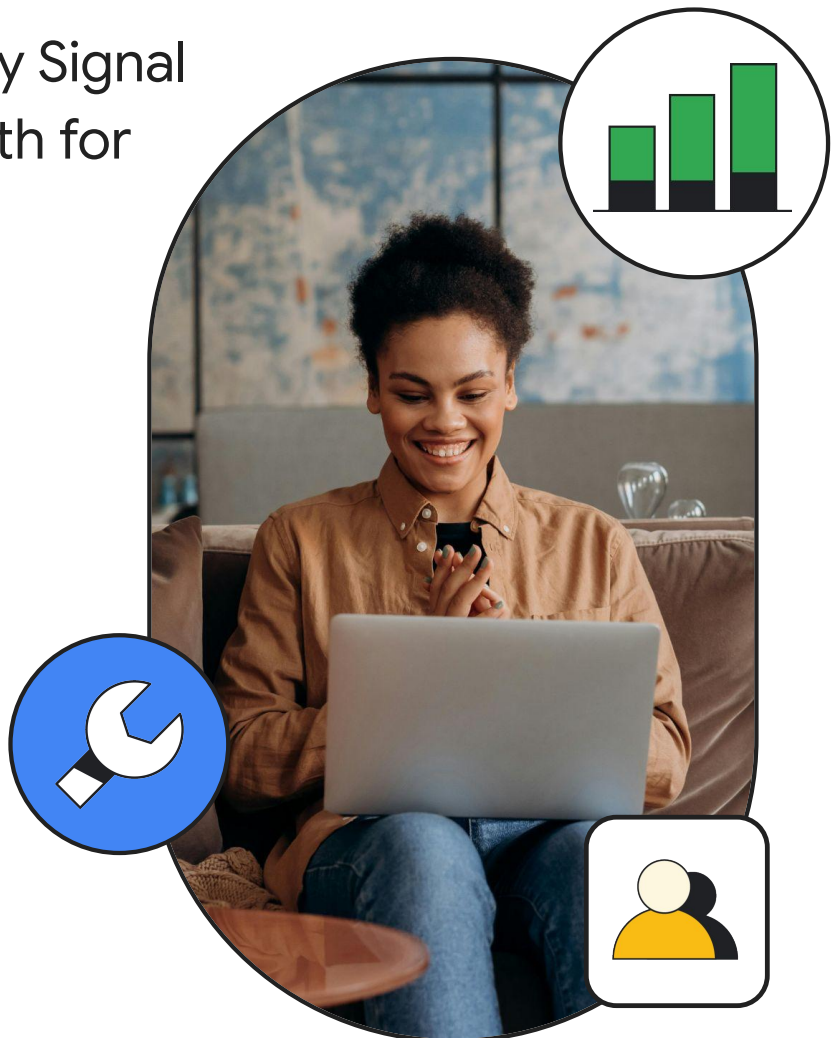




Google Analytics

User-provided data: Implementation Guide

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



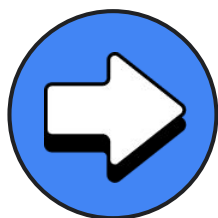


Table of contents

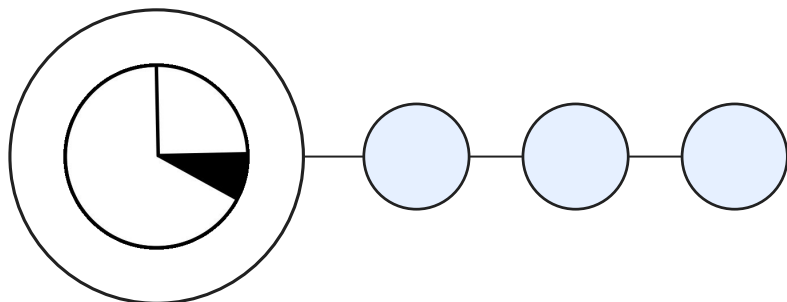
What is User-provided data & Benefits

Implementation

FAQ

Contents

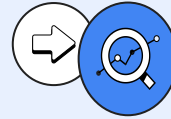
- 01 [What is User-provided data and Why First Party Data Matters Now](#)
- 02 [Benefits of User-provided data](#)
- 03 [Choose your implementation platform](#)
- 04 [Getting Started](#)
- 05 [Accept Terms of Service for User-provided data](#)
- 06 [Enable User-provided data in the Google Tag settings in GA UI](#)
- 07 [Collect and Structure User Data: Google Tag](#)
- 08 [Collect and Structure User Data: Google Tag Manager \(Web\)](#)
- 09 [Collect and Structure User Data: Google Tag Manager \(sGTM\)](#)
- 10 [Collect and Structure User Data: Data Manager API](#)
- 11 [User-provided data FAQs: Collection & Privacy & Data Processing and Protection Guide](#)
- 12 [User-provided data FAQs: Attribution](#)
- 13 [User-provided data FAQs: Reporting](#)
- 14 [User-provided data FAQs: Rollup & Sub-properties](#)



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 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

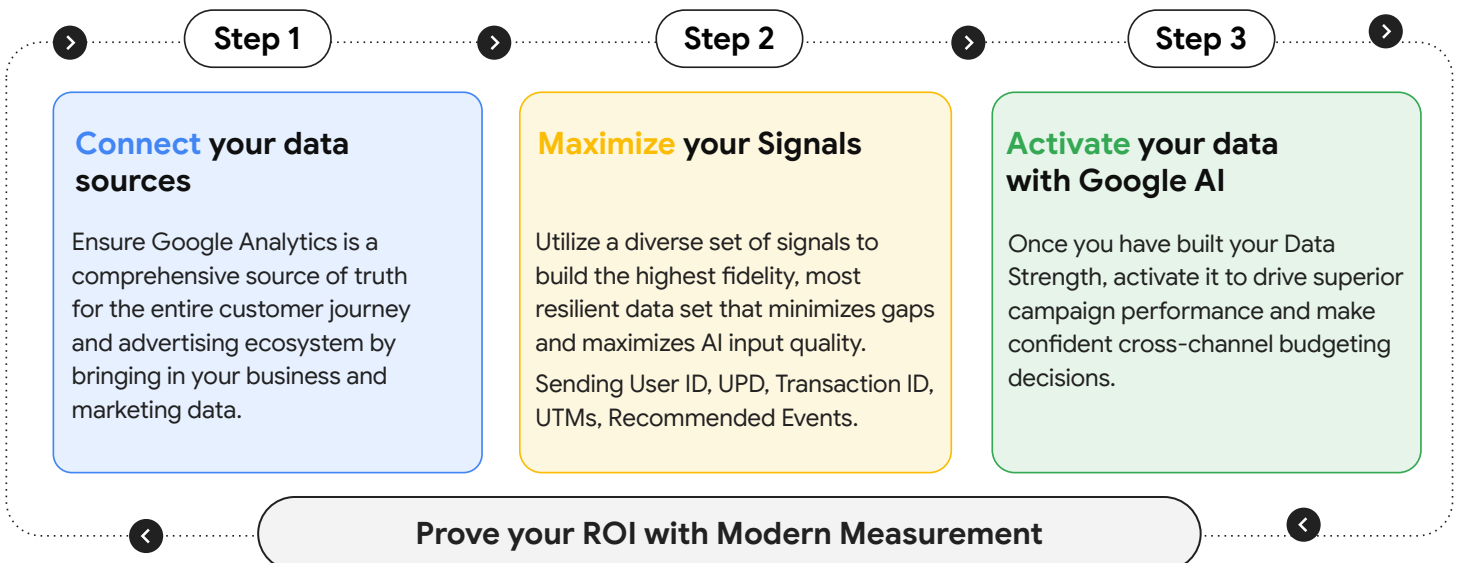
UPD works by filling in signal gaps to ensure more conversions and interactions are observed, attributed, and modeled, making your first-party data improve ROI and Bidding. In addition, UPD improves match rates and expands existing 3P audience lists.

