Device-mode	Cloud-mode
Web	Web
Mobile	Mobile
Server	Server

Firebase is Google's platform for mobile apps. The Segment Firebase destination requires that you bundle the Firebase SDK with your project. The Segment-wrapped destination code then runs on the user's device, and sends its tracking calls to the Firebase API endpoints, and a copy to Segment for archiving.

As of October 2019, Firebase replaced the legacy version of Google Analytics Classic for mobile devices. (If you used Google Analytics for mobile, see the migration guide.)

Segment's Firebase destination code is open source and available on GitHub. You can view these repositories:

- Android

Android

iOS

Kotlin

Swift

Consent mode

Google enforced consent on March 6, 2024 for European Economic Area (EEA) users. Learn more about consent mode and how to set it up.

Getting Started on Android

To start sending data to Firebase Analytics from your Android project, you'll need to follow a few simple steps:

1. Register your mobile app with Firebase at `https://console.firebase.google.com`
2. Once your app is registered, download the `google-services.json` file.
3. Copy the file to your Application's `app` folder. This file contains all necessary configurations and cannot be used across multiple apps. If you're configuring Firebase for other apps, create a new view in your Firebase console and download a unique `google-services.json` file for each.
4. Add the modules to `build.gradle`. Add the Segment-Firebase SDK and apply the Google Services plugin at the end of the file:

```
buildscript {
    dependencies {
        // Add these lines
        implementation 'com.segment.analytics.android:analytics:4.+'
        implementation 'com.segment.analytics.android.integrations:firebase:+@aar'
    }
}

// Add to the bottom of the file
apply plugin: 'com.google.gms.google-services'
```

Note: The Firebase SDK requires android resources which are available on `aar` packages. Use the `aar` package when adding the Segment-Firebase SDK.

5. Add the modules to the Project-level `build.gradle` file. Add Google Services dependency and their Maven repo location to repositories:

```
buildscript {
 dependencies {
   // Add this line
   classpath 'com.google.gms:google-services:3.1.0'
 }
}

allprojects {
 repositories {
   // Add this line
   maven { url 'https://maven.google.com' }
 }
}
```

Add these permissions to your AndroidManifest.xml:

```
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

6. Finally, register the dependency with the Segment SDK in your application subclass, as in our Android library documentation.

```
Analytics analytics = new Analytics.Builder(context, writeKey)
  .use(FirebaseIntegration.FACTORY)
  ...
  .build();
```

Firebase periodically updates the Android configuration requirements for loading their SDK in your app. To check if your Android configuration is compatible with for your version of Firebase, check Google's Firebase release notes.
You can also check the Segment-Firebase changelog to find the version of the Firebase SDK that Segment requires in each of the Segment-Firebase SDK version. For example, Segment-Firebase 1.3.1 includes Firebase Core 17.0.1 as a dependency.
By default, Segment bundles only `Firebase/Core` which is Firebase's mobile analytics offering. You can see the other available Firebase dependencies and features in the Firebase documentation.

Getting Started on iOS

1. Register your app in the Firebase console and add the `GoogleService-Info.plist` to the root of your Xcode project.
2. Add the following dependency to your Podfile:

```
pod 'Segment-Firebase'
```

3. After adding the dependency, import the integration:

```
#import <Segment-Firebase/SEGFirebaseIntegrationFactory.h>
```

4. Finally, register the dependency with the Segment SDK:

