nlytical

Overview

Nlytical is a fully-featured **business listing website** that allows users to easily **discover, explore, and connect** with local or global stores and service providers. With a seamless and intuitive interface, users can:

This platform offers a **centralized solution** for managing the entire user experience through a responsive, unified web-based interface.

Technology Stack

- **Frontend**: Flutter App (✓ *Complete source code is included in this script*)
- **Admin Panel :** ReactJS (✓ *Complete source code is included in this script*)
- **Backend**: Laravel (PHP Framework) (✓ *Complete source code is included in this script*)
- **Database**: MySQL (✓ *As we given command so automatically install all myql database tables*)

Flutter

Flutter is an open-source UI software development toolkit created by **Google**. It allows developers to build **natively compiled applications** for**android and iOS devices** — all from a **single codebase**.

ReactJS(Admin Panel)

React.js, commonly referred to as React, is a popular JavaScript library developed by Facebook for building user interfaces, especially single-page applications. React enables developers to create reusable UI components and efficiently update and render the right components when data changes, making it widely used for dynamic and interactive web applications.

Laravel (Backend API)

Laravel is a powerful PHP framework known for its elegant syntax and developer-friendly tools. It is used to handle backend operations and APIs for full-stack applications.

Requirements

Make sure the following are installed:

- `Laravel`
- `Mysql`
- `Flutter(3.32.8)`
- JDK 17

- Code Editor: **Visual Studio Code (Recommended)**

Here we are providing an explanation for the installation of both VPS and cPanel based servers.

Laravel Installation (Linux)

You need to setup laravel and mysql in VPS server. Here you can find all steps for how to install laravel and mysql in VPS sever.

Connect to sever using ssh.

```
ssh root@server_ip
```

- After execution a prompt will be asked to enter Password
- Execute below command to update the apt package.

```
sudo apt install
sudo apt update
```

- Install apache web server, this is our web server through which our server accept the http request.

```
sudo apt install apache2
```

- Check the apache status

```
sudo systemctl status apache2
```

- Install neccessary php packages

```
sudo apt install php php-cli php-mbstring php-xml php-bcmath php-curl php-mysql php-zip unzip curl php8.3-bcmath git -y
```

MySQL Setup

Installation of My SQL

- run this Command in Linux Terminal to install MySQL

```
sudo apt install mysql-server
```

MySQL Root User Creation

- After the installation of MySQL run this command to start MySQL

```
sudo mysql
```

- To Create Root user for this use this Commmand (This command will be run in MySQL terminal)

```
alter user 'root'@'localhost' identified with mysql_native_password by 'root@123?'
```

> **ROOT USER'S PASSWORD**
>
> Note Down RootUser Password. It will br the main Admin of MySQL. So It's Crucial

- To get exit from MySQL Terminal type `exit`

Laravel Project Extraction using VPS

- First you have to extract the `nlytical.zip` then you will get three zip files.
  - Nlytical-laravel.zip
  - Nlytical-admin.zip
  - Nlytical-app.zip
- Create `nlytical-laravel` folder or any name with below command

```
mkdir -p /var/www/nlytical-laravel
```

Notes: You may append to the end of your folder name as desired. We have used the folder name nlytical-laravel merely as an example.

- Upload `nlytical-laravel.zip` to a location where you want to deploy Project using FileZilla
  - `/var/www/nlytical-laravel` (This is the recommended location)
- Change ownership of the folder to current user

```
chown -R www-data:www-data /var/www/nlytical-laravel
```

- Install unzip to `unzip` the Project . Run following command to install unzip

```
sudo apt install unzip
```

- extract the `nlytical_laravel.zip`

```
unzip nlytical-laravel.zip
```

Laravel Installation (cPanel)

(Notes: Make sure you setup on root Path)

✓ Prerequisites:

- A hosting account with **cPanel** access
- **PHP 8.0 or higher**
- **MySQL database**

Step-by-Step Guide:

1. Laravel Project Zip

- Zip your **entire Laravel project or it's already in zip.** (nlytical-laravel.zip).

2. Upload to cPanel

- Login to **cPanel** → Open **File Manager**
- Go to the root directory (`public_html/` or subfolder)
- Upload the ZIP file and **extract it**

3. Set Folder Permissions

Ensure these folders are **writable (775/755)**:

- `storage/`
- `bootstrap/cache/`

4. Create Database

- Go to **MySQL® Databases** in cPanel
- Create a new DB and user, and assign full permissions

Deploy Process

Laravel Configurations

(It's comman for the both VPS or cPanel).

