# Whoxa – Whatsapp Clone Flutte Android & iOS App

**Whoxa Whatsapp clone** Social Android and iOS App built in Flutter and Firebase. You can reuse more than 120+ widgets to customize your application.

Mobile app developed using flutter framework created by Google is an open-source mobile application development. It is used to develop applications for Android and iOS, as well as being the primary method of creating applications.

**NB: The same code used for both iOS and Android.**

## Install Flutter:

Visit flutter official website : flutter.dev for full install guide. if you prefer video tutorials, we recommend this playlist that will guide throw the full installation process: https://www.youtube.com/playlist?list=PLSzsOkUDsvdtl3Pw48-R8lcK2oYkk40cm

Set up your editor Install the Flutter and Dart plugins.

**Please make sure you have the latest flutter stable version.**

## Android Setup:

**1** – Open Code in visual studio.

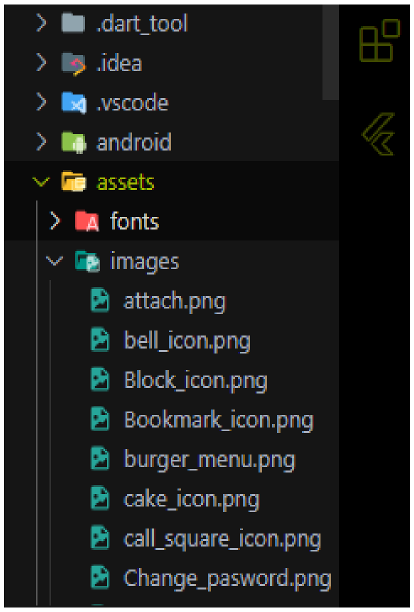
**2** – Create Your own project in firebase. We already provided below step for how to create firebase project.

**3 –** Now You need to add google firebase json file in whoxa project.

**4** – just run the following command.
**Command:** flutter pub get

**5** – If you would like to change app icon then follow below steps.

> **5.1** – Go to Folder: assets > image



**5.2** – Put your PNG icon in above folder.

**5.3** – Open pubspec.yaml, Change below code with your logo image name.

```
flutter_icons:
     image_path: "assets/images/applogo.png"
    android: true
    ios: true
```

**5.4** – Run flutter pub get in terminal.

**5.5** – Run flutter pub upgrade

**5.6** – Run flutter pub run flutter_launcher_icons:main

**4** – If you want to change the package name following files and folders.

**4.1** – Go to Folder android>app>src>main>AndroidManifest.xml

**4.2** – Change the package name in your AndroidManifest.xml file:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android" package="your.package.name">
```

**4.3** – Open /android/app/build.gradle and change the package name.

```
defaultConfig {
        applicationId "your.package.name" minSdkVersion 16
        targetSdkVersion 27
        versionCode 1
        versionName "1.0"
       testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
  }
```

**5** – In your pubspec.yaml in the top of this file, change the name with your app name and the project description.

## That’s Done for Android.

## iOS setup:

1 - Open Code in visual studio.

2 - And just run the following command. Command: Flutter pub get

3 -  Create your iOs project in firebase. We already Provided step below for firebase.

4 – Add JSON file in xcode.

4 - Open terminal and open iOS folder path in terminal and install PODs on your project with below commands.

