



TECHNICAL GUIDE

Track YouTube Video Link Clicks in GA4

A practical guide to tracking which campaigns, channels, and links drive clicks to your YouTube videos.

For marketers, analytics teams, and developers using GA4 and Google Tag Manager.

Prepared by:
TrackFunnels

Website:
<https://trackfunnels.com>

© 2026 TrackFunnels. All rights reserved.



Table of contents

What this guide solves.....	3
How the set up works.....	3
What this set up needs.....	4
Who this guide is for.....	4
Let's Build the Tracking Setup.....	5
Step 1: Measurement Planning.....	6
Option 1: Report clicks as virtual pageviews.....	6
Option 2: Report clicks as GA4 events.....	7
What data should you capture?.....	7
Step 2: Choose your tracking method.....	8
Option 1: Virtual pageviews with UTM parameters.....	8
Option 2: GA4 events with query parameters.....	8
Implementation.....	9
Step 1: Create the intermediate page.....	9
Step 2: Create the GTM variables.....	11
Step 3: Test the variables.....	14
GTM Tags.....	15
Step 4: Create the redirect tag.....	15
Step 5: Create the GA4 tracking tag.....	17
Step 6: Create new trackable links.....	25
Limitations of this setup.....	25



What this guide solves

When you share a **direct YouTube link** in a newsletter, LinkedIn post, paid campaign, QR code, partner campaign, or sales email, YouTube analytics shows you views. But it does not clearly tell you which specific link drove those views.

Did the click come from LinkedIn or other platforms where you shared it?

These sources matter because you want to know which channels are actually helping your videos get watched.

Link shorteners can help you track clicks, and they are a good solution if all you need is basic click data.

But if you want this data inside GA4, alongside your website traffic and campaign data, you need a different setup.

This guide shows how to **track direct YouTube video link** clicks in GA4 using Google Tag Manager and a lightweight redirect page on your own website.

How the set up works

Instead of sharing the direct YouTube URL, you share a trackable URL from your own website.

For example, instead of sharing:

```
HTML
https://www.youtube.com/watch?v=QdJ9Q5WCEV4
```

You share:

```
HTML
https://yourwebsite.com/go-to-video/?video_id=1&utm_source=linkedin&utm_medium=organic&utm_campaign=direct-links
```



When someone clicks this link:

1. The user lands on your intermediate page.
2. GTM reads the query parameters.
3. GA4 receives the click data.
4. The user is redirected to the correct YouTube video.

For the user, the experience stays simple. They click a link and land on YouTube.

For you, GA4 receives the data before the redirect happens.

What this set up needs

You need four things:

1. A reusable intermediate page on your website
2. A video list that maps each video ID to a YouTube URL and video title
3. GTM variables and tags
4. Trackable URLs with query parameters

The intermediate page is created once and can be reused for all future YouTube videos.

Who this guide is for

This guide is useful if you:

- Share YouTube links across marketing channels
- Want to know which source, campaign, or link drove clicks to your videos
- Prefer keeping click data inside GA4 instead of a separate link shortener report

You will need:

- Access to your website or CMS
- Access to GA4 and Google Tag Manager
- Permission to publish GTM changes
- Basic understanding of URL parameters and GTM variables
- A list of YouTube videos you want to track



Let's Build the Tracking Setup

Understanding the sources contributing to your YouTube views remains a challenge. We don't have control over customizing YouTube analytics, but we do have control over our links. If you post a direct link to a YouTube video, like <https://youtube.com>, how would you know if it was clicked?

Besides using URL shortener services, use the method described here to track sources in your GA4 account.

We will build the setup in this order:

1. Decide what information should be captured.
2. Choose how the data should be sent to GA4.
3. Create a reusable intermediate page.
4. Create GTM variables to read the video ID, video title, and video URL.
5. Create the GA4 tracking tag.
6. Create the redirect tag.
7. Test the full flow before publishing.
8. Create new trackable links for future videos.

There are two ways to send the data to GA4.

The first method sends the click **as a virtual pageview using UTM parameters**. This is useful if you want the data to appear under standard GA4 traffic source dimensions.

The second method sends the click **as a GA4 event with event parameters**. This is usually cleaner if you want to keep YouTube link click data separate from normal website pageview data.

Most of the setup is the same for both methods. The main difference is the GA4 tracking tag you create in Google Tag Manager.

By the end of the guide, you will have a reusable setup that can work for all future YouTube videos. You will only need to keep your video list updated and create the right trackable links before sharing them.

