example:

ga('send','event','Play','videos','chappy',100);

Here 'Play' will be treated as an event category and 'videos' will be treated as an event action.

Another example:

ga('send','event','Play','videos',100);

here 100 will be treated as event label instead of event value.

To avoid this issue, you can leave the 'event label' field empty:

```
ga('send','event','Play','videos','',100);
```

#4 Use the correct data type for your field values while setting up event tracking

The *eventCategory*, *eventAction* and *eventLabel* fields accept a value of type text. Whereas, the *eventValue* field accepts a value of type integer.

So following event tracking codes are going to create tracking issues:

ga('send','event',videos,'Play','chappy',100); // incorrect because 'event category' accepts value of type text.

ga('send','event','videos','Play','chappy','100'); // incorrect because 'event action' accepts value of type integer.

Note: There is no hard and fast rule regarding what data you put into 'event category', 'event action', 'event label' and 'event value' as long as you are assigning values of type text to 'event category', 'event action' and 'event label' and values of type integer to 'event value'.

#5 Follow naming conventions for event tracking

Use descriptive names for event categories, event actions, and event labels. So that just by looking at the name, a user can understand what type of data is being collected.

Likewise, use consistent names for event categories, event actions, and event labels.

So if you are using 'videos' as an event category, then do not use 'video', 'Videos', 'VIDEOS' as an event category. Otherwise, you will end up creating several different event categories in Google Analytics that all hold the same type of event data.

One of the best ways to use consistent names is by using only lowercase letters for event categories, event actions, and event labels.

#6 Use Google Tag Manager to track events.

There are two methods to set up event tracking in Google Analytics:

#1 Hard code the Google Analytics event tracking code on your website that tracks a specific user's interaction (aka event) with a web page element.

#2 Use Google Tag Manager to track users' interaction with the web page elements.

The advantage of using GTM to track events is that it is easier to use and is scalable.

So if you want to track clicks on the buttons across your website in one go without manually adding the event tracking code to each and every button, you can easily do that via Google Tag Manager.

#7 Avoid event tracking from negatively impacting your website bounce rate through non-interaction event

The 'nonInteraction' event is used to designate an event as interactive or non-interactive. This field object can have a value of true or false.

The default value is false, which means by default an event is considered as interactive and it impacts the bounce rate calculations. If you want to make an event non-interactive so that it doesn't impact the bounce rate calculations then set the nonInteraction value to true.

The **{'nonInteraction': true}** is used to designate an event as non-interactive. You can also use **{'nonInteraction': 1}** to designate an event as non-interactive.

Example

If you are using the analytics.js library:

```
ga('send','event','videos','Play','chappy',100, {'nonInteraction': 1});

ga('send','event','videos','Play','chappy',100, {'nonInteraction': true});
```
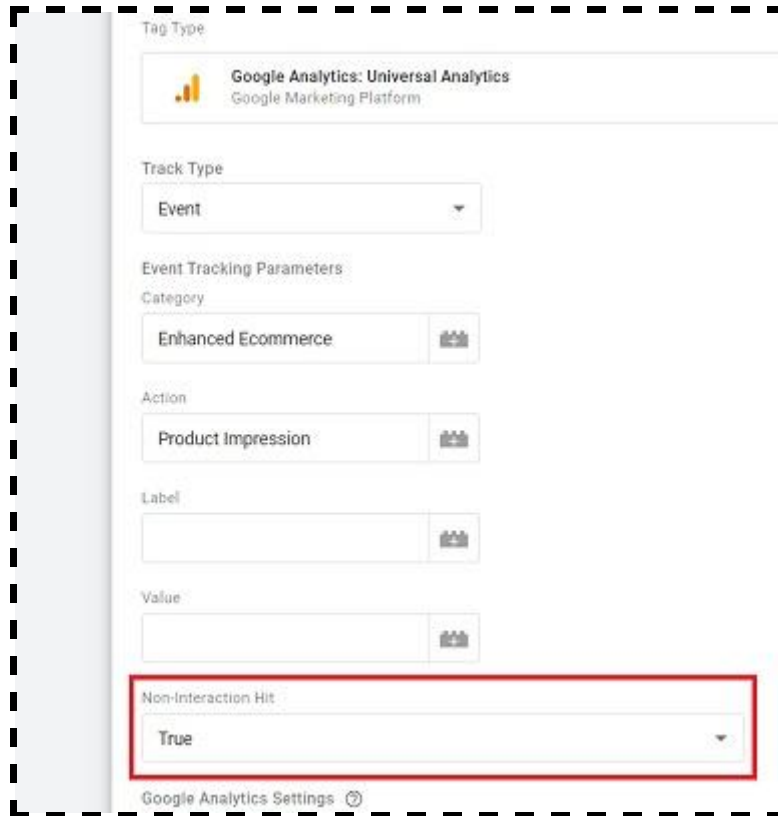
If you are using the gtag.js library:

```
gtag('event', 'Play', {
```

```
'event_label': 'chappy',

'event_category': 'videos',

'value': 100,

'non_interaction': 1

});

gtag('event', 'Play', {

'event_label': 'chappy',

'event_category': 'videos',

'value': 100,

'non_interaction': true

});
```

For [Google Tag Manager](#):

#8 Avoid using duplicate 'event action' names across categories

There is a disadvantage of using duplicate 'event action' names across categories:

ga('send', 'event','videos', '**play**','Spiderman-2',10);

ga('send', 'event','videos', '**play**','Who lets the dog out',10);

ga('send', 'event','videos', '**play**','Alien vs predator',10);

The disadvantage is that the **unique events are incremented by unique action names.**

So if a user clicks on two different videos say 'spiderman-2' and 'Aliens vs predator' which have the same action name 'play', then, in that case, Google Analytics will report only one unique event, even when the user clicked on two different videos.

So action names needs to be unique:

```
ga('send', 'event','videos', 'play-spiderman','Spiderman-2',10);

ga('send', 'event','videos', 'play-dog-out','Who lets the dog out',10);

ga('send', 'event','videos', 'play-aliens-predator','Alien vs predator',10);
```

If you are using the gtag.js libabry then the above code can be written as:

```
gtag('event', 'play-spiderman', {

'event_category' : 'videos',

'event_label' : 'Spiderman-2'

});

gtag('event', 'play-dog-out', {

'event_category' : 'videos',

'event_label' : 'Who lets the dog out'

});
```

```
gtag('event', 'play-aliens-predator', {

'event_category' : 'videos',

'event_label' : 'Alien vs predator'

});
```

#9 Plan out your even tracking structure in advance before you set up event tracking

Before you implement event tracking:

#1 Determine in advance, all of the web page elements (like videos, gadgets, image, external link, etc) you want to track and the type of users' interactions (aka events) you want to track for each web page of your website.

#2 Create a hierarchy of event categories, event actions, and event labels that is scalable. So that you don't need to frequently change the names of event categories, event actions, and event labels to hold different types of event data.

#3 Use an Excel Spreadsheet to decide the names of all your event categories, event actions, and event labels in advance. Then work with your report users, to make sure that your hierarchy is understandable:

Event Category	Event Action	Event Label	Event Value
Guides	Download-Social	Social Media Guide	10
Guides	Download-SEO	SEO Guide	15
Videos	Play-Spider	Spider Man	5
Videos	Play-Batman	Batman Begins	5

How to set up event tracking in Google Analytics?

In order to set up event tracking in Google Analytics, follow the steps below:

Step-1: Determine in advance, all of the web page elements (like videos, gadgets, image, external link, etc) and the type of users' interactions you want to track for each web page of your website.

Step-2: Create a hierarchy of event categories, event actions, and event labels that is **scalable.** So that you don't need to frequently change the names of event categories, event actions, and event labels to hold different types of event data.

Step-3: Use an Excel Spreadsheet to decide the names of all your event categories, event actions, and event labels in advance. Then work with your report users, to make sure that your hierarchy is understandable:

Event Category	Event Action	Event Label	Event Value
Guides	Download-Social	Social Media Guide	10
Guides	Download-SEO	SEO Guide	15
Videos	Play-Spider	Spider Man	5
Videos	Play-Batman	Batman Begins	5

Step-4: Chose a method you want to use for setting up event tracking in Google Analytics.

There are two methods to set up event tracking in Google Analytics:

#1 Hard code the Google Analytics event tracking code on your website that tracks a specific user's interaction with a web page element.

#2 Use Google Tag Manager to track users' interaction with the web page elements.

#1 Hard code the Google Analytics event tracking code on your website

Let us suppose you want to track clicks on a download button (embedded on a web page on your website) in Google Analytics via event tracking.

Follow the steps below:

Step-1: Navigate to the web page (via Google Chrome web browser) which contains the button you want to track via event tracking.

Step-2: Right-click on the button and then click on 'Inspect' from the drop-down menu:

Step-3: Find your button code. It is going to look something like the one below:

Optimize Smart
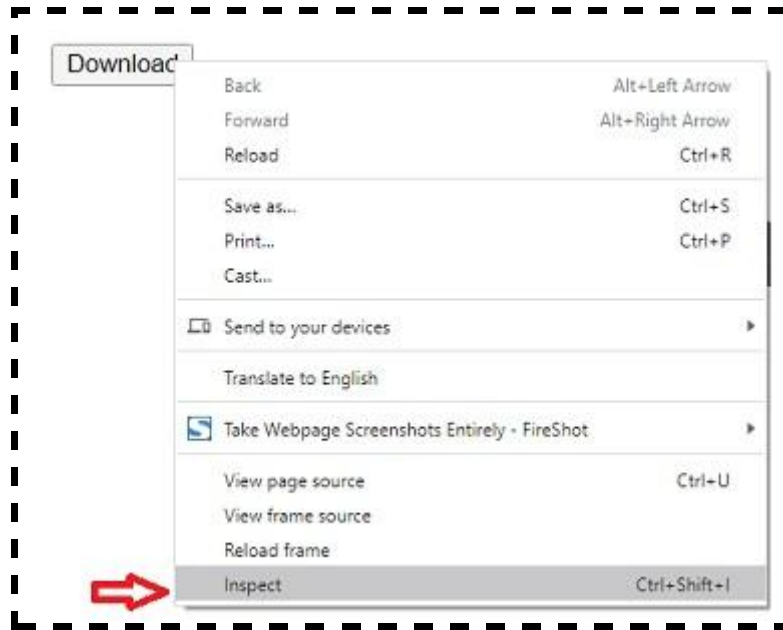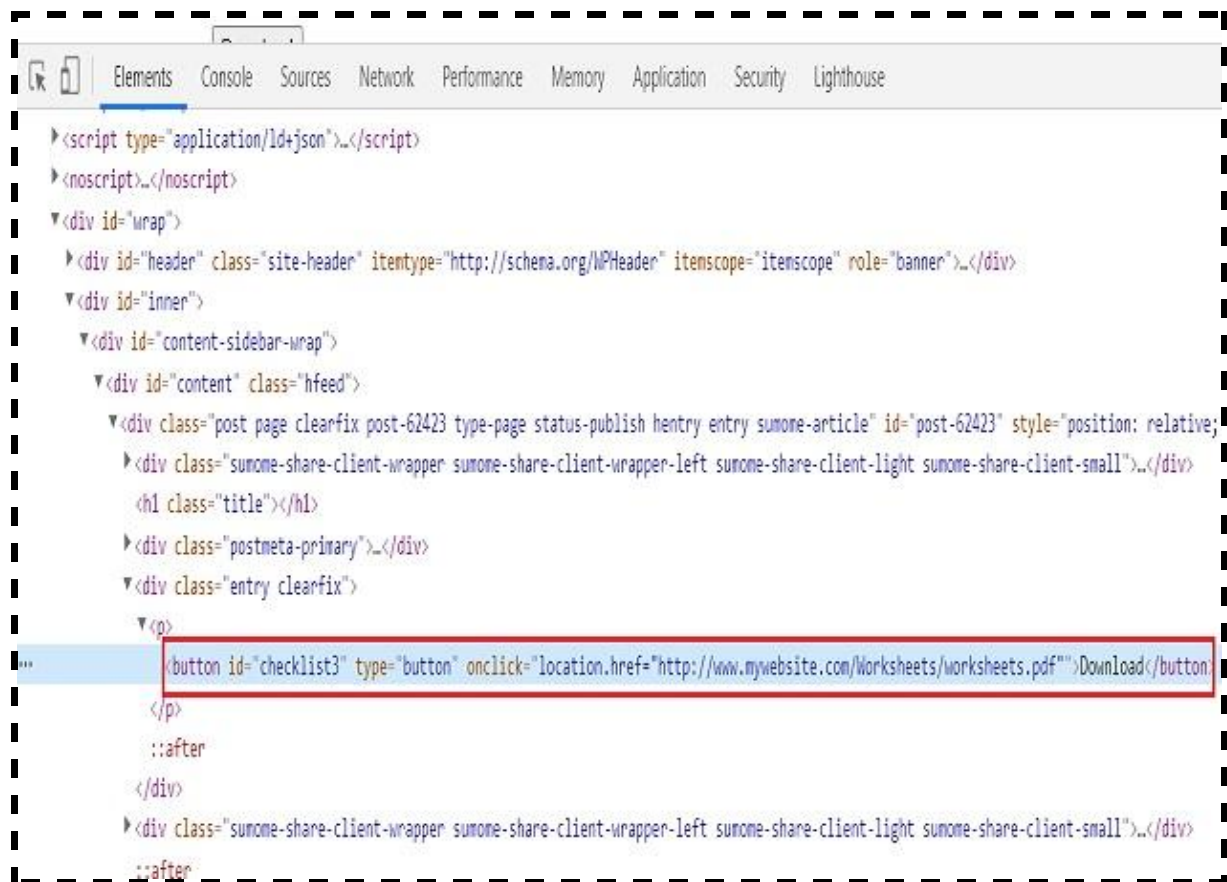
```
<button id="checklist3" type="button"
onClick='location.href="http://www.mywebsite.com/Worksheets/works
heets.pdf"'>Download</button>
```