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 must complete the following three steps:



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

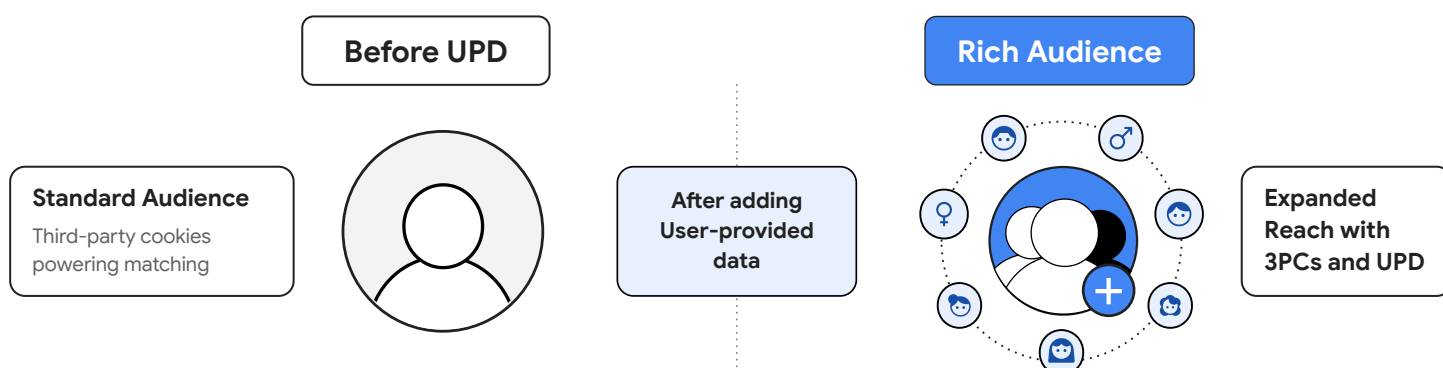
Better data ingested by bidding models (tROAS / tCPA) and attribution models (DDA)



Improved Performance

- Better overall CVRs
- Higher ROAS / Lower CPA

02 UPD unlocks high-value Audience Activation with Customer Match



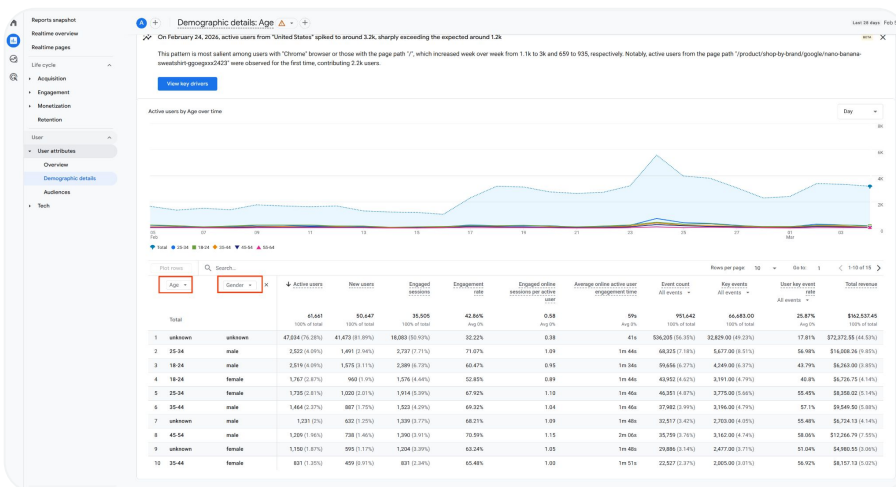
- **UPD improves match rates and expands existing 3P audience lists**, giving them more reach for Google targeting across platforms like Search and YouTube in addition to third-party cookies
- **This makes your list size bigger and increases match rates** leading to even more improved ad relevance and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

Module II: Maximize your ROI with User-provided 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.



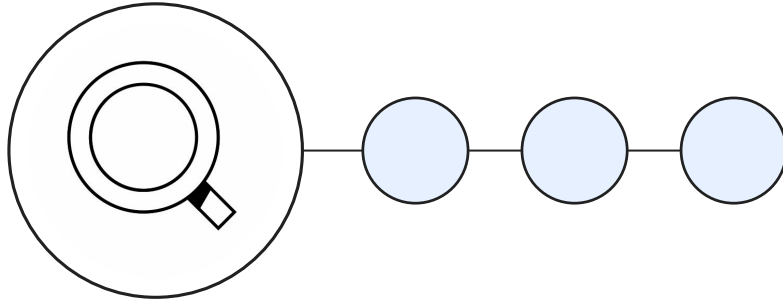
But today, connecting these **offline events to your online marketing efforts** is like trying to piece together a puzzle with missing pieces. This fragmented view means **you're not seeing the full picture** of what drives your business.

Bridging the Gap with User-provided data

In the near future, 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.

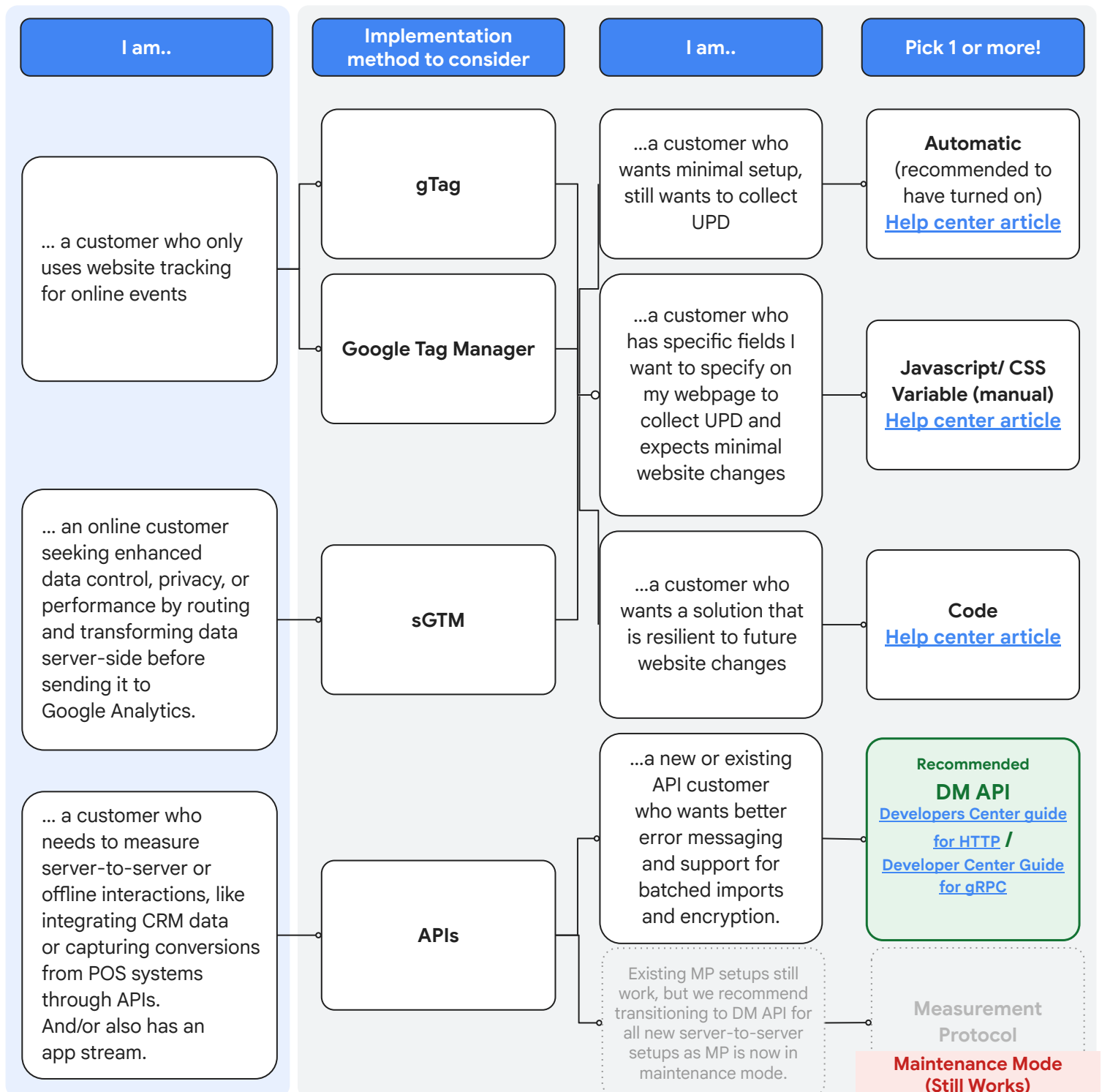


Implementation



Module III: Choose your implementation platform

[UPD Help Center article](#)