Now that the concept is clear, let's walk through the setup in order: what data to capture, how to send it to GA4, how to create the intermediate page, and how to configure GTM so the user is redirected only after the click is tracked.



Step 1: Measurement Planning

Before setting up tags, variables, or redirects, start with the report you want to see in GA4.

This keeps the implementation focused.

For this use case, we mainly want to answer:

- Which video link was clicked?
- Where did the click come from?
- Which campaign or channel drove the click?
- How many clicks did each video receive?

There are two common ways to structure this in GA4.

Option 1: Report clicks as virtual pageviews

In this approach, each YouTube click is sent to GA4 as a pageview.

The YouTube video URL becomes the page URL, and the video title becomes the page title. Source information is passed through UTM parameters such as `utm_source`, `utm_medium`, and `utm_campaign`.

A GA4 exploration that covers all these data points would look like:

Session manual ad content	Session manual campaign	Session manual medium	Session manual source	Page Location	Views
generate-api-secret	organic-post	social	instagram	https://www.youtube.com/watch?v=EezGi1woEUw	23
hubspot-validator	organic-post	social	instagram	https://www.youtube.com/watch?v=1ZdQz77Lrm4	13
avoid-page-refresh-1	organic-post	social	instagram	https://www.youtube.com/watch?v=H82y9gV-7m0	14
generate-api-secret	organic-post	social	linkedin	https://www.youtube.com/watch?v=EezGi1woEUw	3
hubspot-validator	organic-post	social	linkedin	https://www.youtube.com/watch?v=1ZdQz77Lrm4	2
avoid-page-refresh-1	organic-post	social	linkedin	https://www.youtube.com/watch?v=H82y9gV-7m0	1

Do you notice how Page Location shows you the YouTube URL? That's a virtual PageView.

This method is useful if you want your click source data to appear under standard GA4 traffic source dimensions.



Option 2: Report clicks as GA4 events

In this approach, each YouTube click is sent as an event.

The event name could be:

`video_url_click`

Then you send details as event parameters, such as:

Event Name	Parameter 1 (source)	Parameter 2 (Video Title)	Parameter 3 (Video URL)	Parameter 4	Parameter 5	Event Count
video_url_clicks	instagram	Generate an API Secret in GA4 for Measurement Protocol	https://www.youtube.com/watch?v=EezGf1woEUw		use if you will	23
video_url_clicks	instagram	Hubspot Traffic Sources Report - Validate Your URLs	https://www.youtube.com/watch?v=IzdQz77Lrm4			13
video_url_clicks	instagram	Avoid Page Refresh Event Counts I - Variable Set up	https://www.youtube.com/watch?v=H82y9gV-7m0			14
video_url_clicks	linkedin	Generate an API Secret in GA4 for Measurement Protocol	https://www.youtube.com/watch?v=EezGf1woEUw			3
video_url_clicks	linkedin	Hubspot Traffic Sources Report - Validate Your URLs	https://www.youtube.com/watch?v=IzdQz77Lrm4			2
video_url_clicks	linkedin	Avoid Page Refresh Event Counts I - Variable Set up	https://www.youtube.com/watch?v=H82y9gV-7m0			1

The benefit of this option is that you can pass as many parameters you like into an event report.

What data should you capture?

Thanks to our mockup reports, the outcome is clear and visible. That gives our technical work a clear purpose. Before moving further with the setup, let's summarize what data we are capturing.

At minimum, we'll capture two things:

1. **Click source** - For example: LinkedIn, newsletter, Instagram, partner, sales email, paid campaign.
2. **Video information** - For example: Video title, and YouTube URL.

A simple video list can look like this:

Video ID	Video Title	YouTube URL
1	Generate an API Secret in GA4.	https://youtube.com/...
2	HubSpot Traffic Sources Report.	https://youtube.com/...
3	Avoid Page Refresh Event Count	https://youtube.com/...



The Video ID is your own assigned ID (internal) and not to be confused with the YouTube Video ID.

We use the Video ID later to connect the trackable link with the correct video title and YouTube URL.

Step 2: Choose your tracking method

There are two good ways to send this data to GA4.

Option 1: Virtual pageviews with UTM parameters

In this method, the click is sent to GA4 as a pageview.

The source data is passed through UTM parameters, and the video data can be passed as the virtual page URL and page title. Example URL:

HTML

```
https://yourwebsite.com/go-to-video/?video_id=1&utm_source=linkedin&utm_medium=organic&utm_campaign=direct-links
```

Use this method if you want the source data to appear under standard GA4 traffic source dimensions.

The trade-off is that video click data may mix with normal website pageview and traffic source reporting.

