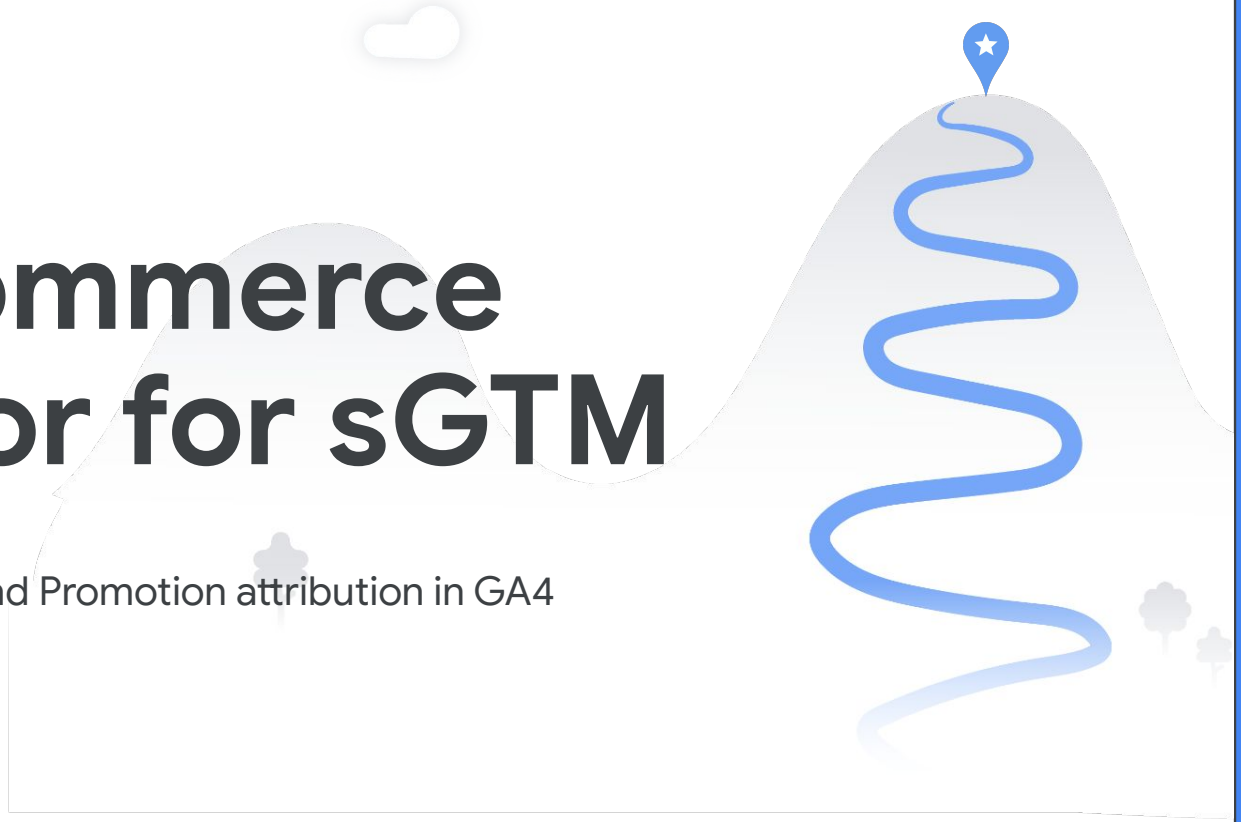
Google

GA4 Ecommerce Attributor for sGTM

Helps you enable Item list and Promotion attribution in GA4



Agenda

01

One Pager

Ecommerce Attributor for sGTM

Requirements:

google/ga4-ecom-attributor

- Server-side GTM container up and running
- Access & billing enabled in Google Cloud Project
- List information available on select_item, view_item or add_to_cart event
- Item ID available on every ecommerce event

Customer Challenge

Advertisers often in their ecommerce implementation have List information only available when user clicks on the item inside the Item List. Similar is for Promotion - information is usually available only when user clicks on specific Promotion.
In order to get full data in GA4 Item List & Promotion reports, Item List and Promotion data needs to be sent with all ecommerce events.

Often, due to the complexity of a website, it is hard for advertisers to provide this information with every single ecommerce event, which creates a challenge how to make these reports actionable.

Item list name	Item-list view events	Item-list click events	Item list click-through rate	Add to baskets	Checkouts	E-commerce purchases	Item revenue
	27,137 100% of total	6,382 100% of total	47.31% Avg 0%	1,387 100% of total	2,351 100% of total	365 100% of total	€150,545.29 100% of total
Home - top products	10,543	312	5.58%	0	0	0	€0.00
Search results	4,554	2,246	57.41%	0	0	0	€0.00
On sale	3,943	308	15.68%	0	0	0	€0.00
You might also like	3,374	755	29.77%	0	0	0	€0.00
Top deals	2,505	1,382	58.61%	0	0	0	€0.00
Basket upsell	1,336	728	58.3%	0	0	0	€0.00
New products	540	208	46.79%	0	0	0	€0.00
(not set)	1	0	0%	1,387	1,999	364	€150,545.29

Solution Description

This solution is designed to pass List and Promotion data with every ecommerce event. Each time when user interacts with item or promotion (which contains List or Promotion information), List and Promotion information will be stored.
Depending on which version of solution you are using, information will be stored either in 1st party cookie or in Firestore. This data is then passed along with all subsequent ecommerce events to GA4.

Item list name	Item-list view events	Item-list click events	Item list click-through rate	Add to baskets	Checkouts	E-commerce purchases	Item revenue
	3,725 100% of total	845 100% of total	47.41% Avg 0%	235 100% of total	398 100% of total	55 100% of total	€2,669.93 100% of total
Home - top products	1,364	40	6.47%	3	2	1	€3.32
Search results	696	317	61.26%	63	94	14	€506.09
On sale	638	143	35.34%	33	65	10	€382.86
You might also like	589	57	20.28%	9	41	4	€91.03
Top deals	178	106	58.33%	28	67	9	€391.79
Basket upsell	143	89	52.75%	12	24	4	€150.51
New products	59	33	61.76%	2	3	1	€10.47
(not set)	0	60	0%	83	194	28	€1,133.86

Impact

Item List and Promotion attribution fully working in GA4, making Item List and Promotion reports fully populated with ecommerce data and ready to be used for analysis.

Limitations

Solution needs to be implemented in server-side Google Tag Manager container.
Data is stored in Firestore, which can generate additional GCP cost.

How to decide which version of solution is right for you?

	Ecom Attributor for web GTM (Github)	Ecom Attributor for sGTM (Github)
Ease of implementation	Implementation in web GTM container. No changes required in the website code.	Implementation in sGTM container. No changes required in the website code. Requires GCP billing enabled and setup of Firestore database.
Requirements	Item ID present in each ecommerce event.	Item ID present in each ecommerce event.
Measurement support	Supports only Universal Analytics & GA4 Data Layer schema. Works only if ecommerce measurement is implemented via Data Layer and web GTM (based on official Google ecommerce implementation documentation). Data can still be sent to sGTM endpoint.	Solution supports all types of ecommerce measurement implementation (gtag.js, web GTM, third party tag management system, custom implementation). Solution will works as long as ecommerce data received in sGTM respects GA4 ecommerce event data model (event and parameters naming).
Storage solution	List and Promotion information is stored in 1st party cookie. Browser cookie limitations could apply (cross-domain measurement restriction, max number of cookies per domain, max cookie size etc.).	List and Promotion information is stored in Firestore. Firestore quotas and limits apply.
Cost of Operation	No additional cost.	Additional cost could occur due to usage of sGTM and Firestore as a storage solution.

02

Resources

Resources

[group/ga4-ecom-attributor]

[google/ga4-ecom-attributor]

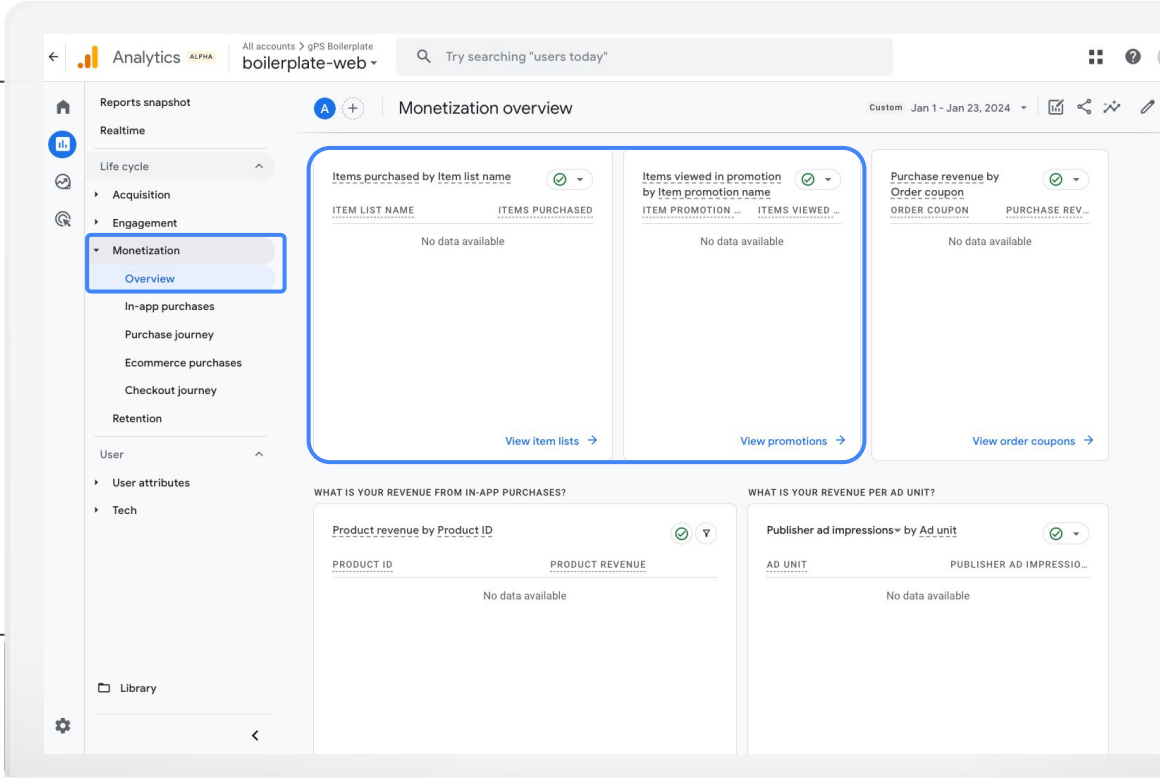
03

What Ecom Attributor can do for you?