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 settings

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](#), [Data Manager API](#)

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 Data Manager API, or Measurement Protocol), the first step is to enable User-provided data toggle in GA UI.**



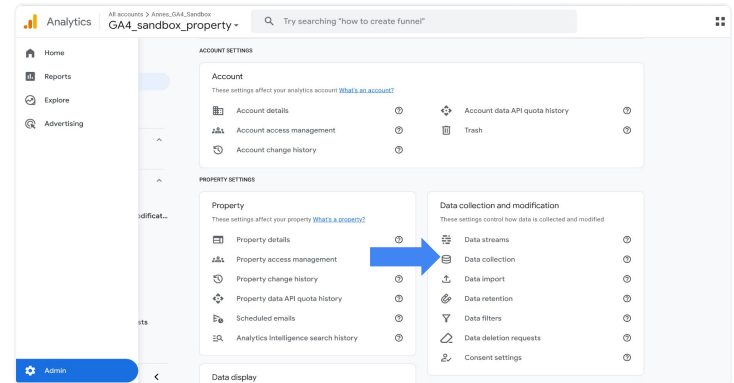
Steps

1

In your Google Analytics account, click **Admin**

2

Data Collection



User-ID collection

The user-ID feature allows you to provide your own identifiers for your site and app users, so you can connect their behavior across different sessions and the various devices and platforms from which they visit. Google Analytics interprets each distinct user-ID as a separate user, which provides you with more accurate user counts and a more holistic story about a user's interactions with your business. [Learn more about user-ID](#)

You're responsible for ensuring that you use the user-ID feature in accordance with the [Google Analytics Terms of Service](#). This includes not sending Google Analytics personally identifiable information as user-ID, and providing appropriate notice of your use of identifiers in your privacy policy. For instructions on how to send user-IDs to Google Analytics see [Send user-IDs](#).

User-provided data collection beta

Improve audience and conversion measurement, and enable additional audience list capabilities in Google Ads. Please review the [beta feature documentation](#) before turning this feature on to understand how this will impact your Analytics property. [Turn on](#)

This activates improved user and conversion measurement by giving you enriched more-detailed audience data. Capabilities such as improved conversion measurement and audience sharing require having linked advertising accounts and [Ads Personalization](#) turned on in your Google Analytics property. User-provided data collection supplements your existing Google Analytics data by allowing you to send your consented, hashed customer data to Google Analytics in a privacy-safe way. [Learn more about user-provided data collection](#).

By turning on user-provided data collection, you acknowledge you adhere to the [user-provided data feature policy](#) and intend to have the necessary privacy disclosures and rights from your end users for such use. Per our policy, use of this feature is prohibited for customers in sensitive categories.

3

Click **"Turn On"**

4

Review the Google Analytics [User-provided-data collection for User-ID Policy](#).

5

Enable User-provided data automatic collection

User-provided data collection beta

Improve audience and conversion measurement and enable additional audience list capabilities in Google Ads. [Turn on](#)

This activates improved user and conversion measurement by giving you enriched more-detailed audience data. Capabilities such as improved conversion measurement and audience sharing require having linked advertising accounts and [Ads Personalization](#) turned on in your Google Analytics property. User-provided data collection supplements your existing Google Analytics data by allowing you to send your consented, hashed customer data to Google Analytics in a privacy-safe way. [Learn more about user-provided data collection](#).

Turn on user-provided data collection

When you choose to turn on user-provided data collection, you may securely send Google Analytics consented, hashed customer data or allow the Google tag on your website to collect on your behalf.

By turning on the user-provided data feature, you acknowledge you adhere to the user-provided data feature policy, including rules around sensitive categories and the need to have the necessary privacy disclosures and rights from your end users for such use.

Optional settings

These settings can be updated after user-provided data is turned on.

☒ **Collect automatically-detected user-provided data.**

When turned on, Google Analytics will receive hashed user-provided data that is automatically detected by the Google Tag on your website.

Note: This feature requires that your Google Tag is configured to automatically detect user-provided data. User-provided data collection can be configured manually with the Google Tag or Google Tag Manager. [Collect user-provided data](#)

[Cancel](#) [Turn on](#)

(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](#).

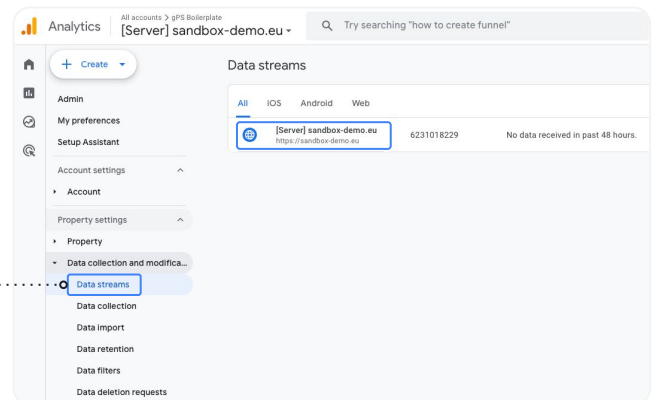
02

Enable User-provided data in the Google tag Configuration

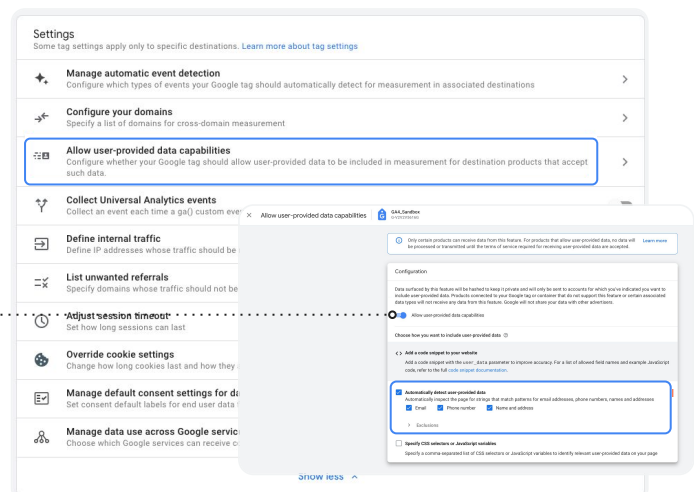
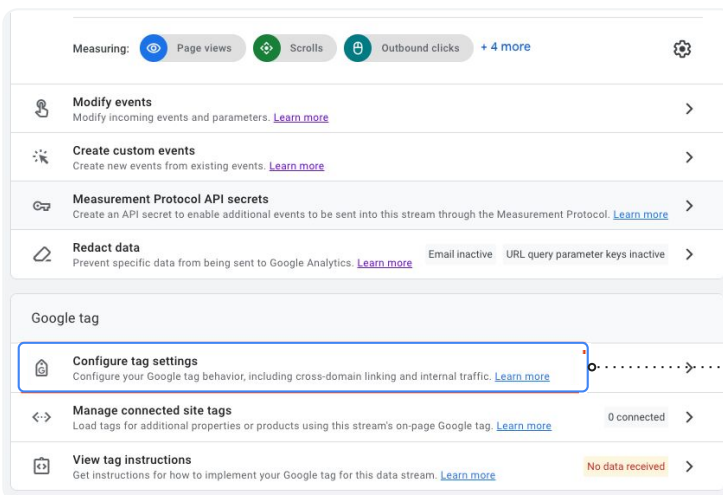
UPD collection is controlled within the **Google tag settings**, which acts as the **master switch**, even for GTM implementations. **Google tag's UPD** setting must be **ON** for User-provided data to be collected from any source (Google Analytics events, GTM tags).

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.

¹ Web+App Streams are also supported. UPD for Apps can be ingested through Data Manager API or 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 Analytics' Google Tag settings

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

Choose how you want to include user-provided data ?

< > Add a code snippet to your website

Add a code snippet with the `user_data` parameter to improve accuracy. For a list of allowed field names and example JavaScript code, refer to the full [code snippet documentation](#).

☒ Automatically detect user-provided data

Automatically inspect the page for strings that match a pattern for email addresses

> Exclusions

☒ Specify CSS selectors or JavaScript variables

Specify CSS selectors or JavaScript variables to identify relevant user-provided data on your page

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

Configuration

Data surfaced by this feature will be hashed to keep it private and will only be sent to accounts for which you've indicated you want to include user-provided data. Products connected to your Google tag or container that do not support this feature or certain associated data types will not receive any data from this feature. Google will not share your data with other advertisers.

