



Google Analytics for Firebase - Migration Guide [from legacy Google Analytics App views - GA SDK]

[Introduction & Important Notes](#)

[Migrating to new app analytics reports](#)

1. [Define business objectives and apps architecture](#)
2. [Understand Google Analytics for Firebase limits, automatically collected events/user properties and reports](#)
3. [Design your data model](#)
 - [Google Analytics for Firebase Events](#)
 - [Google Analytics for Firebase User Properties](#)
 - [Migration Data modeling Questions](#)
4. [Implement your GA for Firebase project](#)
 - [Create a Firebase account](#)
 - [Add an app in your Firebase project](#)
 - [Additional implementation adjustments](#)
 - [Events \(LogEvents\)](#)
 - [User properties](#)
 - [Other implementations](#)
 - [Hybrid Apps](#)
 - [Google Tag Manager](#)
 - [GTM v5 \(with Firebase\)](#)
 - [GTM v4 or older \(no Firebase\)](#)
 - [Migrating GTM v4 or older to GTM v5 \(w/ Firebase\)](#)
5. [Check the Firebase reports](#)
 - [Google Analytics](#)
 - [Firebase](#)
 - [Events, conversions, user properties, and audiences](#)
6. [Explore the GA for Firebase integrations](#)

Introduction & Important Notes

[Firebase](#) is an integrated app development platform with many different modules, one of which is dedicated to app analytics. This is called [Google Analytics for Firebase](#).

Our latest reports represent a new approach to app analytics which better reflects the way that users interact with apps. This means that these reports have different concepts and functionalities when compared to the original app analytics reports in Google Analytics via the legacy Google Analytics Services SDK.

In October 2019, we will start to [sunset Google Analytics mobile-apps reporting](#) based on the legacy Google Analytics Services SDKs for [Android](#) and [iOS](#). Google Analytics 360 customers' properties are not affected on this specific deprecation this time.

In any case, we strongly advise installing GA for Firebase as well. There are certain features, such as the ability to track and share app conversions from Google Ads campaigns, that are unavailable within the legacy GA Services SDK.

Google Analytics for Firebase app reports include the following features:

- **Free:** Our latest app reports are available at no cost
- **Unlimited:** Enjoy unlimited event collection
- **Easy,** out-of-the-box implementation: Automatically report a number of important events and collect several user properties without writing any custom code.
- **Integrated:** Integrates with many Firebase features, such as Cloud Messaging, Remote Config, Predictions, A/B Testing, Dynamic Links, Cloud Functions and Crashlytics
- **Customizable:** Define and capture events and properties that are specific to your user base.
- **App Data where you need it:** Available in both the Google Analytics interface and the Firebase console so everyone in your organization can access and analyze app data including summary dashboards, event reports, audience data, attribution, funnels, retention reporting, and user properties
- **Cross-Platform:** Supported on Android and iOS and de-duplicates user data across OS (if User ID is enabled and implemented).

Migrating to new app analytics reports

This guide is written for users familiar with Google Analytics (web and app analytics solutions) who are considering new implementations or implementation updates to their iOS and Android apps. There are different migration techniques/preferences and this guide is a suggestion about how you could plan your migration.

Google Mobile Analytics Solutions:

- **Google Analytics Services SDK** - Legacy Google Analytics for Apps SDK. Synonymous with GA for web in terms of functionality.
- **Google Analytics for Firebase** - Uses the Firebase SDK and integrates seamlessly with Firebase, BigQuery, & other Google tools.
- **Google Tag Manager + Firebase** - Firebase SDK with GTM for Mobile SDK on top to send data to a Google Analytics web property and elsewhere.

Terminology notes:

Firebase (or Google Analytics for Firebase) reporting: The reports are populated with data from the Firebase console which collects data with the Firebase SDK. These reports are visible in the Firebase console and Google Analytics interface (if linked with GA for Firebase console).

Legacy GA app views reports: The reports populated by data collected using the Google Analytics Services SDK or as a result of data forwarded from Google Tag Manager V5 (GTM w/ Firebase SDK) using the Universal Analytics tag. These reports will already be familiar to anybody who has used app reporting in Google Analytics prior to the launch of GA for Firebase.

1. Define business objectives and apps architecture

As a first step, review your key business objectives and how your apps are structured and implemented.

