Google

Google Analytics User-provided data: Implementation Guide

Maximize your First Party Signal coverage & Data Strength for Enhanced ROI

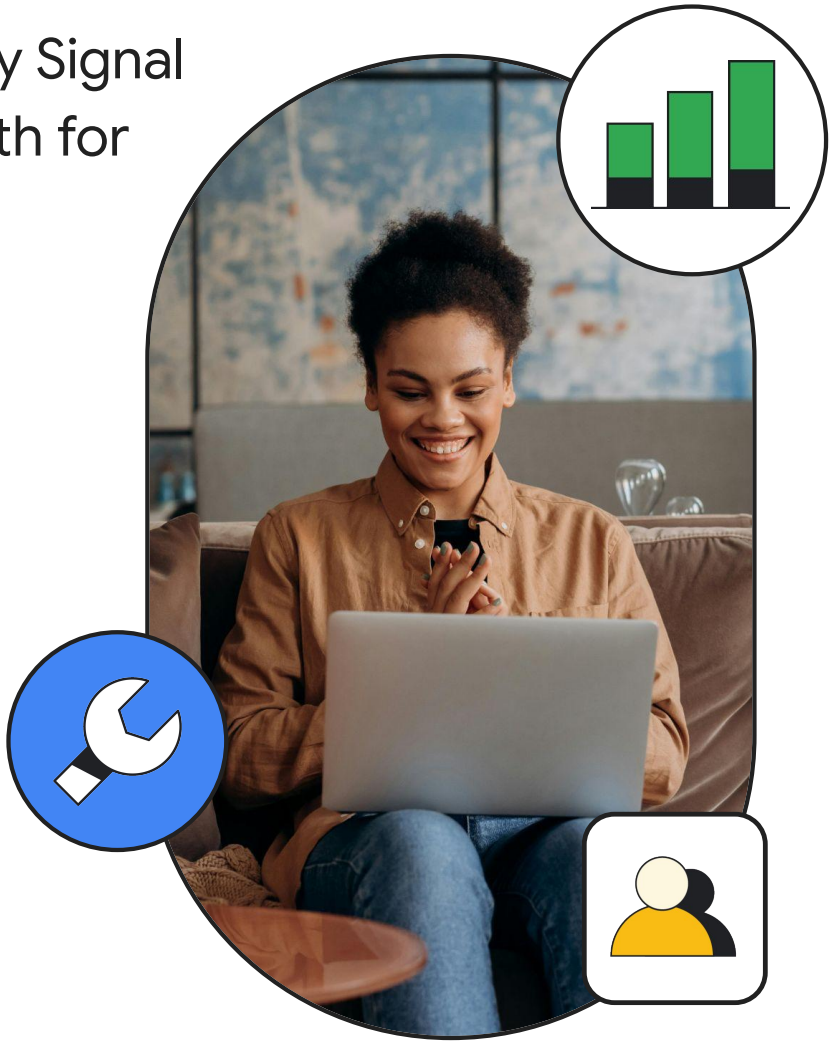
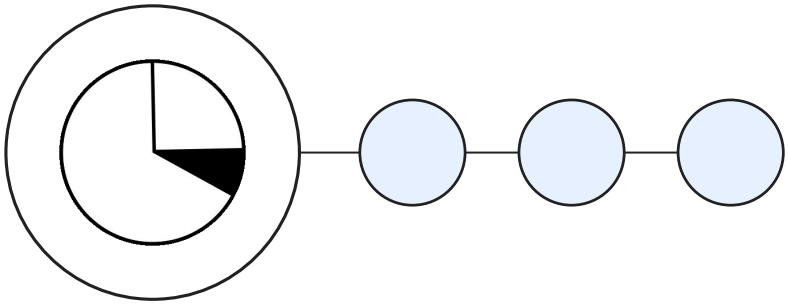


Table of contents

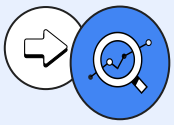
Contents



What is User-provided data & Benefits

Module I: What is User-provided data and Why First Party Data Matters Now

User-provided data(UPD) is hashed, consented customer data (e.g., email, phone, name, address) from a your website or app that is sent to **Google Analytics.** This data is safely matched with Google's signed-in user data, which enhances Google Analytics data used for conversion measurement, bidding and audience targeting.



Improved Bidding & Customer Match List

UPD works by filling in signal gaps to ensure more conversions and interactions are observed, attributed, and modeled, making your first-party data improve ROI and Bidding. In addition UPD improves match rates and expands existing 3P audience lists.

In a perfect world you could get a full view of your customers & conversions without building **Data Strength.** But **signal loss is causing blind spots** in your view of customers & conversions, degrading AI outcomes.

As a result, having the Google tag is the first step, but it is not enough to recover a complete view. Building Data Strength allows you to maximize the view of your customers & conversions

Data Strength is the comprehensive data setup that ensures the necessary input for Google AI to achieve the best ROI for your business.

To achieve Data Strength in Google Analytics and boost your ROI, you require the following three steps:

Step 1

Connect your data sources

Ensure Google Analytics is a comprehensive source of truth for the entire customer journey and advertising ecosystem by bringing in your business and marketing data.

Step 2

Maximize your Signals

Utilize a diverse set of signals to build the highest fidelity, most resilient data set that minimizes gaps and maximizes AI input quality.

Sending User ID, UPD, Transaction ID, UTMs, Recommended Events.

Step 3

Activate your data with Google AI

Once you have built your Data Strength, activate it to drive superior campaign performance and make confident cross-channel budgeting decisions.

Link your Google Analytics with Google Ads.

Prove your ROI with Modern Measurement

This guide focuses on Step 2. Maximizing your Signals after setting up your data sources and implementing GTG

Module II: Benefits of User-provided data

User-provided data feeds more accurate and complete conversion data into Google Ads, fueling smarter bidding (Smart Bidding, Performance Max) and more reliable conversion attribution.

01 UPD powers Enhanced Conversions within GA

Increased conversion observability

- Recover lost conversions due to browser and regulatory changes
- Observe net new conversions from additional 1P data

Better modeling and overall conversion reporting

Improved observability allows our conversion modeling to be more accurate

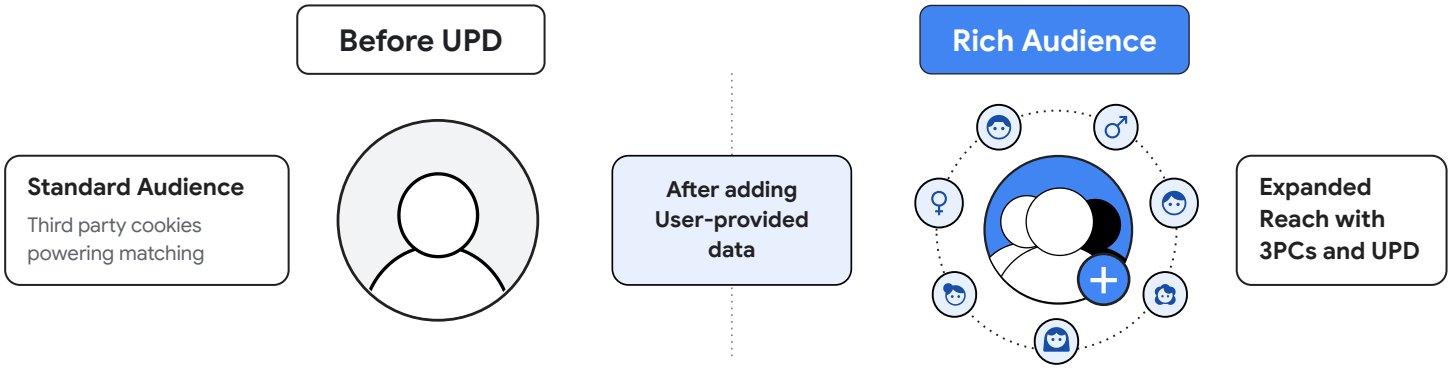
Improved bidding & attribution

Better data ingested by bidding models (tROAS / tCPA) and attribution models (DDA)

Improved Performance

- Better overall CVRs
- Higher ROAS / Lower CPA

02 UPD unlocks high-value Audience Activation with Customer Match

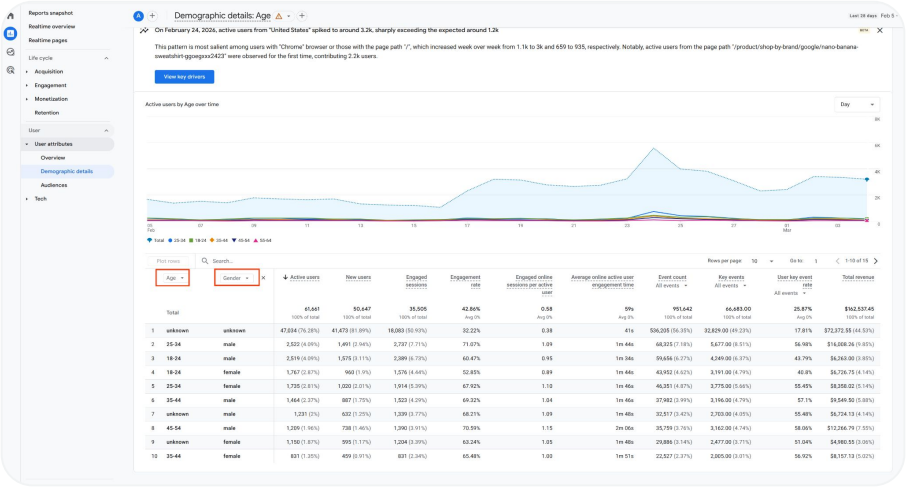


- **UPD improves match rates and expands existing 3P audience lists,** giving them more reach for Google targeting across platforms like Search and YouTube in addition to third party cookies
- **This makes your list size bigger and increases match rates** leading to even more improved ad relevance and campaign efficiency by targeting users based on your first-party data

03 Deeper Demographic & Interest Insights

UPD helps power more accurate and durable Demographics and Interests reporting within Google Analytics based on your first-party data matched with Google's data in addition to 3P cookies.

A **retailer** might find that their highest-converting segment is surprisingly **'Value Shoppers' aged 45-54, rather than the younger demographic** they were targeting.



04 [Future] Key identifier for holistic business measurement

The Modern Customer Journey is More Than Clicks

Customer Journey today

01 A customer **sees your ad and visits your website.**

02 Days later, **they call your support center** with questions.

03 Finally, **they make a purchase in your physical store.**

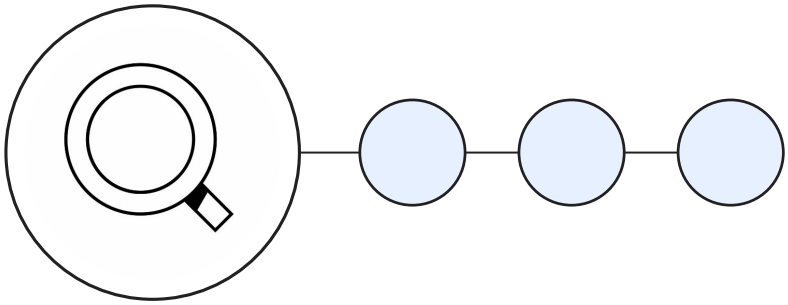
But today, connecting these offline events to your online marketing efforts is like trying to piece together a puzzle with missing pieces. This fragmented view means you're not seeing the full picture of what drives your business.

Bridging the Gap with User-provided data

In the the near future (H2 2026), UPD will serve as a **standalone join key**, allowing you to securely connect offline interactions back to online activities without relying on other identifiers.

Unlocking a Unified View of Your Business:

- **CRM Integration:** Bring in qualified lead events from your CRM (like phone consultations or contract signings) and link them back to the initial online submission to see which ads drive real intent.
- **POS Reconciliation:** Import sales and transaction data directly from your Point of Sale (POS) system. This reconciles your digital reporting with actual revenue, capturing offline conversions that browser tags simply cannot see.



Implementation

Module III: Choose your implementation platform

UPD Help Center acticle

I am..	Implementation method to consider	I am..	Pick 1 or more!
... a customer who only uses website tracking for online events	gTag	...a customer who wants minimal setup, but still collect UPD	**Automatic** (recommended to have turned on) Help center article
	Google Tag Manager	...a customer who has specific fields I want to specify on my webpage to collect UPD and expects minimal website changes	**Javascript/ CSS Variable (manual)** Help center article
.. an online customer seeking enhanced data control, privacy, or performance by routing and transforming data server-side before sending it to Google Analytics.	sGTM	...a customer who wants a solution that is resilient to future website changes	**Code** Help center article
... a customer who need to measure server-to-server or offline interactions, like integrating CRM data or capturing conversions from POS systems through APIs. And/or also have an app stream.	APIs	.. Customers who already use legacy MP API to send server-server events	**Measurement Protocol**
		...new API customers who wants better error messaging and support for batched imports.	**DM API** (coming Soon)

Module III: Getting Started

User-provided data feeds more accurate and complete conversion data into Google Ads, fueling smarter bidding (Smart Bidding, Performance Max) and more reliable conversion attribution.

Get started with User-provided data in 4 steps

User-provided data is **not available to the healthcare vertical and limited use for finance properties** as determined by the property-level industry category.

01

Accept User-provided Terms of Service

Before any User-provided data can be processed by Google Analytics, you must accept the User-provided data collection terms within the Google Analytics **Property settings from the Admin page**

02

Enable User-provided data in the Google Tag interface

The primary control for allowing User-provided data collection resides within the Google tag configuration, "**Allow User-provided data capabilities.**" This setting acts as a master switch and requires no effort because it is **enabled by default.**

03

Collect and Structure User-provided data

Update your tagging implementation to send User-provided data. This can be done via gTag, GTM, sGTM, or Measurement Protocol.

04

Validate implementation

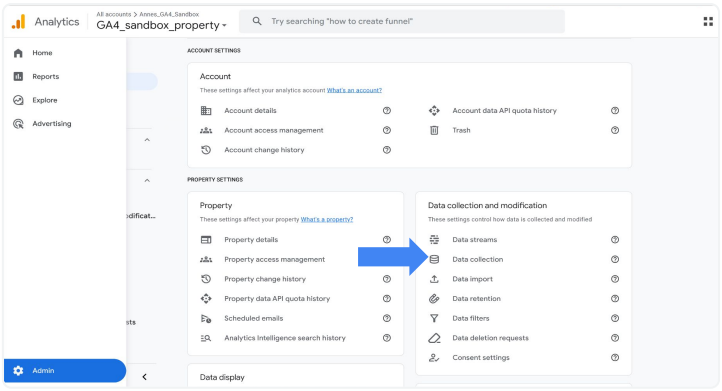
Customers can validate their User-provided data implementations using Chrome Developer Tools or Google Tag Assistant depending on implementation choice.

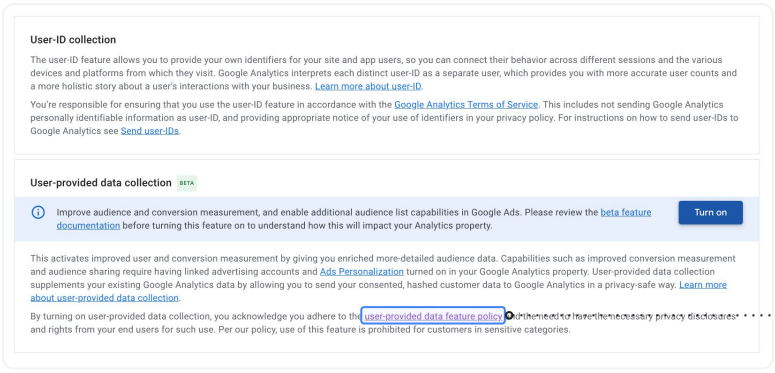
01 Accept Terms of Service for User-provided data

First step for User-provided data collection is accepting User-provided data Terms of Service and enabling UPD in GA Property settings. **Regardless of the implementation method (gtag, GTM or Measurement Protocol)**, the first step is to enable User-provided data toggle in GA UI.