Ecommerce measurement enables Item List & Promotion reports

With GA4 ecommerce measurement, it is possible to send Item List and Promotion information.

Information will be populated in this two specific reports in GA4 and it enables you to analyze success of internal promotions and item lists on the website.

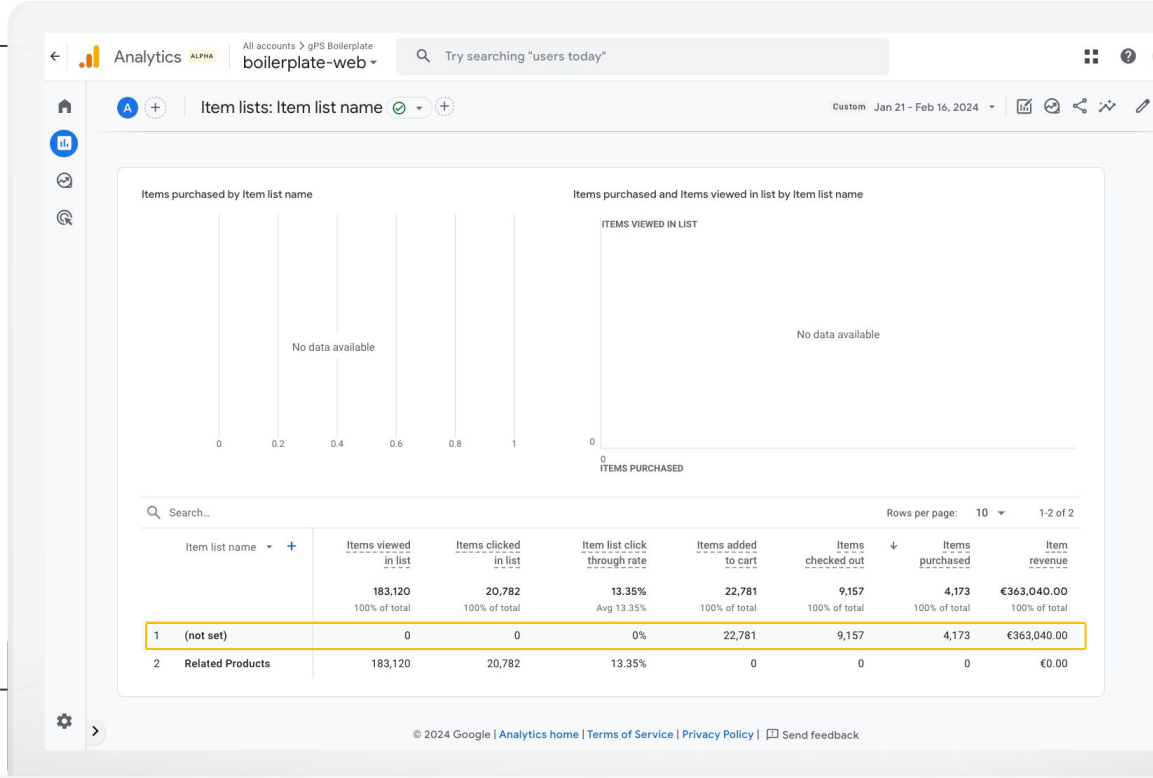


Item List report

How is report populated?

When user clicks on one of the items inside item list on the website (e.g. list of items on homepage that are on sale), Item List information needs to be sent with every ecommerce event in order to attribute Checkout and Revenue data to the appropriate Item List.

In case if List information is not sent with relevant ecommerce events, revenue data could be attributed to the (not set) value.

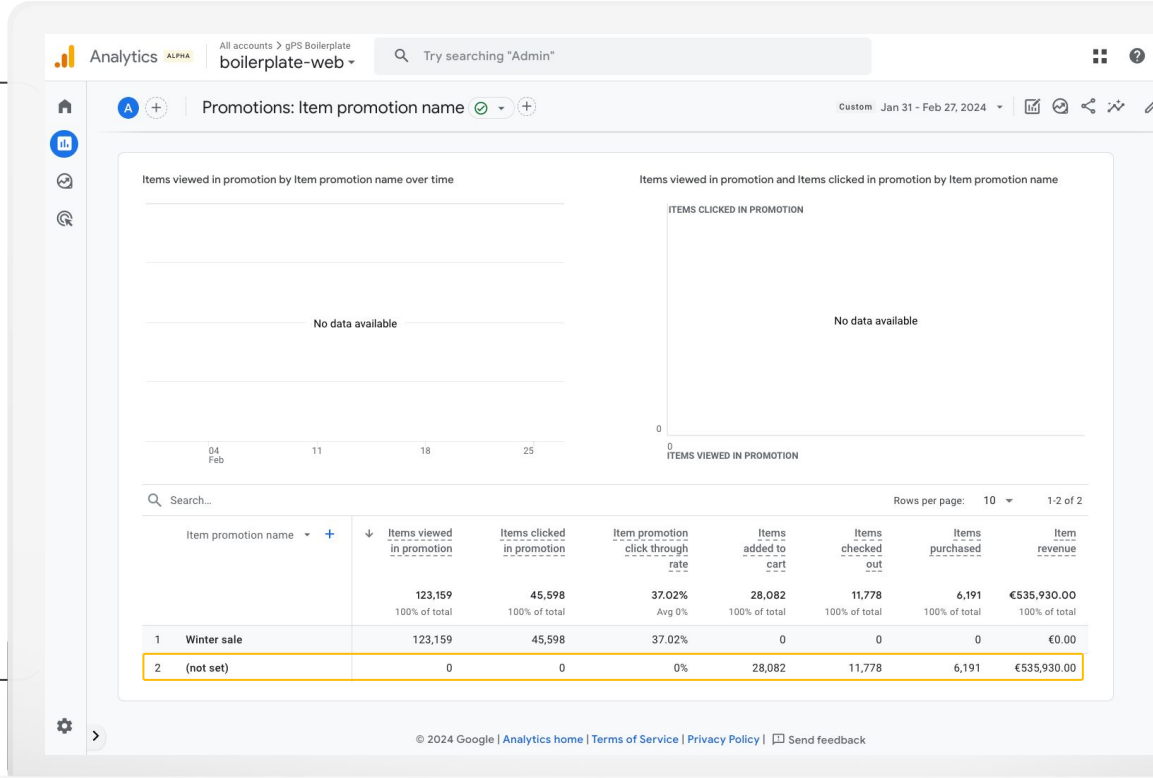


Promotions report

How is report populated?

When user clicks on one of promotions on the website, Promotion information needs to be sent with every ecommerce event in order to attribute Checkout and Revenue data to the appropriate Promotion.

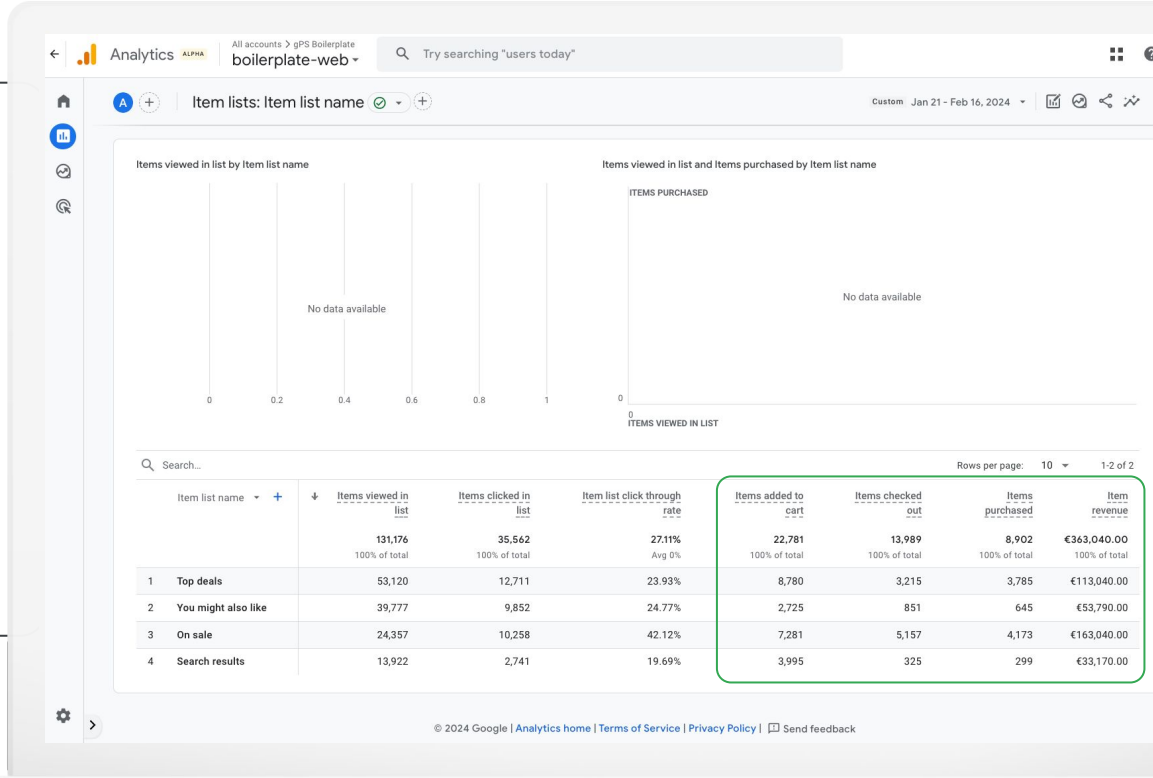
In case if Promotion information is not sent with relevant ecommerce events, revenue data could be attributed to the (not set) value.



Item List report with Ecommerce Attributor solution

Each time when user interacts with item (e.g. clicks on item inside the list), Ecommerce Attributor stores Item List information and sends it with all subsequent ecommerce events.

Data is populated in Item lists report and advertiser can start using report to make business decisions.