Option 2: GA4 events with query parameters

In this method, the click is sent as a GA4 event.

Example URL:

HTML

```
https://yourwebsite.com/go-to-video/?video_id=1&vid_source=linkedin
```



You can send the video title, YouTube URL, source, campaign, or any other useful information as event parameters.

Use this method if you want cleaner separation between website pageviews and video link clicks.

Implementation

The implementation has two main parts:

1. Create the intermediate page
2. Set up GTM variables and tags

Step 1: Create the intermediate page

Create a simple page on your website.

Example:

```
HTML
https://yourwebsite.com/go-to-video/
```

This page should contain your normal GTM container code.

Add a message container to the page (optional - decide if you want to show a message like give below):

```
HTML
<div id="message-container"></div>
```

Then add this JavaScript to the page footer:



JavaScript

```
<script>

    function getQueryParameter(name) {
        const urlParams = new URLSearchParams(window.location.search);
        return urlParams.get(name);
    }

    document.addEventListener("DOMContentLoaded", function() {
        const videoId = getQueryParameter('video_id');
        const messageContainer = document.getElementById('message-container');

        if (videoId) {
            messageContainer.innerHTML = `
                <div>
                    <p class="mt-3"
style="font-weight:200;font-size:1.75rem;color:#6d59ff">Hang on while we take you
to the video :)</p>
                </div>
            `;
        } else {
            messageContainer.innerHTML = `
                <div>
                    <p
style="font-weight:200;font-size:1.75rem;color:#6d59ff">Uh oh! We could not
understand that request. <br>Go to our YouTube channel directly.</p>
                    <a href="https://www.youtube.com/@trackfunnels/videos"
class="btn btn-primary">Go to YouTube Channel</a>
                </div>
            `;
        }
    });
</script>
```

Replace the YouTube channel URL (**highlighted in red**) with your own fallback URL.

This script checks whether `video_id` exists in the URL. If it exists, the page displays a waiting message. If it does not exist, the page shows a fallback message and link.

The actual redirect happens through Google Tag Manager.

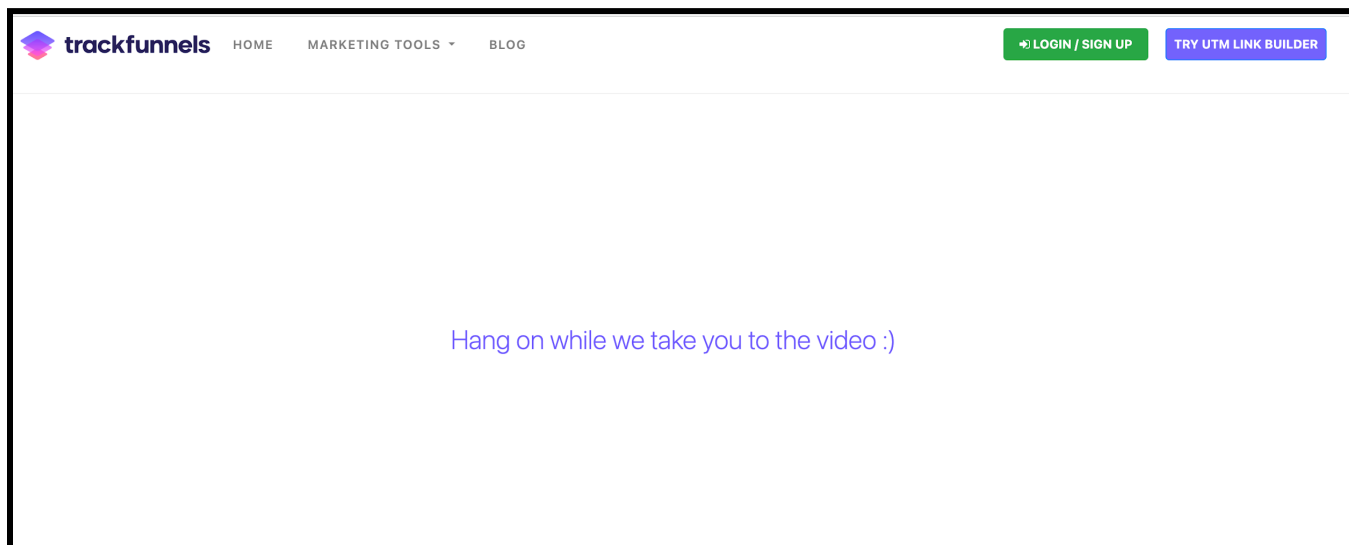


You can style the page however you want. The above Javascript uses some basic styling to the display messages.