☒ Allow user-provided data capabilities

Choose how you want to include user-provided data ?

< > Add a code snippet to your website

Add a code snippet with the user_data parameter to improve accuracy. For a list of allowed field names and example JavaScript code, refer to the full [code snippet documentation](#).

☐ Automatically detect user-provided data
Automatically inspect the page for strings that match patterns for email addresses, phone numbers, names and addresses

☒ **Specify CSS selectors or JavaScript variables**
Specify a comma-separated list of CSS selectors or JavaScript variables to identify relevant user-provided data on your page

☒ Email
Email address
Example: #email or emailJsVar

☐ Phone number

☐ Name and address

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.	a4199369cd4e0e5a5a4b5d48c5754098b6c813a89045b642e4785b9b9a67894a
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.	GB

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.

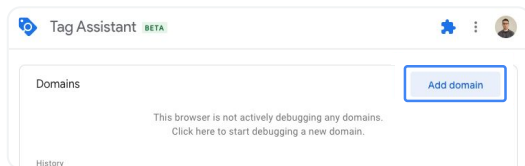
01



Tag Assistant/GTM

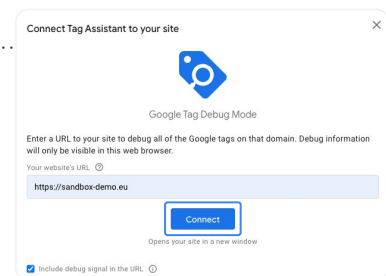
01

Open tagassistant.google.com



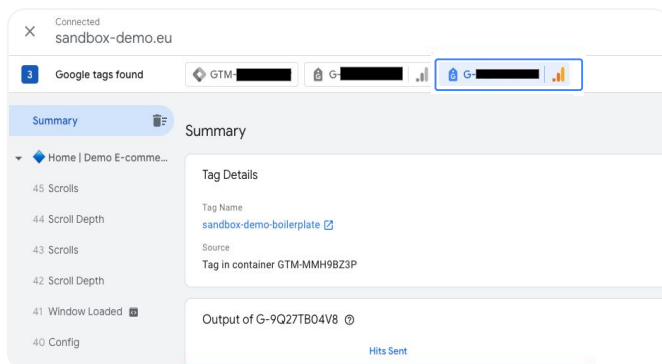
02

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



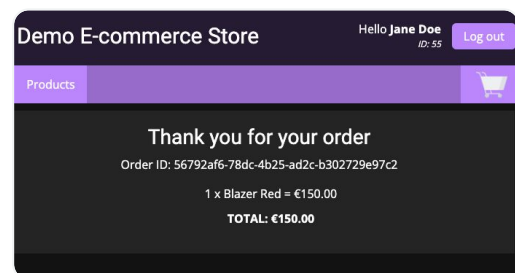
03

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

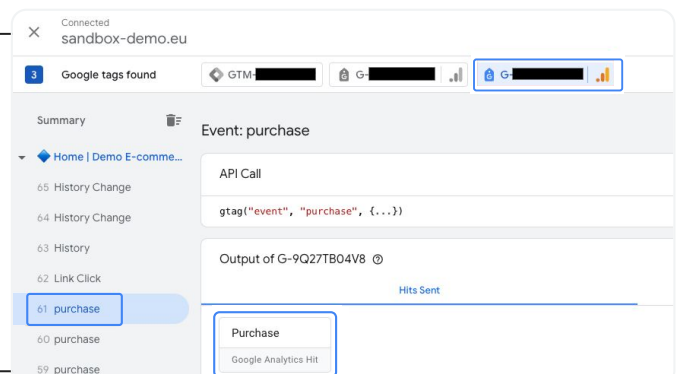
On the left side, click on the event that should carry User-provided data

06

Check if there is hit sent to Google Analytics

07

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





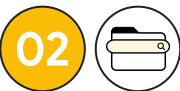
5. Check User-provided data values

- 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-adfb3ccf101
Engagement Time	..et	877
ec_mode	ec_mode	c
em	em	tv.1~em.huC55WwXzE0S0H4Z5bhQ0_vn08POWhGicTqMazGEz0

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
- Select 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

The screenshot shows the Chrome DevTools Network tab with a search filter for 'collect'. A request to 'gtag.js' is selected, and the 'Payload' tab is active. The query string parameters are listed, with the 'em' parameter highlighted in red, showing a hashed value: `em: tv.1~em.huC55WwXzE0S0H4Z5bhQ0_vn08POWhGicTqMazGEz0`. The 'en' parameter is also highlighted, showing the value `en: purchase`.



Validate in Google Analytics UI Diagnostics

Data collection

Google signals data collection ⬆ 🔵

By activating Google signals, you allow Google Analytics to collect data about both your traffic and the data collected through a standard Google Analytics implementation in order to give you additional features. Google signals will not be included in [Google Analytics reporting identity](#).

When Google signals is turned on, Google Analytics will collect visitation information and associate it with Google's information from accounts of signed-in users who have consented to this association. Google's information may include end-user location, search history, YouTube history, and data from sites that partner with Google. By turning on Google signals, you acknowledge you adhere to the [Google Advertising Features Policy](#), including rules around sensitive categories, have the necessary privacy disclosures and rights from your end users for such association, and that such data may be accessed and deleted by end users via [My Activity](#).

Google signals data collection is allowed in **307 of 307** regions. ⚙️

Note: Enabling the above toggle causes Google Analytics to automatically collect additional data about your traffic to provide additional insights about your users. If you don't want to collect data for Google signals, you should disable the toggle above.

User-ID and user-provided data collection ⬇ 🔴 URGENT

Granular location and device data collection ⬇ 🔵

Advanced settings to allow for ads personalization ⬇

User Data Collection Acknowledgement ⬇ 🟢 Acknowledged



Where to access in UI Diagnostics

Follow these steps to check signal status:

- Navigate to the **Admin** section of your property.
- Click on **Data collection**.
- Expand the section for **User-ID and User-provided data collection**.
- Verify if any of your data streams have received User-provided data within the **last 48 hours**.

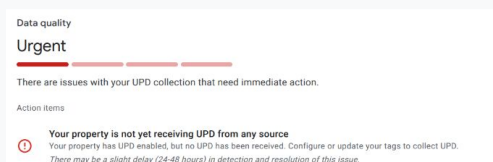
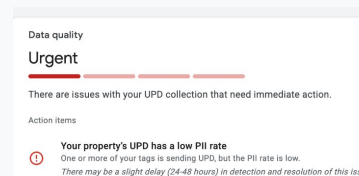
Note on Update Latency:

This message won't update immediately. It may take a few days before status changes are reflected in the UI.



Troubleshoot Alerts

An **"Urgent"** flag will appear if UPD is enabled but no data is detected, or if you are sending UPD but with low PII rate for formatting issues



Troubleshoot Diagnostic Alert



Google Tag Manager

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

[Help Center guide](#)