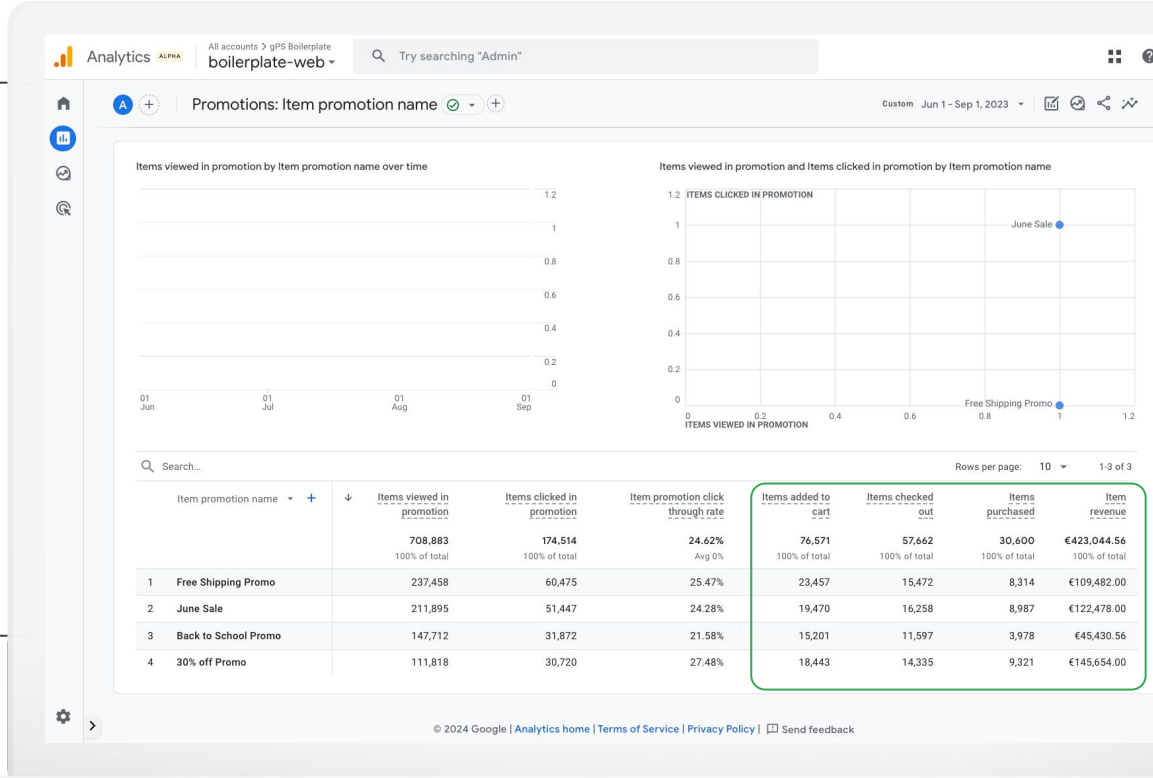


	Item list name	Items viewed in list	Items clicked in list	Item list click through rate	Items added to cart	Items checked out	Items purchased	Item revenue
		131,176 100% of total	35,562 100% of total	27.11% Avg 0%	22,781 100% of total	13,989 100% of total	8,902 100% of total	€363,040.00 100% of total
1	Top deals	53,120	12,711	23.93%	8,780	3,215	3,785	€113,040.00
2	You might also like	39,777	9,852	24.77%	2,725	851	645	€53,790.00
3	On sale	24,357	10,258	42.12%	7,281	5,157	4,173	€163,040.00
4	Search results	13,922	2,741	19.69%	3,995	325	299	€33,170.00

Promotions report with Ecommerce Attributor solution

Each time when user interacts with promotion (clicks on specific promotion on website), Ecommerce Attributor stores Promotion information and sends it with all subsequent ecommerce events.

Data is populated in GA4 Promotion report and advertiser can start using report to analyze Promotions and make business decisions.

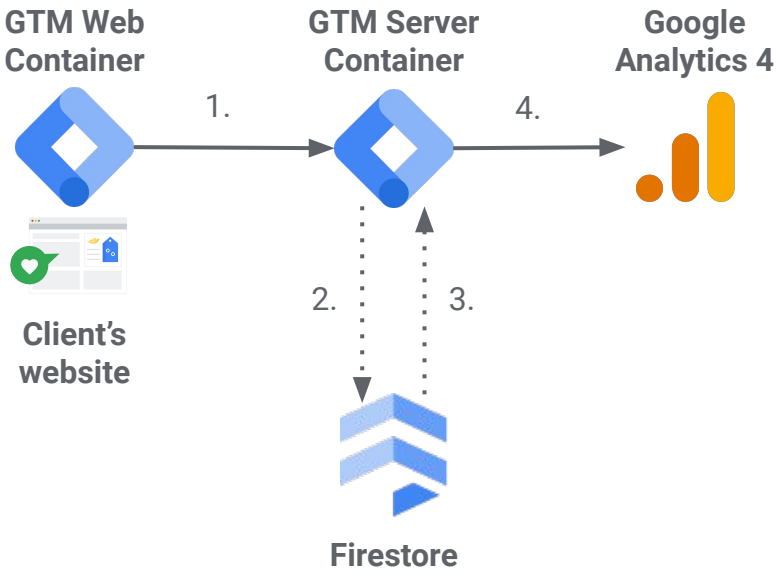


	Item promotion name	Items viewed in promotion	Items clicked in promotion	Item promotion click through rate	Items added to cart	Items checked out	Items purchased	Item revenue
		708,883 100% of total	174,514 100% of total	24.62% Avg 0%	76,571 100% of total	57,662 100% of total	30,600 100% of total	€423,044.56 100% of total
1	Free Shipping Promo	237,458	60,475	25.47%	23,457	15,472	8,314	€109,482.00
2	June Sale	211,895	51,447	24.28%	19,470	16,258	8,987	€122,478.00
3	Back to School Promo	147,712	31,872	21.58%	15,201	11,597	3,978	€45,430.56
4	30% off Promo	111,818	30,720	27.48%	18,443	14,335	9,321	€145,654.00

04 Technical solution explanation

Ecommerce Attributor for sGTM

1. User interacts with items on website, ecommerce event data is sent from website to server-side GTM container
2. If ecommerce event contains List and/or Promotion data, data is stored in Firestore for each item user had interaction with
3. When user progresses through the shopping funnel on your website (e.g. triggers ecommerce checkout events or purchase event), List and Promotion data is fetched from Firestore, appended to the item
4. Item data is sent to GA4, Checkout and Revenue data is attributed to the appropriate List and Promotion



05

Implementation guide

Pre-requisites

Server Side Google Tag Manager
Deployed sGTM container and website data is routed to the sGTM endpoint

Google Cloud Project
Access to a Google Cloud project with Firestore in Native mode

Ecommerce measurement
Active ecommerce measurement implemented on website with List and/or Promotion data

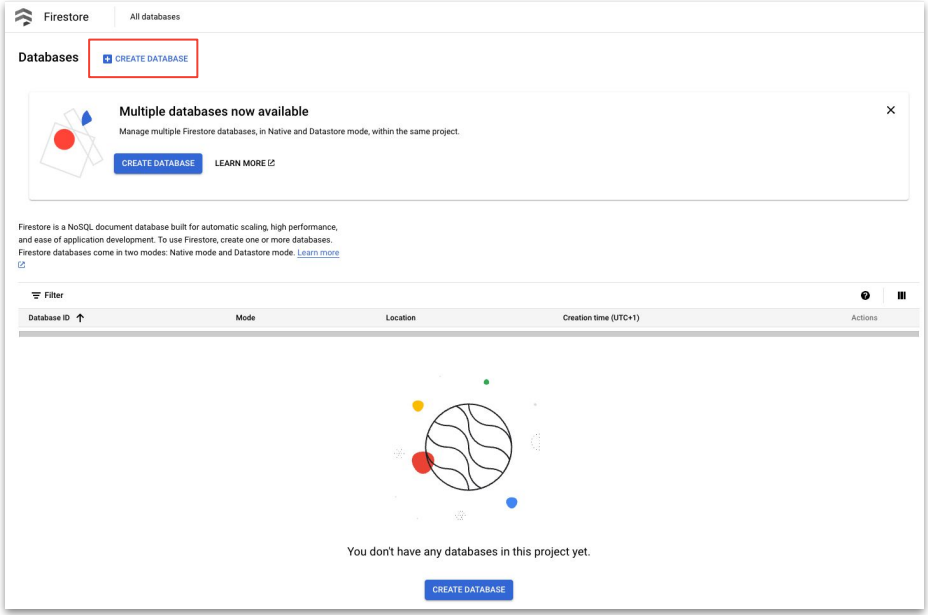
Create Firestore Database

Step 1

Go to Google Cloud Platform, to the Firestore page.

If you don’t have any databases created, click on **Create database** button.

If you already have database created, make sure you have Database with name **(default)**, **Native** mode.



Create Firestore Database

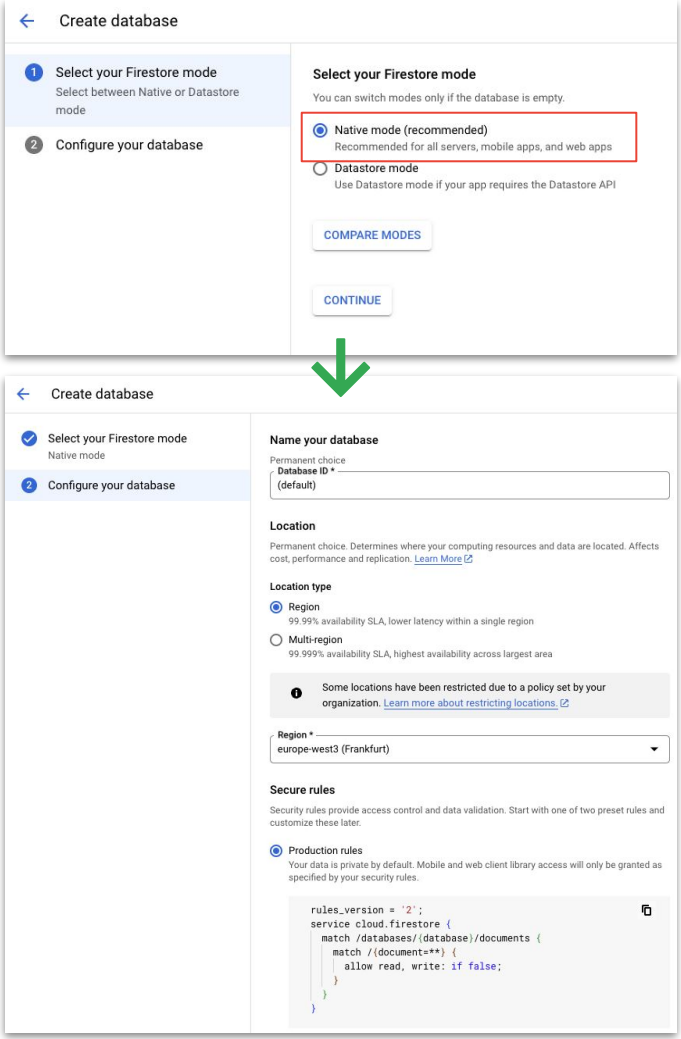
Step 2

In first step, for Firestore mode select **Native mode (recommended)** and click on Continue.