```
[config use:[SEGFirebaseIntegrationFactory instance]];
```

By default, Segment only bundles `Firebase/Core` which is Firebase's Analytics offering. You can see the other available Firebase pods and features here.

Consent mode

Consent mode is a feature provided by Google in the context of its products, particularly the Gtag library and Google Analytics. As of March 6, 2024, Google announced that consent mode must function for European Economic Area (EEA) users, otherwise data from EEA users won't process.

Consent mode in the Gtag library and Google Analytics is designed to help website owners comply with privacy regulations, such as the General Data Protection Regulation (GDPR) in the European Union. It allows website owners to adjust how these tools use and collect data based on user consent.

With consent mode, you can configure your website to dynamically adjust the tracking behavior of the Gtag library and Google Analytics based on the user's consent status. If a user provides consent to data processing, both the Gtag library and Google Analytics can collect and use that data for analysis. If a user doesn't provide consent, both tools limit data collection to essential functions, helping businesses respect user privacy preferences.

Consent mode may involve updates to your sources outside of Segment, such as incorporating a consent management system for consent functionality.

To set up consent mode for Google Firebase:

1. Update your app's SDK to a version that supports consent mode v2.
    - Android apps must use Firebase Android Analytics SDK version 21.5.0 or later.
    - iOS apps must use Firebase Apple SDK version 10.17.0 or later.
2. Set up consent mode for your app if you haven't already set it up.
    - Android: Set up consent mode for Android apps
    - iOS: Set up consent mode for iOS apps
3. If you already set up consent mode for your app, upgrade it to consent mode v2.
    - Android: Upgrade to consent mode v2 for Android apps
    - iOS: Upgrade to consent mode v2 for iOS apps

Setting up Firebase with Analytics-React-Native

If you use Segment's older React Native source library, you must explicitly bundle the mobile SDKs for both iOS and Android with your project.

1. Use yarn to add the `analytics-react-native-firebase` SDKs. (`@segment/analytics-react-native-firebase`)
2. Add `import` statements to your code so you can access the SDKs.
3. Go to the `await analytics.setup` configuration in your code, and find (or add) the `using:` item. Add "Firebase" to the list of device-mode destinations in the `using` item.
4. Change to your iOS directory and run `pod install`.
5. Add the `analytics-react-native-firebase` module to your `build.gradle` file. (See Step.4 of Getting Started on Android)

For React Native 2.0 you can reference the install guide.

Identify

When you call `identify` Segment will map to the corresponding Firebase Analytics calls:

- If there is a `userId` on your `identify` call, Segment triggers `setUserId` using the Firebase SDK
- If there are traits included, Segment will set user properties for each trait you include on the `identify` call

You can use these traits to create audiences and views to analyze your users' behavior.

Note: Google prohibits sending PII to Firebase unless "robust notice" is given to your app users. For iOS apps, some Analytics features, such as audiences and campaign attribution, and some user properties, such as Age and Interests, require the AdSupport framework to be enabled.

Learn more about Firebase's reporting dashboard here.

Firebase has strict requirements for User Property names; they must:

- Begin with a letter (not a number or symbol, including an underscore)
- Contain only alphanumeric characters and underscores
- Be no longer than 40 characters

User Property values must be fewer than 100 characters.

You are limited to 25 unique user properties per Firebase Console.

Segment automatically:

- Trims leading and trailing whitespace from user property names
- Replaces spaces with underscores

- Trims property names to 40 characters (Android only)

Firebase automatically collects these user properties.

Track

When you call `track` Segment will log the event with Firebase. Firebase automatically tracks the events listed here and it will still do so when bundling with Segment.

Firebase has a limit of 500 distinctly named events so it pays off to be intentional in what you track.

When you call `track`, Segment maps from the Segment spec to those that match Firebase's spec. For anything that does not match, Segment will pass the event to Firebase as a custom event. Custom parameters cannot be seen directly in the Firebase Analytics dashboard but they can be used as filters in **Audiences**.

Like with user properties, Segment will perform the following transformations on both your event names and event parameters. Unlike user properties, you do not need to pre-define event parameters in your Firebase dashboard.

- Trims leading and trailing whitespace from property names
- Replaces spaces with underscores
- Trims property names to 40 characters (Android only)

Event parameter values must be fewer than 100 characters.

Event Mappings

Segment adheres to Firebase's semantic event specification and maps the following Segment specced events (left) to the corresponding Firebase events (right):

SEGMENT EVENT	FIREBASE EVENT
Products Searched	search
Product List Viewed	view_item_list
Product Viewed	view_item
Product Clicked	select_content
Product Shared	share
Product Added	add_to_cart
Product Added To Wishlist	add_to_wishlist
Checkout Started	begin_checkout
Promotion Viewed	present_offer
Payment Info Entered	add_payment_info
Order Completed	purchase
Order Refunded	purchase_refund

Property Mappings

Segment maps the followed Segment specced properties (left) to the corresponding Firebase event parameters (right):

SEGMENT PROPERTY	FIREBASE PROPERTY	ACCEPTED VALUE(S)
category	item_category	(String) “kitchen supplies”
product_id	item_id	(String) “p1234”
name	item_name	(String) “Le Creuset pot”
price	price	(double) 1.0
quantity	quantity	(long) 1
query	search_term	(String) “Le Creuset”
shipping	shipping	(double) 2.0
tax	tax	(double) 0.5
total	value	(double) 3.99 or (long) 3.99
revenue	value	(double) 3.99 or (long) 3.99
order_id	transaction_id	(String) “o555636”
currency	currency	(String) “USD”

Passing Revenue and Currency

Ecommerce events containing “revenue” or “total” must also include the appropriate ISO 4217 “currency” string for revenue data to populate to the Firebase dashboard. If a “currency” value is not included, Segment default to “USD”.