Step-4: Copy the button code into your notepad.

Step-5: Create your Google Analytics Tracking Code using the following syntax:

ga('send','event','**<INSERT YOUR EVENT CATEGORY>**','**<INSERT YOUR EVENT ACTION>**','**<INSERT YOUR EVENT LABEL'>**,**<INSERT YOUR EVENT VALUE>**);

For example,

ga('send', 'event', 'guides', 'download-cro', 'cro-beginners-guide.pdf', 10)

Here, Google Analytics will automatically assume that:

event category is 'guides',

event action is 'download-cro'

event label is 'cro-beginners-guide.pdf'

event value is 10

Step-6: Add your Google Analytics Event tracking code to your button code:

<button id="checklist3" type="button" onClick="**ga('send', 'event', 'guides', 'download-cro', 'cro-beginners-guide.pdf',**

10);location.href="http://www.mywebsite.com/Worksheets/worksheets.
pdf">Download</button>

Here, **onClick** is a **mouse event handler.** It executes a method/function
(called 'ga') in response to click on the button. This event handler is used
to tie the event tracking code to the button.

Note: You can also use the **event tracking code builder tool** to generate event
tracking code.

Step-7: Open your FTP tool and then navigate to the web page/file which
contains the button you want to track via event tracking.

Step-8: Download the file, open it, and then replace the button code with your
new button code (the button code with event tracking code).

Step-9: Upload the file to your web server. This will replace the old file with the
new version.

Step-10: Navigate to the web page (via Google Chrome web browser) which
contains the button you want to track.

Step-11: Click on the button.

Step-12: Navigate to your Google Analytics view and then click on 'Events'
under 'Realtime':
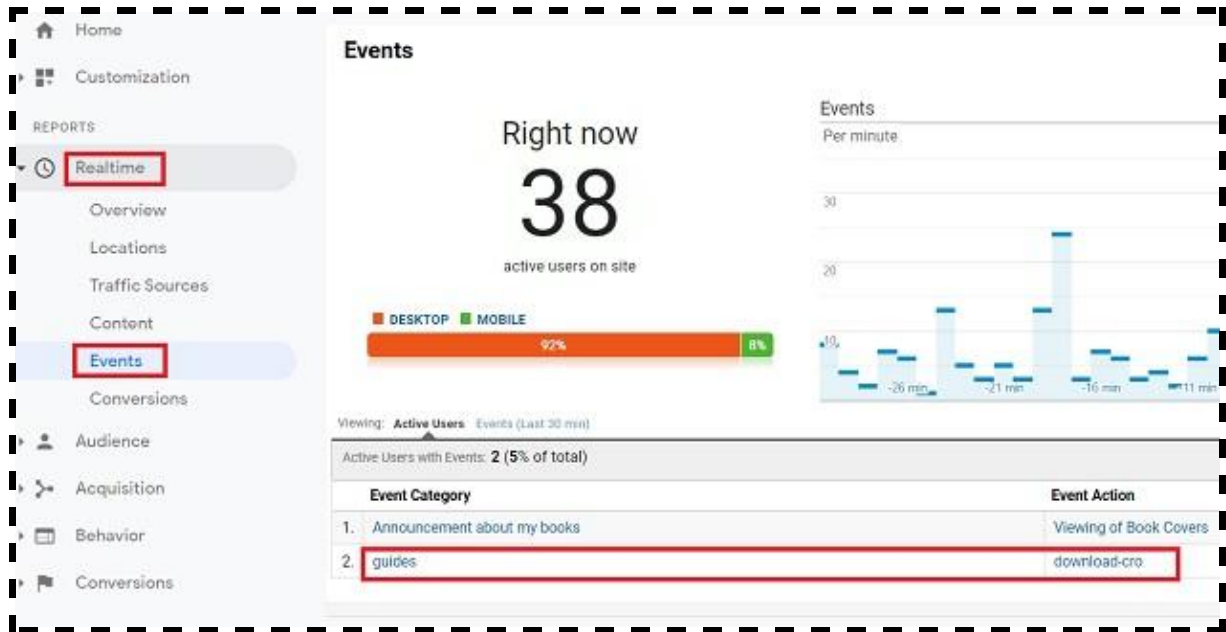
You should now be able to see your event category and event action values being reported.

If you don't see your event category and/or event action then it means your event tracking is not set up correctly.

#2 Use Google Tag Manager to track users' interactions (aka events) with the web page elements.

This is the recommended method to track users' interactions with web page elements as it is easier to use and is scalable.

So if you want to track clicks on the buttons across your website in one go without manually adding the event tracking code to each and every button, you can easily do that via Google Tag Manager.

Optimize Smart

Let us suppose you want to track clicks on download buttons embedded on dozens of web pages on your website via event tracking.

Follow the steps below:

Step-1: Navigate to the web page (via Google Chrome web browser) which contains the button you want to track via event tracking.

Step-2: Right-click on the button and then click on 'Inspect' from the drop-down menu:
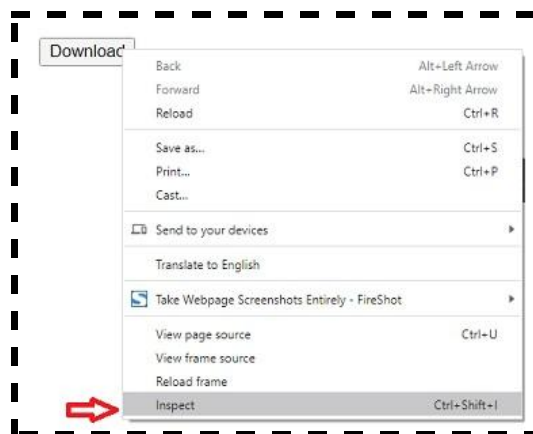
Step-3: Find your button code. It is going to look something like the one below:

<button id="checklist3" type="button"

onClick='location.href="http://www.mywebsite.com/Worksheets/worksheets.p df"'>Download</button>

Step-4: Note down the 'id' attribute of the button. In our case, it is '*checklist3*'.

Note: The 'id' value that you use should be unique. If the 'id' attribute is missing, then ask your developer to add it.

Step-5: [Install Google Tag Manager on your website](link) (if you have not already).

Step-6: Login to your Google Tag Manager account.