- After the extraction you will see the Project files

Configure Environment Variables

- Edit the `.env` file in the root directory of your project. This step is mandatory to ensure the application can connect to your database.
- Update the following lines in the `.env` file with your actual server and database credentials:

```
APP_URL=http://{your_server_ip}
DB_DATABASE={your_database_name}
DB_USERNAME={your_database_username}
DB_PASSWORD={your_database_password}
```

Clear Configuration Caches

- Before running migrations, clear all cached configurations using the following Artisan commands:

```
php artisan config:clear
php artisan cache:clear
php artisan route:clear
php artisan view:clear
```

Once all above command run successfully then run this command::

```
php artisan optimize
```

Database Migration

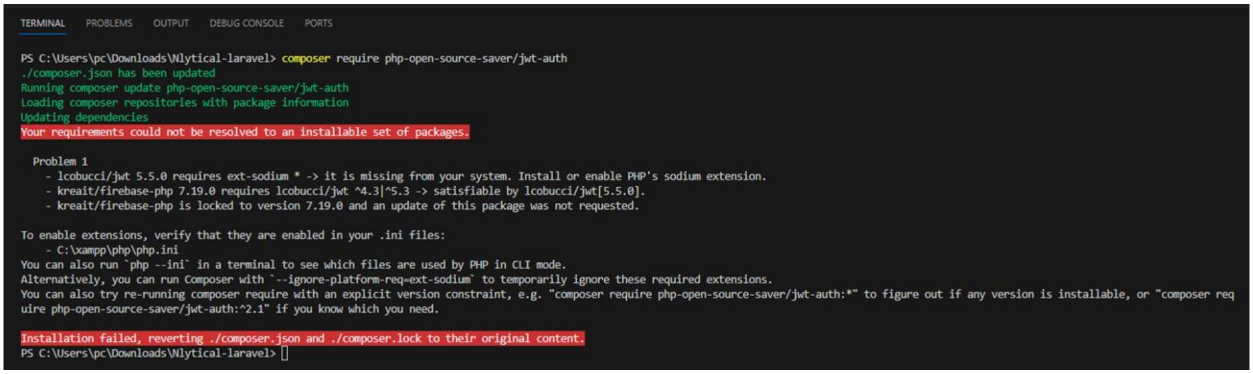
- Below command used to manage your database schema using migration files.:

```
php artisan migrate
```

- During the installation process, you may be prompted with configuration options. Answer "yes" to all questions.
- Once installed all tabales in database then you can run below command to serve installation process.

```
composer require php-open-source-saver/jwt-auth
```

Notes: The error you're facing is due to the missing ext-sodium PHP extension after run above command, which is required by lcobucci/jwt (used by kreait/firebase-php for JWT handling). Here's how to fix it:



```
PS C:\Users\pc\Downloads\Nlytical-laravel> composer require php-open-source-saver/jwt-auth
./composer.json has been updated
Running composer update php-open-source-saver/jwt-auth
Loading composer repositories with package information
Updating dependencies
Your requirements could not be resolved to an installable set of packages.

  Problem 1
    - lcobucci/jwt 5.5.0 requires ext-sodium * -> it is missing from your system. Install or enable PHP's sodium extension.
    - kreait/firebase-php 7.19.0 requires lcobucci/jwt ^4.3|^5.3 -> satisfiable by lcobucci/jwt[5.5.0].
    - kreait/firebase-php is locked to version 7.19.0 and an update of this package was not requested.

To enable extensions, verify that they are enabled in your .ini files:
    - C:\xampp\php\php.ini
You can also run `php --ini` in a terminal to see which files are used by PHP in CLI mode.
Alternatively, you can run Composer with `--ignore-platform-req=ext-sodium` to temporarily ignore these required extensions.
You can also try re-running composer require with an explicit version constraint, e.g. "composer require php-open-source-saver/jwt-auth:*" to figure out if any version is installable, or "composer req
uire php-open-source-saver/jwt-auth:^2.1" if you know which you need.

Installation failed, reverting ./composer.json and ./composer.lock to their original content.
PS C:\Users\pc\Downloads\Nlytical-laravel> 
```