Steps

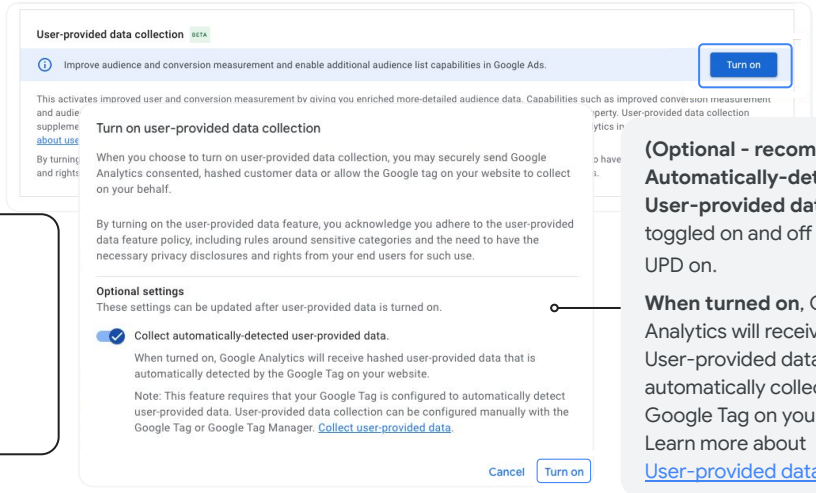
1. In your Google Analytics account, click **Admin**
2. **Data Collection**





3. Click **“Turn On”**
4. Review the Google Analytics User-provided-data collection for User-ID Policy.

5. Enable User-provided data automatic collection



(Optional - recommended) Automatically-detected User-provided data, can be toggled on and off after turning UPD on.

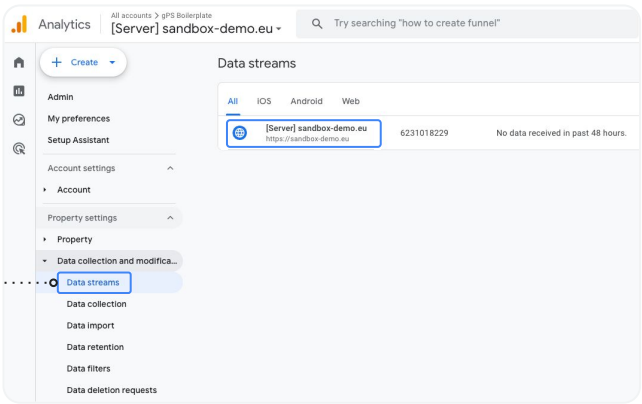
When turned on, Google Analytics will receive hashed, User-provided data that is automatically collected by the Google Tag on your website. Learn more about User-provided data collection.

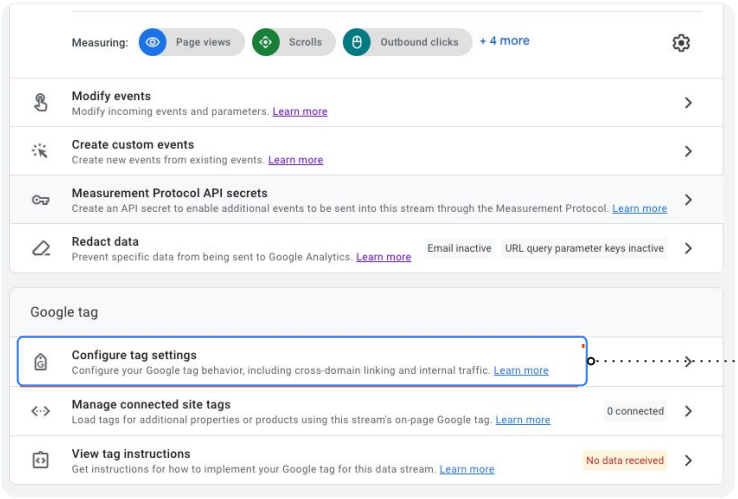
02 Enable User-provided data in the Google Tag Configuration

UPD collection is controlled within the **Google tag settings**, which acts as the **master switch**, even for GTM implementations. **Google tag's UPD** setting must be **ON** for User-provided data to be collected from any source (Google Analytics events, GTM tags).

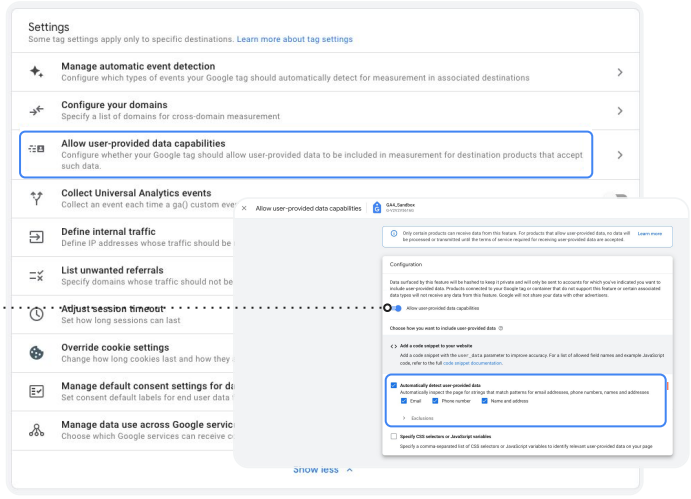
To exclude **UPD for particular events in GTM**, configure those specific event tags to send an empty user_data object (e.g., user_data: {})

1. In Admin, under Data collection and modification, click **Data streams**
2. Select the desired **web data stream**[1]





3. Under **Google tag**, click **Configure tag settings**



1. Click on **“Allow User-provided data capabilities”**
2. Enable **“Allow User-provided data capabilities”** if it is already not enabled.
   Note that this setting is enabled by default.

If you **turn OFF** “Allow User-provided data capabilities,” this will stop the collection of User-provided data, but will not erase any previously collected data. This will be true for any **Google tag** created data AND **Google Tag Manager - Google Analytics event tags**.

NOTE: In case if you are not planning to use the feature to automatically collect User-provided data from your website, uncheck the checkbox next to automatically detect User-provided data.

Google

[1] Web+App Streams are also supported. UPD for Apps can be ingested through Measurement Protocol API only. App only streams do not have UPD support at this time.

03

Collect and Structure User-provided data

Gtag.js

Select this option if you currently use **gtag.js** to collect data from your website

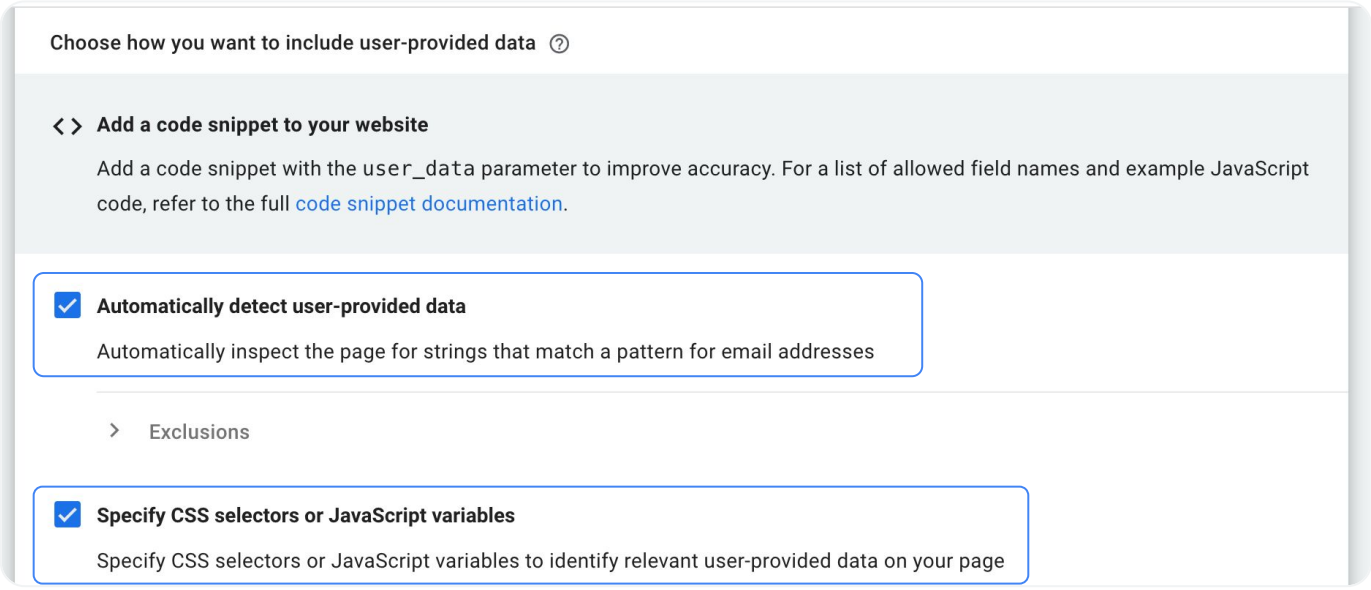
Help Center guide

01

Automatic and CSS Selector configuration for gtag.js

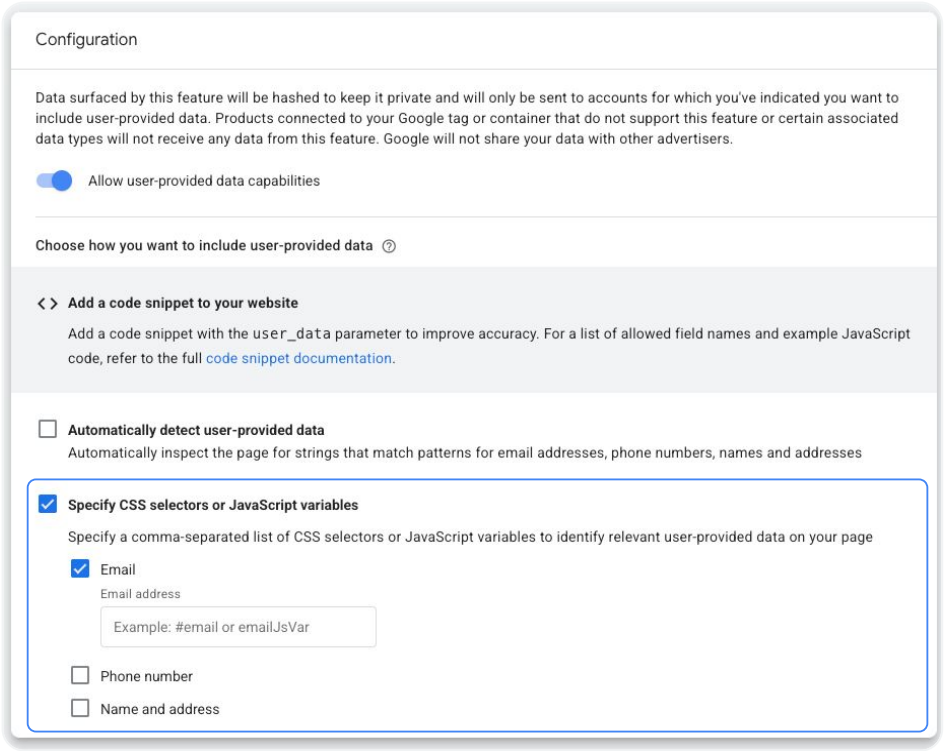
Enable Auto and CSS Selector in Google Analytic's Google Tag settings

To enable Auto, select **Automatically detect User-provided data. We recommend enabling both Auto and CSS Selector implementations.**



02 Specify CSS Selectors or Javascript Variables

- Select **Specify CSS selectors or Javascript variables** to set up a variable for email, phone number or name and address.
  - **Note:** The Javascript variable must be Global
- More than one selector or JS variable can be specified using a ‘,’ delimiter.
- It is recommended to use existing Global Javascript variables when possible



Pro Tip

1. **Tip:** Ensure the CSS selector’s element ID is unique, for example if there are two email tags with same identifier, it may not ingest the correct email.

Note - Considerations for CSS Selector vs. Javascript variables:

CSS Selectors:

1. Only works when the 1P customer data is visible on the page
2. This option is more unstable, can easily break due to website changes
3. Easier to implement
4. **Tip:** Ensure the CSS selector’s element ID is unique, for example if there are two email tags with same identifier, it may not ingest the correct email.

Javascript Variables (recommended)

1. Works even if 1P data is not visible on the page
2. More stable, less like to break due to website changes
3. Requires more technical expertise

03 Code: Custom user_data variable in gtag.js

Identify and define your User-provided data fields

Ensure the fields you need like email, address, and phone number are available on the conversion page where the **gtag.js** event snippet fires.

Note: At least one of the following fields must be provided:

- Email (preferred)
- Address (first name, last name, postal code, and country are required when using this option). You can optionally provide street address, city, and region as additional match keys.
- A phone number can also be provided along with an email or full name and address

You can either send unhashed data, which Google will normalize and hash before the data reaches the servers, or normalized and hashed data.

For normalization:

- Remove leading or trailing whitespaces
- Convert the text to lowercase
- Format phone numbers according to the E.164 standard

For hash:

- Use hex SHA256

Data Field	Key Name	Description	Example
Email Address	`email`	User email.	jdoe@example.com
	`sha256_email_address`	Hashed user email.	a4199369cd4e0e5a5a4b5d48c5754
Phone number	`Phone_number`	User phone number. Must be in E.164 format, which means it must be 11 to 15 digits including a plus sign (+) prefix and country code with no dashes, parentheses, or spaces.	+11231234567
	`sha256_phone_number`	Hashed user phone number.	e9d3ee9a3b19820f92696be53d646
First name	`address.first_name`	User first name.	John
	`address.sha256_first_name`	Hashed user first name.	96d363564cc3032521409cf22a85f
Surname	`address.last_name`	User last name.	Doe
	`address.sha256_last_name`	Hashed user last name	f92a11af918e3fb741df42934f3b58
Street address	`address.street`	User street address	123 New Rd
City	`address.city`	User city name	Southampton
Region	`address.region`	User province, state, or region	Hampshire
Postal Code	`address.postal_code`	User post code	SO99 9XX
Country	`address.country`	User country code. Use 2-letter country codes, per the ISO 3166-1 alpha-2 standard.	UK

Example of gtag.js code

The following sample includes variable names for the customer data. Make sure that the variable names in your code match the attributes on your website. For example, if you store email in a variable named “email_address”, then you should update the snippet with that name (for example, where it says emailVariable). You can provide multiple values (up to 3 phone numbers and email addresses, and 2 addresses) using an array.

You can also hardcode the field with a string or use a function instead of using variables. Learn More.

Example of gtag.js code with hashed data

```
<script>
gtag('set', 'user_data', {
    email: emailVariable,
    phone_number: phoneVariable,
    address: {
        first_name: firstNameVariable,
        last_name: lastNameVariable,
        street: streetNameVariable,
        city: cityVariable,
        region: regionVariable,
        postal_code: postalCodeVariable,
        country: countryVariable
        }
});
</script>
```

```
<script>
gtag('set', 'user_data', {
    sha256_email_address: hashedEmailVariable,
    sha256_phone_number: hashedPhoneVariable,
    address: {
        sha256_first_name:
        hashedFirstNameVariable,
        sha256_last_name: hashedLastNameVariable,
        street: streetNameVariable,
        city: cityVariable,
        Region: regionVariable,
        postal_code: postalCodeVariable,
        country: countryVariable
        }
});
</script>
```

Implementation validation (gtag.js)

Different Validation Methods

01 Available

Tag Assistant

Review outgoing requests and appended user signals via Tag Assistant UI.