Few things to remember

1. Make sure the page contains GTM tracking code just as its recommended to be present on every page
2. Swap the red highlighted URL in the JS code with your own YouTube channel, or if you do not have one - swap it with anywhere you'd like to take users in case they stumble upon this page by accident.

Here is how my intermediate page looks like when a video_id is present in the URL.



Step 2: Create the GTM variables

Create three variables in Google Tag Manager.

1. video_tracking_url_id

Variable type: URL
Component type: Query
Query key: video_id



× video_tracking_url_id

Save

Variable Configuration

Variable Type



URL



Component Type

Query

Query Key

video_id



> More Settings

> Format Value

2. video_click_tracking_title

Variable type: Lookup Table

Input variable: `{{video_tracking_url_id}}`

Use this variable to map each video ID to the correct video title.

Example:

× video_click_tracking_title

Save

Added in this workspace

Abandon changes

Variable Configuration

Variable Type



Lookup Table



Input Variable

`{{video_tracking_url_id}}`



Lookup Table

Input

1



2



3



Output

Generate an API Secret in GA4 for Measurement Prot



Hubspot Traffic Sources Report - Validate Your URLs



Avoid Page Refresh Event Counts I - Variable Set up



+ Add Row



Set Default Value



3. video_click_tracking_url

Variable type: Lookup Table

Input variable: `{{video_tracking_url_id}}`

Use this variable to map each video ID to the correct YouTube URL.

Example:

video_click_tracking_url

Save

Variable Configuration

Variable Type

Lookup Table

Input Variable

{{video_tracking_url_id}}

Lookup Table

Input	Output
1	https://www.youtube.com/watch?v=EezGi1woEUw
2	https://www.youtube.com/watch?v=IzdQz77Lrm4
3	https://www.youtube.com/watch?v=H82y9gV-7m0

+ Add Row

☒ Set Default Value

Default Value

https://www.youtube.com/@trac

Set up default value wherever it makes sense. We set up a default to our YouTube channel URL

> Format Value

Set a default value, such as your YouTube channel URL, so users are not stranded if the lookup fails.

Step 3: Test the variables

Open GTM Preview Mode and test a URL like this:

HTML

`https://yourwebsite.com/go-to-video/?video_id=1`

Change red highlighted part with the ID you used in Google Tag Manager variable Lookup tables.



Check that all three variables return the correct values:

- `video_tracking_url_id`
- `video_click_tracking_title`
- `video_click_tracking_url`

Only continue once the values are correct.

Connected
trackfunnels.com

2 Google tags found

GTM-MC8DMZQ G-7QRCWDQH1J

Summary

Video Redirection Demo

5 Window Loaded

4 DOM Ready

3 Container Loaded

2 Initialization

1 Consent Initialization

Form Classes	Data Layer Variable	undefined	undefined
Form Element	Data Layer Variable	undefined	undefined
Form ID	Data Layer Variable	undefined	undefined
Form ID Mailer Lite	Data Layer Variable	undefined	undefined
Form Name	Data Layer Variable	undefined	undefined
Form Target	Data Layer Variable	undefined	undefined
Form Text	Auto-Event Variable	undefined	undefined
Form URL	Data Layer Variable	undefined	undefined
GA4 Measurement ID	Constant	string	
Page Hostname	URL	string	"trackfunnels.com"
Page Path	URL	string	"/utm-links-builder/demos/go-to-video/"
Page URL	URL	string	"https://trackfunnels.com/utm-links-builder/demos/go-to-video/?video_" + "id=1>m_debug=1717169207672"
Referrer	HTTP Referrer	string	"https://taoassistant.google.com/"
video_click_tracking_title	Lookup Table	string	"Generate an API Secret in GA4 for Measurement Protocol"
video_click_tracking_url	Lookup Table	string	"https://www.youtube.com/watch?v=EezGi1woEUw"
video_tracking_url_id	URL	string	"1"

GTM Tags

We'll now set up 2 types of tags

1. **The Redirecting Tag:** This is a custom HTML tag that makes sure that users are redirecting to the correct video when they click on the link
2. **The Tracking Tag:** This is a GA4 tag that will send data to your GA4 account depending on what method you use.
 1. A Google tag (if you are using UTM Tracking)
 2. An event tag (if you are using Event tracking)



Step 4: Create the redirect tag

Tag type: Custom HTML

Add new tag > choose custom html > paste the below script

```
JavaScript
<script>
var videoUrl = {{video_click_tracking_url}};

if (videoUrl) {
  setTimeout(function() {
    window.location.href = videoUrl;
  }, 3000);
} else {
  setTimeout(function() {
    window.location.href = 'https://www.youtube.com/@trackfunnels/videos';
  }, 3000);
}
</script>
```

Change the red-highlighted parts in the script with your own values e.g. the fallback URL.