You should list specific objectives. Examples include:

- Increase installs
- Increase in-app purchases
- Identify if users are engaging on a certain action (make sure to list the specific actions)
- Identify trends of specific user types/context (make sure to list the specific conditions)
- Identify most viewed screens

Additionally, list your current app structure/implementation conditions. Examples include:

- OS versions (e.g. Android/iOS)
- Implementation frameworks (e.g. Unity, Android studio, xCode)
- Environment strategy (e.g. Dev, Integration, QA, Preprod, Prod)

The details about the objects, app structure/implementation should shape a "data collection specification" for you. This specification should help you anticipate additional implementation actions like adding additional Firebase SDK code lines and/or help you structure your Firebase project and apps.

2. Understand Google Analytics for Firebase limits, automatically collected events/user properties and reports

It's recommended that you review the current Firebase limits and what it already collects automatically. In doing so, you may find that some information you need is already automatically collected and reported by a regular Google Analytics for Firebase implementation without the need of additional implementation adjustments.

Identifying items from your "data collection specification" covered by the automatically collected events/user properties should help you avoid unnecessary implementation work. Additionally, if your specification indicates a chance to reach some of the Firebase limits, you might need to consider different approaches/modeling or to prioritize what you need as data measured by your Google Analytics for Firebase account.

References to review:

- [Firebase collection and configuration limits](#)
- [Automatically collected events](#)
 - [Events and parameter details](#)
- [Automatically collected user properties](#)
- [Reports and configuration](#)

3. Design your data model

The legacy GA app views reports (that collect data with the legacy [iOS/Android](#) GA SDKs) have a data model based on elements like [Sessions](#), [Hits](#) and [Custom dimensions/metrics](#). In comparison, the Google Analytics for Firebase data model is based primarily on [Events](#) (and [event parameters](#)) and [User properties](#).

It's recommended that you define a new data model for your Google Analytics for Firebase implementation given the considerable differences in comparison with the legacy GA App views data model.

Here's a summary of how some of the key entities compare between the legacy Google Analytics App views and Google Analytics for Firebase:

Legacy Google Analytics App views	Google Analytics for Firebase
User Sessions Screenviews Goals Ecommerce Custom dimensions/metrics (hit level) Event	Events (and event parameters)
Custom dimensions/metrics (user level) Location (Country) Language	User properties

Google Analytics for Firebase Events

Whereas legacy Google Analytics App views reports use many different hit types (screenview, event, social, ecommerce, etc), all interactions for the new reports are collected as events.

In the legacy Google Analytics App views reports, events are typically named using fields called Event Category, Event Action, and Event Label (optionally, Event Value and Custom Dimensions may be included).

In the Google Analytics for Firebase reports, all events are categorised using an Event Name (required) and Event Parameters. These values are provided using key-value pairs in the implementation.

To aid with naming convention consistency, there are many suggested events and associated parameters built into the SDK. Where possible, it is recommended to use the suggested event names and parameter combinations.

It is also possible to create custom event names and values if you have specific requirements for your app analytics which are not met by the suggested options. Implementation detail for suggested and custom events can be seen in the resources below:

- [Events: All apps](#)
- [Events: Retail/Ecommerce](#)
- [Events: Jobs, Education, Local Deals, Real Estate](#)
- [Events: Travel \(Hotel/Air\)](#)
- [Events: Games](#)

Google Analytics for Firebase User Properties

In legacy GA app views, you might use User Scope Custom Dimensions to capture custom user properties (for example, Customer Tier). In the Google Analytics for Firebase reports, these values are collected using user properties.

Note again that many user properties (first open time, gender, etc) are collected automatically. Additional custom user properties may be set by additional implementation adjustments.

Migration Data modeling Questions

These are comments to some questions you might have on this stage:

A) We analyze via screenviews, but the same concept doesn't exist in GA for Firebase, instead it's focused on events. How can we analyze?

- Analytics automatically tracks some information about screens in your application, such as the class name of the UIViewController or Activity that is currently in focus.
- A possible approach is to analyze the event "screen_view" and check the "firebase_screen_class" (Screen class) parameter
- If you want to override the class names and use custom screen_names, it's possible to adjust [manual track screens collection](#).

B) We send 50+ Custom Dimensions with every hit, how does that translate to 25 parameters per event?

- User level Custom Dimensions (from the legacy GA app views) map to User Properties (on the GA for Firebase)
- Hit level Custom Dimensions (from the legacy GA app views) can be implemented as a Custom Event (on the GA for Firebase)
- You might need to prioritize the information (events) you want to collect if you see you might be reaching the 500 unique event maximum
- Possible workaround: group some of the values. For example: if you have events for A,B,C,D,E events, you might consider fewer events like A_B and C_D_E.