02 Available

Developer Console

Review outgoing requests and key value pairs via browsers developer console capabilities.

03 Limited

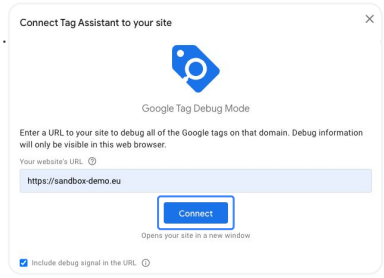
UI Diagnostics

See how many data streams in your Google Analytics property are receiving UPD signals.

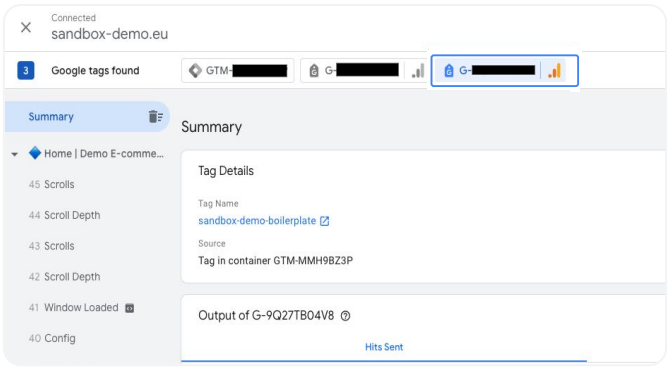
01 Tag Assistant/GTM

01 Open tagassistant.google.com

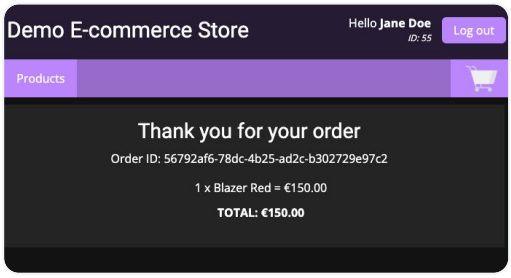
02 Type your domain, and click on **Connect** and the screen will then show connected



03 On top, click on the Measurement ID from your Google tag, in order to see hits that are sent to your GA endpoint



04 Perform action/trigger event on website (e.g. submit the form, purchase, newsletter subscription) which you are using to send User-provided data to GA

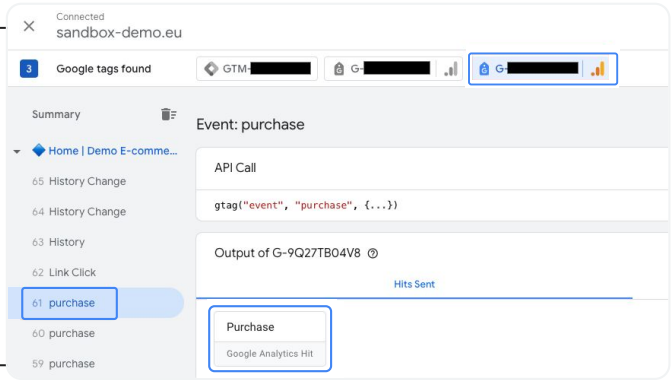


Review the sent hit in Tag Assistant

05 On the left side, click on the event that should carry User-provided data

06 Check if there is hit sent to Google Analytics

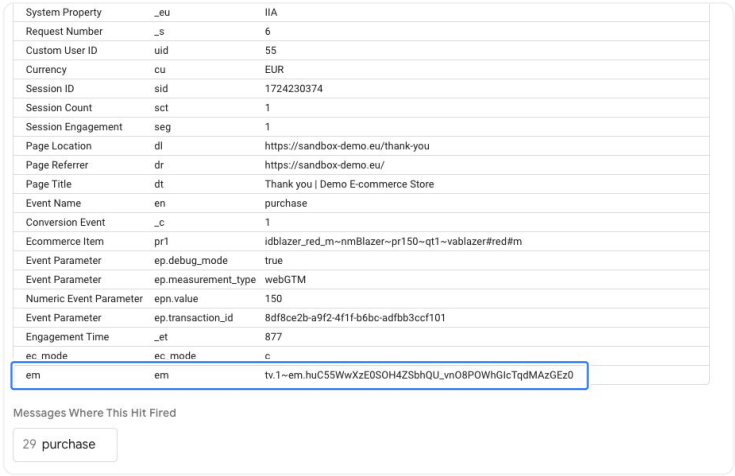
07 Click on the Google Analytics hit - **See next slide**



Google

5. Check User-provided data values

- In the Hit Details, you will see the list of parameters being sent to Google Analytics
- Scroll down to the bottom, here you should see **“em”** parameter
- Look for a parameter **“em”** with a hashed string as the value. The value should start with **“tv.1~em”** followed by a long string of characters.
- If you see the **"em"** parameter, this means that the tag is picking up, hashing and sending user provided data to Google Analytics endpoint



System Property	_eu	IIA
Request Number	_s	6
Custom User ID	uid	55
Currency	cu	EUR
Session ID	sid	1724230374
Session Count	sct	1
Session Engagement	seg	1
Page Location	dl	https://sandbox-demo.eu/thank-you
Page Referrer	dr	https://sandbox-demo.eu/
Page Title	dt	Thank you | Demo E-commerce Store
Event Name	en	purchase
Conversion Event	_c	1
Ecommerce Item	pr1	idblazer_red_m~nmBlazer~pr150~qt1~vablazer#red#m
Event Parameter	ep.debug_mode	true
Event Parameter	ep.measurement_type	webGTM
Numeric Event Parameter	epn.value	150
Event Parameter	ep.transaction_id	8df8ce2b-a9f2-4f1f-b6bc-adfbb3ccf101
Engagement Time	_et	877
ec_mode	ec_mode	c
em	em	tv.1~em.huC55WwXzE0SOH4ZSbhQU_vnO8POWhGIcTqdMAzGEz0

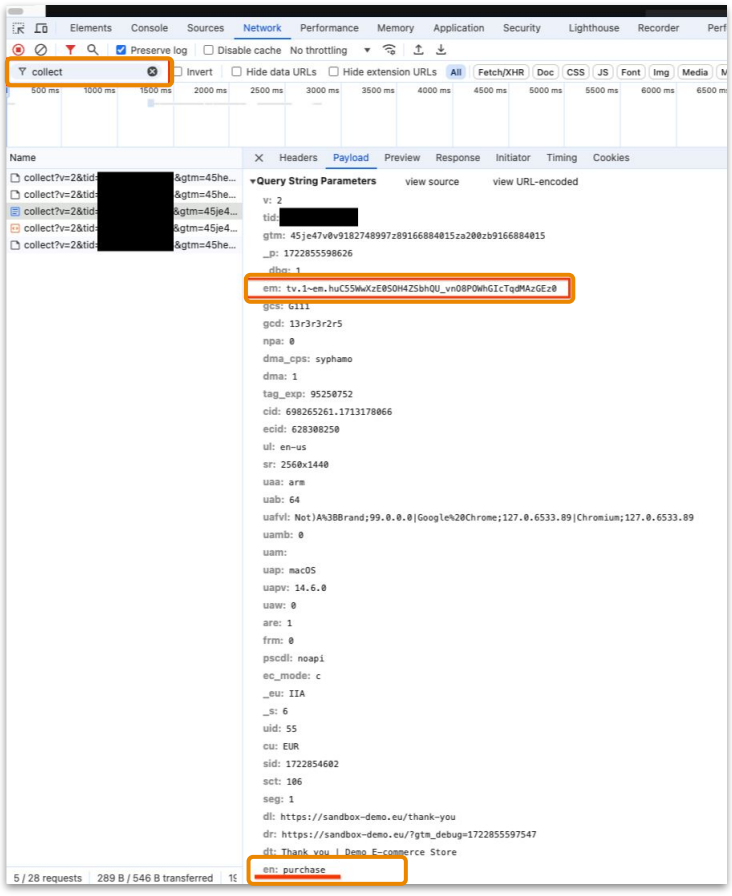
Messages Where This Hit Fired

29 purchase

02 Developer Console gtag.js

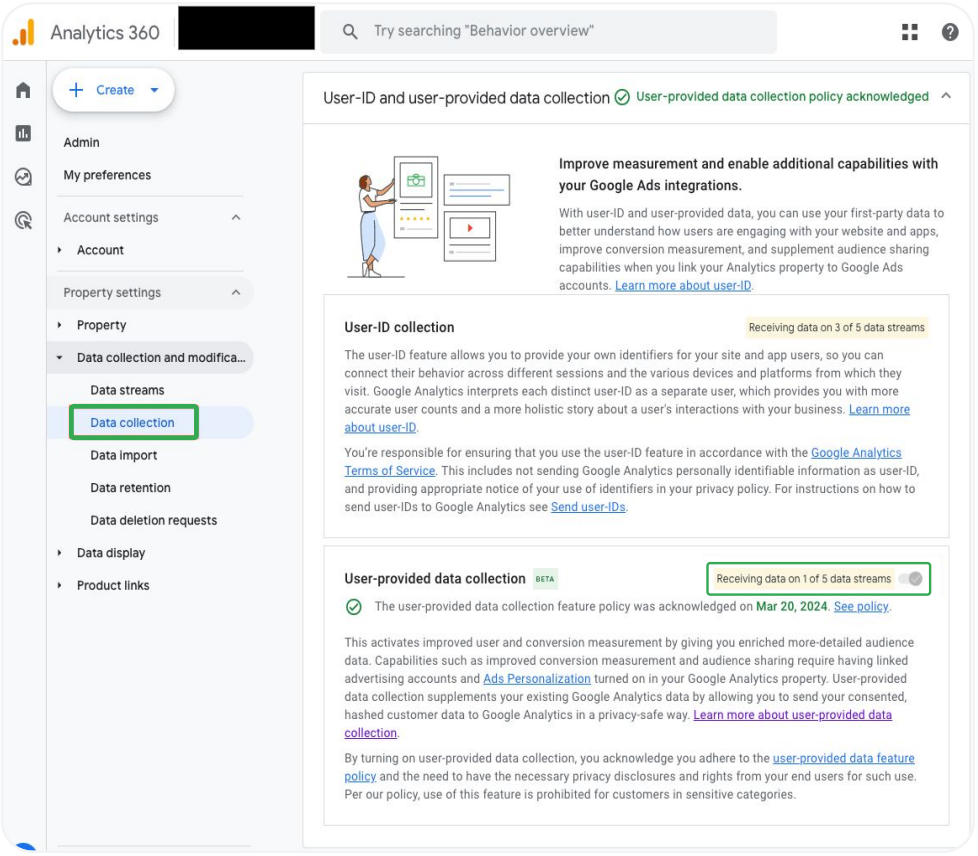
Review outgoing requests and key value pairs via browsers developer console capabilities.

- Right click on your web page
- Select **Inspect**
- Select the “**Network**” tab
- Enter “**collect**” in the search bar
- Selet Payload
- In the Payload details, you will see the list of parameters being sent to Google Analytics
- Scroll down to the bottom, here you should see **“em”** parameter
- Look for a parameter **“em”** with a hashed string as the value. The value should start with **“tv.1~em”** followed by a long string of characters. If you see the **"em"** parameter without the long string of characters, it will have an error code indicating we are not getting hashed user data



03 UI Diagnostics

See how many data streams in your Google Analytics property are receiving UPD signals.



- You can also validate in UI if User-provided data is collected
- Navigate to Admin section, click on **Data collection** and expand the section **User-ID and User-provided data collection**
- Here you should see message if one of your data streams is receiving User-provided data in last 48 hours

NOTE: This message won't be updated immediately. It can take few days before message is updated

Collect & Structure User Data - GTM Web Container

Google Tag Manager

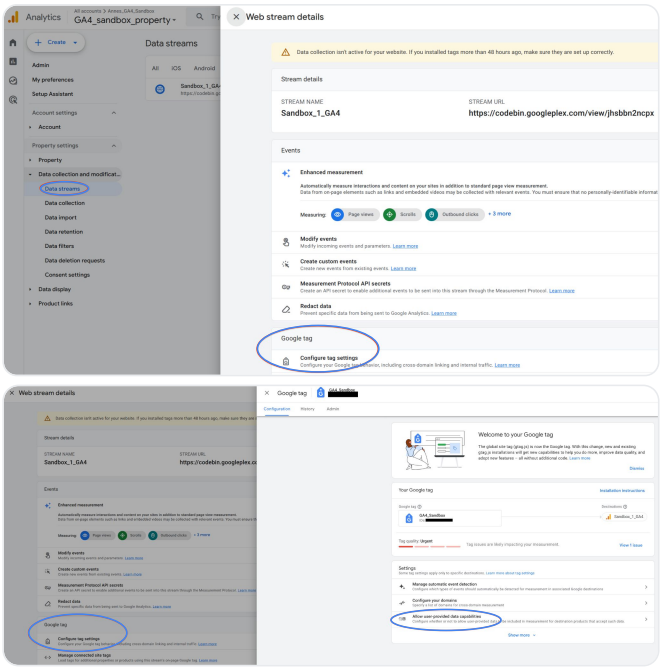
Select this option if you currently use client-side Google Tag Manager to collect data from your website

Help Center guide

Important Note: Allow User-provided data capabilities on Google Tag Settings

Important Note

- UPD collection is controlled within the Google tag settings, which acts as the master switch, even for GTM implementations.
- "Allow User-provided data capabilities" toggle on Google tag settings **MUST be ON collect** User-provided data from any source (Google Analytics events, GTM tags).
- To exclude UPD for particular events in GTM, configure those specific event tags to send an empty **user_data** object (e.g., **user_data: {}**).
- Event tag level settings and parameters will override Google tag settings or other variable settings



Other Key Definitions and Considerations

- **Google Analytics Property Settings** - UPD toggle must always be on for GA to collect UPD from Google Tag. The toggle can be turned on and off, but if disabled, we will stop processing UPD
- **Google Tag container settings** - central place to manage core data collection settings. For UPD, the *"Allow User-provided data capabilities"* toggle (the master switch) must be on. If turned off, no UPD is sent to GA even if available on event tags from GTM
- **Google Tag (Google tag configuration)** - This tag is responsible for loading the entire configuration for that Google tag ID onto your page.
- **User Provided Data Variable** - user-defined variable type to store and format consented first-party user data for Google's measurement products. Required for code based UPD collection and CSS selector.
- **GA Events Tag** - A tag to instruct UPD to be sent on event level.
- **Google tag: Event Settings variable-** you can use this in Google tag type under shared events settings to send User-provided data for all events without the need of adding user_data variable per event or under events tag to control events settings from a central place

Sending User-provided data through Google Tag Manager Flow Chart

Step 1:
Create
User-Provided
Data Variable

Automatic collection

Manual/CSS Collection

Code based collection

Select one or more implementation option for the user-provided data variable

Step 2

Path 1
Events parameter path

Google Analytics: Google Analytics Event tag

- Create or use existing event tag
- Under events parameter add **user_data**
- For user_data **value**, select the **User-provided data** variable created in **Step 1**

Target user - you prefer to add **user_data** as a parameter to each event individually

Recommended: Method for users who want to use user_data parameter on event tag

Path 2
Events settings path

Create GA Events Setting Variable

Add **user_data** as event parameter with value of created user-provided data variable

Pro Tip
Always include user_data on every event or you can use events settings variable

[Recommended]

Google Analytics: Google Analytics Event tag

- Create an event tag
- Add the events settings variable on the event tag

Target user - a user who wants to control which events collects **User Provided Data** with event setting control.

