

# FirestoreNative Manual

V1.0.1

|                                          |   |
|------------------------------------------|---|
| Introduction .....                       | 2 |
| Features .....                           | 2 |
| Project Setup.....                       | 2 |
| iOS Setup.....                           | 2 |
| Usage.....                               | 3 |
| Firestore Notification.....              | 3 |
| Initialize Firestore Notification: ..... | 3 |
| Listen to FCM token: .....               | 3 |
| Firestore Analytics .....                | 3 |
| Initialize Firestore Analytics: .....    | 3 |
| Log Event:.....                          | 3 |

## Introduction

FirebaseNative is a Unity plugin for extending the power of Google's new Firebase toolset for your unity game. It provides unified cross-platform API for easy handling Firebase Notification and Firebase Analytics. It currently supports Android and iOS.

## Features

The current version supports a subset of firebase features, but it will keep expanding depending on the most requested features.

- Obtain Firebase Cloud Messaging (FCM) Token.
- Auto renew FCM token.
- Full Firebase Analytics support.

## Project Setup

1. Create a Firebase project in [Firebase console](#) and add an Android/iOS app to the project.
2. Download **google-services.json** for Android and **GoogleService-Info.plist** for iOS.
3. In Unity, after installing FirebaseNative, it will add an option under *Window* menu. Open the FirebaseNative settings from the Unity menu: **Window -> FirebaseNative -> Settings**.
4. Click "**Load google-services.json**" button and locate the file. After the file is correctly loaded, the **Sender ID** field will be automatically populated.

That's all you need to do in Unity. Very simple.

## iOS Setup

1. After building your iOS project. You can either following the official website (<https://firebase.google.com/docs/ios/setup>) to setup your project or follow the simplified steps below:
  - a. Install pod by running:  

```
$ sudo gem install cocoapods
```
  - b. In your project folder, run:  

```
$ pod init
```
  - c. After the above step, a Podfile will be generated. Add two lines in the file under the proper section in the file:  

```
pod 'Firebase/core'  
pod 'Firebase/messaging'
```
  - d. Run:  

```
$ pod install
```
  - e. After the above command (it may take a while to finish), it will generate a .xcworkspace file in the project folder. Open that file in XCode.
  - f. Add **GoogleService-Info.plist** to the root of your project file.
  - g. Under **Build Settings** of the project, add "**\$(inherited)**" without quotes to the "**Other Linker Flags**" field.
  - h. Build and run.

## Usage

### Firebase Notification

Initialize Firebase Notification:

```
FirebaseNative.FirebaseNotification.Instance.Initialize();
```

Listen to FCM token:

```
FirebaseNative.FirebaseNotification.Instance.OnFCMToken += (string token) => {  
    // Handle your FCM token here  
};
```

This event will be triggered:

- a. For Android, it will be called during the first launch of your Android app after it successfully obtained FCM token from Firebase and when there is a new FCM token available.
- b. For iOS: it will be called every time you start your app or there is a new FCM token.

### Firebase Analytics

Initialize Firebase Analytics:

```
FirebaseNative.FirebaseAnalytics.Instance.Initialize();
```

Log Event:

```
FirebaseNative.EventData data = new FirebaseNative.EventData("test_event");  
data.Add("test_param", "test_value")  
    .Add("test_int", 10);  
FirebaseNative.FirebaseAnalytics.Instance.LogEvent(data);
```