C) We already have a robust Category/Action/Label event hierarchy setup with hundreds/thousands of events, how can we migrate?

- Recommendation is to rethink implementation plan and data structures and retag in new methodology
- Backup/temporary solution could be to use a custom event to capture all previous GA Category/Action/Label data. Recommend using [GTM v5](#) to map these to our suggested schema.

4. Implement your GA for Firebase project

General instructions for the Google Analytics for Firebase implementation are available in the following article: [Get started with Google Analytics app reporting](#)

Summary of necessary steps:

Create a Firebase account

Go to [firebase.google.com](#) and click on "+ Add a project" to create a Firebase account. You will be asked to define a project name and review some project settings. Click on "Create project" and opt-in to Google Analytics.

It's recommended to ensure that all apps within a project are platform variants of the same application from an end-user perspective. It's advisable to connect the Android, iOS, and web versions of the same app or game to the same Firebase project. All the apps in a project generally share the same Firebase resources (database, storage buckets, etc.).

Generally, if a set of apps don't share the same data and configurations, strongly consider connecting each app to a different Firebase project.

See more details on the [Understand Firebase projects](#) article.

Add an app in your Firebase project

1. Sign in to Firebase, then open your project.
2. Click , then select Project settings.
3. In the Your apps card:
 - If you have no apps in your project: Click the icon for the app's platform, then follow the setup workflow.
 - If you already have apps in your project: Click Add app, then follow the setup workflow.

Note that you can also add an app to your project from the console's main overview page of your Firebase project.

The process to include an app in your Firebase project will lead you through the necessary basic implementation steps:

- a) Register the app (in the Firebase project)
- b) Download the config file (and add it to your app project)
- c) Add the Firebase SDK (in your app source code)
- d) Add the initialization code (in your app source code)
- e) Run your app to verify installation

Additional technical implementation details (per framework):

- [iOS](#)
- [Android](#)
- [Unity](#)
- [C++](#)

Additional implementation adjustments

If your "data collection specification" lists the need for data that is not automatically collected, you will need to do additional implementation adjustments for events and/or user properties.

Events (Log Events)

Check the following implementation guidelines (per framework):

- [iOS](#)
- [Android](#)
- [Unity](#)
- [C++](#)

User properties

Check the following implementation guidelines (per framework):

- [iOS](#)
- [Android](#)
- [Unity](#)
- [C++](#)

Other implementations

Hybrid Apps

If your app is hybrid (an app working with WebViews), you may consider the implementation instructions described in the following guides:

- [iOS](#)
- [Android](#)

Google Tag Manager

Full implementation guidelines details:

<https://support.google.com/tagmanager/answer/7035192>

GTM v5 (with Firebase)

You may have GTM v5 already implemented but just not working with the Google Analytics for Firebase tags on your container. IF that's the case, you should be able to skip some of the implementation steps for GTM and mostly focus on the following: [Create the Google Analytics \(Firebase\) tag](#)

GTM v4 or older (no Firebase)

If you're working with Google Tag Manager v4 or older, you will need to also adjust your Google Tag Manager implementation.

This is a summary of the steps needed to implement the Google Analytics for Firebase data collection with Google Tag Manager:

1. Get your app set up with Firebase:
 - Install the Firebase SDK ([Android](#) | [iOS](#)).
 - In the Firebase console, add your app to your Firebase project.
 - Add Firebase Analytics to your app ([Android](#) | [iOS](#)).
2. Create a [Tag Manager account](#) (if you don't have any) and configure a [Tag Manager container](#).
3. Add Tag Manager to your project ([Android](#) | [iOS](#)).
4. [Create the Google Analytics \(Firebase\) tag](#)

Migrating GTM v4 or older to GTM v5 (w/ Firebase)

Existing mobile containers cannot be migrated to the new container type. Certain functionality from legacy containers is not functional in the new Firebase containers (i.e. user-defined [Value Collection](#) variables, [Data Layer](#) variables, and [Content Experiments](#)).

Generally, it's recommended to create a new container and reconfigure your tags, variables and triggers.

Check the following implementation guidelines (per framework) for more details:

- [iOS](#)
- [Android](#)

DataLayer

The DataLayer implementation is likely something you will need to re-model and re-implement when transitioning from a legacy GTM SDK implementation to the GTM v5 (+Firebase) implementation.