NOTE: This is the recommended method and ensures that User-provided data is collected on every event tag.

Google Tag Shared Events Settings

- You can add **events settings variable** on Google tag (configuration) tag under shared events settings variable

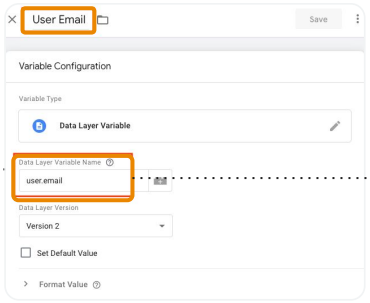
Target user - select this option if **User-provided data** is available on the page at the Google tag config load.

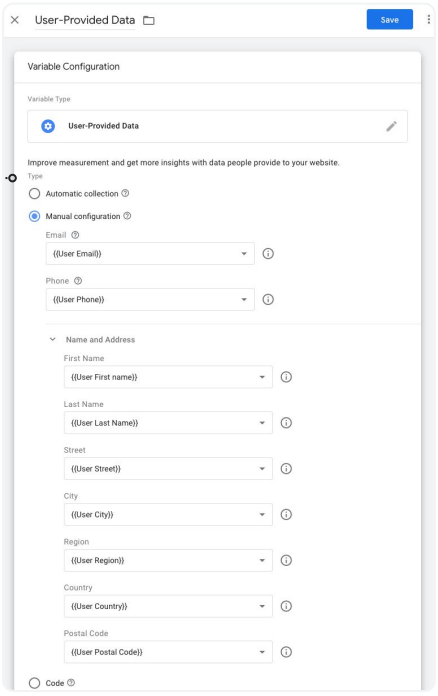
NOTE: This method is not recommended to be used alone. Because **User-provided data** might be available early on. Or there could be no Google Tag on the next page.

Step 1: Create User-provided data variable continued..

Option 2: Manual Configuration: CSS Selector/Javascript Variable

```
dataLayer.push({
  user: {
      id: "55",
      name: "Jane Doe",
      email: "jane.doe@example.com"
      }
});
```



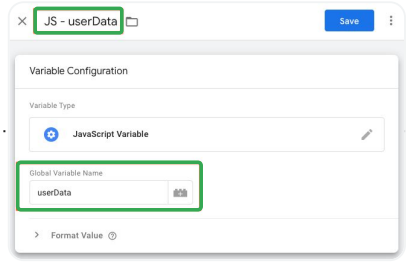


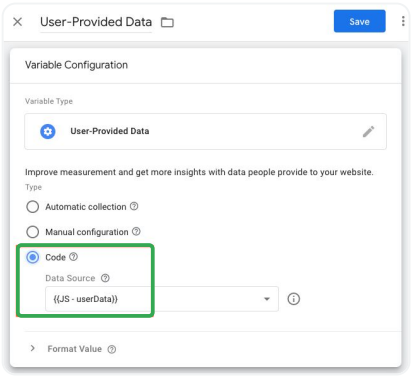
- Manually specify variables where User-provided data should be extracted from.
- Use this option in case if you have user information available in Data Layer or you want to manually create variables to extract information using CSS selectors.
- Add variables that collects corresponding values to each field you want to collect. For example, in Email dropdown, add variable that collects user email from your website.
- After you are done, click on **Save** button.

Option 3: Code Snippet

Example of variable that contains unhashed data

```
<script>
var userData = {
    email: "jane.doe@example.com",
    phone_number: "+11231234567",
    address: {
        first_name: "john",
        last_name: "doe",
        street: "123 new rd",
        city: "Southampton",
        region: "hampshire",
        postal_code: "12345",
        country: "uk"
        }
});
</script>
```





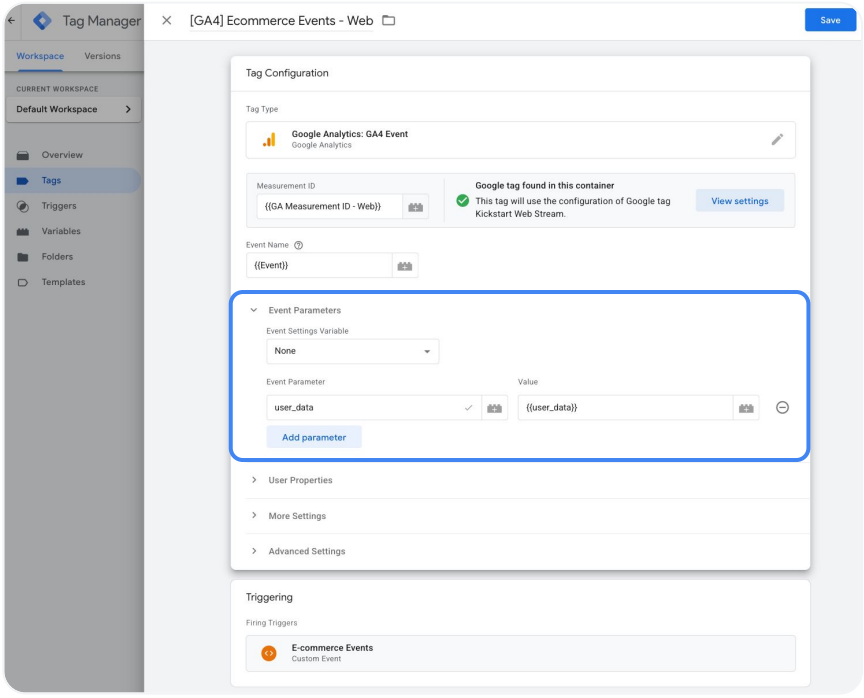
- Select a variable that will return structured User-provided data. You can check here to see how structured User-provided data should look like.
- Use this option in case if you have user information available as an object with User-provided data
- Select variable from **Data Source** dropdown menu. After you are done, click on Save button.

Step 2: Assign User-provided data in two Paths

Path One: On Events Parameter on GA Event Tag

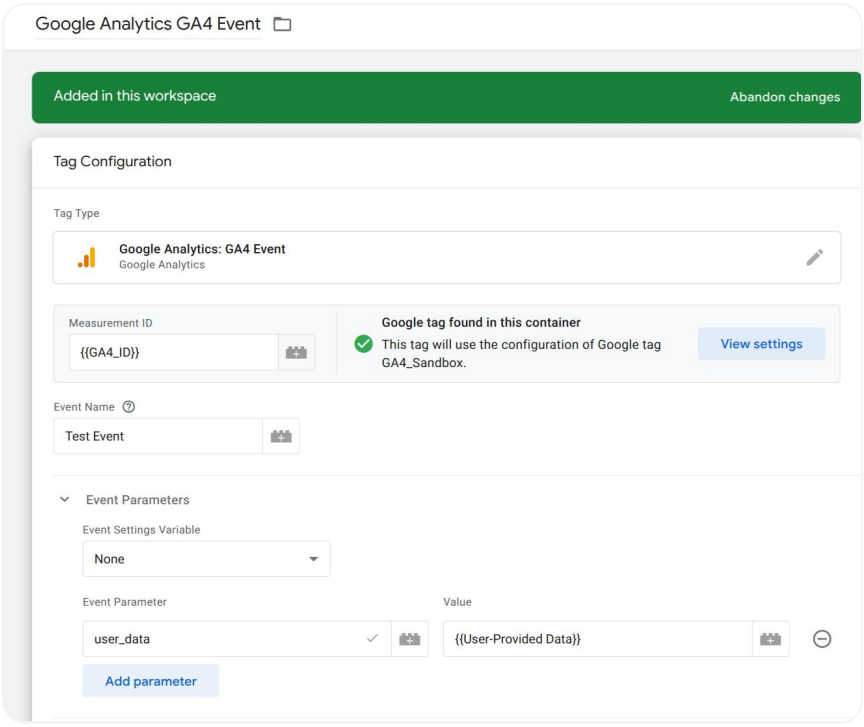
Instructions

- Sign in to Google Tag Manager
- From the Workspace tab, click **Tags** on the left
- Open the **“Google Analytics: GA4 Event”** tag where you want to collect customer data
- Expand the Event Parameters section and click **Add parameter**.
- Enter user_data in the "Event Parameter" field.
- In the **"Value"** field, the **User-provided data variable** created in Step 1



Review the changes

After creating and saving User-provided data variable in GTM, this is how GA Event tag should look like. Save the tag changes.

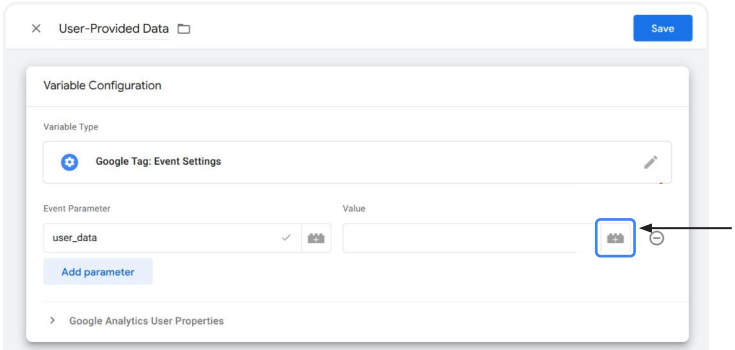


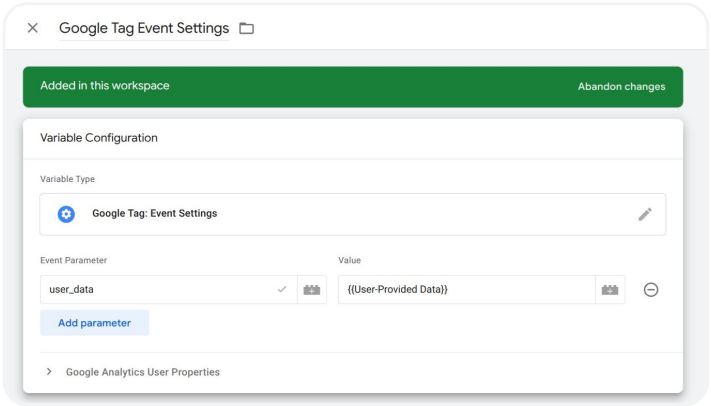
Step 2: Assign User-provided data in two paths continued...

Path Two: Through Events Settings Variable

Instructions

- Sign in to Google Tag Manager
- From the Workspace tab, click **Variables** on the left
- Click new and search for **Google Tag: Event Settings**
- Expand the Event Parameters section and click **Add parameter**.
- Enter **user_data** in the "Event Parameter" field.
- In the **"Value"** field, the **User-provided data variable** created in **Step 1**



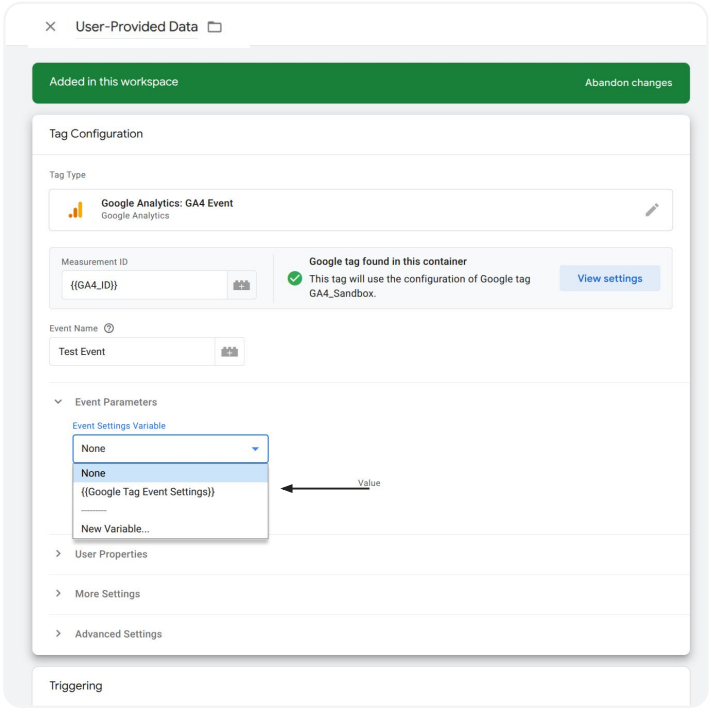


Now add the Events Settings Variable to GA4 Event tag or Google Tag or both

1. Google Analytics Events Tag (Recommended)

Instructions

- Sign in to Google Tag Manager
- From the Workspace tab, click **Tags** on the left
- Open the **"Google Analytics: GA Event"** tag where you want to collect user data
- Expand the **Event Parameters section**
- Under **Events Settings variable**, click the dropdown and select the Events Setting variable you just created



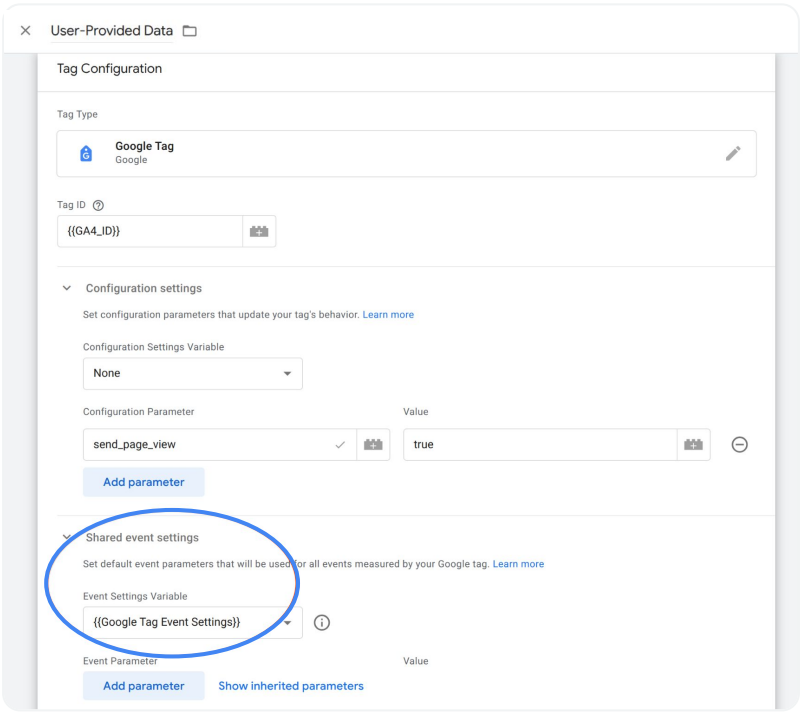
This is the most recommended method to ensure that User-provided data is available at the time event is submitted and persisted across sessions

Add Events Settings Variable to:

2. Google Tag

Instructions

- Sign in to Google Tag Manager
- From the Workspace tab, click **Tags** on the left
- Open the **“Google tag”**
- Expand **“Configuration settings”** and then **“Shared Events settings”**
- Under **Events Settings variable**, click the dropdown and select the Events Setting variable you just created

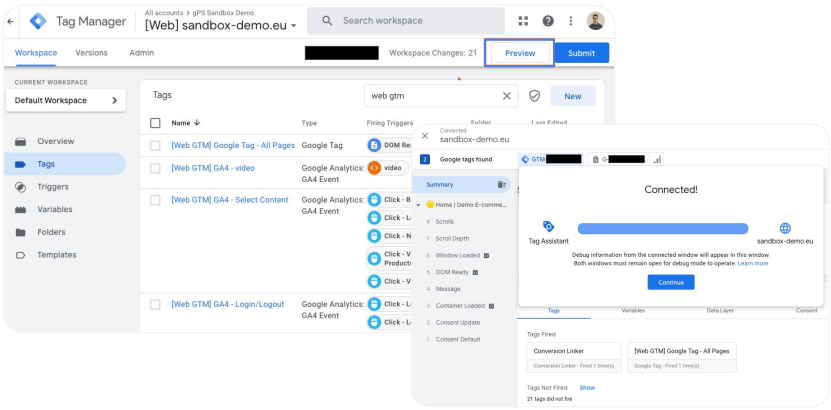


Select this option if User-provided data is available on the page at the Google tag config load.