Step-7: Click on the 'Triggers' link and then click on the 'New' button to create a new trigger:
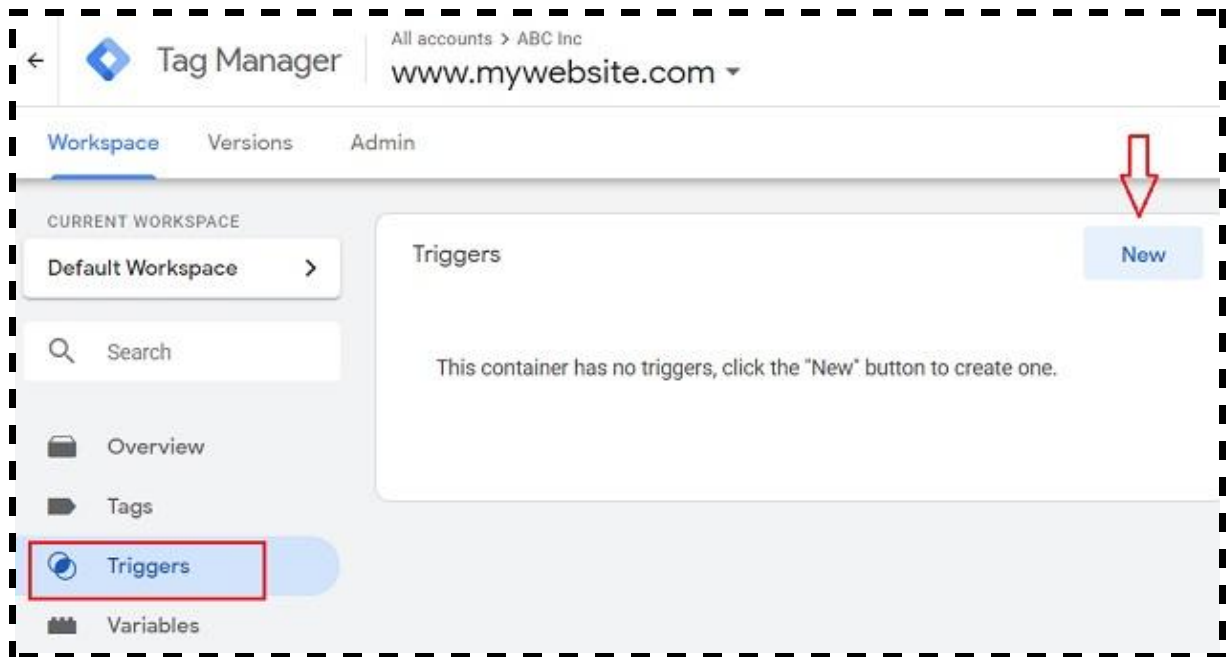
Step-8: Name the new trigger '*Check for clicks on the download button*'
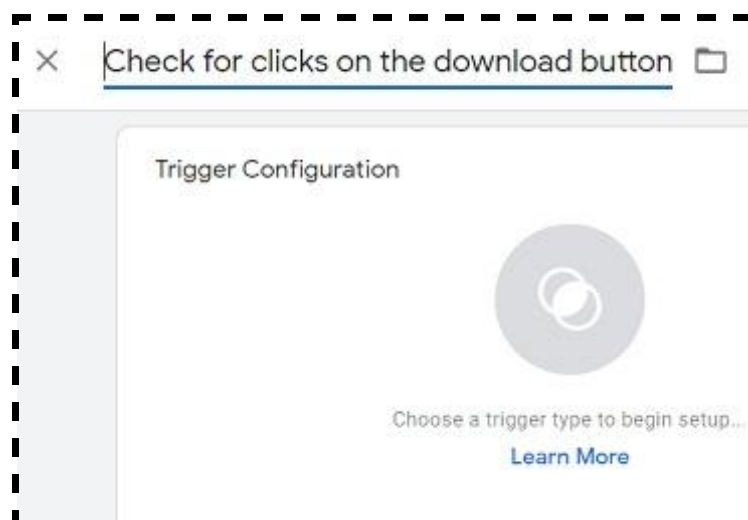
Step-9: Click on '*Choose a trigger type to begin setup*':
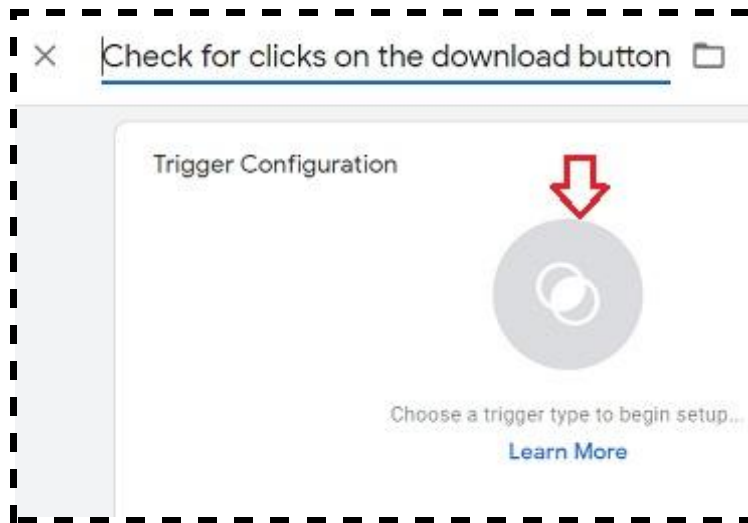
Step-10: Click on 'All Elements':
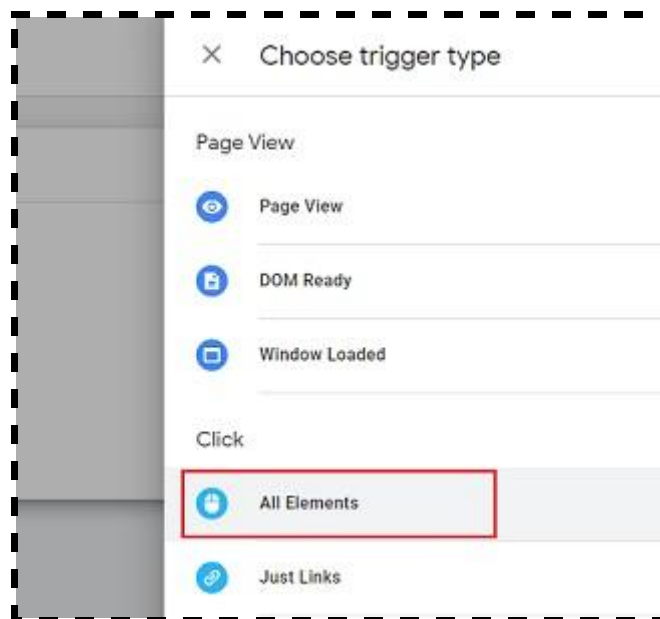
Step-11: Click on 'Some Clicks':
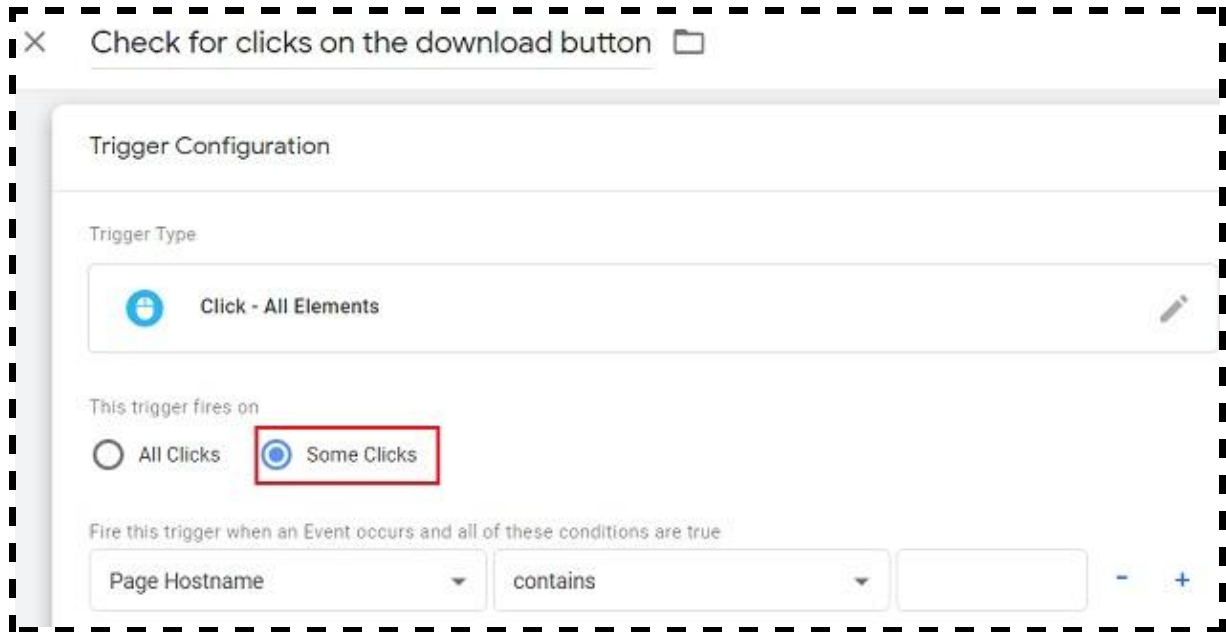
Step-12: Click on 'choose built-in variable' from the drop-down menu:
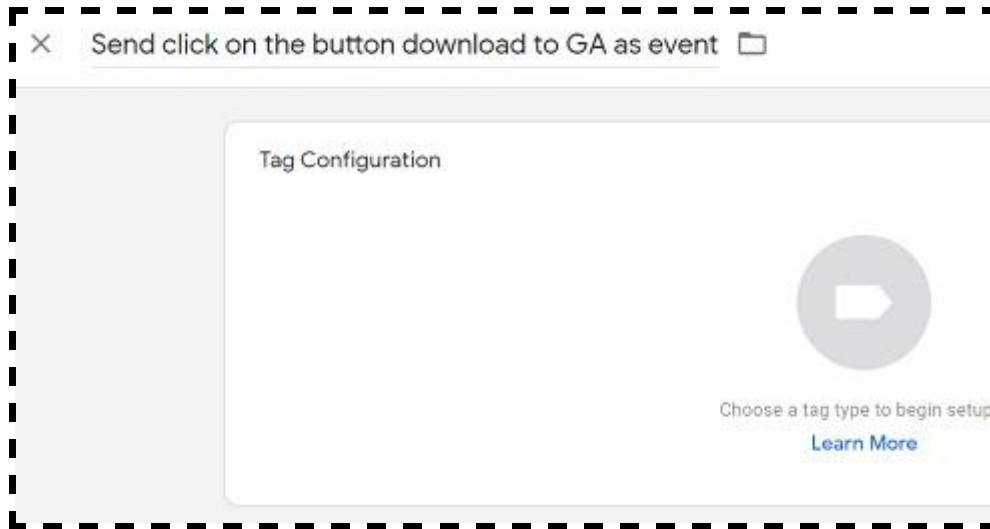

Step-13: Select 'Click ID':

Step-14: Enter the id of your button element in the text box next to 'equals' and then click on the 'Save' button:
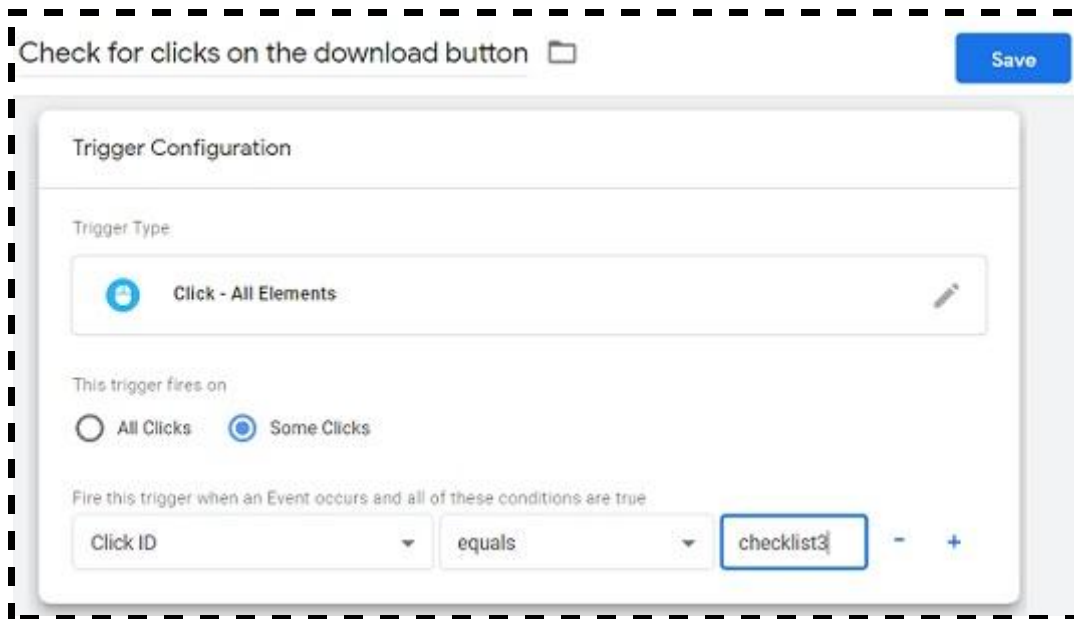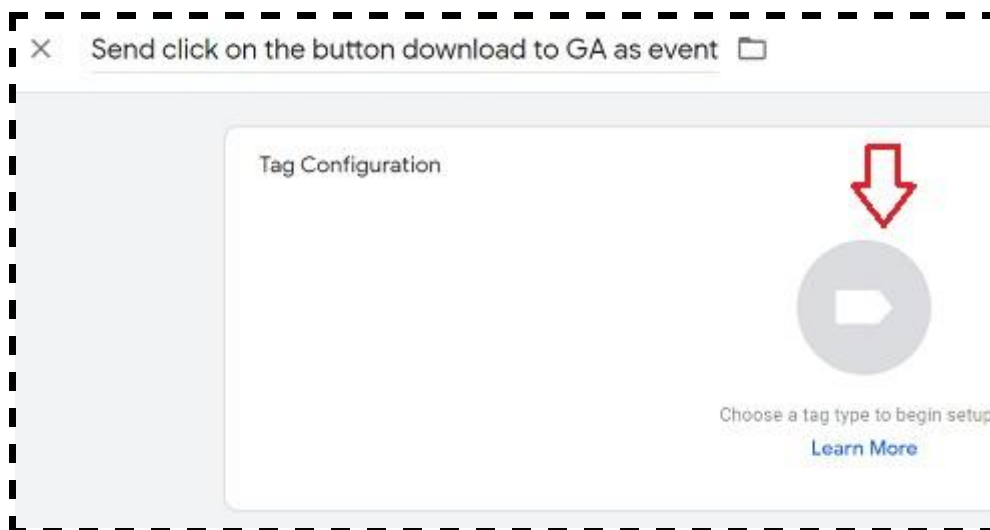
Step-15: Click on the 'Tags' link and then click on the 'New' button to create a new tag:
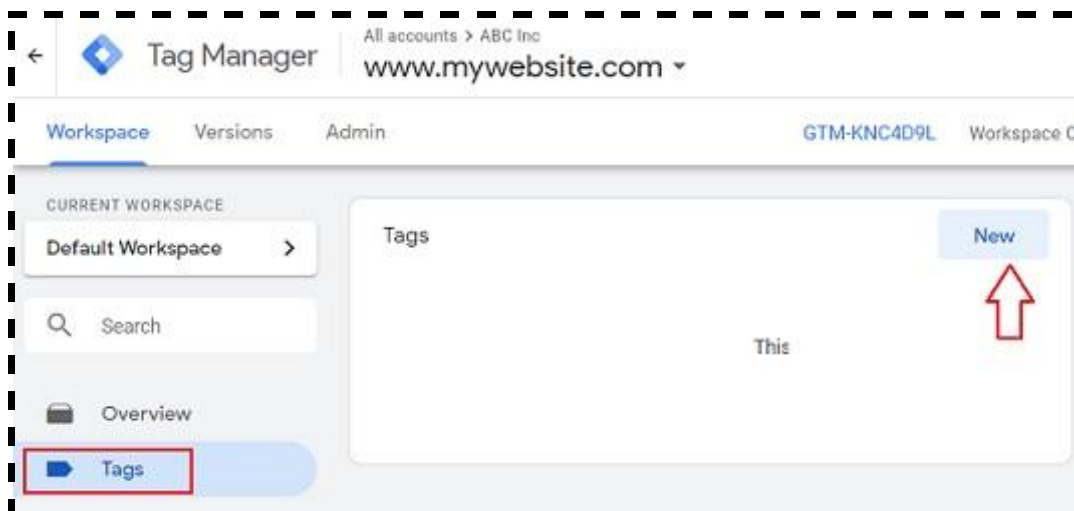
Step-16: Name your new tag 'Send click on the button download to GA as event':