```
pod install
pod update
pod repo update
pod install --repo-update
```

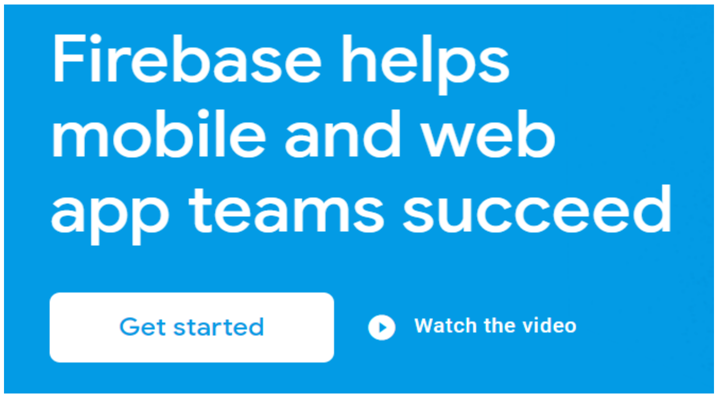
5 - Now Open iOS folder and open Runner.xcworkspace with Xcode. Run your project with simulators.

## That’s done for iOS.

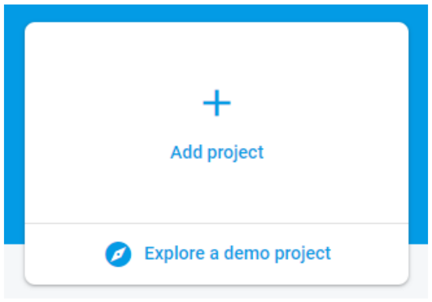
Build & Publish Flutter App:

To build and publish Android App, follow the official Documentation Android and iOS .

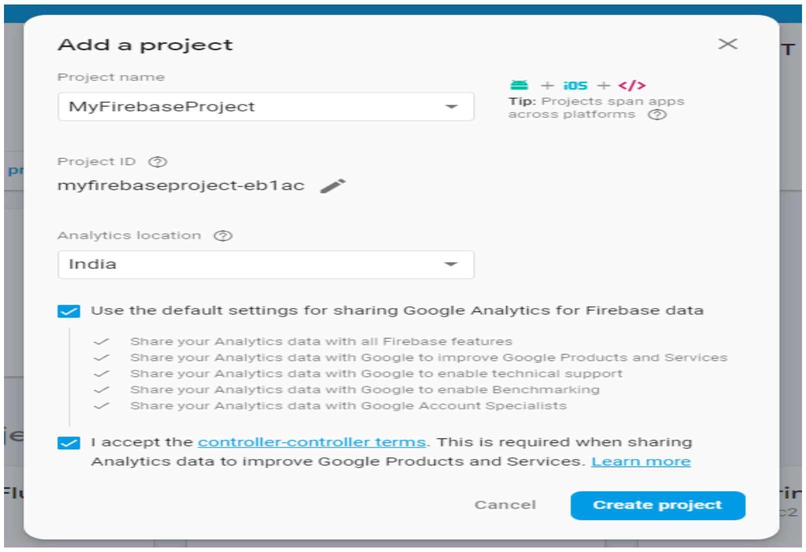
## Create a Firebase Project



From within the Firebase dashboard, select the “Create new project” button and give it a name:



Next up, we’re given the option to enable Google Analytics. It isn’t directly necessary for what we’re trying to do, so do whatever feels right for your use-case here.



Add a project

Project name
MyFirebaseProject

Tip: Projects span apps across platforms

Project ID
myfirebaseproject-eb1ac

Analytics location
India

✓ Use the default settings for sharing Google Analytics for Firebase data
- ✓ Share your Analytics data with all Firebase features
- ✓ Share your Analytics data with Google to improve Google Products and Services
- ✓ Share your Analytics data with Google to enable technical support
- ✓ Share your Analytics data with Google to enable Benchmarking
- ✓ Share your Analytics data with Google Account Specialists

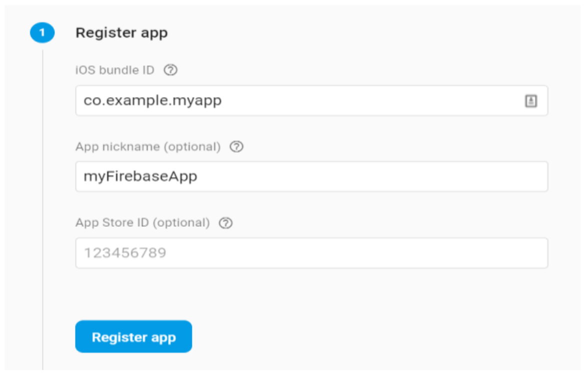
✓ I accept the controller-controller terms. This is required when sharing Analytics data to improve Google Products and Services. Learn more

Assuming that we’re using Google Analytics, we’ll need to review and accept the terms and conditions prior to project creation.