Note: This method is not recommended to be used alone. Because User-provided data might not be available early on the page. Or there could be no Google Tag on the next page. We recommend adding the events settings variable here in conjunction to the Event Tag.

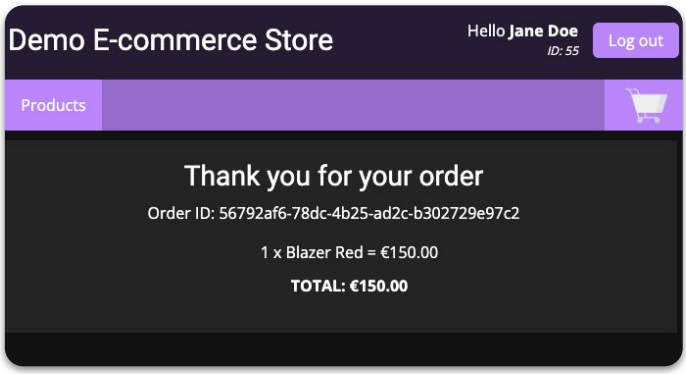
Implementation Validation: Client-side GTM

1. Open Tag Assistant



Go to your client-side GTM container (tagmanager.google.com) and click on **Preview** button.

NOTE: In case if you have sGTM implemented, we still recommend to validate if User-provided data is sent to sGTM endpoint. And in later steps you can find how to validate implementation via sGTM Preview Mode if data is actually sent to Google Analytics.

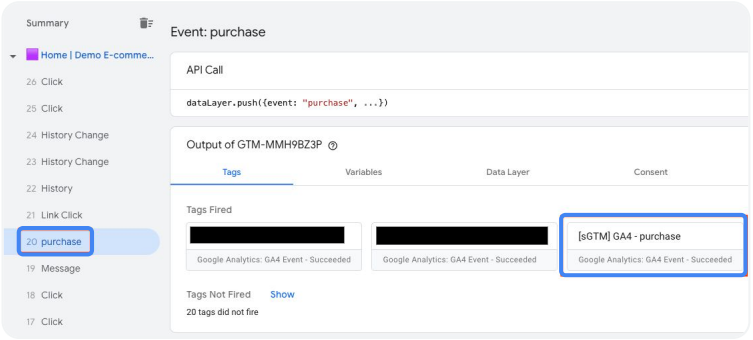


2. Perform test conversion (key event)

Perform **action/trigger event on website** (e.g. submit the form, purchase, newsletter subscription) which you are using to send User-provided data to GA

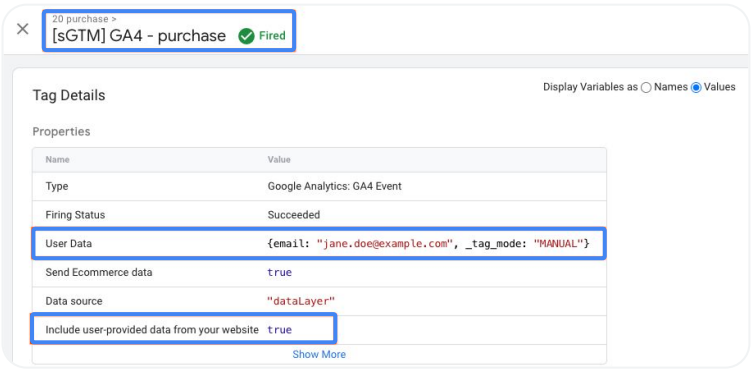
3. Check if the the key event was sent

Check in Tag Assistant if **Google Analytics** event tag was triggered successfully on event



4. Review User data parameter

- Click on the **GA event tag**
- Choose to display variables as **“Values”**
- Review the **User-provided data** to verify that the data was correctly captured
- **_tag_mode** shows you the method how you capture User-provided data from your website (automatic, manual (aka CSS selector/JS variable), code)



Collect & Structure User Data - Server-side tagging

Server Side GTM

User-provided data is fully compatible with server-side GTM

Help Center guide

4 Steps for User-provided data (UPD) Implementation via Server-Side Container

(This assumes your server-side environment is set up with the necessary GA Client and Triggers. For sGTM setup details, see this Developer Article.)

If you are already configured your implementation to send events data to sGTM endpoint and have UPD configured on Web Container, you don't have to configure anything in sGTM Container to collect User-provided data

01

Collect User Data in the Web Container:

First, you need to capture the User-provided data on the client side (your website) and format it. More information on Google Tag Manager implementation section. Refer to GTM section.

02

Set up and Send Data to Server Container:

Configure your Google Tag or Google Analytics in the Web Container to send data to your sGTM container URL. This routes the events, including the user_data, to your server. This is done adding this.

03

Process Data in the Server Container

No changes needs to be done if you already have sGTM configured. But if not set up your sGTM container with the right clients and tags.

04

Validate your implementation

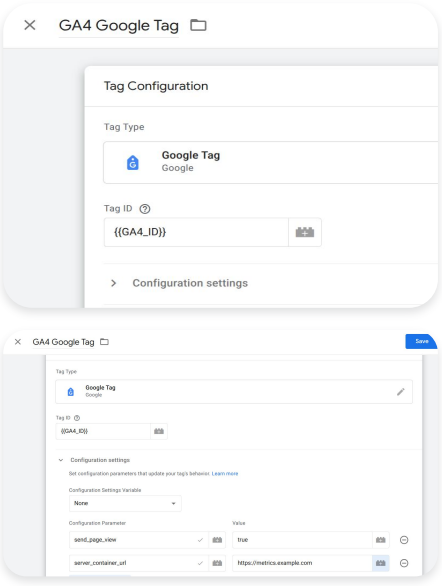
Use GTM Preview mode and Google Analytics's DebugView to confirm everything is working correctly using your Web and Server Container

Step 2. Set up and Send User Data from Web Container to Server Container:

Sending User-provided data to sGTM

1. Open your **Web Container**
2. Go to Tags: Click on **Tags** in the left-hand menu.
3. Select your **Google Tag**
4. Click on **Configuration Settings**
5. Under **Configurations Parameter,** click "Add parameter."
6. Set the Configuration Parameter field to ***server_container_url.***
7. In the **Value** field, enter the full URL of your tagging server endpoint. This is the URL you configured when setting up your sGTM environment (e.g., ***https://metrics.example.com*** if using a custom domain, or the URL provided by your cloud provider like *https://your-project.a.run.app*).

By adding the server_container_url parameter, you are instructing the Google Tag running in the user's browser to send all measurement beacons (Google Analytics events, etc.) to your specified sGTM endpoint instead of sending them directly to Google's collection servers. Your sGTM container will then receive these requests, process them, and forward the data to Google Analytics, Google Ads, etc., based on the tags configured within the server container.



Google

Collect & Structure User Data - Server-side tagging

Exclude User-provided data from being sent an endpoint

Optional:
You can Exclude User-provided data from sending it to specific Google endpoint

Help Center guide

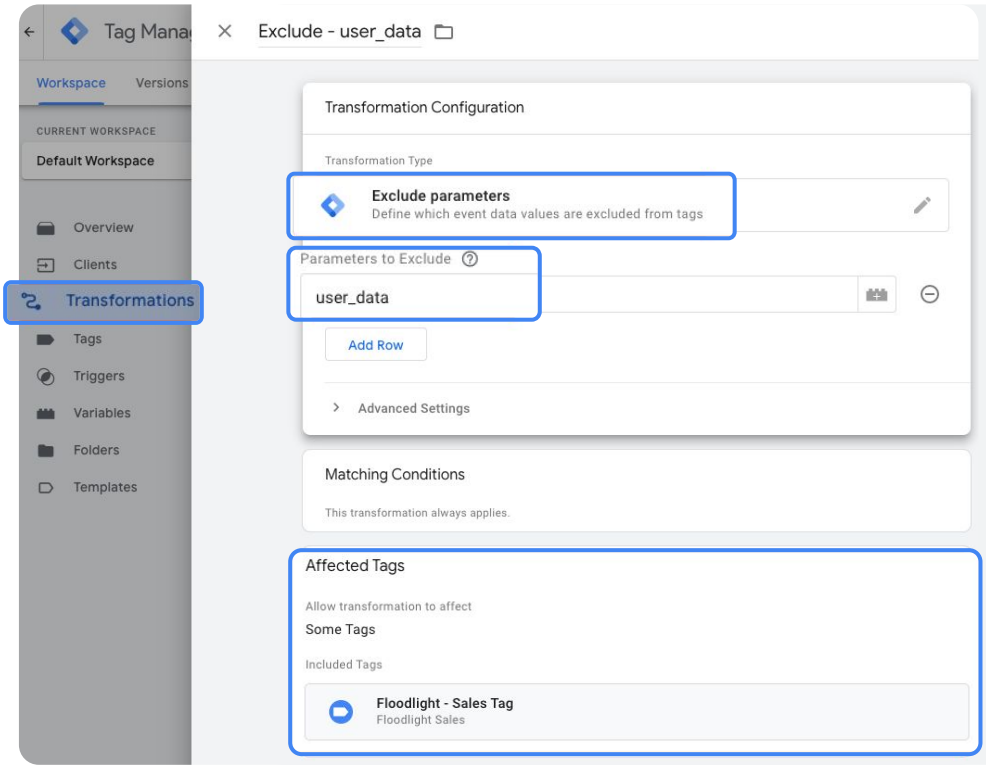
When user_data is sent to sGTM endpoint, Google tags (GA, Google Ads, Floodlight) will automatically pick data and forward to respective endpoint.

In case if you don't want to send User-provided data to specific endpoint, you can set this up using Transformations in sGTM.

To do this, in sGTM:

- Click on Transformations in the left side menu
- Click on **New** button, and select **Exclude parameters**
- Add row, and provide **user_data** parameter in the row
- Under **Affected Tags,** specify on which tags you want to exclude User-provided data

In our example, this Transformation is included in Floodlight tag, which means User-provided data (user_data parameter) won't be sent to Floodlight.



Pro Tip

2. **Tip:** If you do not want to send user_data to a given endpoint. Use Transformations

Implementation Validation - Server-side GTM

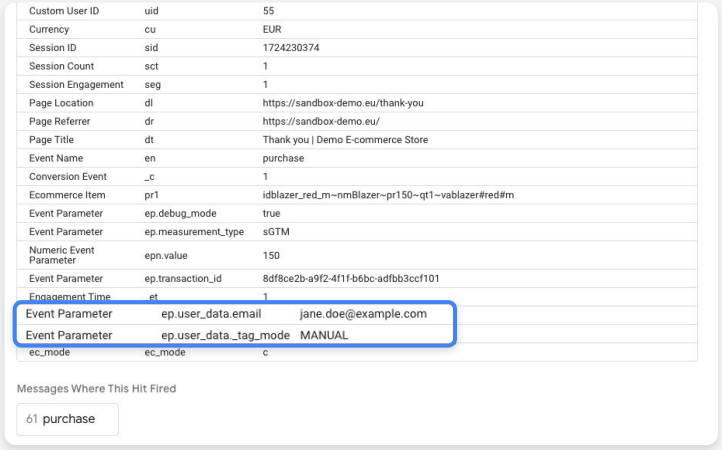
01 Tag Assistant/GTM

Use GTM Preview Mode for Both Web and Server Containers:

- Open Google Tag Manager.
- Enable Preview mode for your **Web Container**.
- In a separate tab, enable Preview mode for your **Server Container**.
- Connect Tag Assistant to your website.

Trigger the UPD Event on Your Website:

- Perform the action that sends UPD (e.g., submit a form with test email, phone number).



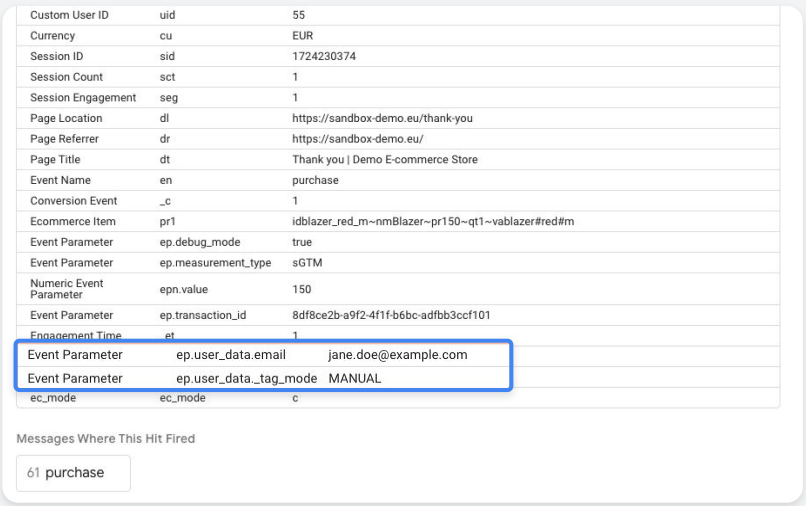
Validate in the Web Container Tag Assistant:

- Select the event in Tag Assistant (e.g., purchase).
- Check the Variables tab to ensure your User-provided data variable is capturing the data correctly from the page.
- Select the Google Tag or Google Analytics Event Tag that sends data to sGTM.
- In the HTTP Request shown in the tag details, verify that the user_data object is included in the payload being sent to your server container URL. At this stage, the data will be unhashed if you haven't pre-hashed it on the client side.

Check User-provided data values

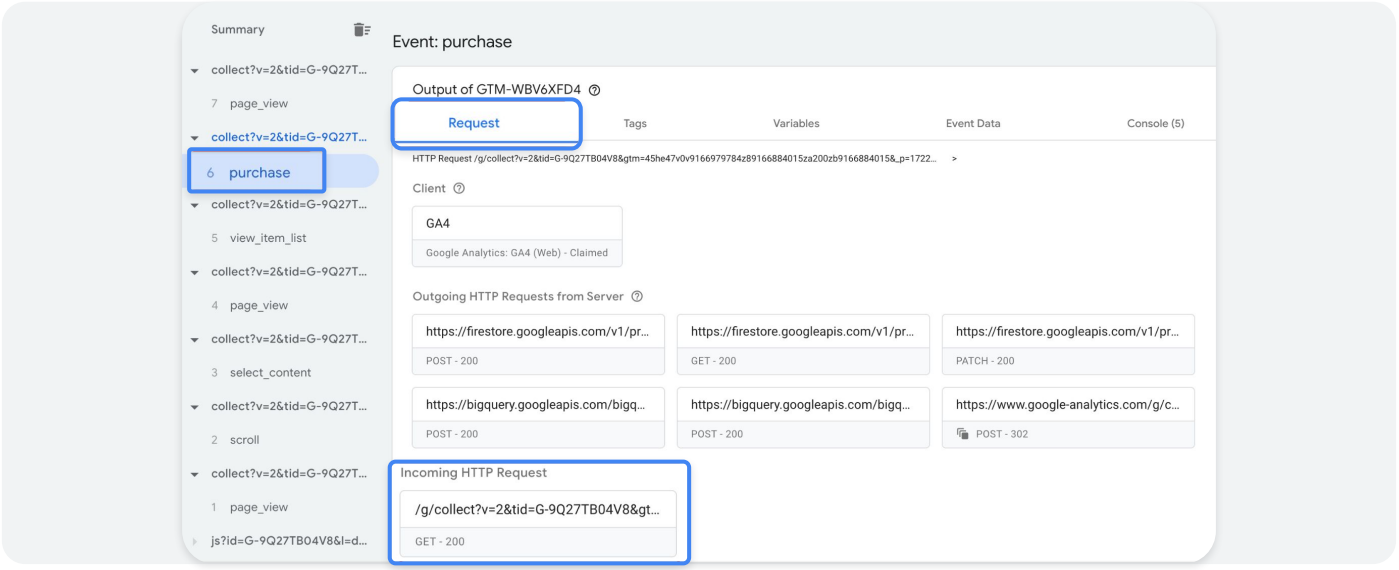
If you are sending data to server-side GTM endpoint:

- In Tag Assistant, in the Hit Details, you will see the list of parameters being sent to sGTM endpoint
- Scroll down to the bottom, here you should see “ep.user_data” parameters
- If you see those parameter, it means values are sent to sGTM
- sGTM will hash the values (unless you a re already sending them hashed) and automatically forward do Google Analytics



Validation in sGTM Tag Assistant

- Switch to the Tag Assistant window connected to your Server Container.
- Click on the event that should carry User-provided data
- In the Request tab, first click on the **Incoming HTTP request**



Incoming HTTP Request

- In incoming HTTP request, you should see ep.user_data parameters that carry User-provided data

NOTE: In case if you are not specifically hashing data on your side, **user_data** sent from your website to sGTM endpoint will arrive unhashed. But, before it is forwarded to Google endpoints, data will be hashed automatically (check next few slides for details).