In second step, keep Database ID as it is (don't change it): **(default)**

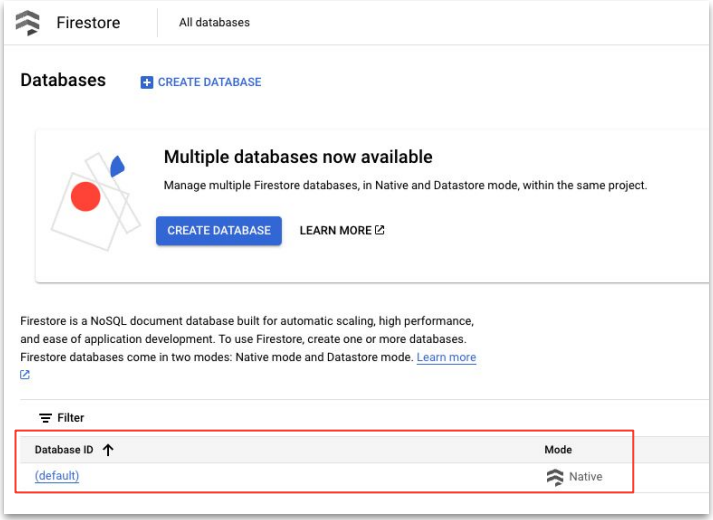
For location type, select what you prefer. In this example, we will select Region and europe-west3.

Once you are finished, click on blue button **Create database**.



Create Firestore Database
Overview

After Firestore Database is created, it should look like this.



Create Time-to-live (TTL) policy

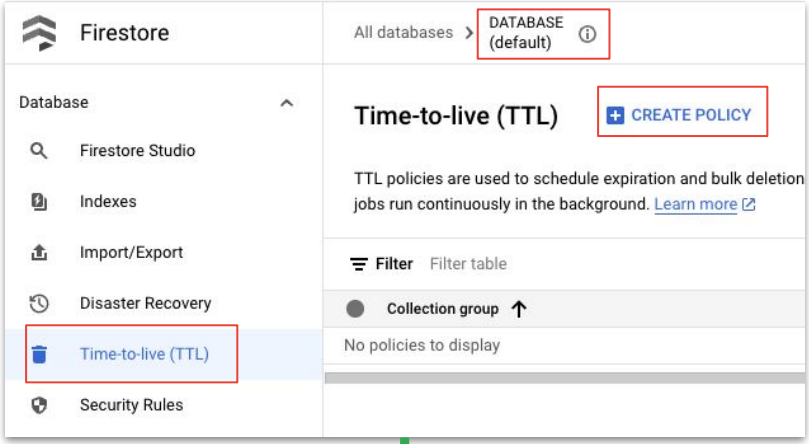
Step 1

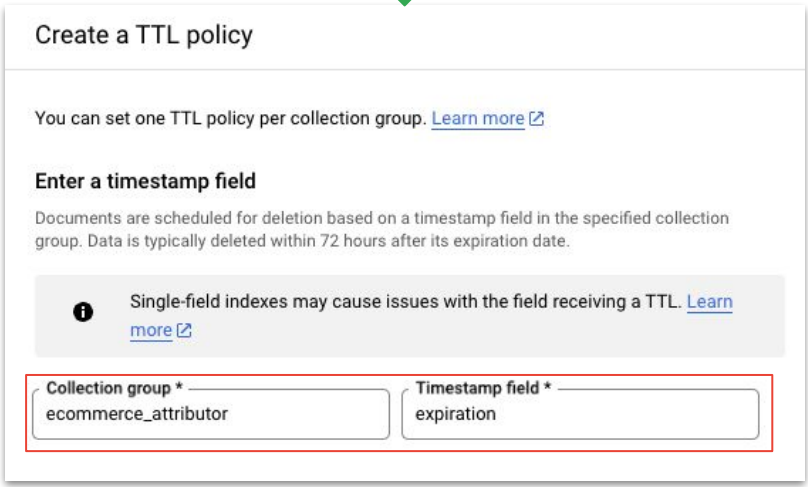
In Firestore, click on **(default)** database, and then on left-side navigation, click on **Time-to-live (TTL)**.

Click on **Create Policy.**

As **Collection group** enter: **ecommerce_attributor**
As **Timestamp field**, enter: **expiration**

Click on **Save** button. It can take some time until policy is created.

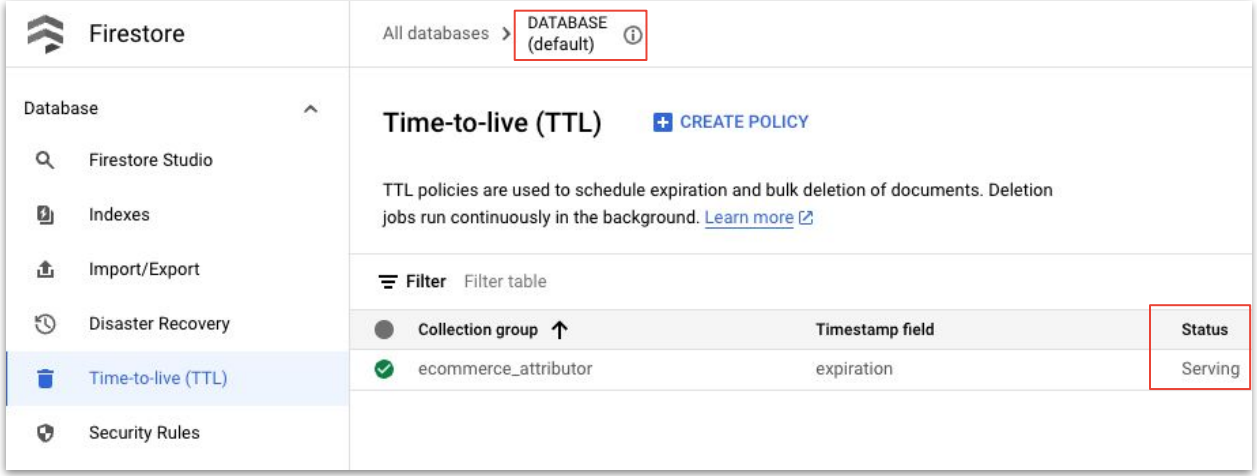




Create Time-to-live (TTL) policy

Overview

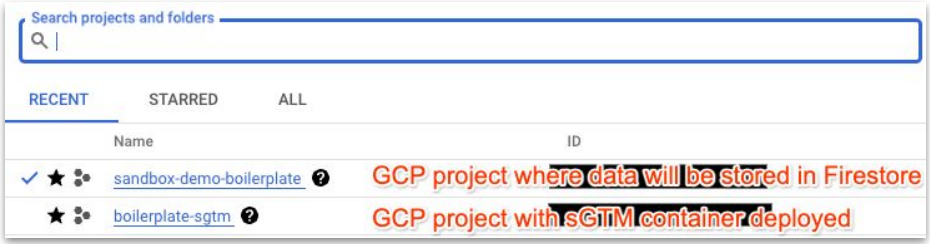
Once TTL policy is created, it should look like this.



Provide sGTM access to Firestore project

This step is necessary only for users who have two separate GCP projects for sGTM container and for Firestore database.

In case if your sGTM container and Firestore database are located in the same Google Cloud Platform project, you can skip this step.



If you have GCP setup just like it is shown on picture, follow steps on next slide.

This is required in order to give server-side GTM access to the Firestore project.

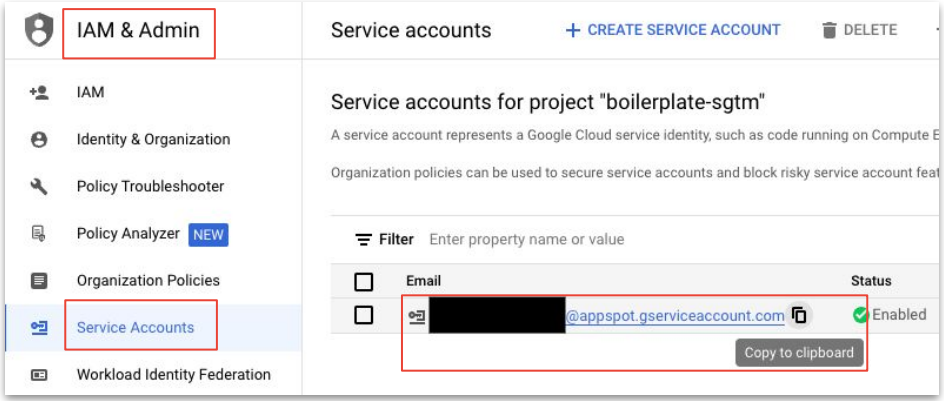
Provide sGTM access to Firestore project

Step 1

Navigate to your GCP project where you have **server-side GTM container deployed**.

Go to **IAM & Admin** and in left-side navigation, click on **Service Accounts**.

Copy your service account email address.