In Google Tag Manager V5 for GA for Firebase, the implementation methodology is different. There are three key entities to consider:

- Event names
- Event parameters
- User properties

Each event implemented will automatically be sent to the new app reports in Google Analytics per the Firebase implementation documentation. From a Tag Manager perspective, event names are used as the primary trigger for tags (the same as in legacy Tag Manager). As in legacy Tag Manager, additional filters may also be used.

One difference to consider is in event parameters and user properties. In Google Tag Manager V5, user properties behave most similarly to the persistent legacy dataLayer variables. Once a user property has been set, it may be accessed in any subsequent tag fired in Tag Manager (e.g. it may be set as a Custom Dimension value in a Google Analytics tag).

Event level parameters, on the other hand, *must be sent with each event* in order for them to be reported against that event in the latest app reports or in order for them to be accessed in any tags associated with the event name in Tag Manager.

Consider the following set of Firebase commands as an example:

```
// Set user property
mFirebaseAnalytics.setUserProperty("customer_type", "Gold");

// Log event 1
Bundle params = new Bundle();
params.putString("currency", "EUR");
params.putString("screen_type", "Product");
params.putString("locale", "UK");
mFirebaseAnalytics.logEvent("example_event_1", params);

// Log event 2
Bundle bundle = new Bundle();
bundle.putString(FirebaseAnalytics.Param.ITEM_ID, "123");
bundle.putString(FirebaseAnalytics.Param.ITEM_NAME, "Example Name");
bundle.putString(FirebaseAnalytics.Param.CONTENT_TYPE, "image");
mFirebaseAnalytics.logEvent(FirebaseAnalytics.Event.SELECT_CONTENT, bundle);
```

In this example, if you trigger a tag in Google Tag Manager for each event, you will have access to the following variables:

Event 1

- Customer Type
- Currency
- Screen Type
- Locale
- Event name (example_event_1)

Event 2

- Customer Type
- Item ID
- Item Name
- Content Type
- Event name (SELECT_CONTENT)

5. Check the Firebase reports

Google Analytics

To see reports in Google Analytics:

1. Sign in to Google Analytics..
2. **Navigate to your app property.** The report navigation will update to reflect the mobile app reports.
If you don't see your app listed in the Property selection menu, you need to add it.
 - Select Create new property from the Property menu and complete the process as instructed
3. Select the report you want from the report navigation (e.g., Dashboard, Events, Audiences).

Firebase

To see reports in Firebase:

1. Sign in to your Firebase account at firebase.google.com.
2. On the Welcome page, select your app.
3. In the left navigation, click Analytics.
4. Click the tab for the report you want (e.g., Dashboard, Events, Audiences).

Events, conversions, user properties, and audiences

There are four building blocks to app analytics: events, conversions, user properties, and audiences.

- *Events* are triggered when users take actions in your app. (already discussed under 3.Data modeling)
- *User properties* are characteristics of your users. (already discussed under 3.Data modeling)
- *Conversions* are the events that matter the most to your business. Think of them as your key performance indicators (KPIs). Marking an event as a conversion lets you send these conversions to the ad networks you already work with, including Google and external networks.
- *Audiences* are segmented groups of users defined by a combination of events and user properties. For example, you might want to see metrics for just your users from a specific country.

You may re-check the [Reports and configuration](#) articles for details about the available reports.

6. Explore the GA for Firebase integrations

Google Analytics for Firebase have integrations with other Google products:

- BigQuery
Allows you access your raw, unsampled event data along with all of your parameters and user properties in BigQuery if you have the Cloud [Blaze plan](#).
More information on the [Link Firebase to BigQuery](#) article.
- Google Ads
Allows exporting event (conversions) and audiences to Google Ads accounts for Ads bidding optimization and ads targeting (remarketing).
More information on the [Link Google Ads to Firebase](#) article.
- AdMob
Allows Google Analytics for Firebase data in AdMob reports.
More information on the [Link AdMob to Firebase](#) article.
- Display & Video 360 and Campaign Manager
Allows conversion attribution details from Display & Video 360 and Campaign Manager campaigns.
More information on the [Link Display & Video 360 and Campaign Manager to Firebase](#) article.
- Google Play
Allows inclusion of Android in-app details on your Firebase reports.
More information on the [Link Google Play to Firebase](#) article.