The 3-second delay gives GA4 enough time to receive the tracking hit before the user is redirected. You can reduce the delay if the page is lightweight and your testing confirms the GA4 hit is sent reliably.

Do not add a normal trigger to this tag.

This redirect tag should fire through tag sequencing after your GA4 tracking tag fires. We'll create this next.



×

Video Tracking Redirect

Save

⋮

Tag Configuration

Tag Type

<> Custom HTML Custom HTML Tag

HTML ⓘ

```
1 <script>
2
3 // Retrieve the value of the GTM variable
4 var videoUrl = {{video_click_tracking_url}};
5
6 // Check if the videoUrl has a value
7 if (videoUrl) {
8   // If videoUrl has a value, set a 3-second delay before redirecting
9   setTimeout(function() {
10     window.location.href = videoUrl;
11   }, 3000);
12 } else {
13   // If videoUrl does not have a value, redirect to the default YouTube channel URL
14   setTimeout(function() {
15     window.location.href = 'https://www.youtube.com/@trackfunnels/videos';
16   }, 3000);
17 }
18 </script>
19
```

another failsafe

Triggering

Firing Triggers

This tag has no firing triggers

No trigger needed

Your tag will fire right after these tags

Step 5: Create the GA4 tracking tag

You should now choose the tracking method.

Method A: Virtual pageview with UTMs

1. Create a new Google Tag. Name it ""

In the tag configuration, override:

page_location = {{video_click_tracking_url}}

page_title = {{video_click_tracking_title}}



Video Click Tracking Tag

Save

Tag Configuration

Tag Type

Google Tag
Google

Tag ID ⓘ

{{GA4 Measurement ID}}

Configuration settings

Set configuration parameters that update your tag's behavior. [Learn more](#)

Configuration Settings Variable

None

Configuration Parameter	Value
page_title ✓	{{video_click_tracking_title}}
page_location ✓	{{video_click_tracking_url}}

Add parameter

2. Create a trigger for the intermediate page. Add to the above tag.

Example trigger condition:

Page Path contains **/go-to-video/**

Intermediate Page Views

Save

Added in this workspace Abandon changes

Trigger Configuration

Trigger Type

Consent Initialization

This trigger fires on

Page Path contains /utm-links-builder/demos/go-to-video/

References to this Trigger

- Google Tag
- Video Click Tracking Tag

Use the same Trigger type you use for your Main Google Tag

Add condition based on your intermediate page path



Finally, use tag sequencing to fire the redirect tag after this tracking tag.

The screenshot shows the Google Tag Manager configuration interface for a tag. The top section, 'Configuration Parameter', lists two parameters: 'page_title' with value '{{video_click_tracking_title}}' and 'page_location' with value '{{video_click_tracking_url}}'. Below this is an 'Add parameter' button. The 'Advanced Settings' section includes 'Tag firing priority' (empty), 'Enable custom tag firing schedule' (unchecked), and 'Only fire this tag in published containers.' (unchecked). The 'Tag firing options' dropdown is set to 'Once per event'. The 'Tag Sequencing' section shows two options: 'Fire a tag before 01 GA4 - 01 virtual Pageview for Video Clicks fires' (unchecked) and 'Fire a tag after 01 GA4 - 01 virtual Pageview for Video Clicks fires' (checked). Below this, the 'Cleanup Tag' dropdown is set to 'Video Tracking Redirect', and the option 'Don't fire Video Tracking Redirect if 01 GA4 - 01 virtual Pageview for Video Clicks fails or is paused' is unchecked.

3. **Change your main Google Tag:** Add this trigger as an exception to your main Google Tag, so the main tag does not also fire on the intermediate page.

This avoids double-counting pageviews.