Step-17: Click on 'Choose a tag type to begin with':
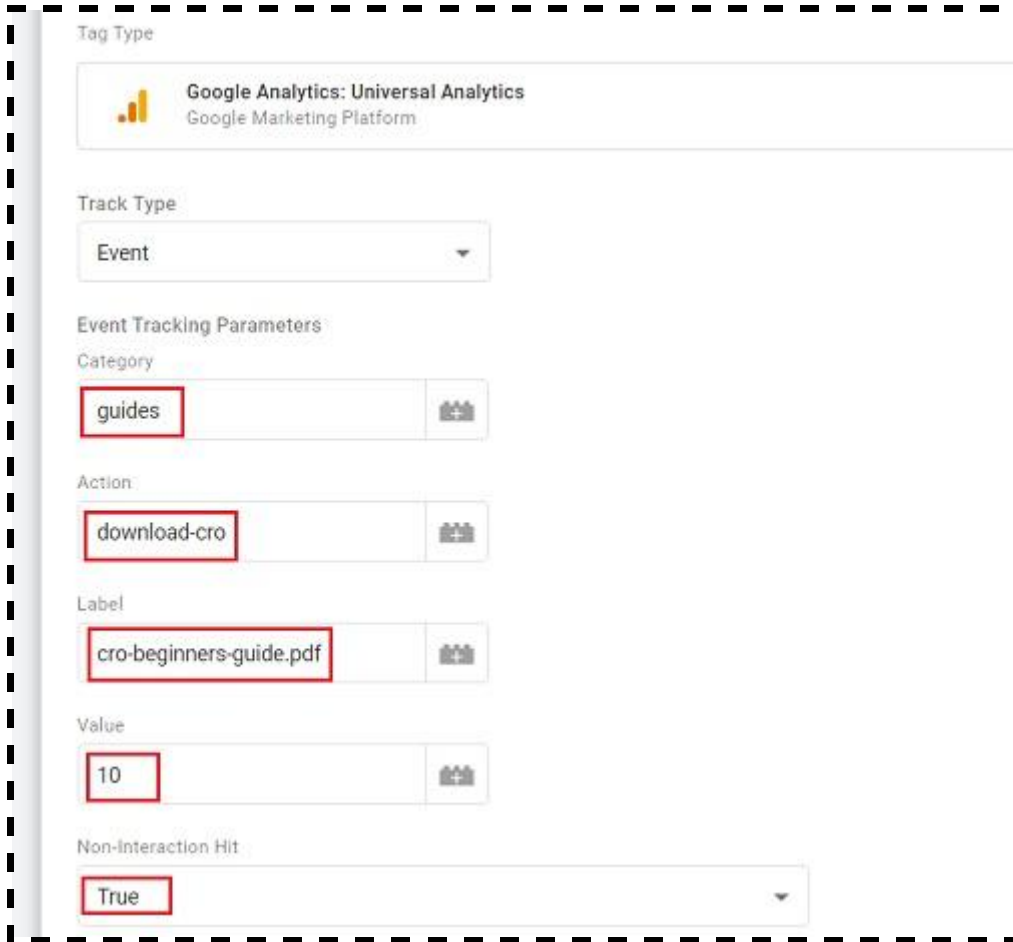
Step-18: Select 'Google Analytics: Universal Analytics' as a tag type:
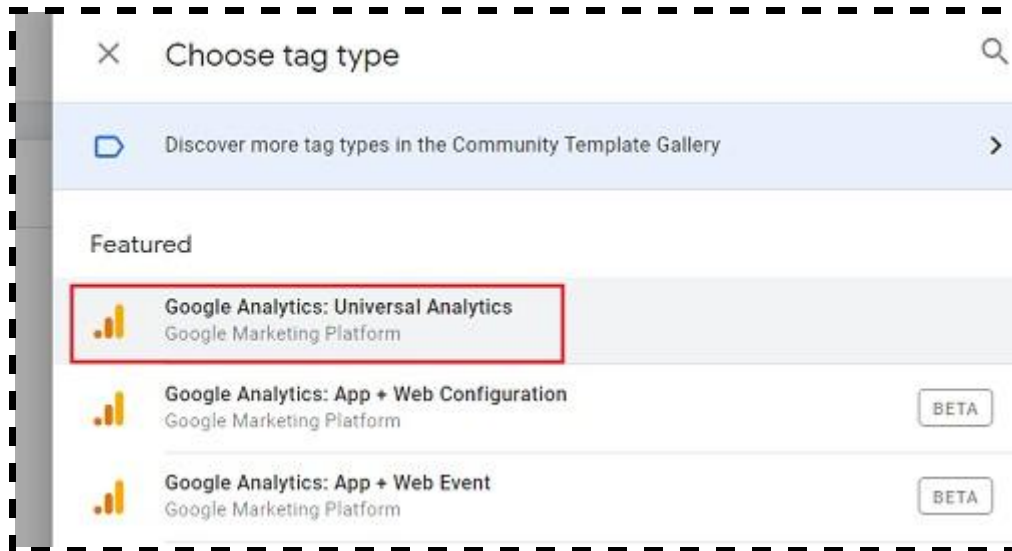
Step-19: Select 'Event' from the 'Track Type' drop-down menu:
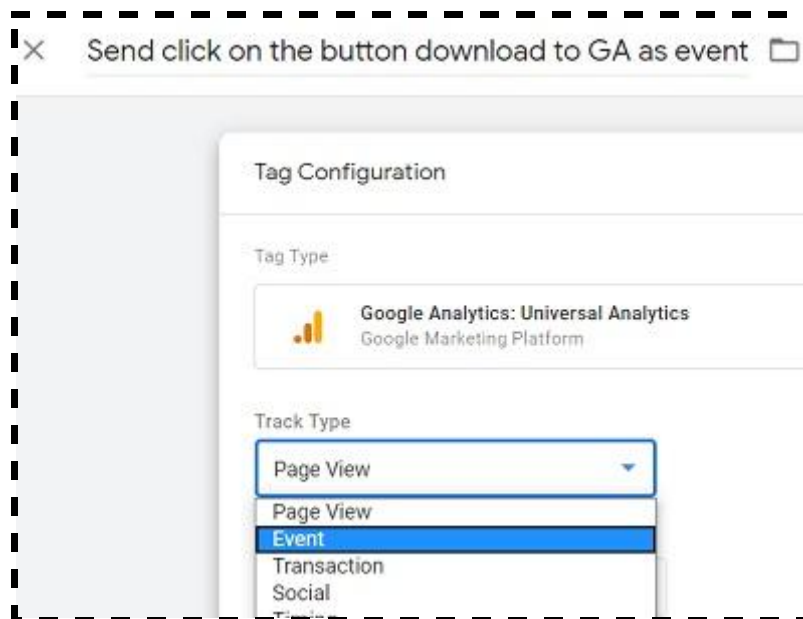
Step-20: Set 'Category' to 'guides', 'Action' to 'download-cro', 'Label to 'cro-beginners-guide.pdf', 'Value' to 10 and 'Non-Interaction Hit' to 'True':
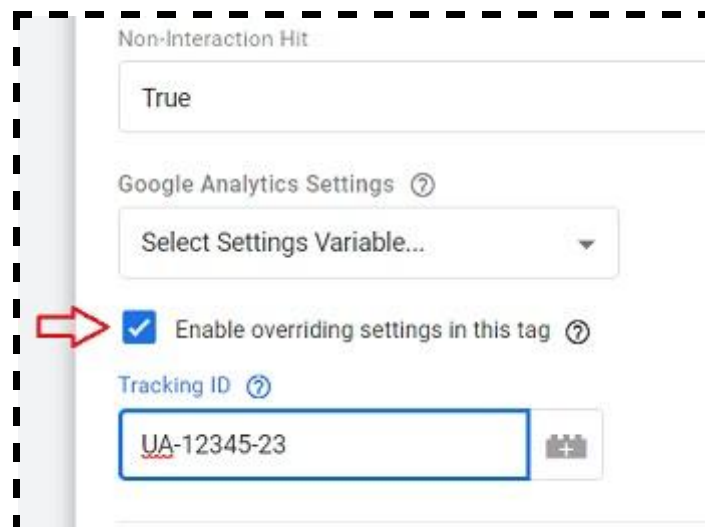
Step-21: Click on the checkbox '***Enable overriding settings in this tag***' and then enter your **GA tracking ID**:

Step-22: Click on the 'Choose a trigger to make this tag fire...':
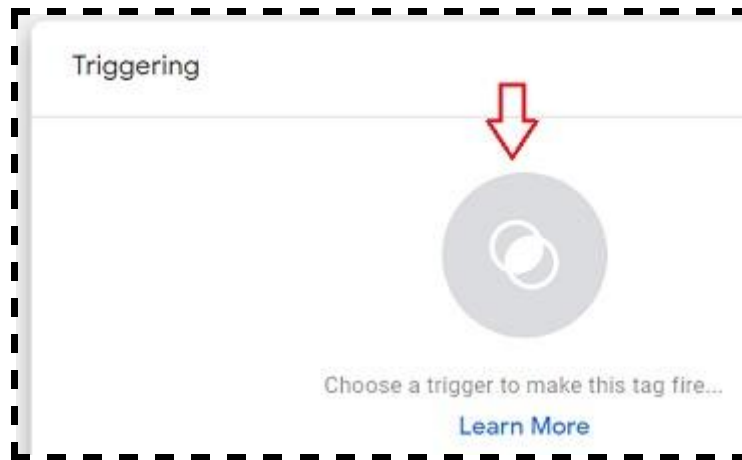
Step-23: Find and click on the trigger 'Check for clicks on the download button':
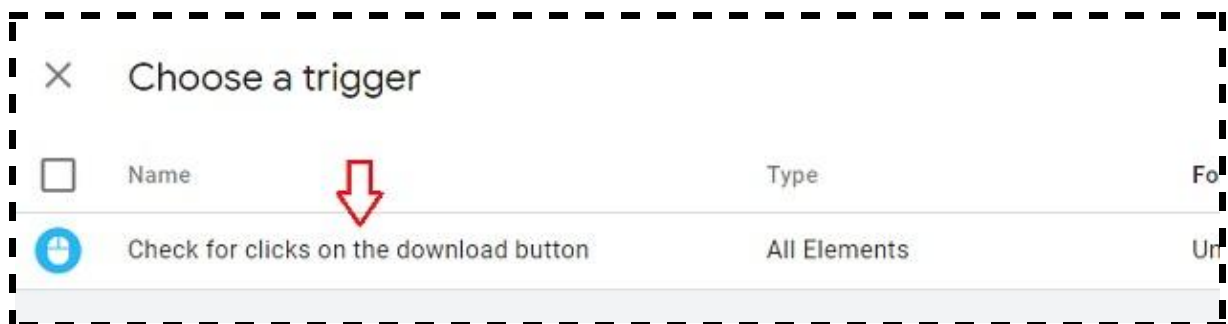
Step-24: Click on the 'Save' button to save your new tag.
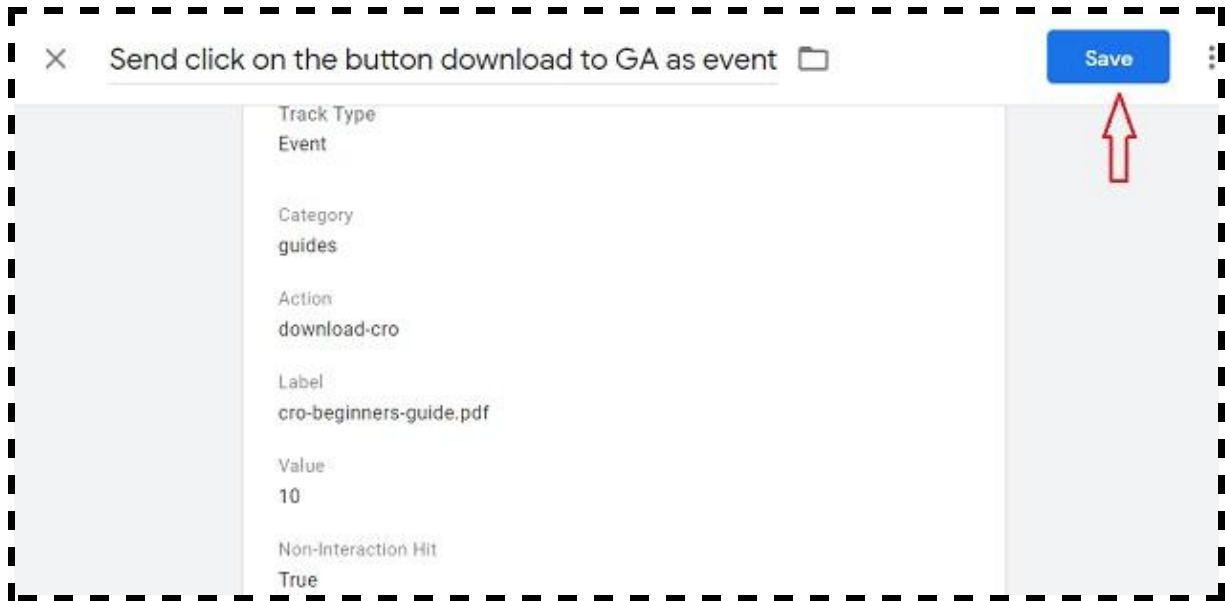
Step-25: Click on the 'Preview' button on the top right-hand side:
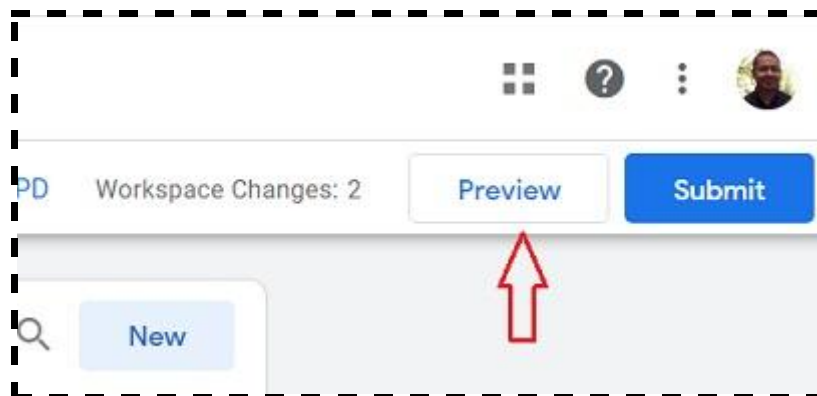
You should now see a message like the one below:
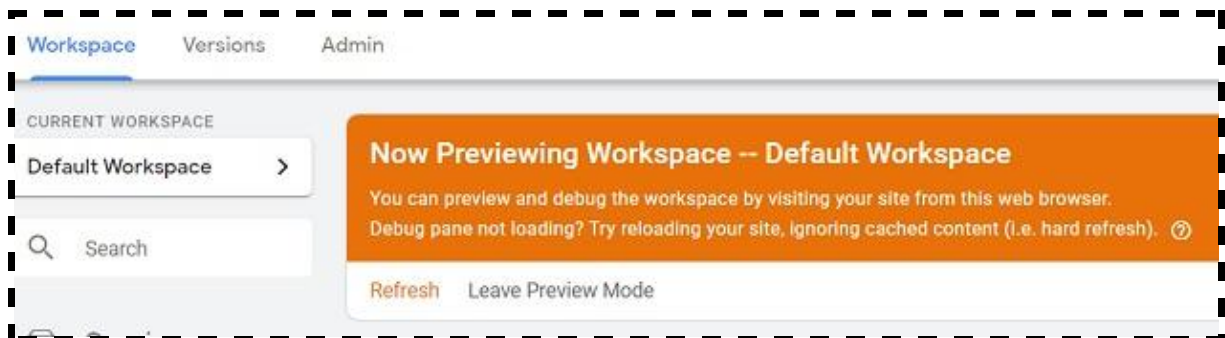
Step-26: Refresh the web page that contains the button you want to track. You should now be able to see the GTM debug console window at the bottom of the page:
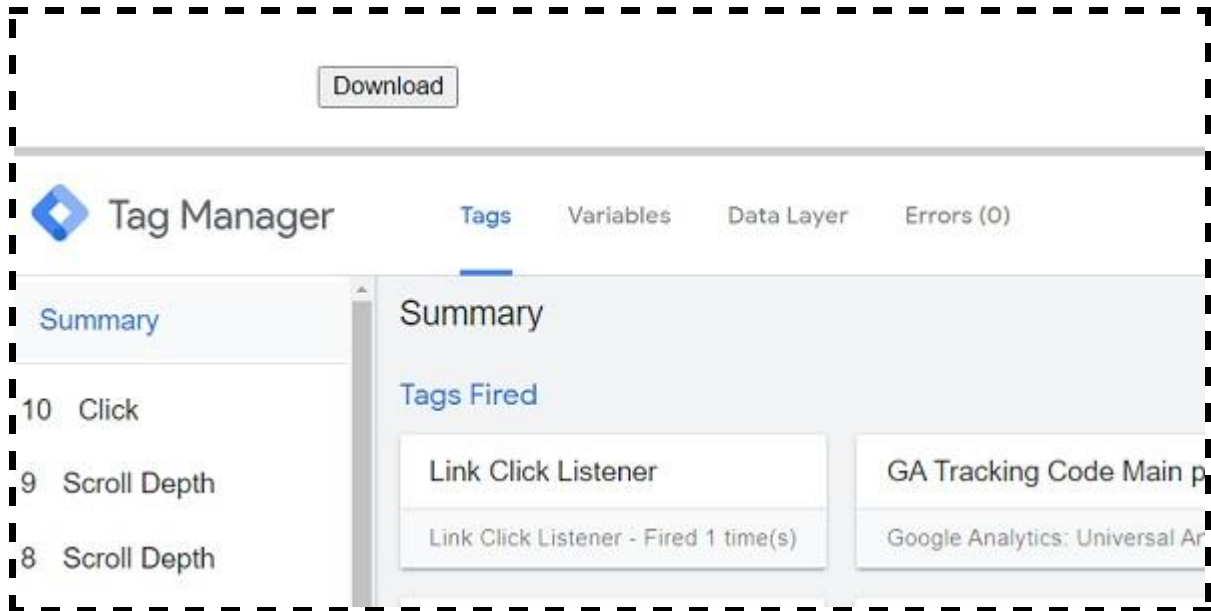
Step-27: Click on the button. Now if you see your tag listed under the 'Tag Fired' section then it means you tag is firing:
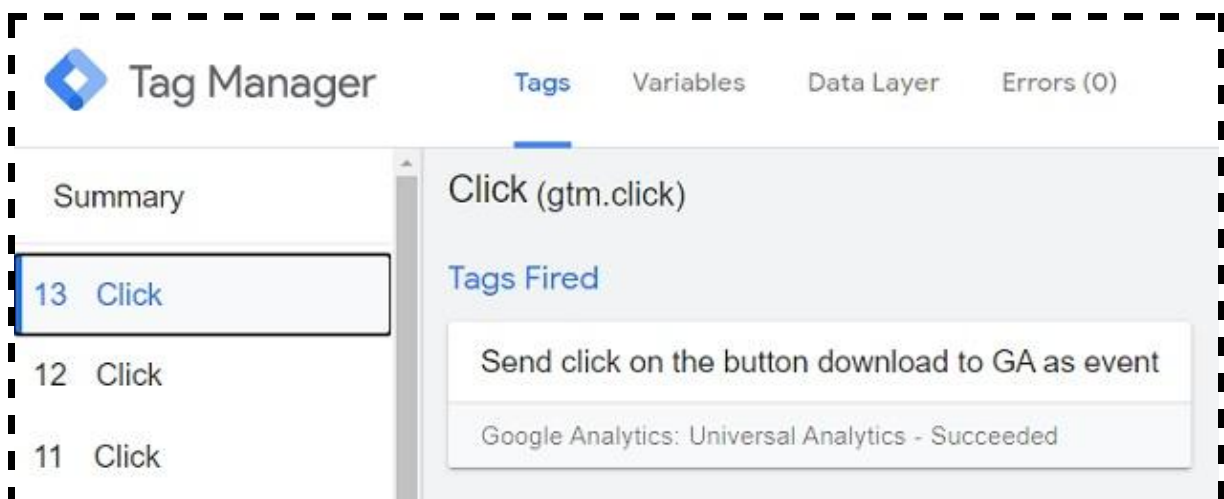
If you don't see your tag listed under the 'Tag Fired' section then it means you tag is not firing and you did something wrong while setting up your tag or trigger.