HTTP Request Details

Request

Request URL

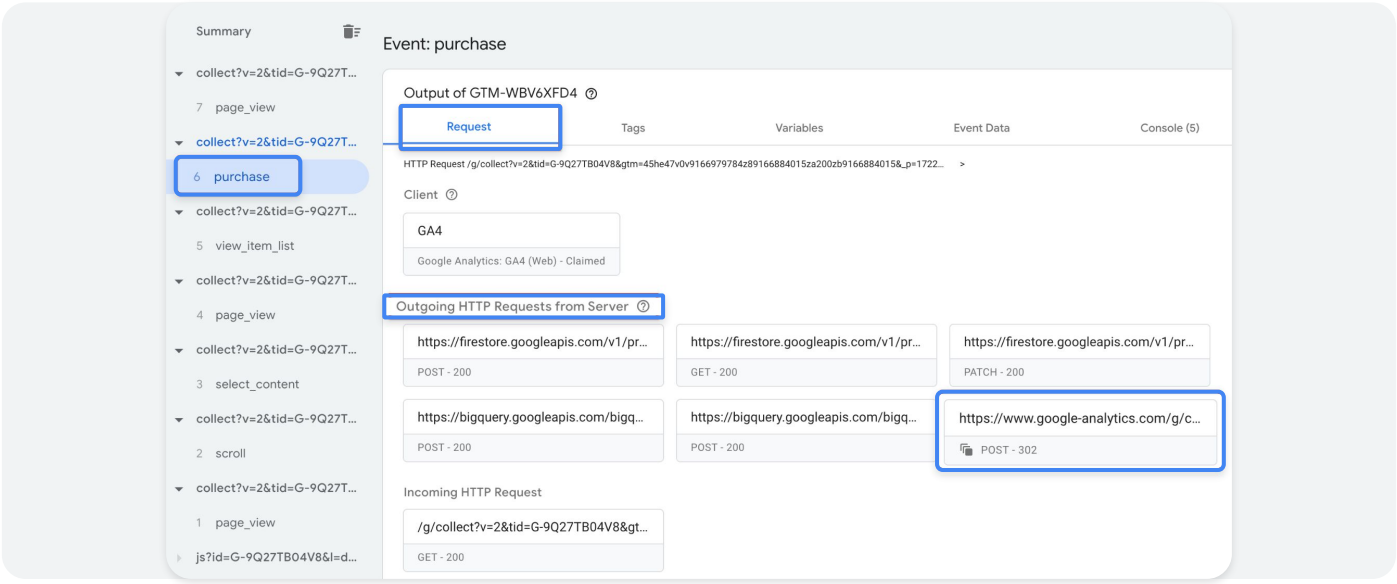
GET

```
/g/collect?v=2&tid=G-
9Q27TB04V8&gtm=45he4880v9166979784z89166884015za200zb9166884015&_p=1723466512750&gcs=G100&gcd=13q3q
Pq2q5&npa=1&dma_cps=-&dma=1&tag_exp=0&cid=1232460501.1723466513&ecid=1673133751&ul=en-
us&sr=2560x1440&_fplc=0&ur=&uaa=arm&uab=64&uafvl=Not)A%253BBrand%3B99.0.0.0%7CGoogle%2520Chrome%3B1
27.0.6533.100%7CChromium%3B127.0.6533.100&uamb=0&uam=&uap=macOS&uapv=14.6.1&uaw=0&are=1&frm=0&pscdl
=denied&ec_mode=c&_eu=IA&sst.rnd=426313654.1723466513&sst.gcd=13q3qPq2q5&sst.tft=1723466512750&sst.
ude=1&_s=7&uid=55&cu=EUR&sid=1723466513&sct=1&seg=1&dl=https%3A%2F%2Fsandbox-demo.eu%2Fthank-
you&dr=https%3A%2F%2Fsandbox-demo.eu%2F&dt=Thank%20you%20%7C%20Demo%20E-
commerce%20Store&en=purchase&_c=1&pr1=idblazer_red_m~nmBlazer~pr150~qt1~vablazer%23red%23m&ep.debug
_mode=true&ep.measurement_type=sGTM&epn.value=150&ep.transaction_id=ea724881-dd90-4dd4-b03a-
c80114c83075&ep.user_data.email=jane.doe%40example.com&ep.user_data._tag_mode=MANUAL&tfd=4318&richs
stsse
```

User-provided data implementation - Server-side tagging

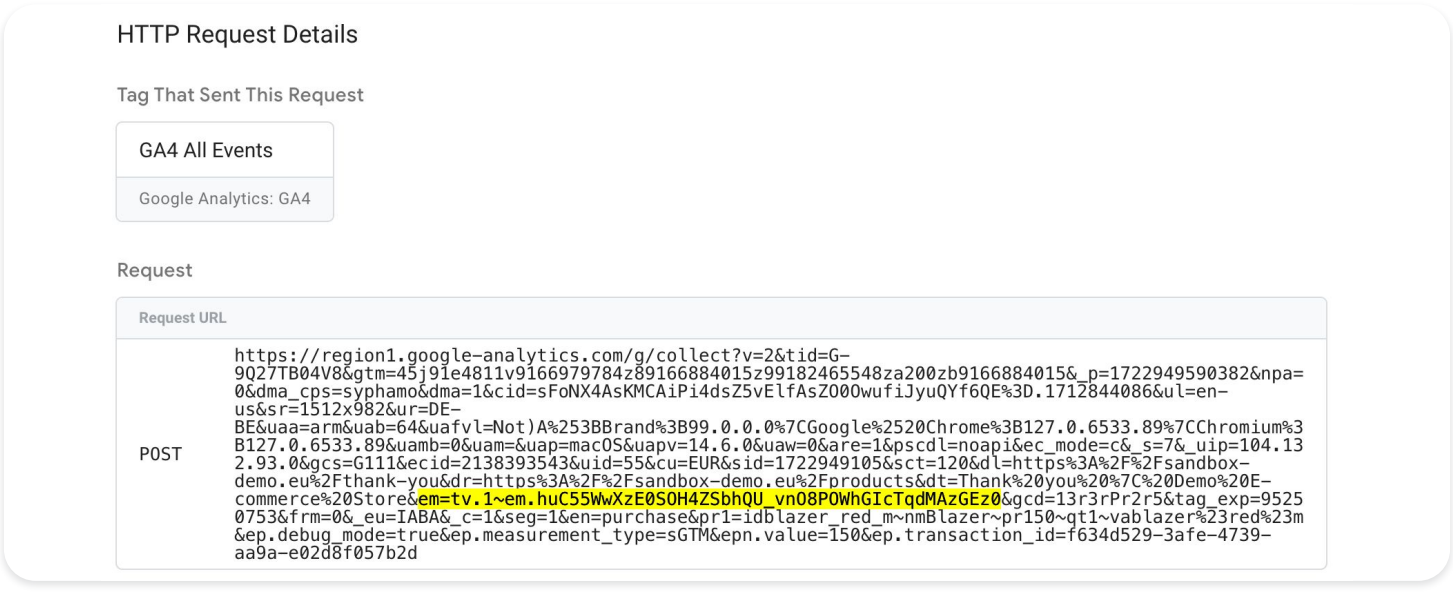
Outgoing HTTP Request

- Next, go back and click on **Outgoing HTTP Request** that is send to Google Analytics endpoint



Checking the “em” parameter

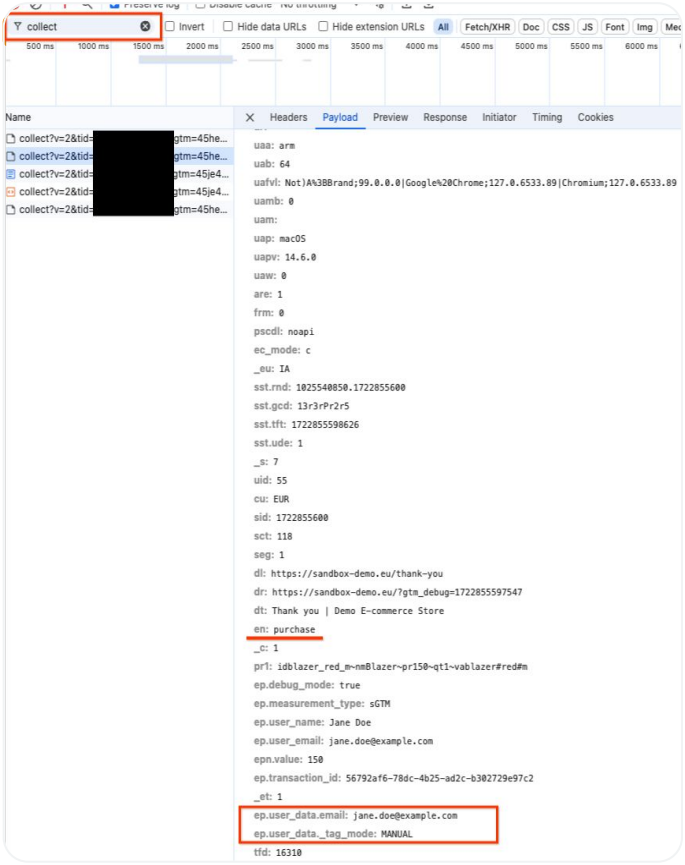
- In the Outgoing HTTP request, look for a parameter “em” with a hashed string as the value. The value should start with **“tv.1~em” followed by a long string of characters.** If you see the "em" parameter with long string character, this means that the tag is picking up and hashing the user-provided data



02 Validation through Developer Console for server-side GTM

Review outgoing requests and key value pairs via browsers developer console capabilities.

- Right click on your web page
- Select **Inspect**
- Select the “**Network**” tab
- Enter “**collect**” in the search bar
- Find the network request that sends event with User-provided data to sGTM endpoint
- Click **Payload** to view the list of query string parameters
- Look for a parameters “**ep.user_data**”
- If you see those parameter, it means values are sent to sGTM
- sGTM will hash the values (unless you are already sending hashed) and automatically forward to GA.
- On next slides, you will find instructions how to validate implementation in sGTM



Important Note:

Browser Network Tab: Filter for your sGTM transport URL. Look for **ep.user_data** or a `user_data` JSON object instead of the **em parameter.** The **`ec_mode`** parameter confirms UPD processing.

- **Why:** In sGTM mode, raw emails aren't sent in the `em` parameter for security. Data is transmitted within a `user_data` object.
- **Processing:** The sGTM GA4 Tag extracts and reformats this back into the standard `em` parameter before the final hit to Google Analytics 4.

sGTM Preview Mode: The definitive proof is the **Outgoing Request** from sGTM to GA. Verify the outgoing hit contains hashed data in the `em` parameter.

Collect & Structure User Data - Measurement Protocol

Developers Center guide

Measurement Protocol

Measurement Protocol for GA4 allows developers to send data directly to Google Analytics servers via HTTP POST requests. It's designed to capture events from environments where standard client-side tracking (like gtag.js, Google Tag Manager, or the Firebase SDK) isn't feasible or sufficient. Commonly used for tracking offline conversions, integrating data from CRM systems, and measuring server-side processes.

Purpose: It enables you to measure interactions that happen server-to-server or from various internet-connected devices.

Function: It supplements and enriches the data collected by your existing Google Analytics tags (gtag.js, GTM, Firebase SDKs), rather than replacing them. It's crucial to have client-side tagging in place for most features to work correctly.

How is Measurement Protocol different from sGTM

Both Measurement Protocol and server-side Google Tag Manager (sGTM) involve server-side data handling, but they serve different primary purposes and operate differently.

Feature	Measurement Protocol	sGTM
Primary Mechanism	Direct HTTP POST requests from your server/device to GA servers.	Acts as a server-side proxy. Receives data from client-side tags (or other sources) in a customer-controlled server container, then transforms and routes it.
Infrastructure	Requires a system/server capable of making HTTP requests.	Requires deploying and managing a tagging server in a cloud environment (e.g., Google Cloud).
Data Control	You programmatically construct and send each event payload.	Centralized control within the sGTM container to modify, enrich, and govern data before sending it to GA or other destinations.
Input Format	Specifically structured JSON payload for GA.	Specifically structured JSON payload for GA.
Customer Segment/Use Case	Customers who need to send specific events from your backend systems or offline sources directly to GA.	Customers who want a secure, customer-controlled server environment to process, modify, and route all or some of the data originating from your website or app before it reaches Google's servers or third parties.

Terms of Service for Measurement Protocol

The use of the Measurement Protocol is governed by the standard Google Analytics Terms of Service. Additionally, there are specific policies and guidelines for using the Measurement Protocol itself, which are outlined in the developer documentation:

- **Prohibited Actions:** You must not send any Personally Identifiable Information (PII) such as names, email addresses, etc., directly to Google Analytics. Hashing is **required** for sensitive data like in User-provided data features.
- **Data Accuracy:** You are responsible for the accuracy and format of the data sent.
- **API Secrets:** GA4 Measurement Protocol requires an api_secret to help prevent spam and ensure data comes from authorized sources.
- **User-provided data (UPD):** If sending hashed user data, you must adhere to the User-provided data Collection policies and have accepted the necessary terms within the Google Analytics interface.

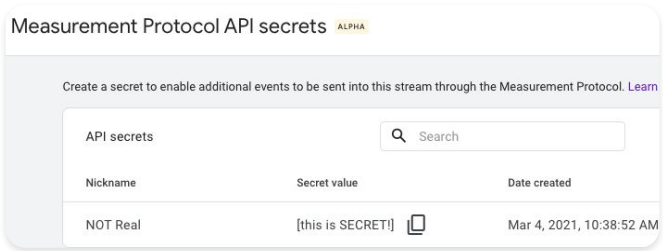
Measurement Protocol - Web

Requirements begin

Review the following requirements before starting setup

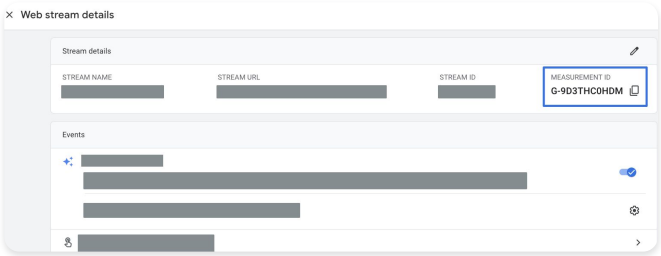
01 Required - API Secret

To create a new secret, navigate in the Google Analytics UI to: **Admin > Data Streams > choose your stream > Measurement Protocol > Create**



02 Required-Measurement ID (Not Stream ID)

To grab your measurement ID associated with a stream, in the Google Analytics UI go to **Admin > Data Streams > choose your stream > Measurement ID**.