After a relatively painless setup process, we should now be navigated to the dashboard for our project.

## Adding Android support

Step 1: Register app



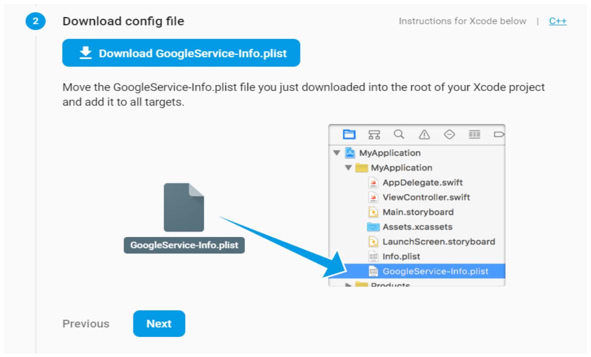
1 Register app

iOS bundle ID: co.example.myapp

App nickname (optional): myFirebaseApp

App Store ID (optional): 123456789

The most important thing here is to match up the **Android package name** that you choose here with the one inside of our application. The structure of this is done like so: `com.example.app`.



2 Download config file

Instructions for Xcode below | C++
Download GoogleService-Info.plist

Move the GoogleService-Info.plist file you just downloaded into the root of your Xcode project and add it to all targets.

The next step is to add the Firebase configuration file into our Flutter project. This is important as it contains the API keys and other critical information for Firebase to use.

Select “Download `google-services.json`” from this page.

Once you’ve got this, take the file and place it inside of the `android/app` directory within the Flutter project and it must replace the existing one.

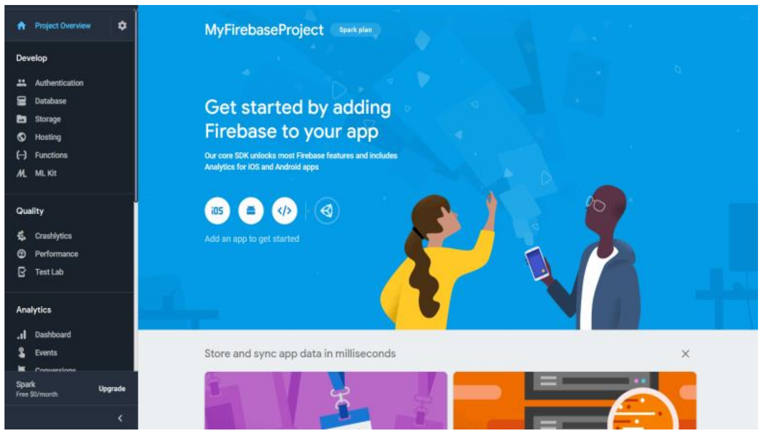
From here, run your application on an Android device or simulator. If everything has worked correctly, you should get the following message in the dashboard:

Next up, let's add iOS support!

## Adding iOS support

In order to add Firebase support for iOS, we have to follow a similar set of instructions.

Head back over to the dashboard and select `Add app` and then iOS icon to be navigated to the setup process.