1. Edit `php.ini`

- Open this file:
  `C:\xampp\php\php.ini`
- Search for:

```
;extension=sodium
```

- **Uncomment** the line by removing the semicolon (`;`), so it becomes:

```
extension=sodium
```

2. Restart Apache

- Open **XAMPP Control Panel**
- Click **Stop** next to Apache, then **Start** again

- This command generates a unique JWT secret key and automatically add in your .env file.

```
php artisan jwt:secret
```

- Starting the Laravel Development Server To run the Laravel app locally, use the built-in Artisan command.

```
php artisan serve
```

Server Starting (if VPS Server)

To Serve our project you have to edit apache virtualhost file.

- Open default virtualhost file.

```
sudo nano /etc/apache2/sites-available/000-default.conf
```

- Replace existing Directory and DocumentRoot with below ones.

```
  ServerAdmin webmaster@localhost
  ServerName your_server_ip

  DocumentRoot /var/www/nlytical_laravel/public

  <Directory /var/www/nlytical_laravel/public>
      Options Indexes FollowSymLinks
      AllowOverride All
      Require all granted
  </Directory>

  ErrorLog ${APACHE_LOG_DIR}/error.log
  CustomLog ${APACHE_LOG_DIR}/access.log combined
```

- then after enable rewrite module of apache

```
a2enmod rewrite
```

  - First check for any syntax error with below command.

```
apachectl configtest
```

  - If syntax is ok then after restart the server.

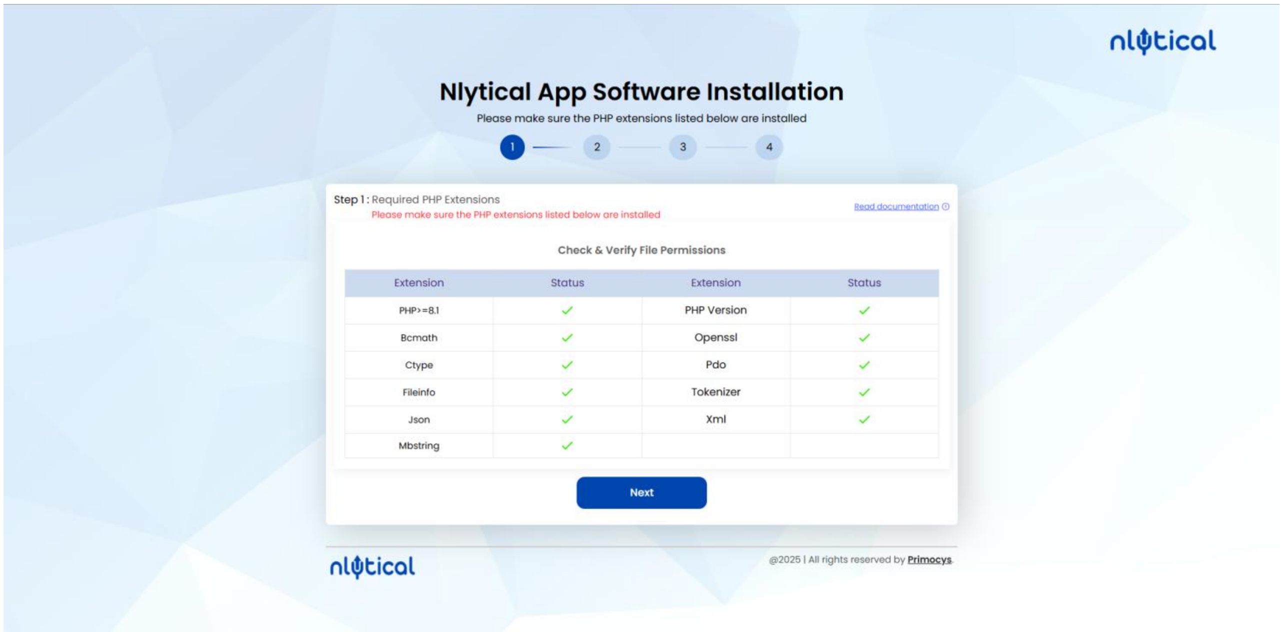
```
sudo systemctl restart apache2
```

Setup Nlytical Software

Open below url in the web browser.

```
http://{your_server_ip}
```

- If everything was installed correctly then you will see page like below.