03 Client ID is required

Client ID is a required field to send web events via Measurement Protocol. Identifiers collected during online interactions are automatically joined with Measurement Protocol events using client_id. Learn more here.

Pre-Hashing is *required* for Name, Address, Phone, and Email in Measurement Protocol

- Email, phone number, first & last name, and street address must be hashed using the **SHA-256** algorithm before being uploaded.
- The hashed value should be encoded in **hex string format** (string object containing only hexadecimal digits), such as 88d7ecb5c5b21d7b1.

Field	Key	Unhashed Values	Hashed Values
Email	user_data.sha256_email_address[]	jdoe@example.com	a8af8341993604f29cd4e0e5a5a4b5d48c575436c38b28abbfd7d481f345d5db
Phone	user_data.sha256_phone_number[]	+11231234567	e9d3eef677f9a3b19820f92696be53d646ac4cea500e5f8fd08b00bc6ac773b1
First name	user_data.address[].sha256_first_name	john	96d363564cc3032521409cf22a85f
Last name	user_data.address[].sha256_last_name	doe	f92a11af918e3fb741df42934f3b58
Street	user_data.address[].sha256_street	123 New Rd	77f9a3b19820f92696be53d646ac4ce

Measurement Protocol Body Example

- Set up a HTTPS post request to www.google-analytics.com/mp/collect with the api_secret & measurement_id as query parameters.
- You can add user provided information to your payload under the **user_data** object.
  User_id is no longer required to send user_data but we highly recommend sending User ID for cross-device session and user attribution along with other identifiers available (gclid, Transaction ID, etc)

Example JSON body with user provided data

```
{
  "client_id":"123232321.12323425",
  "user_id":"12323123233434",
  "events":[{
       "name":"purchase"
   }],
  "user_data": {
        "sha256_email_address":"a8af8341993604f29cd4e0e5a5a4b5d48c575436c38b28abbfd7d481f345d5db",
        "sha256_phone_number":"e9d3eef677f9a3b19820f92696be53d646ac4cea500e5f8fd08b00bc6ac773b1",
         "address":{
                "sha256_first_name":"96d363564cc3032521409cf22a85f",
                "sha256_last_name":"f92a11af918e3fb741df42934f3b58",
                "postal_code":"SO99 9XX",
                "country":"UK"

                }
        }
}
```

Formatting

(!) Pre-hashed format - Required to be Hashed when sent

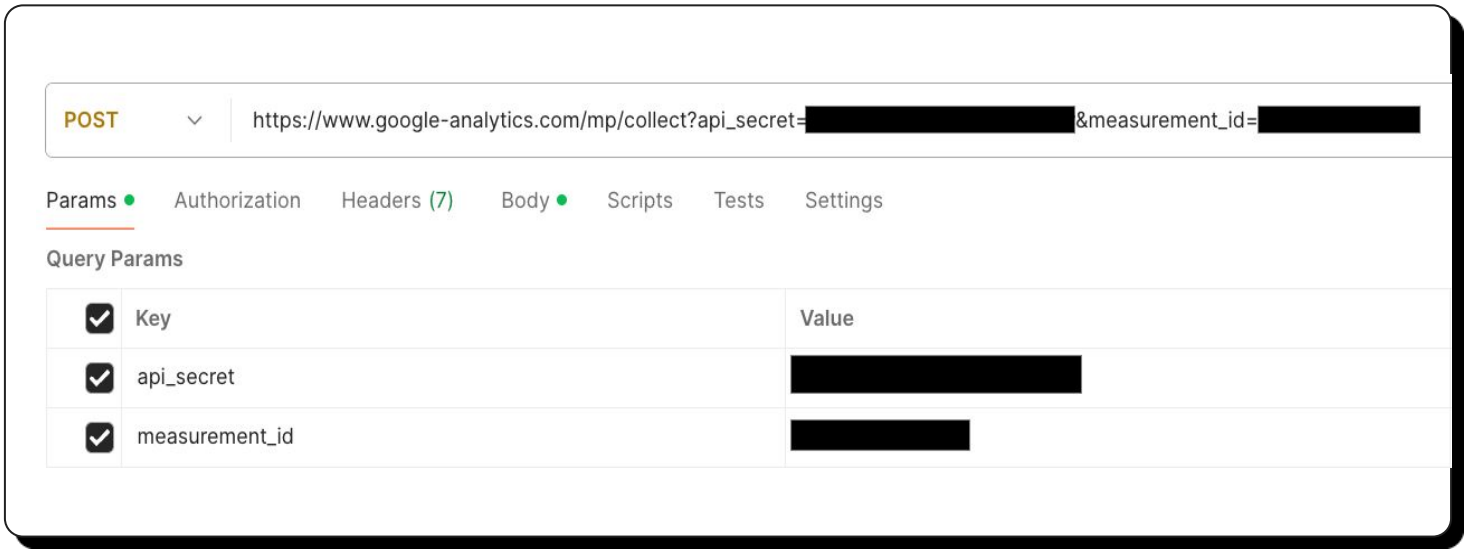
Format reference from dev center

Field	Example Value	Format
Email	jdoe@example.com	Lowercase, no periods before @ for gmail.com/googlemail.com addresses, no spaces
Phone	+11231234567	remove all non digit characters, add + prefix
First name	john	no digits / symbol characters, lowercase, no leading / trailing spaces
Last name	doe	no digits / symbol characters, lowercase, no leading / trailing spaces
Street	123 New Rd	no symbol characters, lowercase, no leading / trailing spaces
City	southampton	no digits / symbol characters, lowercase, no leading / trailing spaces
Region	hampshire	no digits / symbol characters, lowercase, no leading / trailing spaces
Postal Code	SO99 9XX	remove . and ~ characters, no leading / trailing spaces
Country	UK	ISO 3166-1 alpha-2 standard country code

Hashing NOT required

Example Measurement Protocol Request

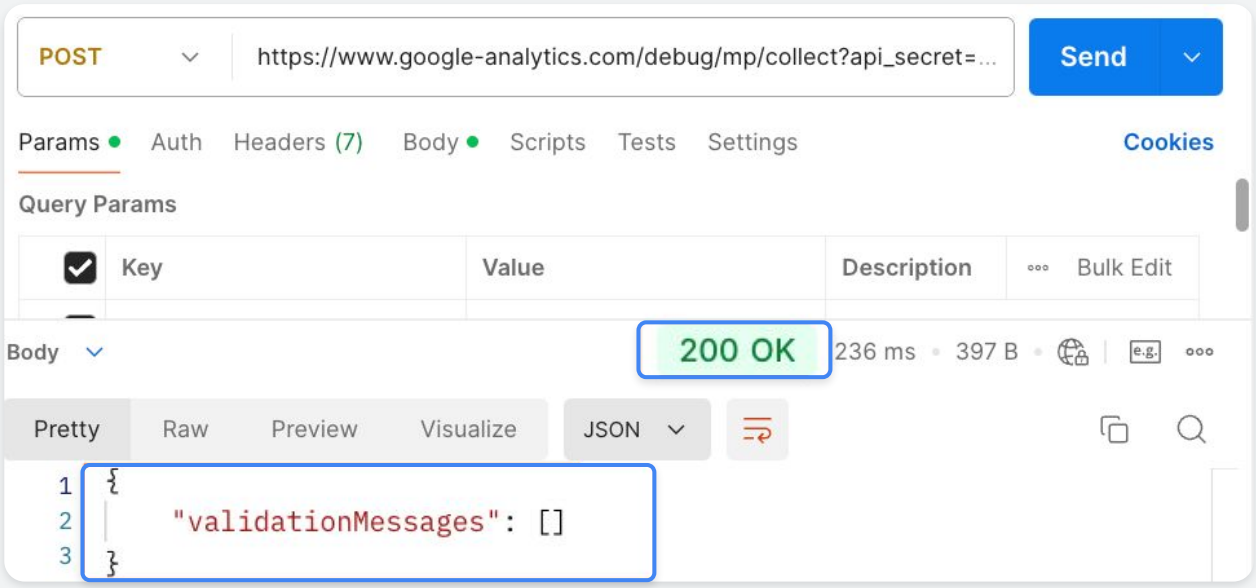
Set up an HTTPS post request to the endpoint: www.google-analytics.com/mp/collect
Add the api_secret and measurement_id as query parameters.



Note that the Event Builder tool does not yet accept user provided data but you can use it to validate a basic request before adding any user_data.

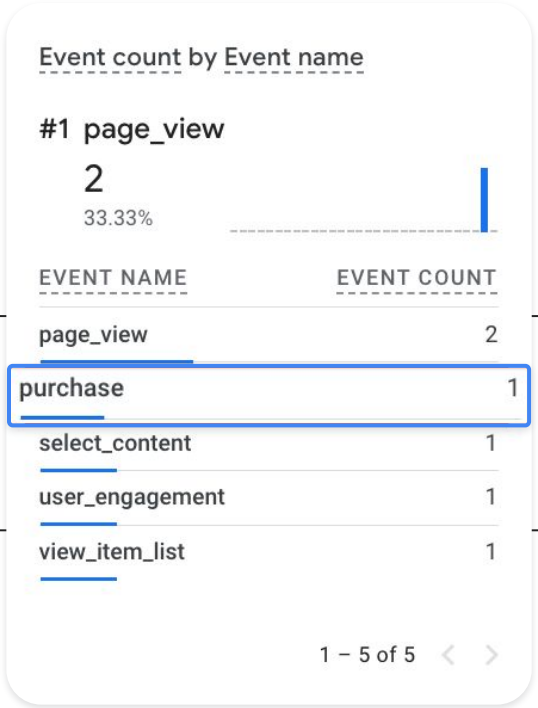
Validation Measurement Protocol

- To validate an event, send HTTPS post requests to the endpoint: www.google-analytics.com/debug/mp/collect
- There is no message confirming a correct request. The validation Messages array should be empty and response code should be **200**.
- When you sent event to www.google-analytics.com/mp/collect endpoint, if request is correct, response code will be **204**.



Realtime Report

After sending the request, check for the event name in the Realtime report under *Event count* by *Event Name*.



User-provided data implementation - Measurement Protocol (App)

App Implementation

Hybrid support for User-provided data

01

GA properties with App & Web Data Streams

User-provided data implementation via Measurement protocol

02

GA properties only with App Streams

Recommended to implement on-device measurement.

Data collection & implementation guide (app)

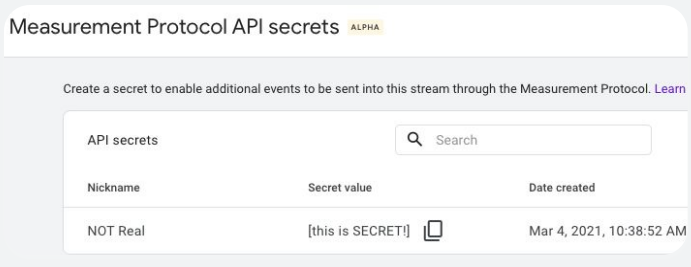
Before you begin

Review the following requirements before starting setup

01

API Secret

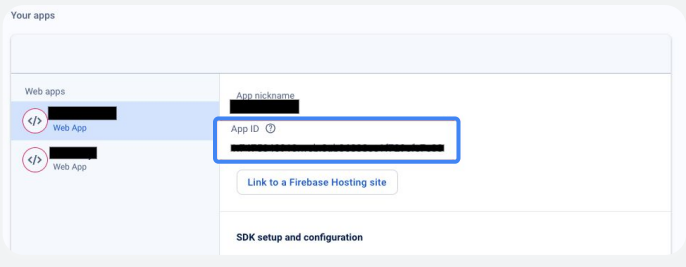
To create a new secret, navigate in the Google Analytics UI to: **Admin > Data Streams > choose your stream > Measurement Protocol > Create**



02

Firebase App ID

To find the App ID, go to Firebase console. Open your project. Click on gear icon next to “Project overview” > Project Settings > General > Your Apps > App ID.



03

App_Instance_id is required

App instance id is a required field to send app events via Measurement Protocol. Identifiers collected during online interactions are automatically joined with Measurement Protocol events using app instance id. Learn more here.

Measurement Protocol Body Example

- Set up a HTTPS post request to www.google-analytics.com/mp/collect with the api_secret & firebase_app_id as query parameters.
- You can add user provided information to your payload under the **user_data** object.
  User_id is no longer required to send user_data but we highly recommend sending User ID for cross-device session and user attribution along with other identifiers available (gclid, Transaction ID, etc)

Example JSON body with user provided data

```
{
  "app_instance_id":"12345678901234567890123456789012",
  "user_id":"12323123233434",
  "events":[{
        "name":"purchase"
   }],
  "user_data": {
        "sha256_email_address":"a8af8341993604f29cd4e0e5a5a4b5d48c575436c38b28abbfd7d481f345d5db",
        "sha256_phone_number":"e9d3eef677f9a3b19820f92696be53d646ac4cea500e5f8fd08b00bc6ac773b1",
         "address":{
                "sha256_first_name":"96d363564cc3032521409cf22a85f",
                "sha256_last_name":"f92a11af918e3fb741df42934f3b58",
                "postal_code":"SO99 9XX",
                "country":"UK"

                }
         }
}
```

Formatting

(!) Pre-hashed format - Required to be Hashed when sent

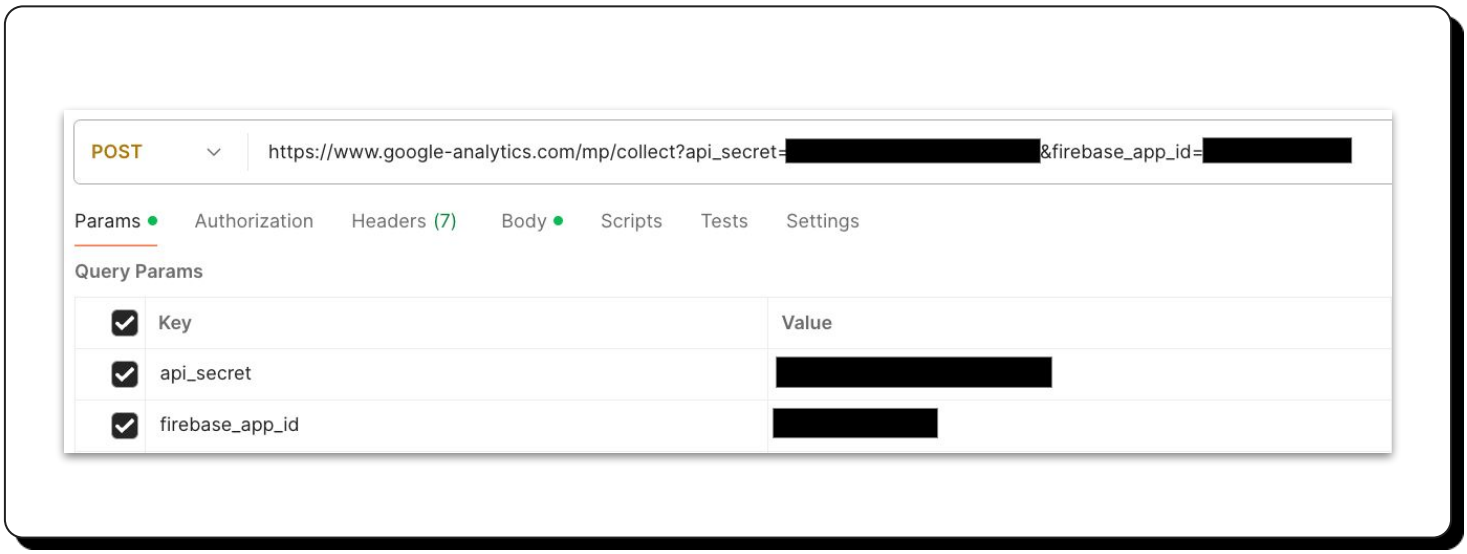
Format reference from dev center

Field	Example Value	Format
Email	jdoe@example.com	Lowercase, no periods before @ for gmail.com/googlemail.com addresses, no spaces
Phone	+11231234567	remove all non digit characters, add + prefix
First name	john	no digits / symbol characters, lowercase, no leading / trailing spaces
Last name	doe	no digits / symbol characters, lowercase, no leading / trailing spaces
Street	123 New Rd	no symbol characters, lowercase, no leading / trailing spaces
City	southampton	no digits / symbol characters, lowercase, no leading / trailing spaces
Region	hampshire	no digits / symbol characters, lowercase, no leading / trailing spaces
Postal Code	SO99 9XX	remove . and ~ characters, no leading / trailing spaces
Country	UK	ISO 3166-1 alpha-2 standard country code