× Google Tag □

Save ⋮

Modified in this workspace View changes Abandon changes

Tag Configuration

Tag Type

Google Tag
Google

Tag ID ?
{{GA4 Measurement ID}}

Triggering

Firing Triggers

All Pages
Page View

Exceptions

Intermediate Page Views
Consent Initialization

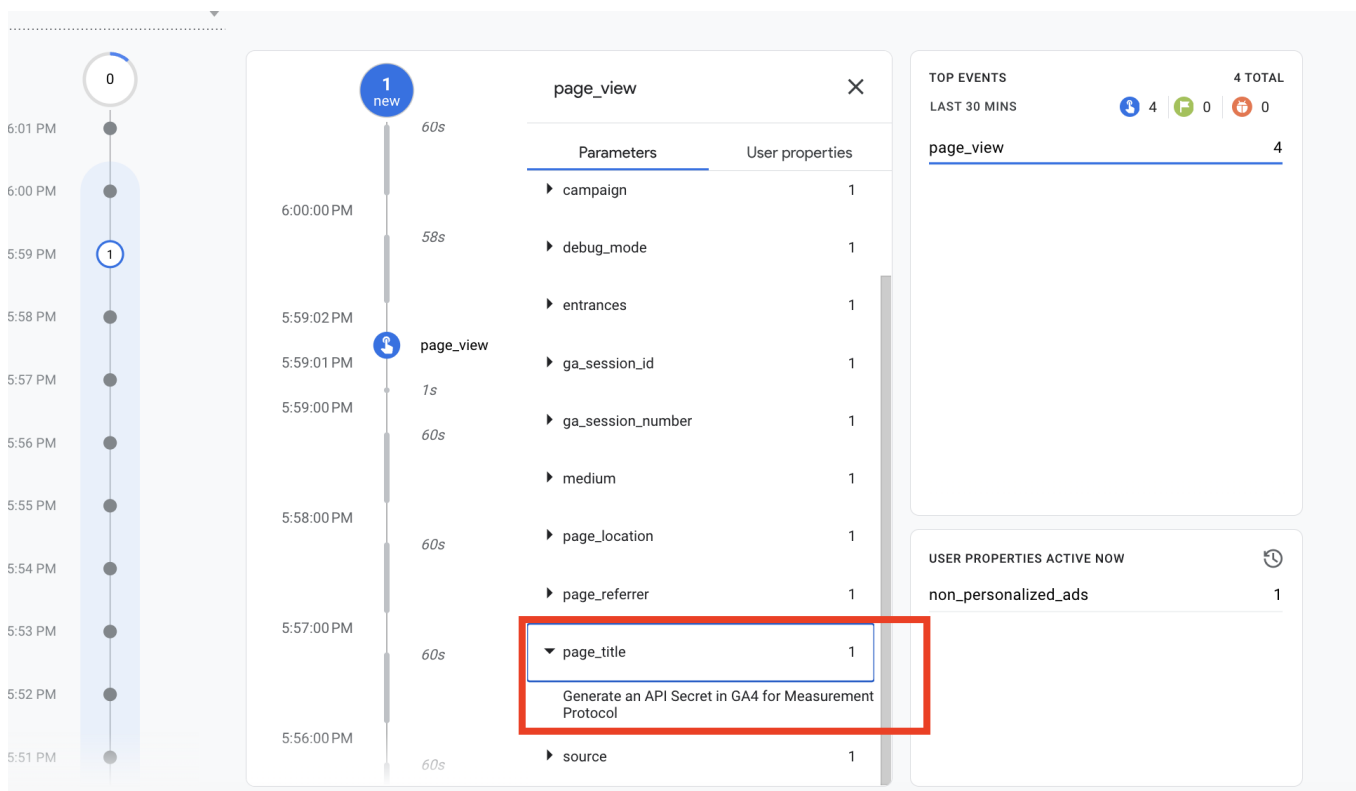
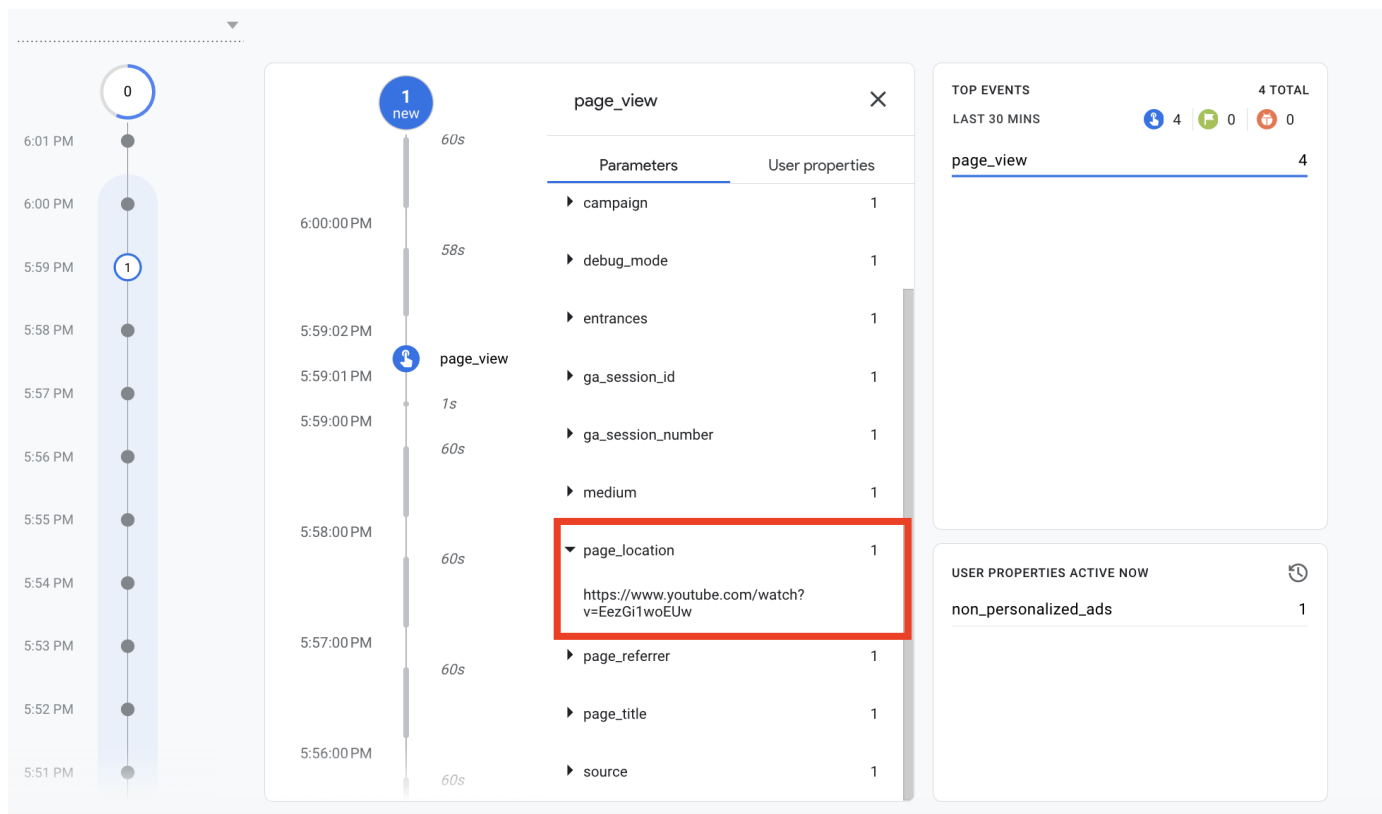
Add an Exception to your main Google tag to avoid multiple page views