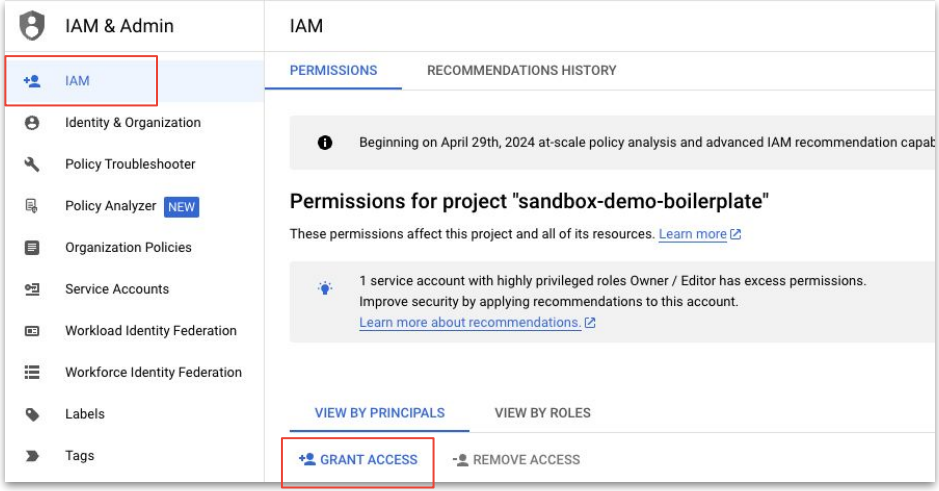
Provide sGTM access to Firestore project

Step 2

Now, navigate to your GCP project which you will use to store data in Firestore (GCP project where you created Database and TTL policy in Firestore in previous step).

Go to **IAM & Admin** and in left-side navigation, click on **IAM**.

Click on **Grant Access**.



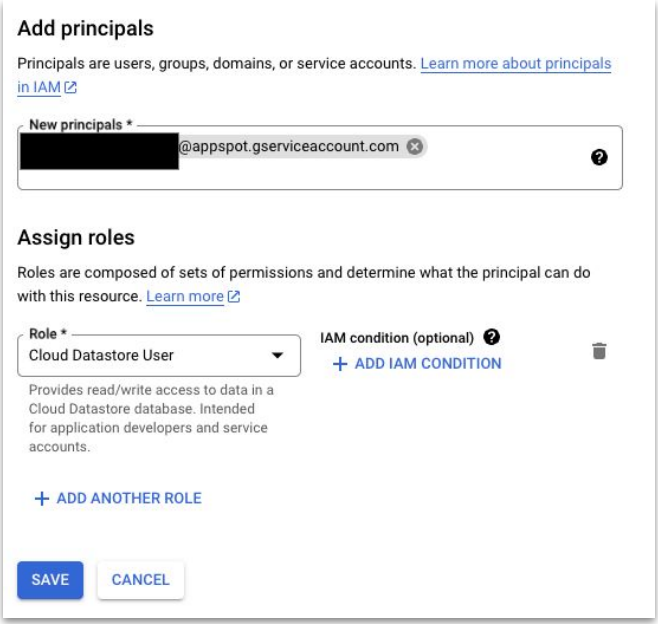
Provide sGTM access to Firestore project

Step 3

Paste your service account in **New Principals** field.

As a Role, select **Cloud Datastore User**.

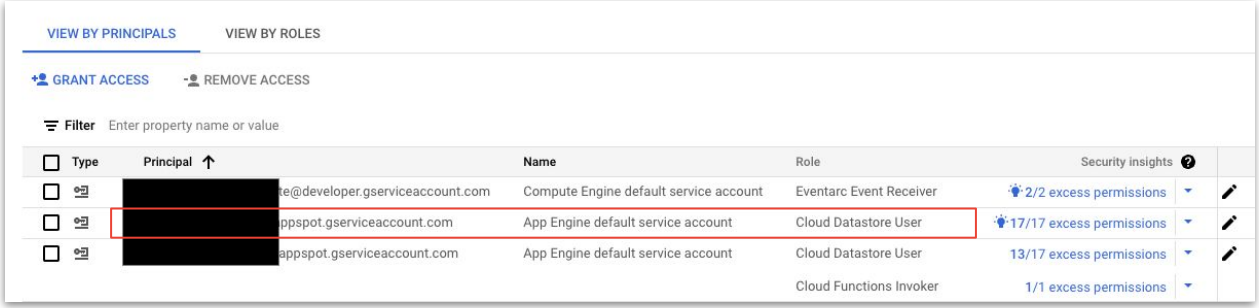
Click **Save** button.



Provide sGTM access to Firestore project

Overview

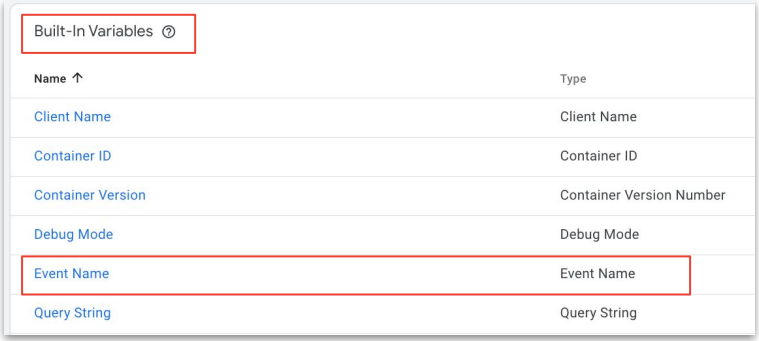
After you provide access, you should see your Service account email listed with Cloud Datastore User access.

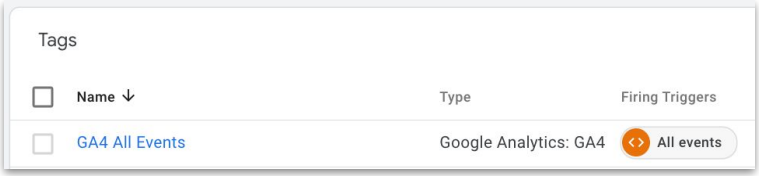


Prerequisites before importing solution in sGTM

Before you import JSON file with solution, go to your sGTM container and make sure you have:

- Built-in Variable **Event Name** enabled
- GA4 event tag that forwards data to GA4





sGTM JSON import

Step 1

Join Google Group, to be able to download JSON file and to get notifications about solution updates.

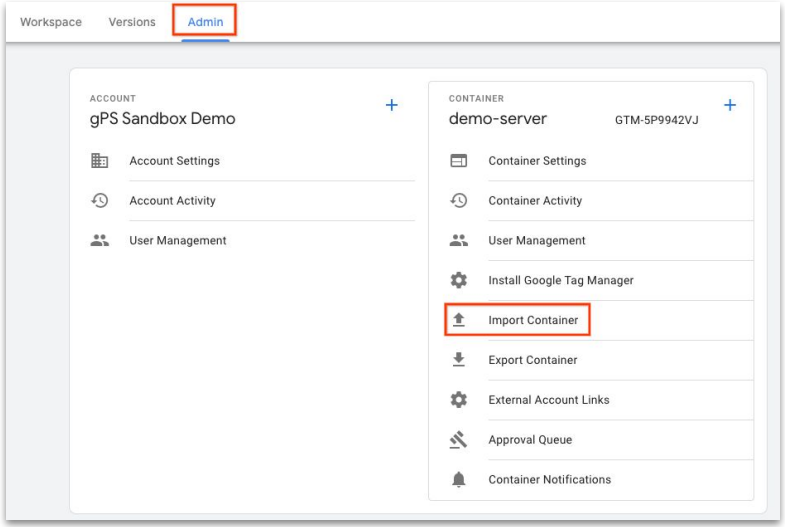
Download the JSON file (available on Github).

Make sure it is saved as **.json** file, otherwise it will not work.

The file is exported sGTM container, which contains tags, variables and custom built templates necessary for this solution to work.

sGTM JSON import
Step 2

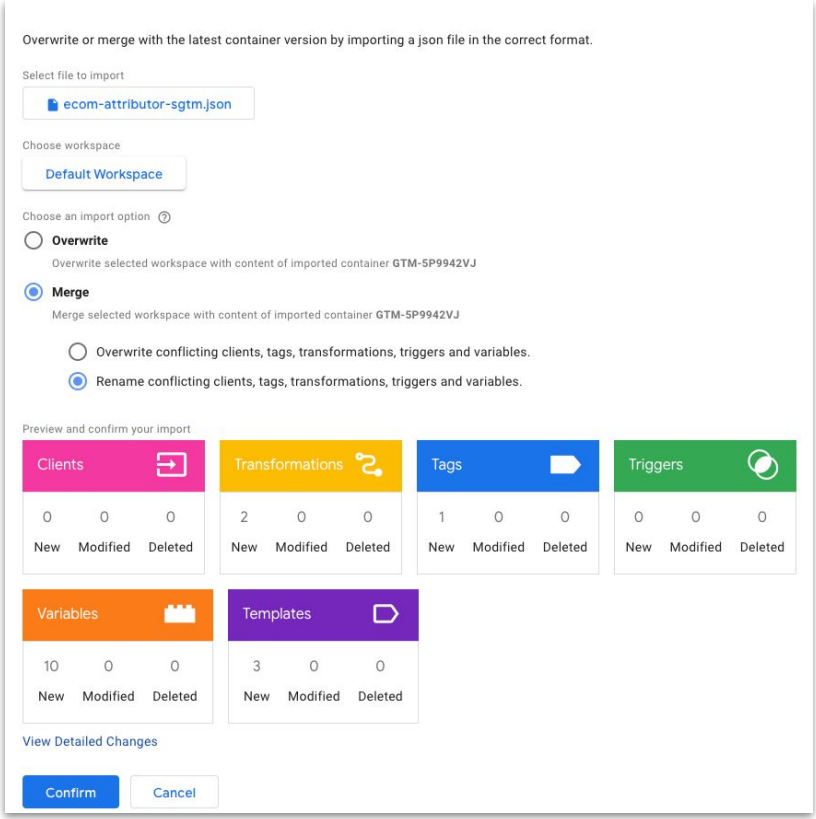
1. Go to Google Tag Manager
2. Open your server-side GTM container where you want to implement solution
3. Click **Admin** → **Import Container**



sGTM JSON import

Step 3

1. Select downloaded JSON file to import
2. Select in which workspace you want to import the solution
3. Select **Merge** and then **Rename conflicting tags, triggers and variables**
4. Click on **Confirm** button



sGTM JSON import
Overview

After you import JSON file, you should see these changes in your workspace.