Step-28: If the tag is firing then click on the 'Leave Preview Mode' link in your GTM account and then click on the 'ok' button:
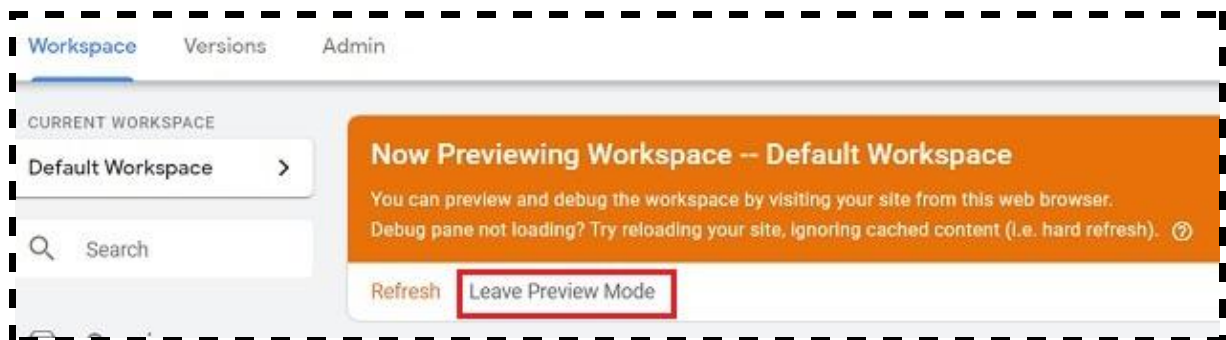
Step-29: Click on the 'Submit' button on the top right-hand side:
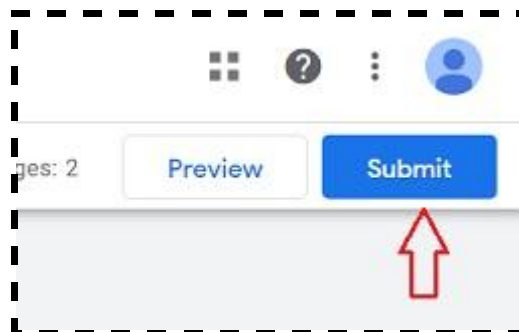
Step-30: Enter 'Added a new tag to track clicks on 'download' buttons as version name and then click on the 'Publish' button:
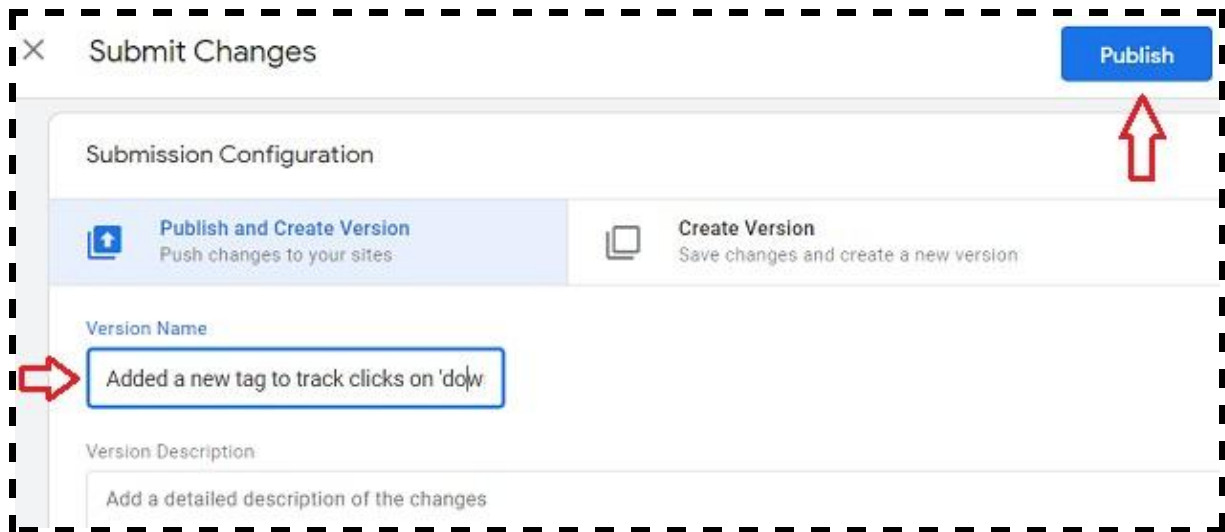
How to make sure that event tracking is working

There are two ways in which you can make sure that event tracking is working as intended:

#1 By looking at the real-time events' report.

Trigger the event you are tracking (via GA event tracking code) and then look at the real-time events report to check whether the event data (category, action, label) is being passed to Google Analytics.

#2 Through Google Developer Console

The second way to see the real-time firing of events is through '**Google developer console**'.

Trigger the event you are tracking (via GA event tracking code) and then look at the Google developer console to check whether the event data (category, action, label, value) is being passed to Google Analytics:
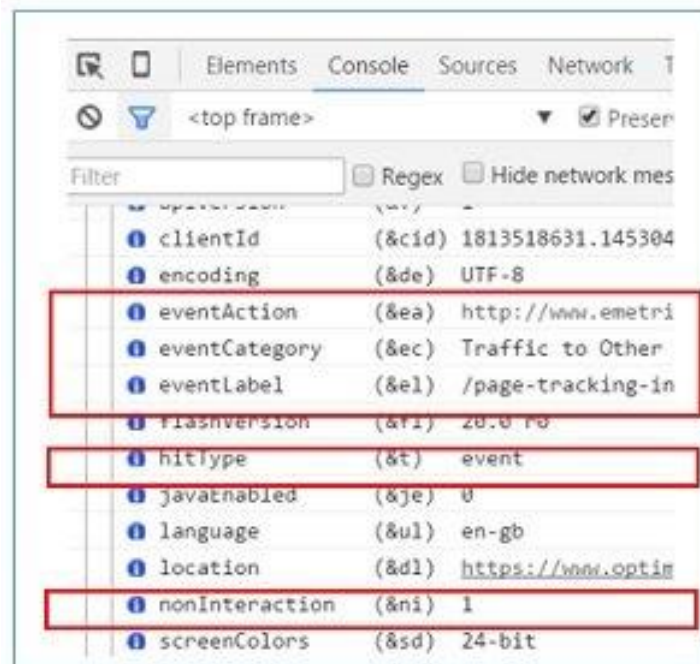
I prefer to use this method, as the developer console provides much more detailed information about fired events than the real-time event's report. The Google developer console also provides error messages in case the event does not fire.

The real-time events report in GA does not provide any such information. So if your event is not firing and you just rely on real-time events report, then you will have a hard time diagnosing event tracking issues.

If you do not supply event category value while setting up event tracking (**in case of analytics.js**), then the tracked event will not be counted in the 'Total events' metric and will not be reported in GA event reports.

The Google developer console will also show you the error message "**Missing required field 'eventCategory' for hit of type 'event':**

Just like the event category, the event action is also a required field.

But unlike event category, if you do not supply the even action value while setting up event tracking (**in case of analytics.js**), then the tracked event (without event action) will still be counted in the 'Total events' metric.

Google developer console will show you the error message "**Missing required field 'eventAction' for hit of type 'event':**

Understanding Events reports in Google Analytics

The events reports are available under the '**Behavior**' menu in Google Analytics:
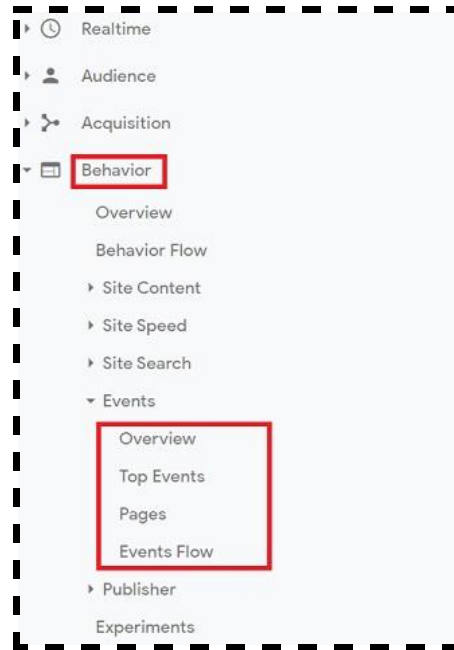
There are 4 event tracking reports available in Google Analytics:

1. Overview report

2. Top Events report

3. Pages report

4. Events Flow report

Events Overview report

This report provides an overview of tracked events on your website:

Total events - It is the total number of users' interactions with tracked webpage elements. For example, if a visitor clicks on the 'play' button of the same video 5 times, then GA reports total events as 5

Unique events - It is the total number of unique users' interactions with tracked webpage elements in a web session. For example, if a user clicks on the 'play' button of the same video 5 times, then GA reports total events as 5 but unique events (number of unique interactions that occurred in a web session) as 1.

Event Value - It is the total value of an event or set of events.
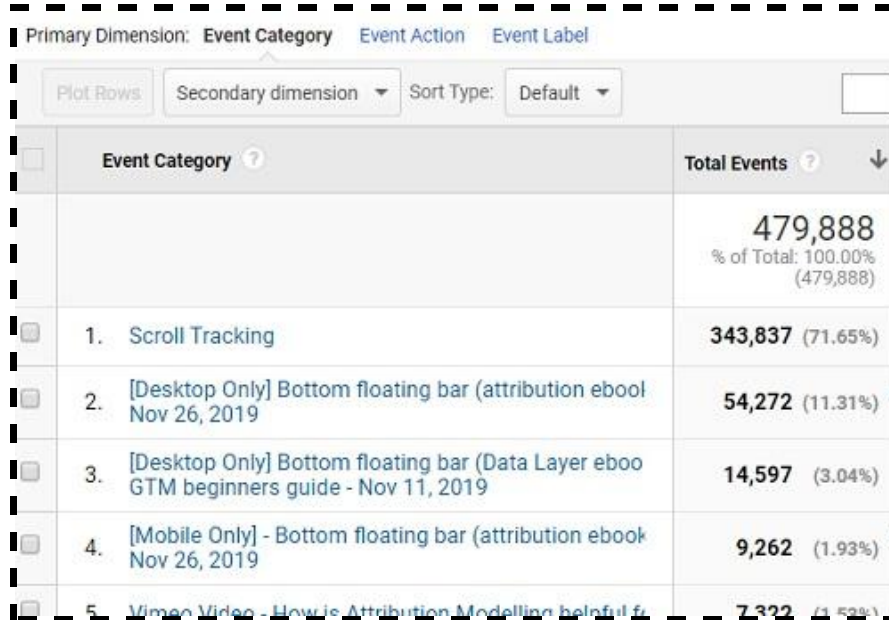
Avg. Value - It is the average value of an event or set of events.

Sessions with an event - These are those Google Analytics sessions in which at least one tracked event occurred.

Events/Session with an event - It is the average number of tracked events that occurred in a Google Analytics session.

Top Events report

Through this report, you can determine the most popular event categories, event actions and even labels:

The popularity is measured in terms of total events.

Event Pages report

Through this report, you can determine the pages of your website on which the maximum number of events were recorded by GA:

These are the pages that provide maximum user engagement.

Events Flow report

Through this report, you can visualize the order in which visitors triggered events on your website.
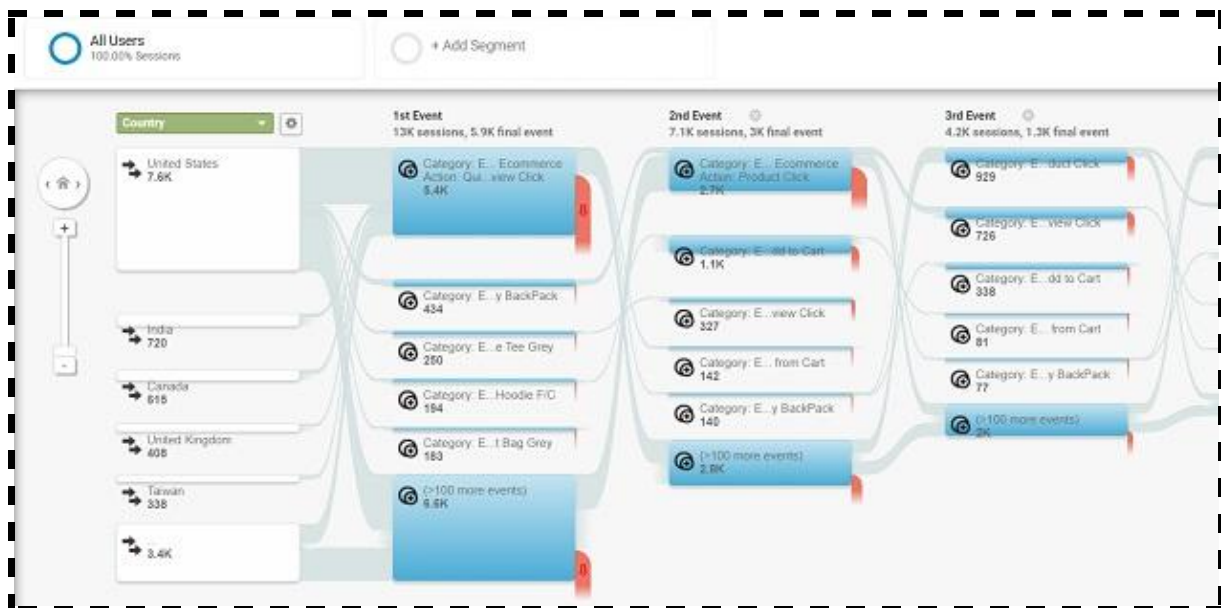
In order to interpret the events flow report accurately, it is critical that you use clear and consistent naming conventions for event categories, actions, and labels.