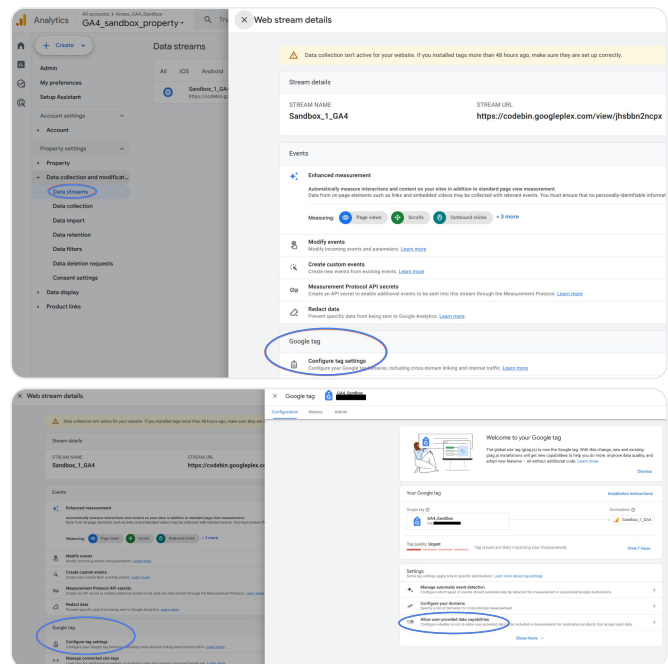


Important Note: Allow User-provided data capabilities on Google Tag Settings



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



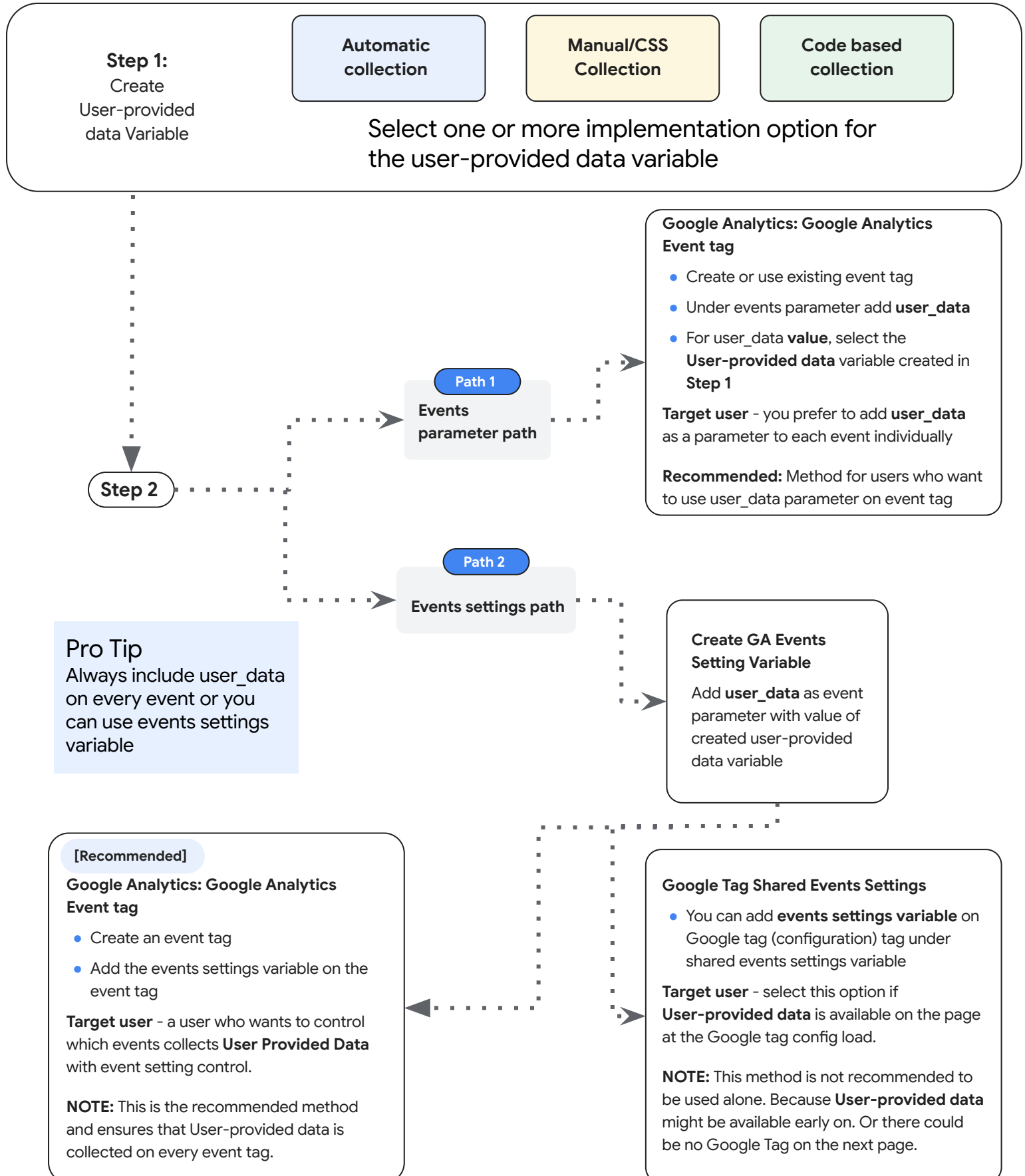
Other Key Definitions and Considerations

- **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

Proprietary + Confidential





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

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 collect 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.

Code Snippet

Example of variable that contains unhashed data

- 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.

```
<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: "gb"
  }
};
</script>
```



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

Tag Configuration

Tag Type: Google Analytics: GA4 Event

Measurement ID: {{GA Measurement ID - Web}}

Event Name: {{Event}}

Event Parameters

Event Settings Variable: None

Event Parameter: user_data

Value: {{user_data}}

Triggering: E-commerce Events



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.

Google Analytics GA4 Event

Added in this workspace

Tag Configuration

Tag Type: Google Analytics: GA4 Event

Measurement ID: {{GA4_ID}}

Event Name: Test Event

Event Parameters

Event Settings Variable: None

Event Parameter: user_data

Value: {{User-Provided Data}}



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

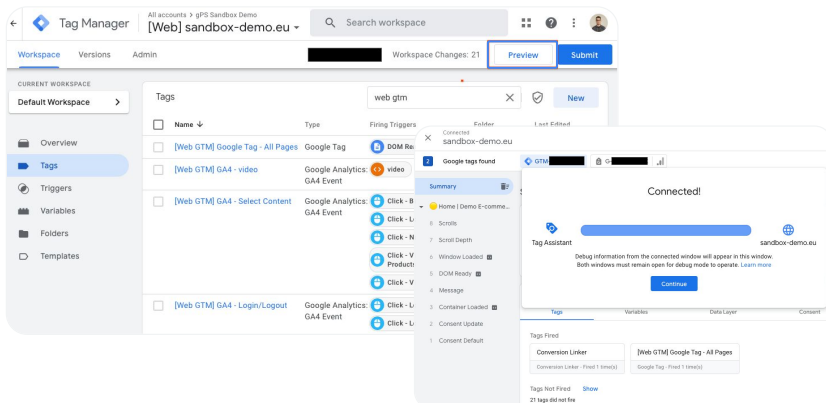
Select this option if User-provided data is available on the page at the Google tag config load.

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.



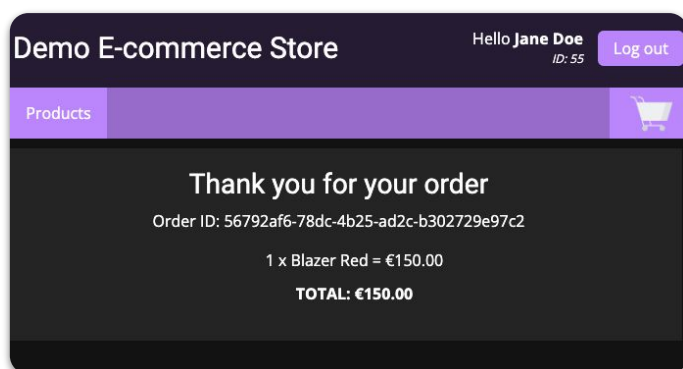
Implementation Validation: Client-side GTM

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.

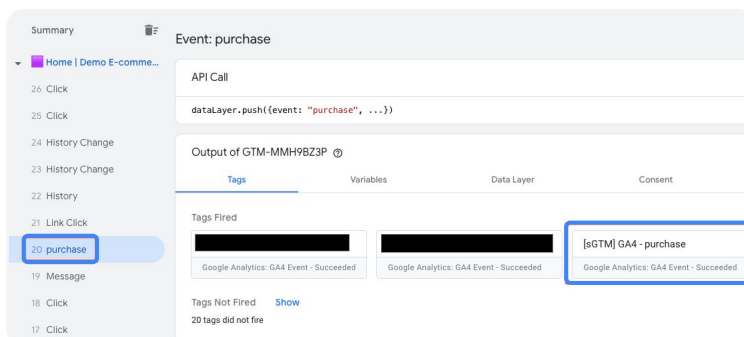


2. Perform test conversion (key event)

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

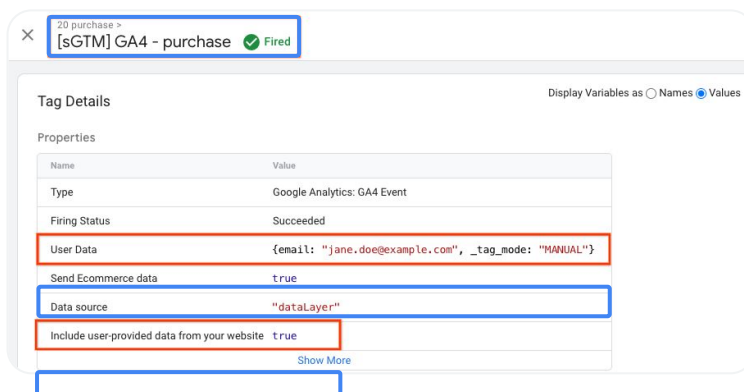
3. Check if the the key event was sent

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)