Next, we need to set up following things:

- Add your GCP Project ID in **Constant - GCP Project ID** variable
- Configure **Ecom Attributor - Write to Firestore** tag
- Configure triggers for **Ecom Attributor - Write to Firestore** tag
- Optional: configure **Identifier - hashed** variable

Workspace Changes

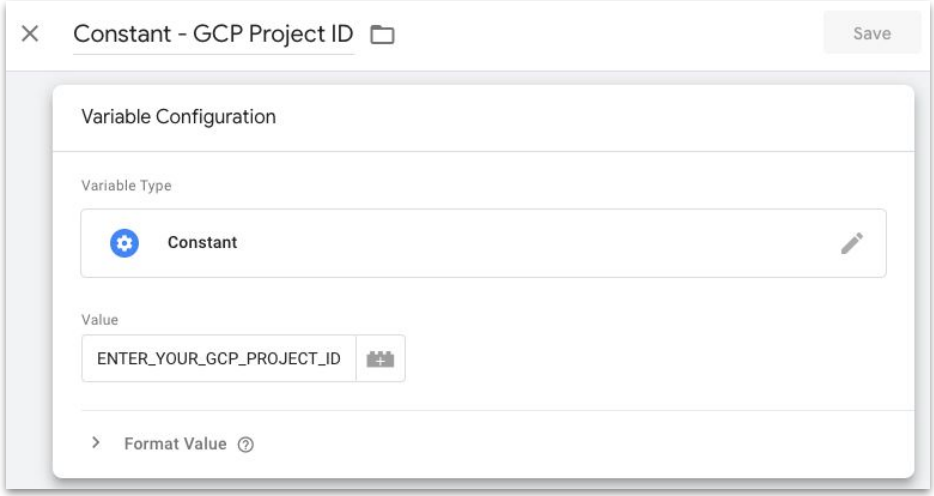
Name	Type ↑	Change
Ecom Attributor - Identifier hasher	Custom Template	Added
Ecom Attributor - Items array	Custom Template	Added
Ecom Attributor - Write to Firestore	Custom Template	Added
Ecom Attributor	Folder	Added
Ecom Attributor - Write to Firestore	Tag	Added
Ecom Attributor - Augment - Item List data	Transformation	Added
Ecom Attributor - Augment - Promotion data	Transformation	Added
Identifier - Event Data - client_id	Variable	Added
Identifier - hashed	Variable	Added
Constant - FS Collection Path	Variable	Added
Constant - GCP Project ID	Variable	Added
FS Lookup - promotion_id	Variable	Added
FS Lookup - attribution	Variable	Added
Ecom Attributor - Items array	Variable	Added
FS Lookup - creative_name	Variable	Added
FS Lookup - promotion_name	Variable	Added
FS Lookup - creative_slot	Variable	Added

Configure Constant - GCP Project ID variable

Open variable called **Constant - GCP Project ID**

Delete everything from the input field, and **provide your Google Cloud Project ID**.

NOTE: This should be your GCP Project ID where you configured Firestore database in previous steps.



Configure Ecom Attributor - Write to Firestore tag

Step 1

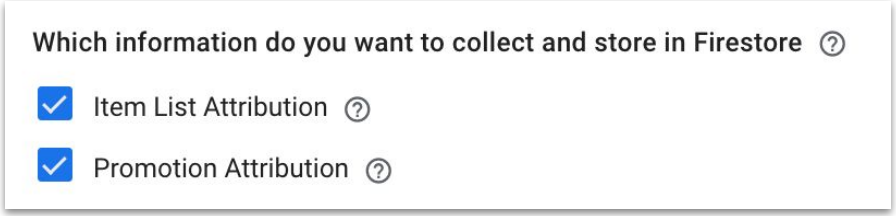
Open tag called **Ecom Attributor - Write to Firestore**

In the tag, you don't have to configure any GCP data or Identifier variable (this is optional, it will be explained in later steps).

Scroll down to the section **Which information do you want to collect and store in Firestore.**

Mark checkboxes for which Attribution type you want to use this solution:

- Item List Attribution
- Promotion Attribution



Configure Ecom Attributor - Write to Firestore tag

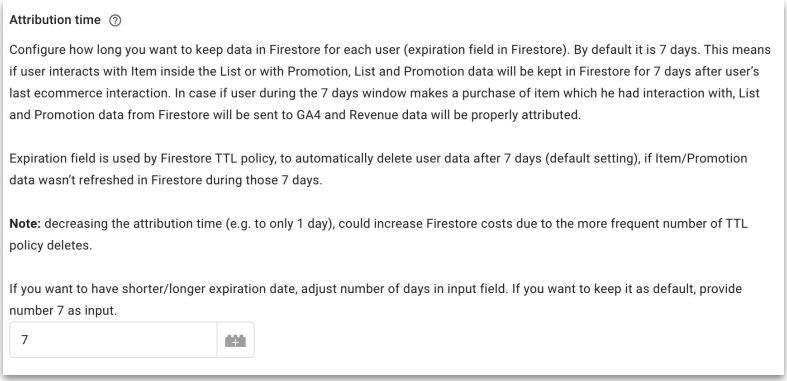
Step 2

Next, go to section called **Attribution time**.

Configure how long you want to keep data in Firestore. By default it is 7 days. This means if user interacts with Item inside the List or with Promotion, List and Promotion data will be kept in Firestore for 7 days after user's last ecommerce interaction. In case if user inside the 7 days window makes a purchase of item which he had interaction with, List and Promotion data from Firestore will be sent to GA4 and Revenue data will be properly attributed.

NOTE: When data is written in Firestore, there is field called **expiration**, which is timestamp set in the future (by default 7 days in the future). Expiration field is used by Firestore TTL policy, to automatically delete user data after 7 days, if Item/Promotion data wasn't refreshed in Firestore during those 7 days.

Decreasing the attribution time (e.g. to only 1 day) can increase Firestore costs due to more frequent TTL policy deletes.



Configure Ecom Attributor - Write to Firestore tag

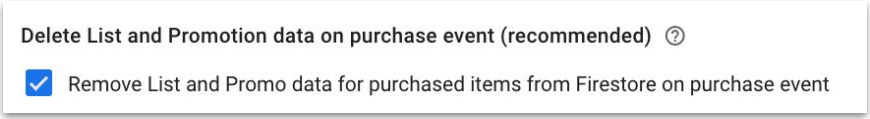
Step 3

Next, go to section called **Delete List and Promotion data on purchase event**.

What happens if you turn on this option:

When user makes a purchase, if user purchased itemA and itemB, and if there is Promotion and List data stored in Firestore for these two items, this information will be deleted from Firestore.

NOTE: In case if user has List data available for items which he didn't purchase currently, this information will stay in Firestore.

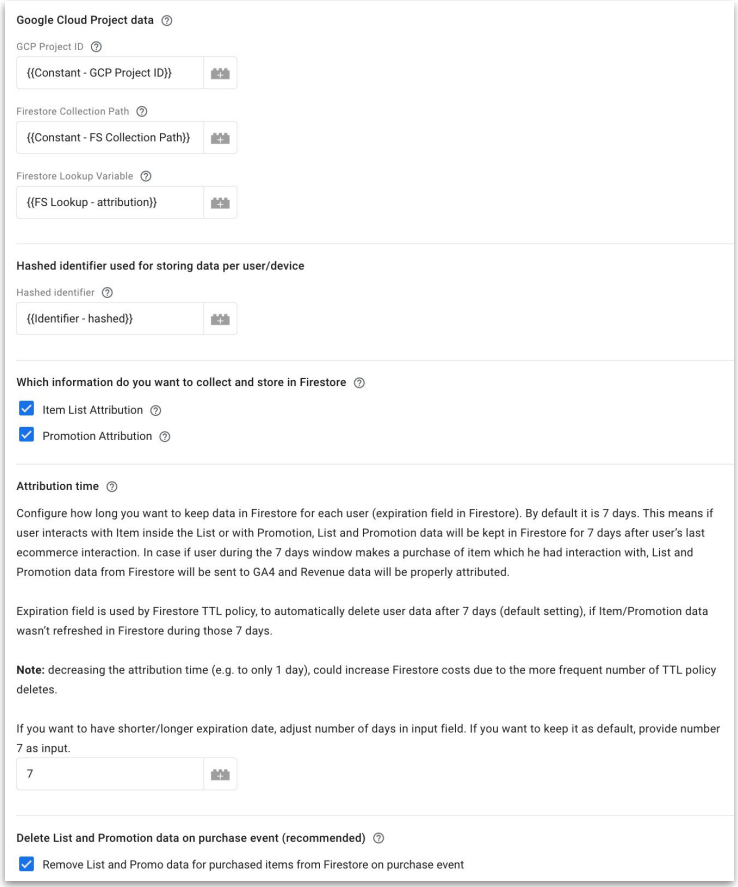


Configure Ecom Attributor - Write to Firestore tag

Overview

After you configure all settings, this is how Ecom Attributor - Write to Firestore tag should look like.

Depending on which settings you selected, your setup will be of course slightly different compared to this example.



Add triggers to Ecom Attributor - Write to Firestore tag

Step 1

Next, we need to provide triggers for **Ecom Attributor - Write to Firestore** tag.

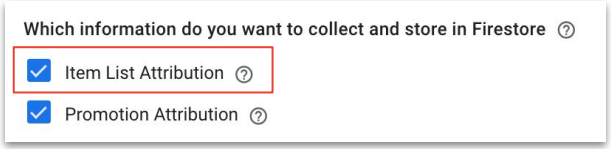
If you selected option **Item List Attribution**, you need to know on which event you have List information available in Event Data model. Solution supports collection of List data on following events:

- select_item
- view_item
- add_to_cart

NOTE: If you have cases on website when List data is available on all three mentioned ecommerce events, solution supports this as well. Just add all three events as a trigger to the tag.

If List data is available outside of items array, solution supports this as well and it will collect List information.

If Promotion data is available with Item, this will be collected and stored in Firestore as well.