Nlytical App Software Installation

Please make sure the PHP extensions listed below are installed

Step 1 : Required PHP Extensions

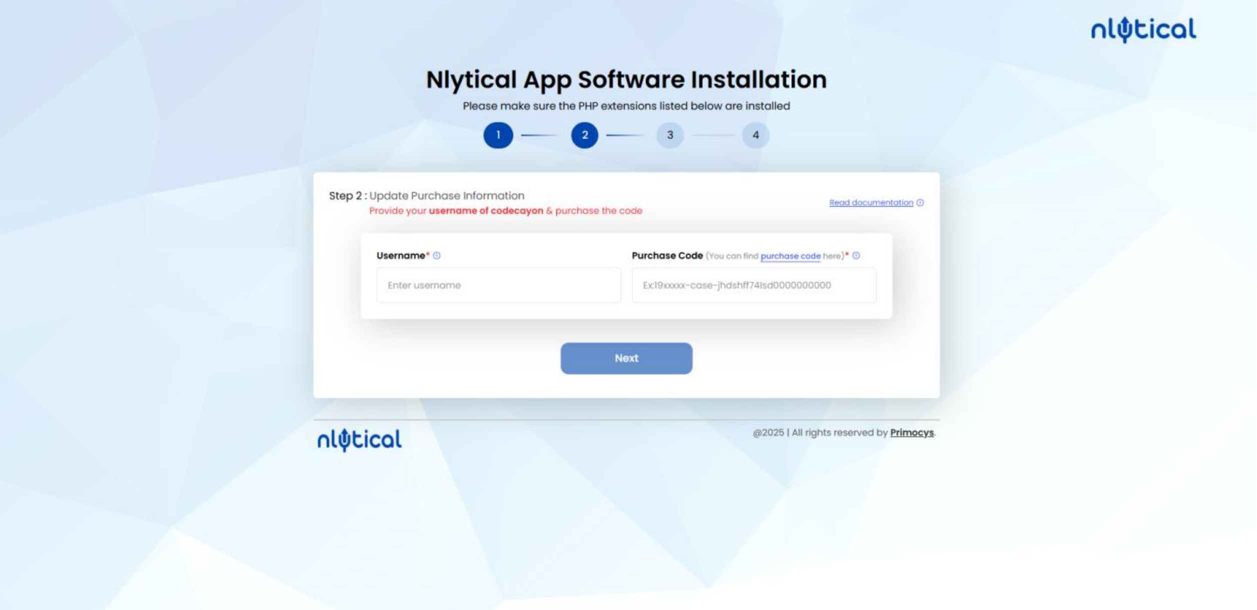
Please make sure the PHP extensions listed below are installed

Check & Verify File Permissions

Extension	Status	Extension	Status
PHP>=8.1	✓	PHP Version	✓
Bcmath	✓	Openssl	✓
Ctype	✓	Pdo	✓
Fileinfo	✓	Tokenizer	✓
Json	✓	Xml	✓
Mbstring	✓		

@2025 | All rights reserved by Primocys.

- Requirements PHP Extensions.



nlytical

Nlytical App Software Installation

Please make sure the PHP extensions listed below are installed

1 — 2 — 3 — 4

Step 2 : Update Purchase Information
Provide your **username of codecayon** & purchase the code

Read documentation

Username*	**Purchase Code** (You can find purchase code here)*
Enter username	Ex:19xxxxx-case-jhdshff74lsd0000000000