5. Validate using diagnostics



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 need to be made 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

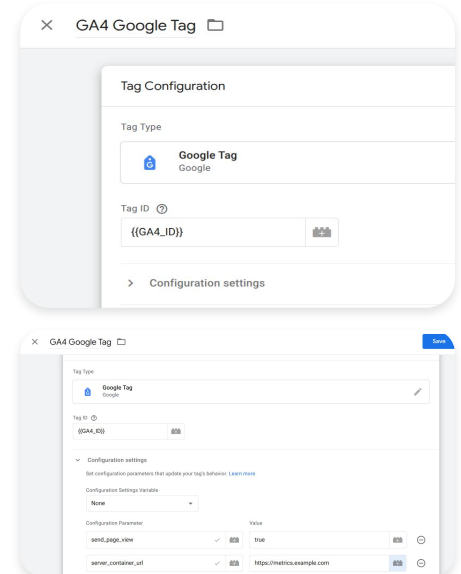


Step 2. Set up and Send User Data from Web Container to Server Container:

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.





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](#)

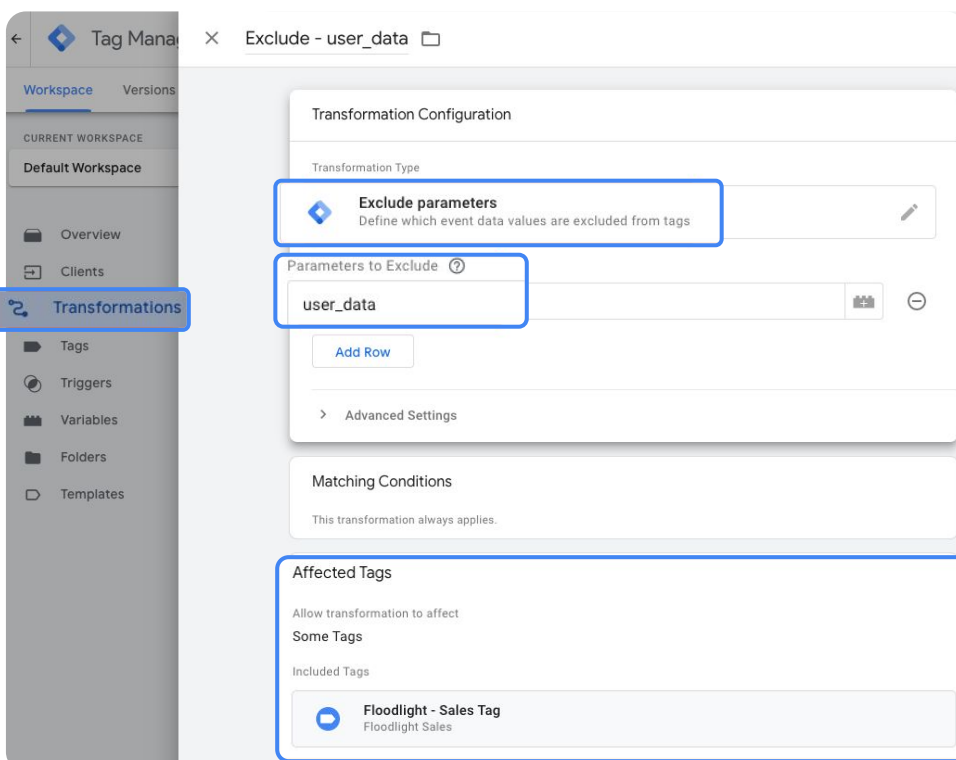
When `user_data` is sent to sGTM endpoint, Google tags (GA, Google Ads, Floodlight) will automatically pick data and forward to respective endpoint.

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.

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	sGTM
Numeric Event Parameter	epn.value	150
Event Parameter	ep.transaction_id	8df8ce2b-a9f2-4f1f-b6bc-adfb3ccf101
Engagement Time	et	1
Event Parameter	ep.user_data.email	jane.doe@example.com
Event Parameter	ep.user_data_tag_mode	MANUAL
ec_mode	ec_mode	c

Messages Where This Hit Fired

61 purchase



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 are already sending them hashed) and automatically forward to Google Analytics

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	sGTM
Numeric Event Parameter	epn.value	150
Event Parameter	ep.transaction_id	8df8ce2b-a9f2-4f1f-b6bc-adfb3ccf101
Engagement Time	et	1
Event Parameter	ep.user_data.email	jane.doe@example.com
Event Parameter	ep.user_data_tag_mode	MANUAL
ec_mode	ec_mode	c