Hashing NOT required

Example Measurement Protocol Request

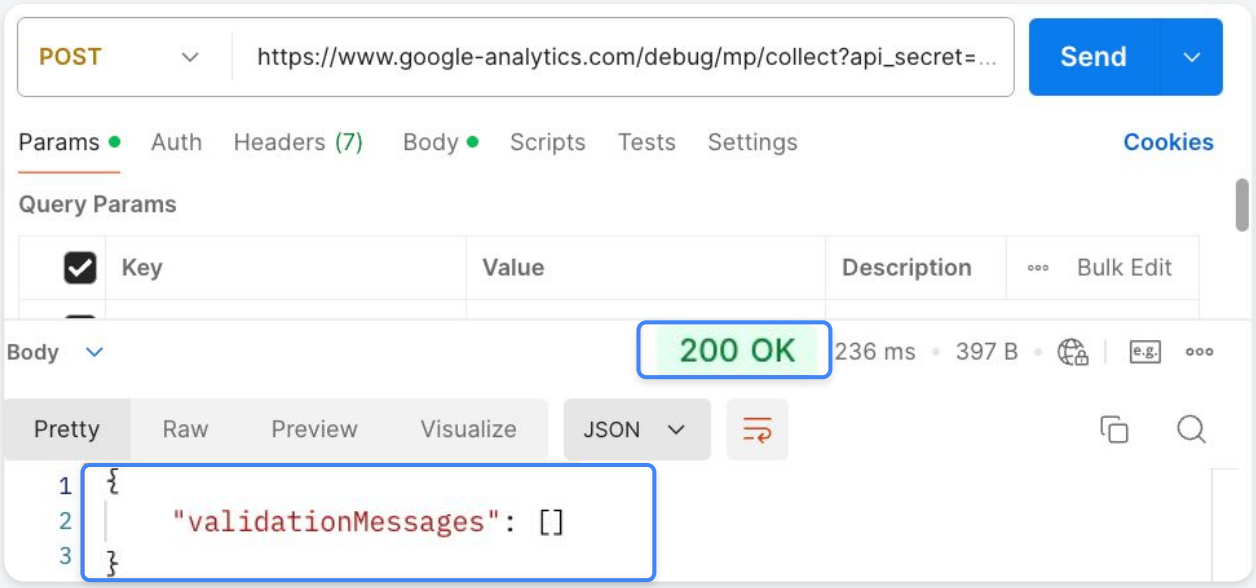
Set up an HTTPS post request to the endpoint: www.google-analytics.com/mp/collect
Add the api_secret and firebase_app_id as query parameters.



Note that the Event Builder tool does not yet accept user provided data but you can use it to validate a basic request before adding any user_data.

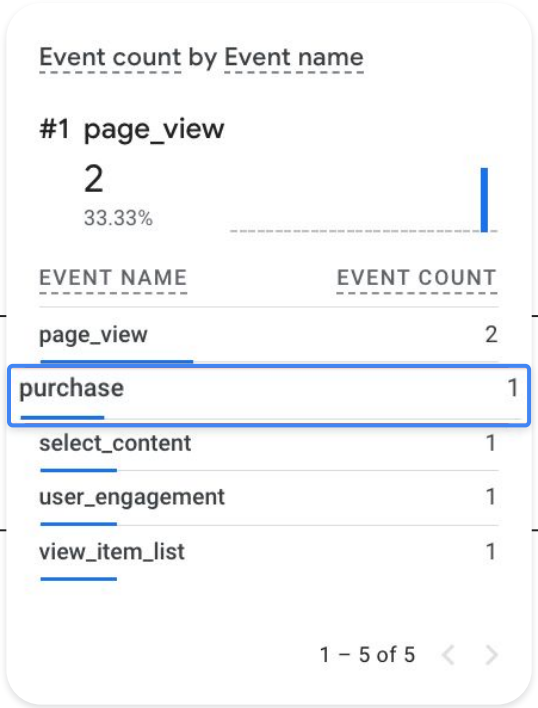
Validation Measurement Protocol

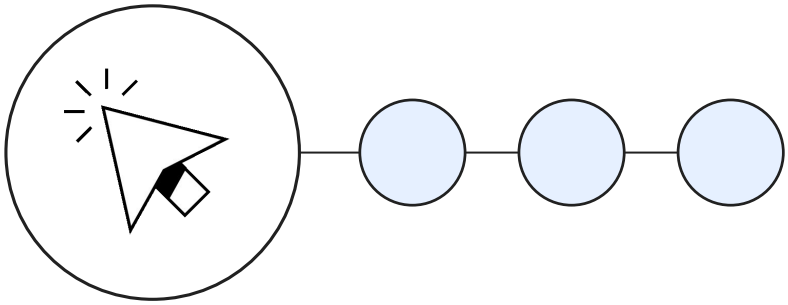
- To validate an event, send HTTPS post requests to the endpoint: www.google-analytics.com/debug/mp/collect
- There is no message confirming a correct request. The validation Messages array should be empty and response code should be **200**.
- When you sent event to www.google-analytics.com/mp/collect endpoint, if request is correct, response code will be **204**.



Realtime Report

After sending the request, check for the event name in the Realtime report under *Event count* by *Event Name*.





FAQ

User-provided data: Collection & Privacy

How can I implement UPD collection?

- **Google Tag (gtag.js):** By modifying your existing tag configuration with user_data parameter
- **Google Tag Manager (GTM):** By configuring variables and updating your Google Analytics event tags.
- **Measurement Protocol:** For sending data from server-side or offline sources (requires pre-hashing).

How is UPD different from the User-ID feature?

- **User-ID:** Is your own unique, non-personally identifiable identifier that you assign to your signed-in users (e.g., from your CRM). It's primarily for tracking a user's journey across devices and sessions within GA4 and is essential for user and session attribution
- **UPD:** Consists of specific, hashable fields (email, phone, address) sent to Google to match with Google's user data to power EC, Customer Match, and Demographics. UPD is essential for conversion attribution
- They can be used together; sending UPD alongside User-ID is recommended for the most accurate and durable measurement.

Is User-ID required to send UPD?

While initially User-ID was a strong recommendation or requirement for some UPD implementations, the system has evolved. It's best practice to use User-ID alongside UPD when available to provide the most robust data. However, UPD can provide benefits even for users not signed into your site, by matching with Google's signed-in user base. Always check the latest implementation guides for specific scenarios (gtag, GTM, Measurement Protocol).

What types of data can be sent as UPD?

You can send email addresses, phone numbers, and name/address details. This data must be hashed using the SHA256 algorithm before Google processes it for matching. You can hash it client-side or let Google hash it during collection via the tag.

Is sending UPD privacy-safe?

Yes, UPD is designed with privacy in mind. Please note that you still need to implement Consent Mode to adhere with GDPR rules and regulations: UPD data is:

- Collected only from users who have given consent.
- Hashed using SHA256 (a one-way hashing algorithm) before matching.
- Matched against hashed Google user data.
- Google does not have access to the raw PII (unhashed PII)

FAQ

User-provided data: Attribution

Is User-provided data still used as a reporting identity key?

User-provided data is not a reporting identity as of October 2025. Please see the help center article for current reporting identities.

Does UPD affect User or Session attribution?

No. UPD is not used for standard User Acquisition or Session Acquisition attribution in Google Analytics reports. Those primarily rely on User-ID and Device ID. UPD's impact on attribution is specifically for conversion attribution through Enhanced Conversions.

Is User-provided data used for user deduplication?

The matching process with UPD does not directly alter the user counts or deduplication logic in standard Google Analytics behavioral reports. Those reports still rely on the User-ID/Signals/Device ID hierarchy.

In summary, while UPD helps connect conversions to users for attribution and audience purposes by matching with Google's user space, it doesn't function as a primary method for deduplicating users within your Google Analytics property's standard reporting. User deduplication for consistent user counts and journey analysis across devices primarily relies on the User-ID feature.

***Note**: In our future roadmap, we would use User-provided data as a join key for offline events and online interactions to unlock Enhanced Conversions for Leads in Google Analytics.*

Initially, we noticed situations where the default lookback window may credit conversions to older touchpoints rather than the most recent paid interactions when a customer has enabled UPD. Is this still true?

There were some instances in the past where UPD in conjunction with other settings might have led to unexpected attribution shifts. However, Google Analytics attribution logic has been refined. UPD is now specifically focused on enhancing conversion attribution through EC, and does not influence standard User or Session source attribution.

How does UPD improve conversion attribution?

UPD helps Enhanced Conversions link conversions back to ad interactions that might otherwise be missed due to cookie limitations or cross-device behavior. By increasing the ability to match conversions to signed-in Google users who interacted with your marketing, Google Analytics attribution models can provide a more complete and accurate picture of which channels drive conversions.

FAQ

User-provided data: Attribution

Does UPD help measure both paid and organic channels?

Yes. While Enhanced Conversions in Google Ads focuses on ad-driven conversions, User-provided data in Google Analytics helps attribute conversions across all channels, including organic search, direct, referral, email, as well as paid campaigns. It provides a more holistic view of the entire or total customer journey.

Should the customer implement Google Analytics EC (UPD) or Google Ads EC?

If you use Google Analytics as your source of truth for conversion bidding, you should implement Google Analytics EC (UPD). If the you use Google Ads conversion tracking as your source of truth for bidding, you should use Google Ads EC. However, you can also implement both Google Ads EC and Google Analytics UPD and set whichever platform to Primary. If you use Google Analytics EC (UPD) you will get other benefits beyond Enhanced Conversions such as Customer Match and Demographics and Interest Reports because Google Analytics also offers comprehensive total journey and cross-channel measurement. Please check the help center article on when to choose which method.

Could you precisely explain how UPD is used for cross-channel or total journey measurement?

UPD for Google Ads Conversion attribution: In a world with signal loss and fragmented user journeys, UPD provided at the time of conversion can be matched against Google's signed-in user data. This allows Google Analytics to link conversions back to ad interactions (clicks, views) that might otherwise be unobservable (especially cross-device), which directly fuels improved bidding in Google Ads.

UPD Conversion attribution across channels: With this more complete view of the touchpoints, Google Analytics' Cross-channel attribution can more accurately distribute conversion across all the different marketing touchpoints a user interacts with before converting. This includes organic search, paid search, social, email, direct traffic, referrals, display, etc. Help Center article for more details.

FAQ

User-provided data: Reporting

How is UPD used to populate Demographics & Interests reports?

UPD is one of the signals used to derive Age, Gender, and Interests data. By matching hashed UPD with Google's data, Google Analytics can infer these characteristics for users, offering insights that are more resilient than those based solely on traditional methods like third-party cookies. This works alongside Google Signals and device advertising IDs.

Does the Demographics & Interest report fully populate based on UPD alone, or is Google Signals still a prerequisite?

Google Signals is not required for Demographics & Interest reports if a customer has UPD enabled. UPD is an additional source for widening reporting coverage and helps power more accurate and durable Demographics and Interests reporting based on first-party data matched with Google's data, in addition to 3P cookies

Which reports are impacted by User-provided data?

User-provided data will impact Event and Conversion attribution in Advertising Workspace and Explore reports, Demographic reports and Audiences list exported to Google Ads (Customer Match is only applied during Audience export). Other behavioral reports are not impacted by UPD

Are you able to see UPD or modelled conversions in BQ exports

No, hashed User-provided data is not exported to BQ and neither is modelled conversions.

FAQ

User-provided data: Sub & Rollup Properties

Is User-provided data collection configured within Subproperties or Roll-up Properties?

No. UPD collection is configured at the ordinary (source) property level, where the data streams are located. Subproperties and Roll-up properties do not have their own data streams; they inherit data from source properties.

Does UPD collected in a source property flow into its Subproperties?

Yes. The event data, including any parameters carrying hashed UPD, flows from the source property to any associated Subproperties, provided the events meet the Subproperty's filter criteria.

Does UPD from multiple source properties flow into a Roll-up Property?

Yes. A Roll-up Property aggregates event data from all its linked source properties. This includes any hashed UPD collected by those source properties.

How does Enhanced Conversions utilize UPD within a Subproperty?

The Subproperty can leverage the UPD passed from its source property for Enhanced Conversions. The matching process for EC will occur based on the data present in the Subproperty. For EC data to be used in Google Ads bidding, the Subproperty itself must be linked to the Google Ads account, and the relevant conversions (Key Events) must be imported from the Subproperty to Google Ads.

How does Enhanced Conversions (EC) utilize UPD within a Roll-up Property?

The Roll-up Property uses the combined UPD from all its source properties. EC matching and modeling benefit from this aggregated dataset. Similar to Subproperties, for EC to impact Google Ads, the Roll-up Property must be linked to Google Ads, and its Key Events imported.

Can I build Customer Match audiences using UPD in Subproperties and Roll-ups?

Yes. You can create audiences within Subproperties and Roll-up Properties. If these audiences utilize the inherited UPD, they can be exported to Google Ads for Customer Match, provided the specific Subproperty or Roll-up Property is linked to the Google Ads account.

FAQ

User-provided data: Sub & Rollup Properties

Will the impact of UPD on conversion attribution be the same in a Subproperty/Roll-up as in the source property?

Not necessarily. Each property (source, sub, or roll-up) has independent attribution settings and models. While all can benefit from the richer data provided by UPD, the models are trained on the specific dataset within that property:

- **Subproperty**: Filtered data, potentially leading to different attribution credit.
- **Roll-up Property:** Aggregated data from multiple sources, potentially leading to different cross-property attribution insights.

The specific product links and imported Key Events from the Subproperty or Roll-up to Google Ads will determine how EC data influences bidding.

If I have UPD enabled in a source property, do I need to do anything extra for my Subproperties/Roll-ups to benefit?

Beyond ensuring the data flows (which it does by default, subject to filters for subs), you need to:

- Configure settings within the Subproperty/Roll-up (e.g., Key Events, Audience definitions).
- Activate Google Signals in the Sub/Roll-up if desired.
- Establish product links (e.g., to Google Ads) from the Sub/Roll-up to export conversions or audiences.

Do Subproperties and Roll-ups automatically use Google Signals if the source property does and UPD is enabled?

No. Google Signals needs to be activated independently for each Subproperty and Roll-up Property where you want to leverage it for demographics, interests, and audience building, even if it's active in the source properties.