Configuring the Events Flow report includes:

#1 Examining Nodes and Connections

#2 Applying Default or custom Segments

#3 Comparing data of different time range

#4 Selecting dimensions

#5 Exporting the events funnel report etc.

You can learn more about this configuration here.

Event dimensions and metrics

You also need to remember, that 'event category', 'event action' and 'event label' are reported as **dimensions** in GA reports. Whereas 'event value' is reported as a **metric** in GA reports:

Note: 'event category', 'event action' and 'event label' are 'hit' level dimensions. In other words, you can't use 'event category', 'event action' and 'event label' as metrics in GA reports.

To understand the difference between dimensions and metrics, read the article:

Complete Guide to Dimensions and Metrics in Google Analytics

Setting up events as goal conversions in Google Analytics

You can also set events as Goals in Google Analytics and assign a monetary value to it. Follow the steps below:

Step-1: Login to your Google Analytics account and then navigate to the main reporting view.

Step-2: Navigate to the admin section and then click on 'Goals' under the 'View' column:
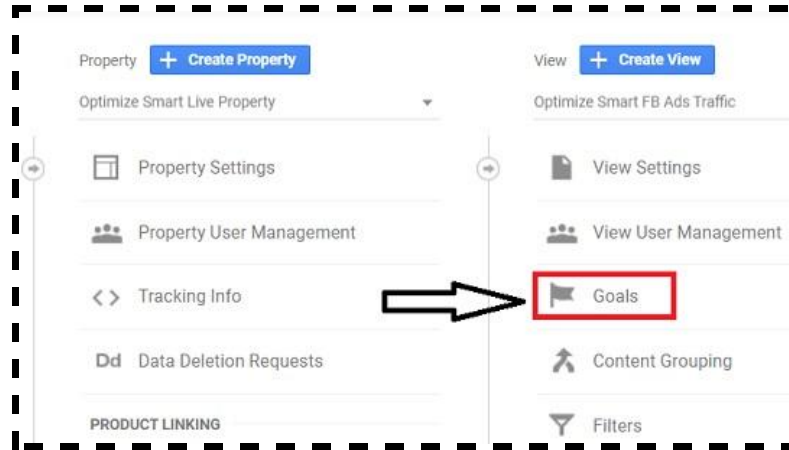
Step-3: Click on '+New Goal' button:
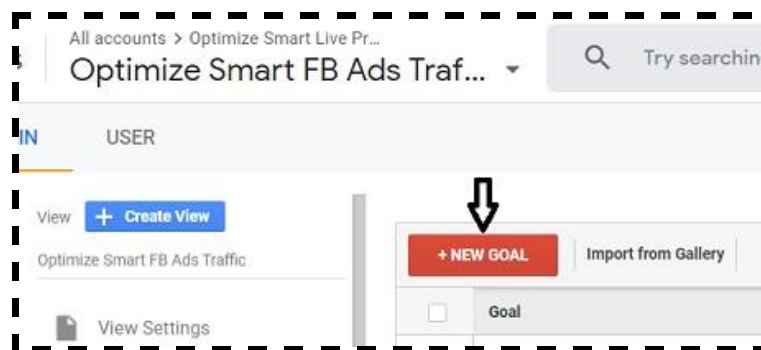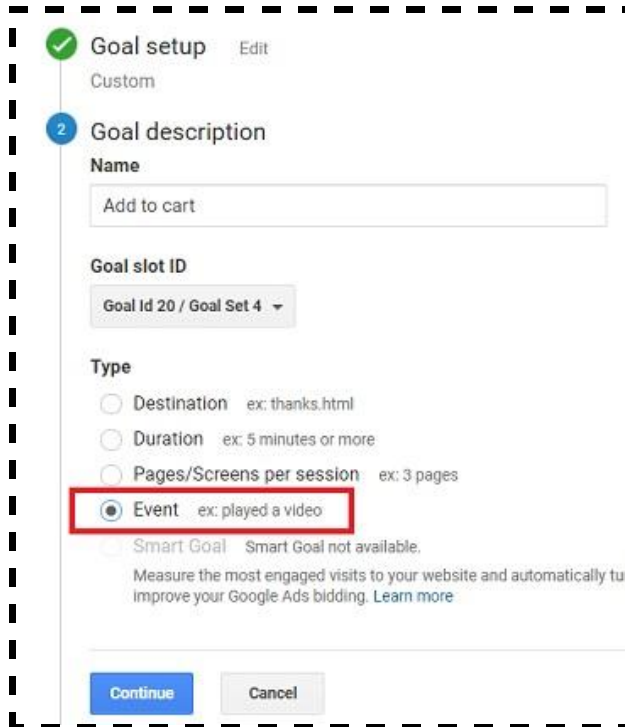
Step-4: Scroll down and then select the 'custom' option (under 'Goal Setup):
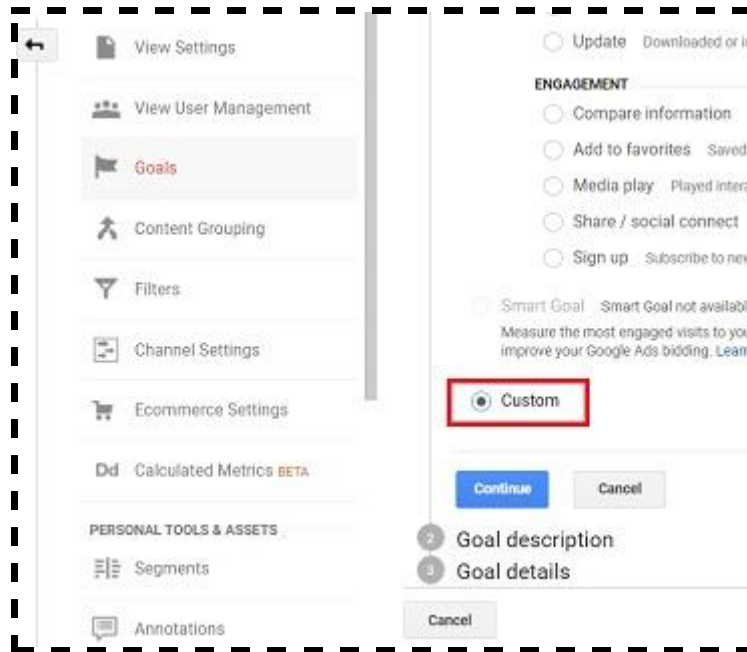
Step-5: Click on the 'continue' button and then give your goal a name (like 'Add to Cart'):
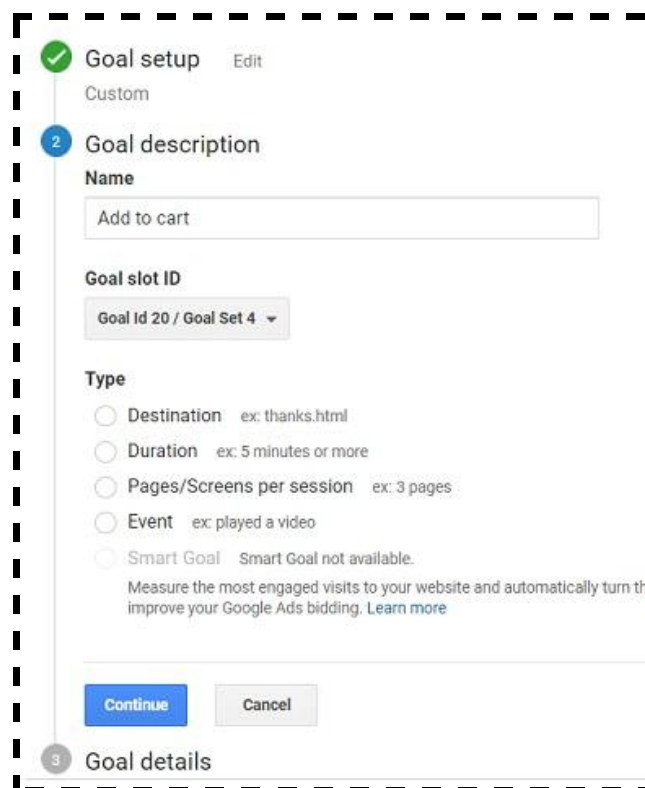
Step-6: Select goal type to 'Event' and then click on the 'Continue' button:
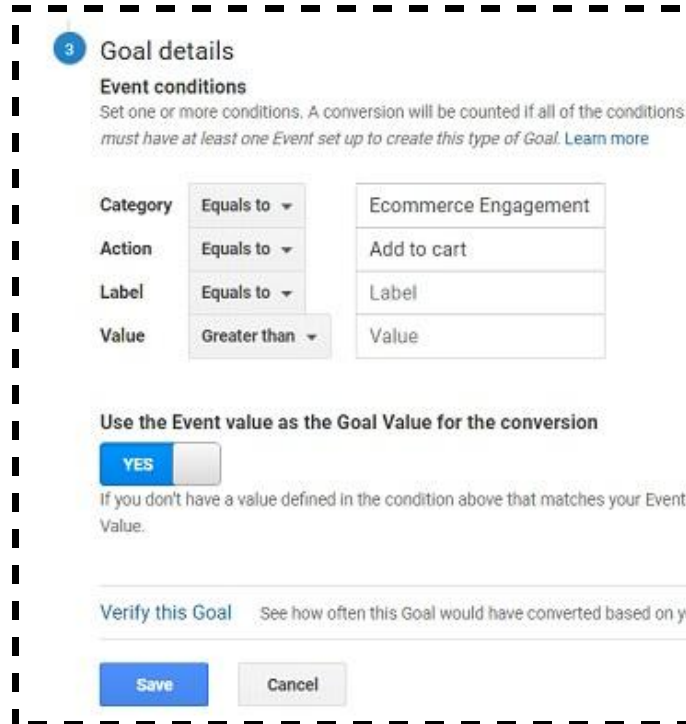
Step-7: Enter the values for 'event category', 'event action', 'event label' (optional) and 'event value' (optional):

Step-8: Click on the 'Save' button.

Types of events in Google Analytics

A user can interact with a web page element via a mouse, keyboard, frame or form. Consequently, we can have:

- **Mouse events** - These are users' interactions with a mouse.
- **Keyboard events** - These are users' interactions with a keyboard.
- **Frame events** - These are users' interactions with a frame/iframe.
- **Form events** - These are users' interactions with a form embedded on a web page.

All of these events can or can not generate pageviews when they occur.

Setting up event tracking in Google Analytics (analytics.js)

In order to set up event tracking in GA, you need to send an event hit whenever the users' interaction/activity you want to track, occurs.