```
Properties properties = new Properties()
        .putValue("orderId", "p966540")
        .putValue("revenue", 25.00)
        .putCurrency("USD");


Analytics.with(this).track("Order Completed", properties);
```

Screen

Segment doesn’t map screen events to Firebase - that’s because Firebase’s SDK collects screen information out of the box for you.

For Android, Segment passes contextual screen information into each screen view on each activity’s `onResume` callback. To ensure that screen names are labeled properly, Segment recommends adding a `label` value to each of your activities in your app’s `AndroidManifest.xml` file. At the moment, Firebase does not allow disabling automatic screen tracking for Android.

For iOS, you can configure `recordScreenViews` which will automatically track screen views, or pass in a screen manually using a screen call. You should be able to disable the Automatic Screen reporting by adding the plist flag `FirebaseScreenReportingEnabled` to `Info.plist` and set its value to `NO` (Boolean).

Google Analytics for Firebase iOS does NOT support the case of manual-only screen reporting. Firebase only supports automatic + manual screen reporting or no screen reporting at all.

Firebase Dynamic Linking (iOS only)

Firebase Dynamic Links are smart URLs that can change behavior dynamically depending on the platform where the user clicks them. Use them in web, email, social media, referral and physical promotions to increase user acquisition, retention and lifetime value. Key features include ability to survive app installs, controlling user

experience depending on what platform they access the link on and knowing which content and campaigns are working using tracking in the Firebase console. Check out Firebase's Docs here.

To use Firebase Dynamic Links, add the below to your podfile.

```
pod 'Firebase/DynamicLinks'
```

Then, enter the deep link URL scheme in your Segment Firebase destination settings. Here's a sample app delegate that shows how to implement the Dynamic Linking Logic.

Conversion Tracking and Adwords Conversions

Firebase is Google's recommended method for reporting conversions to Adwords. To use Firebase, track the conversion events as you normally would with Segment and Segment will send them through to Firebase.

Troubleshooting

Firebase has great logging. If you are having any issues, you can enable debug mode as outlined in Google's Debug events documentation.

Changes from iOS v1 to v2 Beta

Segment has been working hard bringing the Firebase iOS beta integration up to date with the native Firebase SDK. The new version 2.0.0-beta has a number of changes that you should be aware of before you upgrade.

- Bumps to Firebase version 4.0. (Segment's integration was a major version behind)
- Removes `subspec` which pulls in the deprecated `pod appIndexing` .
- Fixes a crash when passing a non NSString value through `traits` on `Identify`.
- Fixes Mapping to Firebase `logEvent` and Firebase reserved Params and Constants.

The last point is important, as the mappings are different in this new version and will change which events you seen in your Firebase dash. Segment recommends that you make this upgrade, as this new naming convention coincides with Firebase's semantic Constants and Params.

Even more exciting is that this new iOS SDK will have parity with the new Segment-Firebase Android SDK.

As a current user of Segment-Firebase iOS, you will be able to pull in the latest version by pinning `pod 'Segment-Firebase', '~>2.0`. While this is not recommended, if you are not ready to upgrade you can pin the old beta version at `pod 'Segment-Firebase', '~>1.0.0'`

For details on the new mapping, you can check out Segment's Event mappings documentation.

Segment recommend upgrading as soon as possible. Reach out to support if you have any feedback about both the Firebase iOS and Android betas.

Settings

Segment lets you change these destination settings from the Segment app without having to touch any code.

SETTING	DESCRIPTION
Deep Link URL Scheme (iOS)	`string`. For iOS, if you're using Firebase for Deep Linking, we'll set this as the deepLinkURLScheme before initializing Firebase.

This page was last modified: 09 Aug 2024

Need support?

Questions? Problems? Need more info? Contact Segment Support for assistance!

Visit our Support page

Help improve these docs!

Edit this page

Request docs change

Was this page helpful?

Yes

No

Get started with Segment

Segment is the easiest way to integrate your websites & mobile apps data to over 300 analytics and growth tools.

Your work e-mail

Request Demo

or

Create free account

Privacy

Terms

Website Data Collection Preferences