Test with a URL like:

HTML

```
https://yourwebsite.com/go-to-video/?video_id=1&utm_source=linkedin&utm_medium=social
```

Check GA4 DebugView to confirm the pageview contains the correct source, page URL, and page title.





Method B: GA4 event tracking

Create another URL variable for the source.

Variable name: video_click_tracking_source

Variable type: URL

Component type: Query

Query key: vid_source

The screenshot shows the GA4 Variable Configuration interface for the variable 'video_click_tracking_source'. At the top, there is a green banner that says 'Added in this workspace' and a button 'Abandon changes'. Below this, the 'Variable Configuration' section shows the 'Variable Type' as 'URL' (indicated by a globe icon) and the 'Component Type' as 'Query'. The 'Query Key' is set to 'vid_source'. Below the configuration section, the 'References to this Variable' section shows a single reference: 'Video Clicks Tracking Event Tag' (indicated by a play button icon).

Create a GA4 Event tag.

Event name: video_url_clicks

Parameter	Value
video_title	{{video_click_tracking_title}}
video_url	{{video_click_tracking_url}}
video_source	{{video_click_tracking_source}}
video_id	{{video_tracking_url_id}}



Video Clicks Tracking Event

Added in this workspace Abandon changes

Tag Configuration

Tag Type

Google Analytics: GA4 Event
Google Analytics

Measurement ID
{{GA4 Measurement ID}}

Google tag found in this container

Event Name ⓘ
video_url_clicks

Event Parameters

Event Parameter	Value
video_clicked_url	{{video_click_tracking_url}}
video_clicked_title	{{video_click_tracking_title}}
video_click_source	{{video_click_tracking_source}}

Tag Sequencing ⓘ

Cleanup Tag
Video Tracking Redirect

An event name to identify

Other data points added as parameters

Use Tag sequencing to fire the redirect handler

Use tag sequencing to fire the redirect tag after this GA4 event tag.