Messages Where This Hit Fired

61 purchase

User-provided data implementation - Server-side tagging



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=13q3qPq2q5&npa=1&dma_cps=-&dma=1&tag_exp=0&cid=1232460501.1723466513&ecid=1673133751&u! =en-us&sr=2560x1440&fpplc=0&ur=&uaa=arm&uab=64&uafvl=Not)A%253BBrand%3B99.0.0.0%7CGoogle%2520Chrome%3B127.0.6533.100%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%2Fdt=Thank%20you%20%7C%20Demo%20E-commerce%20Store&en=purchase&_c=1&pr1=idblazer_red_m~nmBlazer~pr150~qt1~vablayer%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&richsstsse
```

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

Summary

Event: purchase

Output of GTM-WBV6XFD4

Request	Tags	Variables	Event Data	Console (5)
HTTP Request /g/collect?v=2&tid=G-9Q27TB04V8>m=45he47v0v9166979784z89166884015za200zb91668840158_p=1722...				
Client				
GA4				
Google Analytics: GA4 (Web) - Claimed				
Outgoing HTTP Requests from Server				
https://firestore.googleapis.com/v1/pr...	https://firestore.googleapis.com/v1/pr...	https://firestore.googleapis.com/v1/pr...	https://firestore.googleapis.com/v1/pr...	
POST - 200	GET - 200		PATCH - 200	
https://bigquery.googleapis.com/bigq...	https://bigquery.googleapis.com/bigq...		https://www.google-analytics.com/g/c...	
POST - 200	POST - 200		POST - 302	
Incoming HTTP Request				
/g/collect?v=2&tid=G-9Q27TB04V8>...				
GET - 200				



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

HTTP Request Details

Tag That Sent This Request

GA4 All Events

Google Analytics: GA4

Request

Request URL

POST

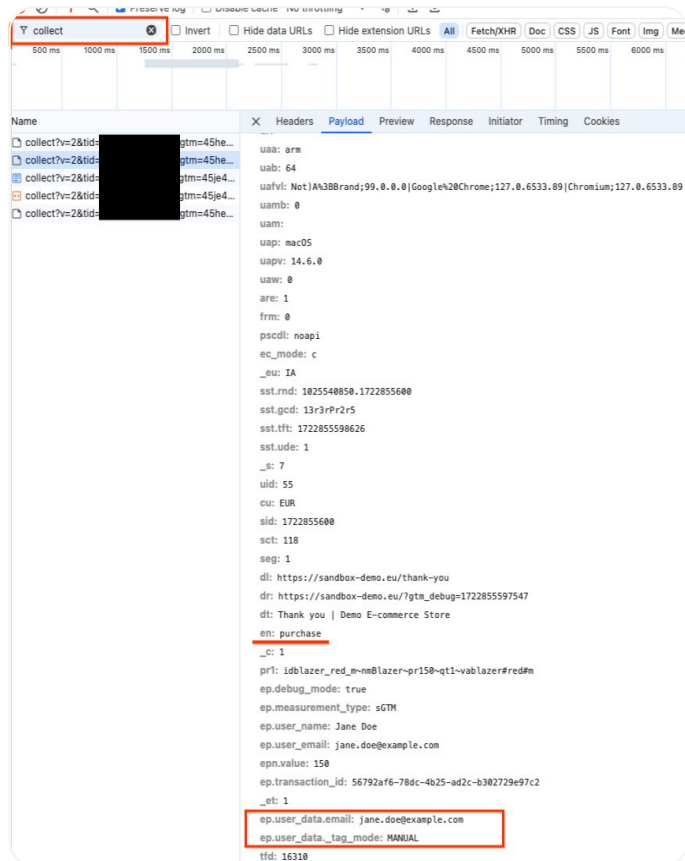
https://region1.google-analytics.com/g/collect?v=2&tid=G-9Q27TB04V8>m=45j91e4811v9166979784z89166884015z99182465548za200zb91668840158_p=1722949590382&nps=0&dma_cps=syphamo&dma=1&cid=sFoNX4AsKMCAiP14dsZ5vElfAsZ000wufiJyQYf6QE%3D.1712844086&ul=en-us&sr=1512x982&ur=DE-BE&uaa=arm&uab=64&uafvl=Not)A%253BBbrand%3B99.0.0.0%7CGoogle%2520Chrome%3B127.0.6533.89%7CChromium%3B127.0.6533.89&uamb=0&uam=&uap=mac05&uapv=14.6.0&uaw=0&are=1&pscdl=noapi&ec_mode=c&s=7&_uip=104.132.93.0&gcs=6111&ecid=2138393543&uid=55&cu=EUR&sid=1722949105&sct=120&dl=https%3A%2F%2Fsandbox-demo.eu%2Fthank-you&dr=https%3A%2F%2Fsandbox-demo.eu%2Fproducts&dt=Thank%20you%20%7C%20Demo%20E-commerce%20Store&em=tv.1~em.huC55WwXzE0S0H4ZSbhQU_vn08P0WhGicTqdMAzGEz0&gcd=13r3rPr2r5&tag_exp=95250753&frm=0&_eu=IABA&_c=1&seg=1&en=purchase&pr1=idblazer_red_m~nmblazer~pr150~gt1~vablazer%23red%23m&ep.debug_mode=true&ep.measurement_type=sGTM&epn.value=150&ep.transaction_id=f634d529-3afe-4739-aa9a-e02d8f057b2d



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.

Next: [Validate using diagnostics](#)



Data Manager API

Data Manager API for GA4 allows developers to send data directly to Google Analytics servers via HTTP POST or gRPC requests. It's designed to capture events from environments where standard client-side tracking (like gtag.js, Google Tag Manager, or the GA4F 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, GA4F SDK), rather than replacing them. It's crucial to have client-side tagging in place for most features to work correctly.



How does Data Manager API differ from Measurement Protocol & sGTM?

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

Feature	Data Manager API	Measurement Protocol API (Maintenance Mode only)	sGTM
Primary Mechanism	Supports direct HTTP POST and gRPC requests from your server/device to GA servers.	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/gRPC requests.	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. Data Manager API supports encryption on the payload for added security to your payloads.	You programmatically construct and send each event payload. Measurement Protocol will encrypt server-side.	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, or a Protocol Buffer (proto).	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 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 Data Manager API

The use of the Data Manager API's governed by the standard [Google Analytics Terms of Service](#). Additionally, there are specific policies and guidelines for using the Data Manager API 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 or encryption 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.
- **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.



Data Manager API for Web and App



API Setup Steps

Review the following requirements for setting up data manager API.

01

Set-up API access

Follow the instructions on the [Data Manager Developer site](#). [Enable the API](#) and [choose your authentication method](#).

Prerequisites

- 1) Have a [Google Cloud project](#).
- 2) Google account with `serviceusage.services.enable` in Google Cloud project ([View current access](#).)
- 3) The Google Cloud CLI [installed](#).

02

Install the Client Library

Use the [samples](#) on the developer website to choose which library works best for your setup.

Installation steps

On your local machine, install the client library of your choice:

[REST](#) .NET Go Java Node.js PHP Python Ruby



Requirements begin

These are required fields for the payload

03

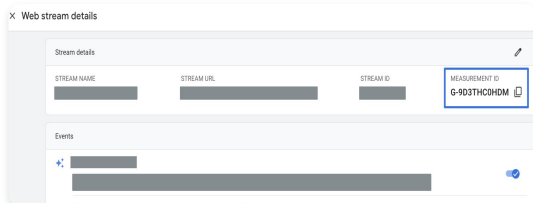
Required- Property ID

To grab your property ID associated with a an account, in the Google Analytics UI go to **Admin > Property Settings > Property> Property Details**

04

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**.



OR

Required-Firebase App ID (Not Stream ID)

To grab your Firebase App ID associated with a stream, go to the [Firebase Console](#) and log in with your Google account. Click on the specific Firebase project that contains your app. Click the **gear icon** (⚙️) next to "Project Overview" in the top-left navigation menu, and select **Project settings**. Stay on the default **General** tab and scroll down until you see the **Your apps** card. Select your specific app (Android, iOS, or Web) from the list. You will see the **App ID** clearly listed in its configuration details

05

Client ID or App Instance ID is required

Client ID (web) and App Instance ID (App) is a required field to send events via data manager API. Identifiers collected during online interactions are automatically joined with API events using `client_id` or `app_instance_id`. [Learn more here](#).



Data Manager API - Web



Pre-Hashing is *required* for Name, Street, Phone, and Email in Data Manager API

- Email, phone number, first & last name, and street address must be hashed using the **SHA-256** algorithm before being uploaded. See [User Data](#) for more details.
- 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_identifiers.email_address	jdoe@example.com	a8af8341993604f29cd4e0e5a5a4b5d48c575436c38b28abbfd7d481f345d5db
Phone	user_identifiers.phone_number	+11231234567	e9d3eef677f9a3b19820f92696be53d646ac4cea500e5f8fd08b00bc6ac773b1
First name	user_identifiers.address.given_name	john	96d363564cc3032521409cf22a85f
Last name	user_identifiers.address.family_name	doe	f92a11af918e3fb741df42934f3b58
Street	user_identifiers.address.address_line	123 New Rd	77f9a3b19820f92696be53d646ac4ce



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, remove "+" symbols, SHA - 256 format
Phone	+11231234567	Remove all non digit characters, add + prefix for country code, use E.164 format , SHA - 256 format
First name	john	No digits / symbol characters, lowercase, no leading / trailing spaces, SHA - 256 format
Last name	doe	No digits / symbol characters, lowercase, no leading / trailing spaces, no suffixes, SHA - 256 format
Street	123 New Rd	No symbol characters, lowercase, no leading / trailing spaces, SHA - 256 format
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, including 4 digit extension code can improve match rate
Country	GB	ISO 3166-1 alpha-2 standard country code
State / Province	ca	Administrative area (state or province) Use either the two-letter abbreviation (for example, ca) or the full name (for example, california)., No digits / symbol characters, lowercase, no leading / trailing spaces



Data Manager API Web Body Example

- The request is sent to For HTTP REST ([schema](#)), ingestion is done via a POST to `https://datamanager.googleapis.com/v1/adevents:ingest`.
- For gRPC ([schema](#)), ingestion is done via a gRPC call to `google.ads.datamanager.v1.IngestionService.IngestAdEvent`.
- You can add user provided information to your payload under the **userIdentifiers** object.
User_id is not required to send userIdentifiers but we highly recommend sending User ID for cross-device session and user attribution along with other identifiers available (gclid, gbraid, wbraid, Transaction ID, etc)

Example JSON body with user provided data via HTTP