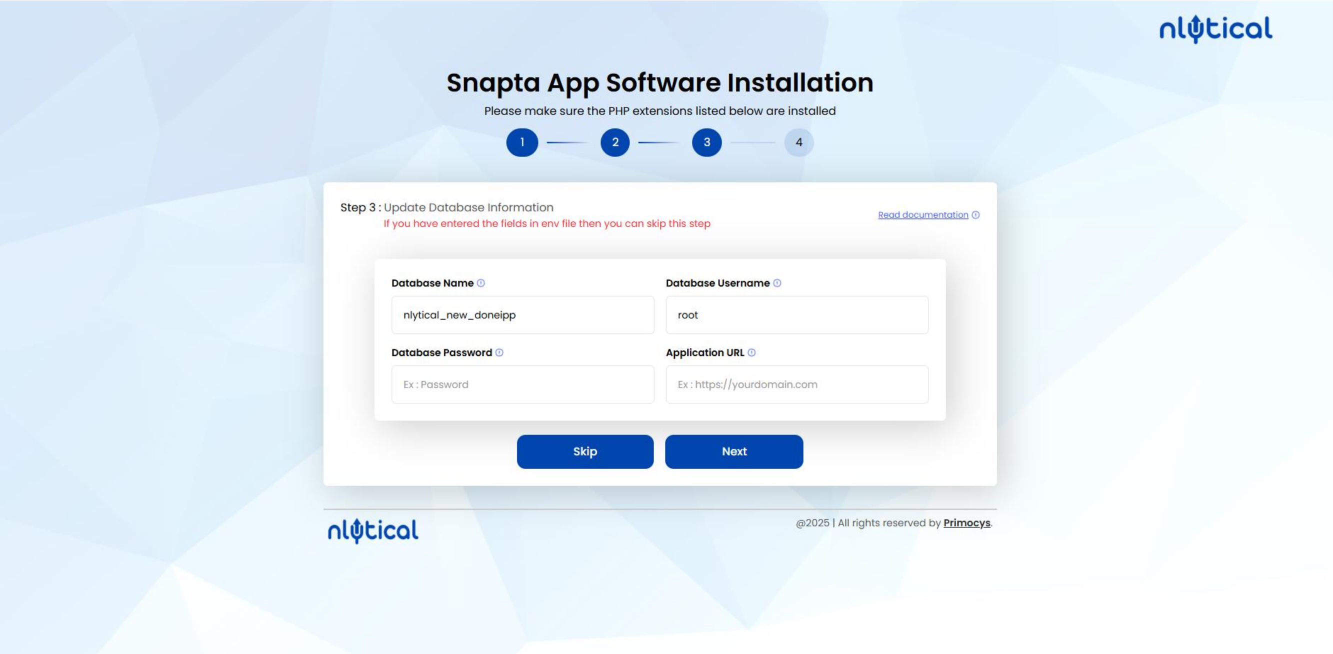
Next

nlytical @2025 | All rights reserved by **Primocys**.

- In here you have to enter envato `Username` and `purchase code`.

> **PURCHASE CODE**
>
> You can only use this purchase code one time.

- Third steps is for database configuration.



nlytical

Snapta App Software Installation

Please make sure the PHP extensions listed below are installed

1 — 2 — 3 — 4

Step 3 : Update Database Information
If you have entered the fields in env file then you can skip this step