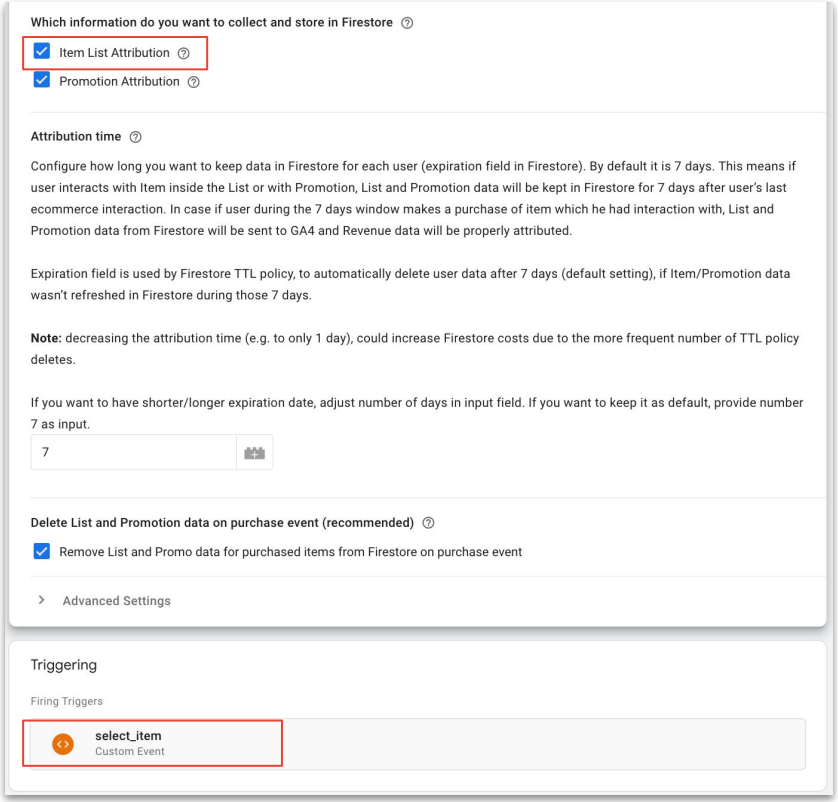

```
gtag("event", "select_item", {
    items: [{
        item_id: "SKU_12345",
        item_name: "Stan and Friends Tee",
        item_brand: "Google",
        item_category: "Apparel",
        item_category2: "Adult",
        item_category3: "Shirts",
        item_list_id: "related_products",
        item_list_name: "Related Products",
        location_id: "ChIJIQBpAG2ahYAR_6128GcTUEo",
        index: 1,
        promotion_id: "P_12345",
        promotion_name: "Summer Sale",
        creative_name: "summer_banner2",
        creative_slot: "featured_app_1"
      }]
  });
```

Add triggers to Ecom Attributor - Write to Firestore tag

Step 1

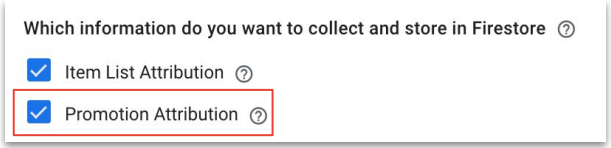
If you selected **Item List Attribution** option, and if you have List data available only on **select_item** event, add this event as a trigger to the tag.

If you have case where List information is also available on view_item and add_to_cart event, add those events are trigger as well.



Add triggers to Ecom Attributor - Write to Firestore tag

Step 2



If you selected **Promotion Attribution** option, you have to provide **select_promotion** trigger to the tag.

```
gtag("event", "select_promotion", {
  creative_name: "Summer Banner",
  creative_slot: "featured_app_1",
  promotion_id: "P_12345",
  promotion_name: "Summer Sale",
    items: [{
        item_id: "SKU_12345",
        item_name: "Stan and Friends Tee",
        item_brand: "Google",
        item_category: "Apparel",
        item_category2: "Adult",
        item_category3: "Shirts",
        item_list_id: "related_products",
        item_list_name: "Related Products",
        location_id: "ChIJIQBpAG2ahYAR_6128GcTUEo",
        index: 1
      }]
  });
```

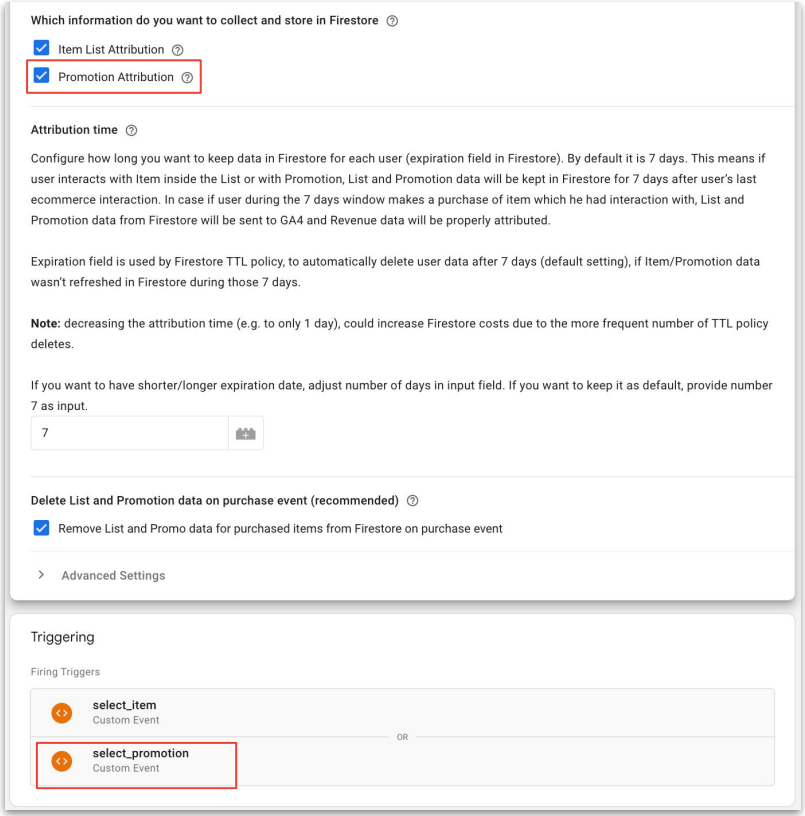
NOTE: Solution supports collection of Promotion data on select_promotion event only.

Solution will collect Promotion data if it is available inside items array but also it supports case if you are sending it outside of array.

Add triggers to Ecom Attributor - Write to Firestore tag

Step 2

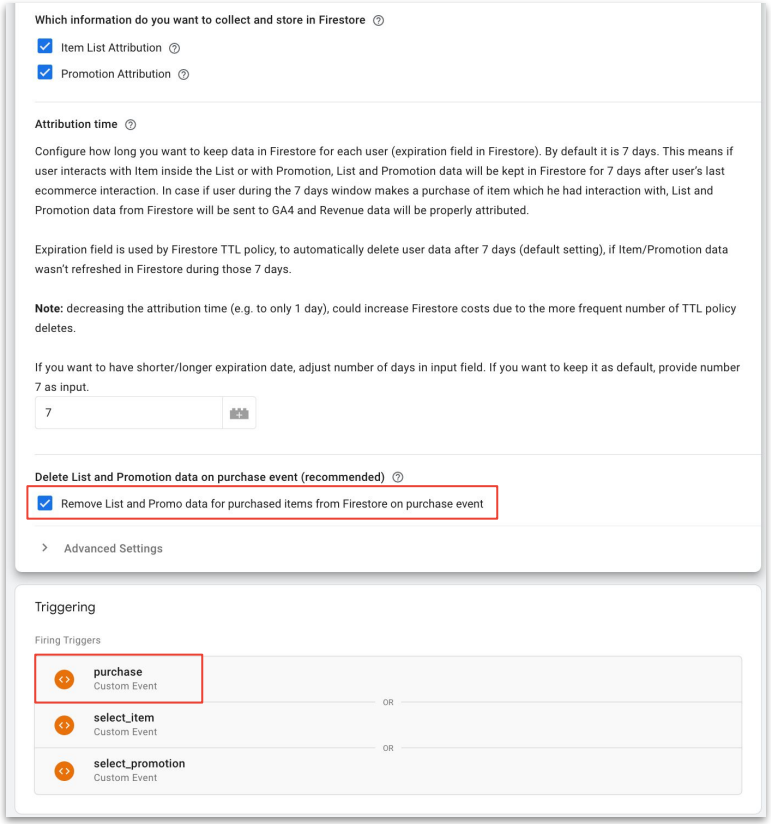
If you selected **Promotion Attribution** option, provide **select_promotion** event as a trigger to the tag.



Add triggers to Ecom Attributor - Write to Firestore tag

Step 3

If you selected option **Remove List and Promo data for purchased items**, provide **purchase** event as a trigger to the tag.



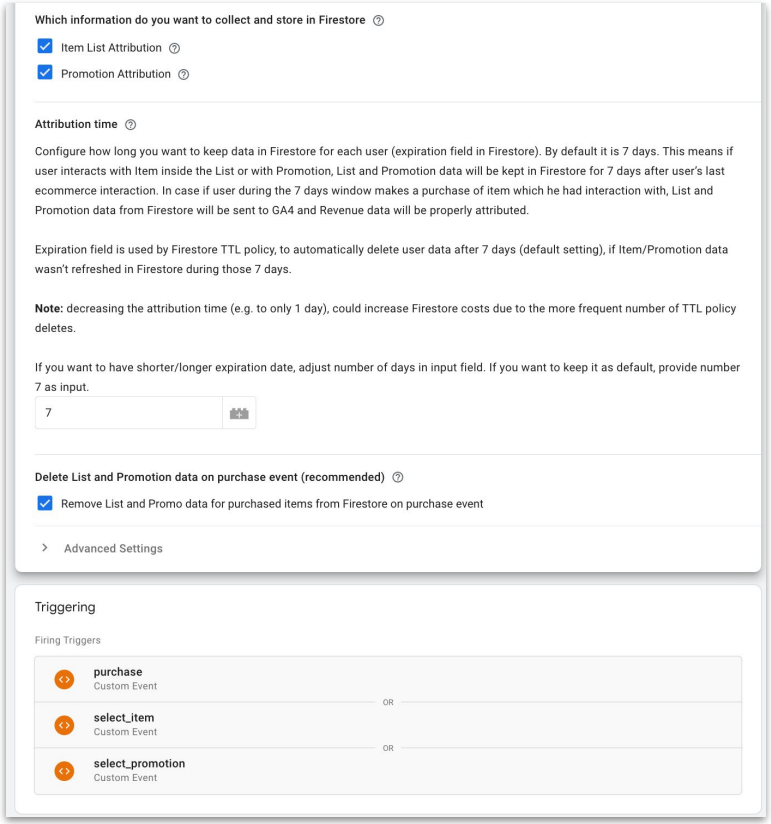
Add triggers to Ecom Attributor - Write to Firestore tag

Overview

Depending on which setting you selected, you should have:

- If you selected **Item List Attribution**, you should have **at least 1 trigger** (select_item, view_cart or add_to_cart)
- In case if you selected **Promotion Attribution**, you should have **select_promotion event as trigger**
- If you selected to **Remove List and Promo data after purchase**, you should have **purchase event as trigger**

IMPORTANT: If you have advanced Consent Mode implemented, it is worth to note that this solution won't work for unconsented pings. We recommend to **not trigger** Write to Firestore tag on unconsented pings.