Following is the syntax for sending an event hit in Universal Analytics (analytics.js)

```
ga('send', {

'hitType': 'event',

'eventCategory': [category],

'eventAction': [Action],

'eventLabel': [Label],

'eventValue': [Value],

'nonInteraction': true

});
```

Here, eventCategory, eventAction, eventLabel and eventValue are the **fields** used for sending event hit and 'nonInteraction' is a **field object**.

*If you are not sure of the difference between a field and field object in ga() command queue function, then read this article: **Introduction to Google Analytics Commands***

Note: Specifying 'event category' and 'event action' is mandatory. Whereas, specifying 'event label', 'event value' and 'nonInteraction' is optional.

Following is an example of an event hit in universal analytics (analytics.js):

```
ga('send', {

'hitType': 'event',

'eventCategory': 'videos',

'eventAction': 'Play',

'eventLabel': 'chappy',

'eventValue': 100,

'nonInteraction': true

});
```

You can also write this line of code without using any field names.

For example:

```
ga('send','event','videos','Play','chappy',100,{'nonInteraction': true});
```

Here, Google Analytics will automatically assume that:

- event category is 'videos',

- event action is 'Play'

- event label is 'Chappy'

- event value is 100

When you do not specify field names, you then need to supply field values in a particular order.

In case of event tracking, you need to specify field values in the following order:

Event Category > Event Action > Event Label > Event Value

Since field values are executed in the order in which they are supplied, if I changed the order, my event tracking will work differently or worse won't work at all.

For example:

```
ga('send','event','Play','videos','chappy',100);
```

Here 'Play' will be treated as an event category and 'videos' will be treated as an event action.

Another example:

```
ga('send','event','Play','videos',100);
```

here 100 will be treated as event label instead of event value. To avoid this issue, you can leave the 'event label' field empty:

```
ga('send','event','Play','videos','',100);
```

Another example,

```
ga('send','event','Play','videos','',100ex);
```

here event value is of type text. Since event value accepts a value of type integer, the code above will create tracking issues.

You can bypass all of these order rules by using field names:

```
ga('send', {

'hitType': 'event',

'eventCategory': 'videos',

'eventAction': 'Play',

'eventValue': 100

});
```

I prefer using field names because it makes the code easy to understand. Remember:

1. The **eventCategory**, **eventAction** and **eventLabel** fields accept a value of type text.
2. The **eventValue** field accepts a value of type integer.
3. The default values of **eventCategory**, **eventAction**, **eventLabel** and **eventValue** fields is '(not set)'.

Setting up event tracking in Google Analytics (gtag.js)

Following is an example of an event hit in Google analytics (gtag.js):

```
gtag('event', <action>, {

'event_category': <category>,

'event_label': <label>,

'value': <value>

});
```

Note: If you are using the global site tag (gtag.js) then specifying the 'event category', 'event label' and 'event value' is optional. However, specifying 'event action' is mandatory. The default value of the 'event category' is 'engagement' and the default value of the 'event label' is (not set).

The following example code fires a Google Analytics Event with an action of 'Play", a category of 'videos', and a label of 'Chappy':

```
gtag('event', 'Play', {

'event_category' : 'videos',

'event_label' : 'Chappy'

});
```

Default Google Analytics Events (gtag.js)

The default Google Analytics Events are the events which are pre-set with default categories and labels. Google recommends using the default events in order to maintain consistent reporting and interoperability with future functionality.

The following table lists the default Google Analytics Events, their default categories, and default label types (if available):

Event name	Default category	Default label type
add_payment_info	ecommerce	
add_to_cart	ecommerce	
add_to_wishlist	ecommerce	
begin_checkout	ecommerce	
checkout_progress	ecommerce	
generate_lead	engagement	
login	engagement	method
purchase	ecommerce	
refund	ecommerce	

remove_from_cart	ecommerce	
search	engagement	search_term
select_content	engagement	content_type
set_checkout_option	ecommerce	
share	engagement	method
sign_up	engagement	method
view_item	engagement	
view_item_list	engagement	
view_promotion	engagement	
view_search_results	engagement	search_term

For event names not listed in this table (e.g. arbitrary event names that you create), the default category is 'engagement' and the default label is 'not set'.

For example, the following code fires a Google Analytics Event with an action of 'Play', a default category of 'engagement, and a default label of '(not set)':

    gtag('event', 'Play');

Tying events to web page elements via event handlers

In order to track users' interactions with web page elements, you need to tie events to the page elements. You can do that by using, what is called '**event handler**' (or '**event listener**'). Since a user can interact with a web page element

via mouse, keyboard, frame or form, **we can have four categories of event handlers:**

1. Mouse event handlers
2. Keyboard event handlers
3. Frame event handlers
4. Form event handlers

Mouse event handlers

The most common mouse event handler is **onClick**. It executes a method/function in response to click on a web page element like a button or link. This event handler is often used to tie the event tracking code to a button/link.

For example:

> <button onClick="**ga('send', 'event', 'guides', 'download-cro', 'cro-beginners-guide.pdf', 10);**">Download CRO Beginners Guide</button>

Here, we are tracking the number of downloads of '*CRO Beginners Guide*' **by tracking clicks on a button.**

Another example,

> Download Social Media Guide

Here, we are tracking number of downloads of social media guide **by tracking clicks on a link.**

Another example,

> <a href="mailto:himanshu@google.com " onClick="**ga('send', 'event', 'Email', 'click to email',**
> **'himanshu@google.com');**">himanshu@google.com

Here, we are tracking the number of people who sent emails by **tracking clicks on emails**.

The other but less popular mouse event handlers are *ondblclick,onmouseover* and *onmouseout*. All of these mouse event handlers can be used to execute the event tracking code.

Note: You should avoid tracking highly repetitive events like mouse movements because Google Analytics can't track more than 500 events per session.

If you want to track clicks on external links, clicks on file downloads and other buttons in **Google Tag Manager**, then read the following articles:

> **#1 Tracking clicks on external links across a website**

> **#2 Tracking File downloads and other button clicks**

Keyboard event handlers

I have not found any good use of keyboards event handlers in event tracking.

Frame event handlers

The most common frame event handler is **onLoad.**

It executes a method after a page, frame or image has completely loaded. For example:

> <img src="popup.gif" onload="**ga('send', 'event', 'popups',**
> **'popup-homePage');**">

Form event handlers

The most common form event handler is **onSubmit**. This event handler executes a method/function when a form is submitted and is often used to tie event tracking code to a form embedded on a web page.

For example:

> <form name="processOrders" action="process.php"
> onsubmit="**ga('send', 'event', 'Forms', 'Order Submitted');**">

Other commonly used form event handlers are ***onBlur, onChange, onReset,*** and ***onSelect***. All of these event handlers are frequently used for **advanced form tracking**.

If you want to track forms via Google Tag Manager then read the following articles:

#1 Tracking Form Submissions via Google Tag Manager

#2 Tracking Form Fields via Google Tag Manager

Scale and automate event tracking via Google Tag Manager

The purpose of this ebook is to teach you the fundamentals of event tracking so that you can use this information to deploy the event tracking code via Google Tag Manager.

I certainly don't want you, to manually add event tracking code on each and every web page element, throughout your website. That would be insane, esp. when now you have got the mighty Google Tag Manager at your disposal.

To learn more about event tracking in Google Tag Manager, read the following article: **Event Tracking in Google Tag Manager – Complete Guide**

Implement video tracking via Google Tag Manager

Video tracking is one of the advanced applications of event tracking, where you track the player state (play, pause and the percentage of video played) of single/multiple videos embedded on a web page or across a website of any

size. However, if you want to scale and automate video tracking, then you must deploy this tracking via GTM.

Here is an article which can help you do that: **Video Tracking in Google Tag Manager**

Implement scroll tracking via Google Tag Manager

Scroll tracking is one of the advanced applications of event tracking, where you can track the length of a page scrolled by users.

Through scroll tracking, you can get an insight into, how people are consuming your website contents. Again, GTM is the best tool to scale and automate scroll tracking on a website.

To learn more about implementing scroll tracking via GTM, read the following article: **Scroll tracking in Google Tag Manager**

Enhanced ecommerce tracking

Enhanced ecommerce tracking is a very advanced application of event tracking. Though it is strictly, not an event tracking application, it heavily uses the logic behind event tracking, to track various types of **ecommerce events**.

These ecommerce events could be:

1. Viewing an internal promotion campaign.

2. Clicking on an internal promotion campaign.

3. Viewing a product in a product list.

4. Clicking one of the product links in the product list.

5. Viewing a product detail page.

6. Adding/removing products from the shopping cart.

7. Starting, completing and/or abandoning the checkout process.

8. Asking for a refund.

In the case of enhanced ecommerce, various ecommerce events are usually tracked via **enhanced ecommerce tracking code** and not via the traditional event tracking code. However, in some cases, traditional event tracking code can also be used.

Cross-device tracking

Cross-device tracking is the most advanced application of event tracking, where you send event hits from devices other than laptops, mobiles, desktops, and tablets via **measurement protocol**.

So if you want to send event data from your cash register or washing machine to the Google Analytics server, you can do that by writing a script that sends the **payload data** to the GA server.

To learn more about, how you can send event data via measurement protocol, read this article: **Understanding Google Analytics Measurement Protocol**

You are doing Google Analytics all wrong. Here is why...

I have dealt with hundreds of Google Analytics accounts in my career.

I have seen many issues, from incorrect tracking code, selecting the wrong KPIs, to analyzing data without using custom reports or advanced segments.

But do you know the biggest issue of all in Google Analytics?....

It is the "misinterpretation of analytics data".

Many marketers make the mistake of crediting conversions to the wrong marketing channel.

And they seem to be making this mistake over and over again.

They give the credit for conversions to the last touchpoint (campaign, ad, search term...).

They can't help themselves because they believe that the Google Analytics reports are 'what you see is what you get'.

But they are actually 'what you interpret is what you get'.

This has resulted in marketers making wrong business decisions and losing money.

All the data you see in Google Analytics reports today lies to you unless you know exactly how to interpret it correctly.

For example, let's talk about direct traffic.

All untagged or incorrectly tagged marketing campaigns, from display ads to emails, could be reported as direct traffic by Google.

Whenever a referrer is not passed, the traffic is reported as direct traffic by Google.

Mobile applications don't send a referrer. Word/PDF documents don't send a referrer.

'302 redirects' sometimes cause the referrer to be dropped. Sometimes browsers don't pass the referrer.

During an HTTP to HTTPS redirect (or vice versa), the referrer is not passed because of security reasons.

All such traffic is reported as direct traffic by Google.

So on the surface, it may look like most people are visiting your website directly, but this is not usually the case.

But this analysis does not end here because you are still not looking at the complete picture.

People do not always access your website directly and then make a purchase straight away.

They are generally exposed to multiple marketing channels (display ads, social media, paid search, organic search, referral websites, email, etc.) before they access your website directly.

Before they make a purchase.

So if you are unaware of the role played by prior marketing channels, you will credit conversions to the wrong marketing channels.

Like in the present case, to direct traffic.

To get this type of understanding, you need to understand and implement web analytics.

But **you learn data analysis and data interpretation from web analytics and not from Google Analytics.**

The direction in which your analysis will move will determine the direction in which your marketing campaigns will move.

You get that direction from 'web analytics' and not from 'Google Analytics'.

Web/digital analytics is not about Google Analytics (GA) or Google Tag Manager (GTM). It is about analyzing and interpreting data, setting up goals, strategies and KPIs.

It's about creating a strategic roadmap for your business.

That's why the knowledge of web/digital analytics is so important.

So what I have done is put together an ebook for you.

This ebook will teach you what digital analytics really is and how I have been able to leverage it to generate floods of news sales and customers.

You can literally copy what I have done to get similar results.

You can download this ebook from here:

https://learn.optimizesmart.com/sales-roi-accelerator-ebook

I hope you find it helpful.

All the best,

Himanshu