Project Overview
Develop
Authentication
Database
Storage
Hosting
Functions
ML Kit
Quality
Crashlytics
Performance
Test Lab
Analytics
Dashboard
Events
Spark
Free $0/month
Upgrade
MyFirebaseProject
Spark plan
Get started by adding Firebase to your app
Our core SDK unlocks most Firebase features and includes Analytics for iOS and Android apps
iOS
Add an app to get started
Store and sync app data in milliseconds

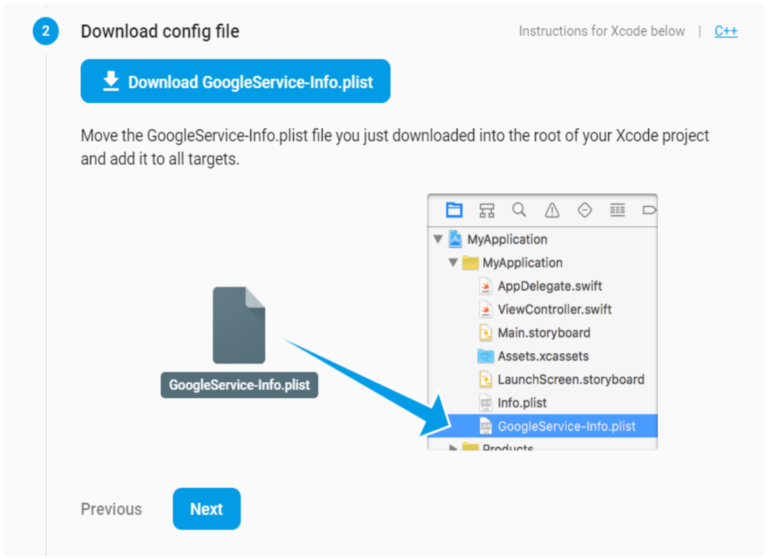
**Step 1:** Register an app

Once again, we'll need to add an iOS Bundle ID which I'm keeping the same as Android for consistency:

You'll then need to make sure this matches up by opening the iOS project up in Xcode at ios/Runner/Runner.xcodeproj and changing the Bundle identifier under General:

Click "Register app" to move to the next screen.

**Step 2:** Download config file.
In this step we'll need to download the configuration file and add this to our Xcode project.



2 Download config file

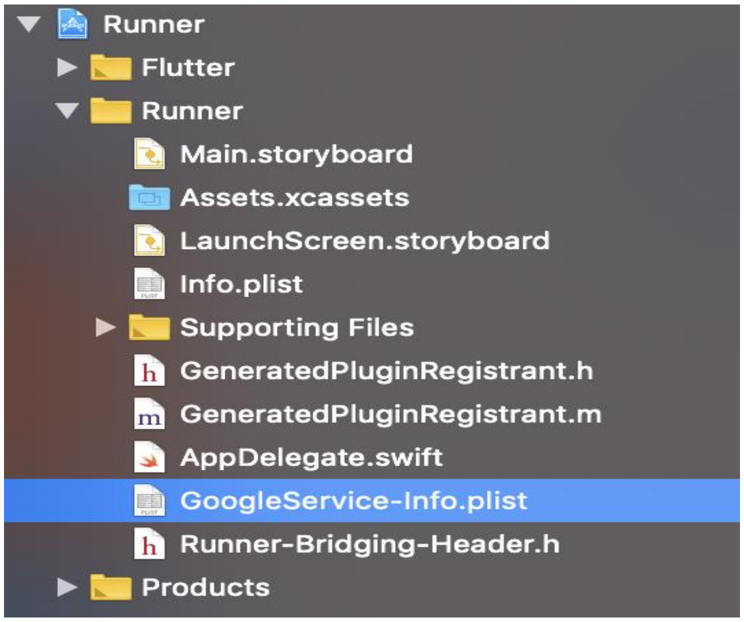
Instructions for Xcode below | C++

Download GoogleService-Info.plist

Move the GoogleService-Info.plist file you just downloaded into the root of your Xcode project and add it to all targets.

Previous
Next

Download `GoogleService-Info.plist` and drag this into the root of your Xcode project within `Runner`:



It's important that you don't simply drag this into the folder without going through Xcode, as this will not work.

**That's it! You'll also be happy to know that we can skip the rest of the steps from here on out.**