Trigger:

Trigger this tag only on the intermediate page.

Intermediate Page Views

Added in this workspace Abandon changes

Trigger Configuration

Trigger Type

Page View

This trigger fires on

☐ All Page Views ☒ Some Page Views

Fire this trigger when an Event occurs and all of these conditions are true

Page Path	contains	/utm-links-builder/demos/go-to-vid	-	+
-----------	----------	------------------------------------	---	---



Check GTM Preview Mode and GA4 DebugView to confirm the event and parameters are sent correctly.

My test URL is:

http://trackfunnels.com/utm-links-builder/demos/go-to-video/?video_id=1&vid_source=linkedin

3 Container Loaded >
Video Clicks Tracking Event Fired

Tag Details Display Variables as ☐ Names ☒ Values

Properties

Name	Value
Type	Google Analytics: GA4 Event
Firing Status	Succeeded
Send Ecommerce data	false
Include user-provided data from your website	false
eventSettingsTable	<pre>[{ parameter: "video_clicked_url", parameterValue: "https://www.youtube.com/watch?v=EezGi1woEUw" }, { parameter: "video_clicked_title", parameterValue: "Generate an API Secret in GA4 for Measurement Protocol" }, { parameter: "video_click_source", parameterValue: "linkedin" }]</pre>
Event Name	"video_url_clicks"

[Show More](#)

Step 6: Create new trackable links

After the setup is complete, creating links becomes a repeatable process.

For every new video:

1. Add the video ID, title, and YouTube URL to your GTM lookup variables.
2. Create a trackable link using your intermediate page.
3. Add the correct **video_id**.
4. Add source, medium, campaign, or other query parameters.
5. Test the link before sharing it.



Example:

None

```
https://yourwebsite.com/go-to-video/?video_id=4&utm_source=newsletter&utm_medium=email&utm_campaign=product-demo
```

Or, if you use event tracking:

None

```
https://yourwebsite.com/go-to-video/?video_id=4&vid_source=newsletter
```

Limitations of this setup

This setup gives you cleaner GA4 reporting, but it adds some maintenance.

The main limitation is that your video list must stay updated in GTM. Every time you create a new video, you need to add a new video ID, title, and URL.

The second limitation is link creation. Instead of copying the YouTube URL directly, you need to create a trackable URL using the intermediate page and query parameters.

These are the trade-offs for getting YouTube link click data inside GA4.

If you want to reduce manual work, you can automate link creation with Zapier. For example, you could submit a YouTube URL through a form and receive the final trackable link automatically.



Need help implementing this?

If you want this set up without going through the technical process yourself, TrackFunnels can help.

We can set up the intermediate page, configure GTM, send the data to GA4, test everything, and document the process so your team can create new trackable YouTube links later.

Book a free implementation call

<https://calendly.com/consultation-with-malik/discovery-call-30-mins>

Or Contact me directly

malik@trackfunnels.com



Copyright © 2026 TrackFunnels. All rights reserved.

You are welcome to use this guide to understand and implement YouTube link click tracking for your own website or organization.

Please do not copy, resell, republish, or present this guide, its screenshots, examples, or implementation process as your own. If you reference it publicly, please credit TrackFunnels and link back to the original article or PDF.

For reuse, training, workshops, or commercial redistribution, please request written permission first.

Contact: malik@trackfunnels.com

Disclaimer

This guide is provided for educational and informational purposes only. The implementation examples are based on a general GA4 and Google Tag Manager setup and may need to be adapted to your website, CMS, consent setup, analytics configuration, and internal tracking requirements.

TrackFunnels is not responsible for errors, data loss, incorrect reporting, compliance issues, or tracking discrepancies that may result from implementing this setup without proper testing.

Before publishing changes, always test your setup in Google Tag Manager Preview Mode, GA4 DebugView, and your browser. If your website uses a consent management platform, make sure your tracking setup respects your consent rules and applicable privacy requirements.

Implementation note

This guide includes technical steps involving website changes, Google Tag Manager, URL parameters, JavaScript, and GA4 configuration. Please review the setup with your developer, analytics specialist, or legal/privacy team before deploying it on a production website.

The examples in this guide are intended as a starting point. Your final setup may differ based on your CMS, GTM container structure, GA4 property, consent mode setup, and reporting requirements.

Trademark notice

Google, Google Analytics, GA4, Google Tag Manager, YouTube, and other product names mentioned in this guide are trademarks or registered trademarks of their respective owners. TrackFunnels is not affiliated with, endorsed by, or sponsored by Google, YouTube, or any other third-party platform mentioned in this guide.