```
{
  "destinations": [
    {
      "operatingAccount": {
        "accountType": "GOOGLE_ANALYTICS_PROPERTY", //required
        "accountId": "363XXXXXX" //required
      },
      "productDestinationId": "G-NXXXXXXX" //required-Measurement ID
    }
  ],
  "events": [
    {
      "eventName": "purchase" //required
      "eventTimestamp": "2025-06-10T23:42:33-05:00", //required
      "eventSource": "WEB",
      "clientId": "12345.67890", //required
      "userId": "user_01", //Optional, but recommended
      "userIdentifiers": [
        {
          "emailAddress": "a8af8341993604f29cd4e0e5a5a4b5d48c575436c38b28abbfd7d481f345d5db",
          "phoneNumber": "e9d3eef677f9a3b19820f92696be53d646ac4cea500e5f8fd08b00bc6ac773b1",
          "address": {
            "givenName": "96d363564cc3032521409cf22a85f",
            "familyName": "f92a11af918e3fb741df42934f3b58",
            "regionCode": "ca",
            "postalCode": "T2P 2M5",
            "addressLine": "77f9a3b19820f92696be53d646ac4ce",
            "city": "alberta",
            "administrativeArea": "calgary"
          }
        }
      ]
    }
  ]
},
{
  "consent": {
    "object (Consent)" // optional, but recommended
  },
  "validateOnly": boolean, // If set to True, will enable validation, but not submission
  "encoding": "HEX", // Required with userIdentifiers object (Encoding)
  "encryptionInfo": {
    "object (EncryptionInfo)" //Optional but recommended for userIdentifiers Objects
  }
}
```

Tip: You can set multiple destinations for your data if you have other Google product linked accounts, learn more [here](#)! Only One Google Analytics property destination is supported at this time, though.



Data Manager API App Body Example

- The request is sent to For HTTP REST ([schema](#)), ingestion is done via a POST to `https://datamanager.googleapis.com/v1/adevents:ingest`.
- For gRPC ([schema](#)), ingestion is done via a gRPC call to `google.ads.datamanager.v1.IngestionService.IngestAdEvent`.
- You can add user provided information to your payload under the **userIdentifiers** object.
User_id is not required to send userIdentifiers but we highly recommend sending User ID for cross-device session and user attribution along with other identifiers available (gclid, gbraid, wbraid, Transaction ID, etc)

Example JSON body with user provided data via HTTP

```
{
  "destinations": [
    {
      "operatingAccount": {
        "accountType": "GOOGLE_ANALYTICS_PROPERTY", //required
        "accountId": "363XXXXXX" //required
      },
      "productDestinationId": "1:123456789012:android:a1b2c3d4e5f67890" /required-FirebaseAppID
    }
  ],
  "events": [
    {
      "eventName": "purchase" //required
      "eventTimestamp": "2025-06-10T23:42:33-05:00", //required
      "eventSource": "APP",
      "appInstanceId": "f8a9e2d3b4c5d6e7f8a9b0c1d2e3f4a5", //required
      "userId": "user_01", //Optional, but recommended
      "userIdentifiers": [
        {
          "emailAddress": "a8af8341993604f29cd4e0e5a5a4b5d48c575436c38b28abbfd7d481f345d5db",
          "phoneNumber": "e9d3eef677f9a3b19820f92696be53d646ac4cea500e5f8fd08b00bc6ac773b1",
          "address": {
            "givenName": "96d363564cc3032521409cf22a85f",
            "familyName": "f92a11af918e3fb741df42934f3b58",
            "regionCode": "ca",
            "postalCode": "T2P 2M5",
            "addressLine": "77f9a3b19820f92696be53d646ac4ce",
            "city": "alberta",
            "administrativeArea": "calgary"
          }
        }
      ]
    }
  ]
},
{
  "consent": {
    "object (Consent)" // optional, but recommended
  },
  "validateOnly": boolean, // If set to True, will enable validation, but not submission
  "encoding": "HEX", // Required with userIdentifiers object (Encoding)
  "encryptionInfo": {
    "object (EncryptionInfo)" //Optional but recommended for userIdentifiers Objects
  }
}
```

Tip: You can set multiple destinations for your data if you have other Google product linked accounts, learn more [here](#)! Only One Google Analytics property destination is supported at this time, though.



Encryption in Data Manager API

Overview & Support

- The Data Manager API supports encrypting data for Google Analytics using Google Cloud Key Management Service (KMS).
- It does **not** currently support AWS.
- Encrypted fields are wrapped in a **wrapped_key** field. Exactly one populated field is required because only one is currently supported.

Google Cloud KMS Architecture

- **Key Encryption Key (KEK):** Acts as the "root of trust" using a digital envelope encryption scheme.
- The KEK is used strictly to wrap local Data Encryption Keys (DEKs) and never leaves the secure hardware security modules (HSMs) of Cloud KMS.
- **Security Benefit:** By hosting the KEK in KMS, raw key exposure is eliminated. Applications only interact with the key via resource names, meaning primary cryptographic material remains secure even if the application server is compromised.

Data Manager API Payload Structure

The **encryptionInfo** block is used to deliver the cryptographic context. When **encryptionInfo** is present, the API ignores standard **HEX** or **BASE64** hashing fields.

JSON Payload Example (**IngestEventsRequest**)

```
{
  "destinations": [{
    "operatingAccount": {
      "accountType": "GOOGLE_ANALYTICS_PROPERTY",
      "accountId": "55555555"
    },
    "productDestinationId": "G-XXXXXXXX"
  }],
  "encryptionInfo": {
    "kmsKeyName":
"projects/my-project/locations/us-central1/keyRings/ga-encryption-ring/cryptoKeys/ga-root-kek",
    "wrappedKey": "BASE64_ENCODED_WRAPPED_DEK"
  },
  "events": [{
    "eventTimestamp": "2026-05-10T13:00:00.000Z",
    "userData": {
      "userIdentifiers": [
        { "emailAddress": "BASE64_ENCRYPTED_PII_STRING" }
      ]
    }
  ]
}]
```

This structure ensures a higher tier of privacy-centric data ingestion compared to standard hashing, as it allows for secure matching without the irreversibility of a one-way hash.



Validation data manager API

1. Set **validateOnly** = true. This will not send the request to the server and only check for syntax, invalid values, and validation errors.

Failure Response example

In this example, an emailAddress containing a plain text string instead of a hex encoded value produces the following response.

Ensure that the email has been hashed and encoded in SHA256 Hex and send the request again.

```
"error": {
  "code": 400,
  "message": "There was a problem with the request.",
  "status": "INVALID_ARGUMENT",
  "details": [
    {
      "@type": "type.googleapis.com/google.rpc.ErrorInfo",
      "reason": "INVALID_ARGUMENT",
      "domain": "datamanager.googleapis.com"
    },
    {
      "@type":
"type.googleapis.com/google.rpc.BadRequest",
      "fieldViolations": [
        {
          "field":
"events.events[0].user_data.user_identifiers",
          "description": "Email is not hex encoded.",
          "reason": "INVALID_HEX_ENCODING"
        }
      ]
    }
  ]
}
```

1. Set **validateOnly** = false. This will send the request to the server. It will return a requestId. This value can be used to validate against the status API ([RetrieveRequestStatusRequest](#))
2. A successful request has a [Status](#) with code equal to 0 (enum value OK, HTTP response 200 OK), and returns a [IngestEventsResponse](#).
3. If a request isn't successful, modify the request to address the error and send the request again.

HTTP Request Example

GET

`https://datamanager.googleapis.com/v1/requestStatus:retrieve?requestId={currentRequestID}`

HTTP Response

[RequestStatusPerDestination](#)

```
{
  "destination": {
    object (Destination)
  },
  "requestStatus": enum
(RequestStatus),
  "errorInfo": {
    object (ErrorInfo)
  },
  "warningInfo": {
    object (WarningInfo)
  },
  "eventsIngestionStatus": {
    object (IngestEventsStatus)
  },
}
```



Realtime Report Validation

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

Event count by Event name

#1 page_view

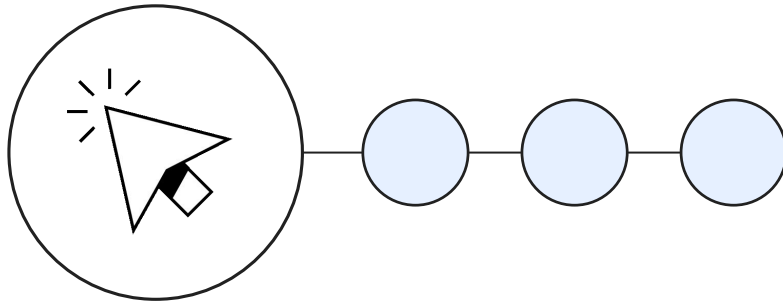
2

33.33%

EVENT NAME	EVENT COUNT
page_view	2
<u>purchase</u>	1
select_content	1
user_engagement	1
view_item_list	1

1 – 5 of 5 < >

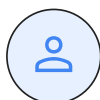
Next: [Validate in the Google Admin UI using diagnostics](#)



FAQ

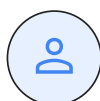


User-provided data: Collection & Privacy



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.
- **Data Manager API (DM API):** 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.



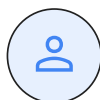
Is User-ID required to send UPD?

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, Data manager API).



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)



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.



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.](#)



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.



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.



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.