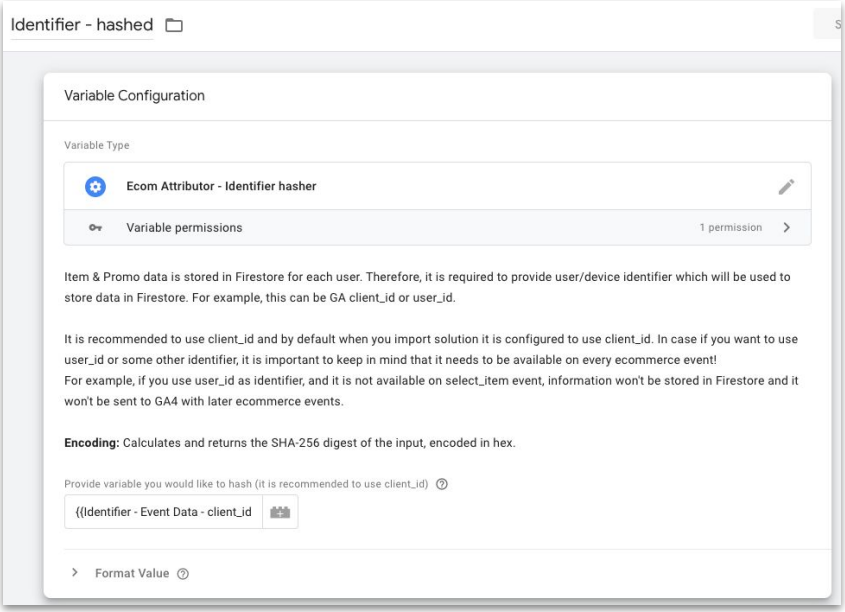


Optional #1: Configure Identifier - hashed variable

This step is optional.

By default, solution is using hashed **client_id** to store data in Firestore (SHA-256, encoded in hex).

In case if you want to use some other device/user identifier, then open variable called **Identifier - hashed.** By default, as an input client_id variable is provided. Remove it, and provide variable with identifier you would like to use.



IMPORTANT: if you want to use some other device/user identifier (e.g. user_id), this identifier has to be available on every ecommerce event, otherwise solution won't work.

Optional #2: Remove Promotion Transformation & Variables

This step is optional and should only be done if you don't have Promotions measurement implemented on your website

In case if you don't have promotion measurement implemented on your website and you are not storing any Promotion information in Firestore, you can delete following things from your sGTM container:

- Ecom Attributor - Augment - Promotion data
- FS Lookup - promotion_name
- FS Lookup - promotion_id
- FS Lookup - creative_slot
- FS Lookup - creative_name

Transformations

Name ↑	Type
Ecom Attributor - Augment - Item List data	Augment event
Ecom Attributor - Augment - Promotion data	Augment event

User-Defined Variables

Name	Type
FS Lookup - promotion_name	Firestore Lookup
FS Lookup - promotion_id	Firestore Lookup
FS Lookup - creative_slot	Firestore Lookup
FS Lookup - creative_name	Firestore Lookup

Summary

Final checklist what you should have done using this step-by-step walkthrough

01

Google Cloud Platform setup

- Create **(default)** Firestore database (Native mode)
- Create TTL policy for **expiration** field in **ecommerce_attributor** collection

02

Import JSON file & Configure Constant - GCP Project ID variable

- Join Google Group
- Download JSON file from Github & import in your sGTM container
- Provide your GCP Project ID in **Constant - GCP Project ID** variable

03

Configure settings in Ecom Attributor - Write to Firestore tag

- Which information you want to collect in Firestore: Item List and/or Promotion data
- How long you want to keep user data in Firestore (default 7 days)
- Do you want to remove List & Promo data for purchased items from Firestore after purchase event

04

Add triggers to the Ecom Attributor - Write to Firestore tag

- If you selected to collect Item List data, you need to provide at least 1 trigger (select_item, view_item and/or add_to_cart)
- If you selected to collect Promotion data, you need to provide select_promotion event as trigger
- If you selected to remove List & Promo data after purchase event, you need to provide purchase event as trigger

*OPTIONAL STEP: Configure **Identifier - hashed** variable*

List of Tags, Transformations and Variables that are part of JSON file you have to import in sGTM:

Name	Type	Description
Ecom Attributor - Write to Firestore	Tag	Tag used to collect List and Promotion information and write it in the Firestore database
Ecom Attributor - Augment - Item List data	Transformation	Responsible for augmenting items array on ecommerce events which do not contain List information
Ecom Attributor - Augment - Promotion data	Transformation	Responsible for adding Promotion information on ecommerce events if promotion data is not attached to specific item
Ecom Attributor - Items array	Variable	Variable responsible for creating new items array with List/Promotion data from Firestore
Identifier - Event Data - client_id	Variable	Client_id variable from event data model
Identifier - hashed	Variable	Variable for hashing the client_id
Constant - GCP Project ID	Variable	Contains your GCP Project ID. You need to manually paste your GCP Project ID inside this variable
Constant - FS Collection Path	Variable	Contains Firestore Collection Path (by default Collection Path name is: ecommerce_attributor)
FS Lookup - attribution	Variable	Checks Firestore database and returns all List/Promotion data (if it exists for user)
FS Lookup - promotion_id	Variable	Checks Firerstore database and returns promotion_id if it exists
FS Lookup - promotion_name	Variable	Checks Firerstore database and returns promotion_name if it exists
FS Lookup - creative_name	Variable	Checks Firerstore database and returns creative_name if it exists
FS Lookup - creative_slot	Variable	Checks Firerstore database and returns creative_slot if it exists

06

Calculating potential Firestore cost

Firestore cost

Cloud Firestore offers free quota that allows you to get started with your (default) database at no cost. The free quota amounts can be found on this link. Quotas are applied daily and reset around midnight Pacific time. Only the (default) database qualifies for the free quota.

Depending on amount of ecommerce events and users on your website, Ecom Attributor for sGTM solution can produce additional Firestore cost. Below you can find details how to calculate potential cost.

Firestore writes

The cost can be calculated based on events you use as trigger for **Ecom Attributor - Write to Firestore** tag.

For example, if you added following events as triggers: select_item, select_promotion and purchase events.

In that case, number of writes would be around the same count of select_item, select_promotion and purchase.

Find what is the average daily amount of those events for your website and calculate potential cost using Pricing table.

Firestore TTL policy deletes

Solution uses Firestore TTL policy to automatically delete data.

TTL delete operations count towards document delete cost..

This is bit harder to estimate (because if users return on your website daily, it is possible you will have less TTL deletes then you estimated here), but you can calculate by looking at how many users clicked on select_item or select_promotion events (in 7 day period, you selected to delete data after 7 days).

For delete operations pricing, see Cloud Firestore pricing table.

07

Frequently Asked Questions

Frequently Asked Questions

Does solution support custom ecommerce implementation?

- Solution supports all types of ecommerce measurement implementation (gtag.js, web GTM, third party tag management system, custom implementation). Solution will works as long as ecommerce data received in sGTM respects GA4 ecommerce event data model (event and parameters naming).

Does solution work with advanced Consent Mode?

- Solution doesn't work with unconsented pings. List and Promotion data is stored in Firestore based on hashed client_id, therefore it won't be possible to stitch List and Promotion data from unconsented pings with events after user gave the consent.

Which information is stored in Firestore if I select option "Item List Attribution"?

- Following parameters will be stored in Firestore: **item_list_id**, **item_list_name**, **index**, **location_id**. In case if you have promotion data inside the items array, following parameters will also be collected: **promotion_id**, **promotion_name**, **creative_name**, **creative_slot**. If certain parameters are not set, it simply won't be collected and stored in Firestore.

Which information is stored in Firestore if I select option "Promotion Attribution"?

- Following parameters will be stored in Firestore: **promotion_id**, **promotion_name**, **creative_name**, **creative_slot**. If certain parameters are not set, it simply won't be collected and stored in Firestore.

Will List/Promotion information be collected if it is not located inside items array, if it is actually located inside ecommerce object?

- Yes, solution supports this, information will still be collected. Following logic is applied: solution will always first look into the items array and pick if any information is available. In case if certain parameter is not available inside items array (e.g. item_list_name), solution will check inside ecommerce object and pick information if it is available.

Why I need to have Item ID on every ecommerce event that is fired on my website?

- Item ID is required on every ecommerce event (except promotions events) because this is the key how List information is stored in Firestore for every item. If you don't have Item ID on one of your ecommerce events (e.g. item ID missing on begin_checkout event), then List information won't be retrieved from Firestore and sent to GA4.

On which events solution supports collection of List data?

- Solution will support collection of List data (and Promotion data if case if it is available) on following events: **select_item**, **view_item** and **add_to_cart** event. Depending on which event you have List information, those events needs to be provided as triggers to the **Ecom Attributor - Write to Firestore** tag.

On which events solution supports collection of Promotion data?

- Solution will support collection of Promotion data (and List data if case if it is available) only on **select_promotion** event. This event needs to be provided as trigger to the **Ecom Attributor - Write to Firestore** tag.

What if I have only Promotion data on select_promotion event (no item data)?

- Solution supports this type of implementation as well. It will collect only Promotion data on **select_promotion** event, and Promotion data will be sent on event-level with any subsequent ecommerce events, which means on purchase event, whole Items revenue will be attributed to the last clicked Promotion.