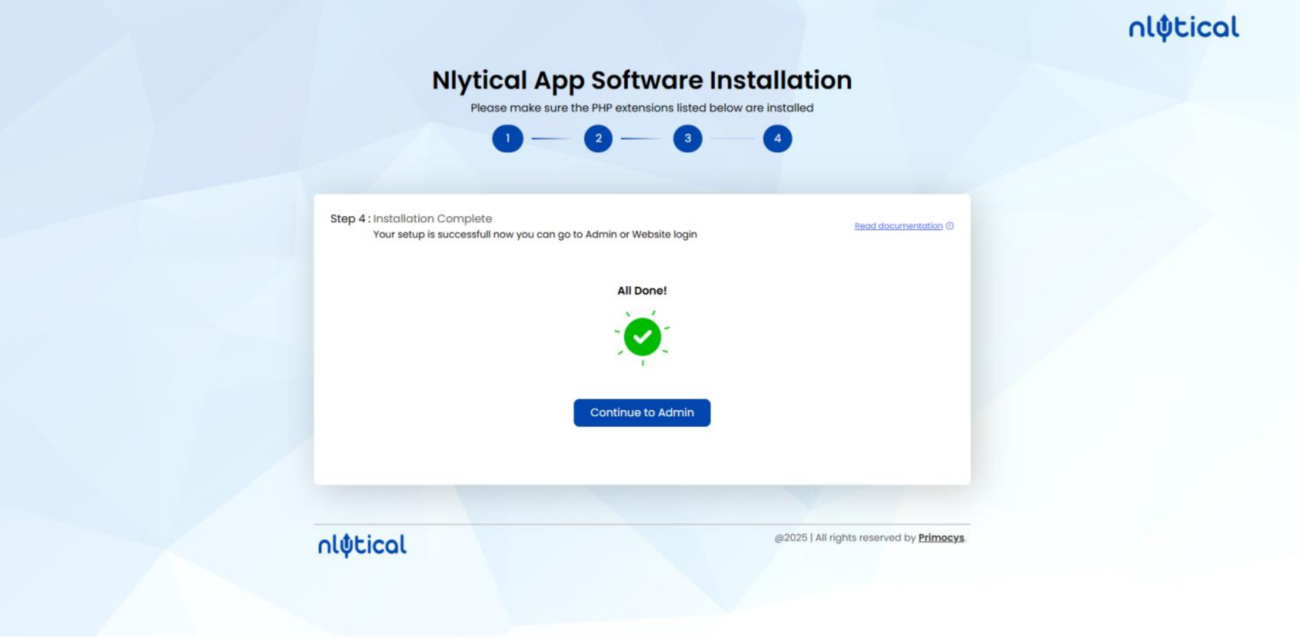
Read documentation

Database Name	**Database Username**
nlytical_new_doneipp	root
Database Password	**Application URL**
Ex : Password	Ex : https://yourdomain.com

Skip Next

nlytical @2025 | All rights reserved by **Primocys**.

- You can see here your added details which you add in .env file so you can skip it.
- After all the installation process completed you will see this page. You can click on ‘Continue to Admin’ then it will redirect to admin page.



Nlytical App Software Installation

Please make sure the PHP extensions listed below are installed

Step 4 : Installation Complete

Your setup is successfull now you can go to Admin or Website login

All Done!

@2025 | All rights reserved by Primocys.

What you'll need Flutter App Setup (Prerequisites)

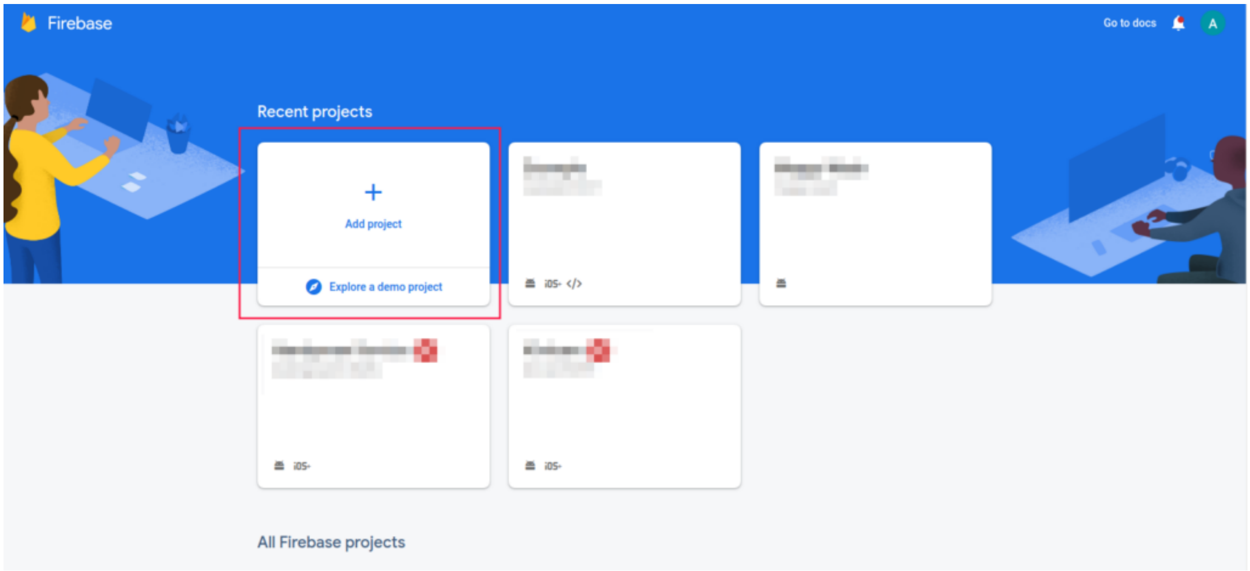
- IDE (Android Studio or Vs code. Android Studio recommended)
- Ensure that you download and install JDK 17 on your device.
- Flutter SDK setup (version 3.29.0 Stable)
- JDK with path setup (only for vs code)
- Xcode 16.2 for IPA file build

Create Firebase Project

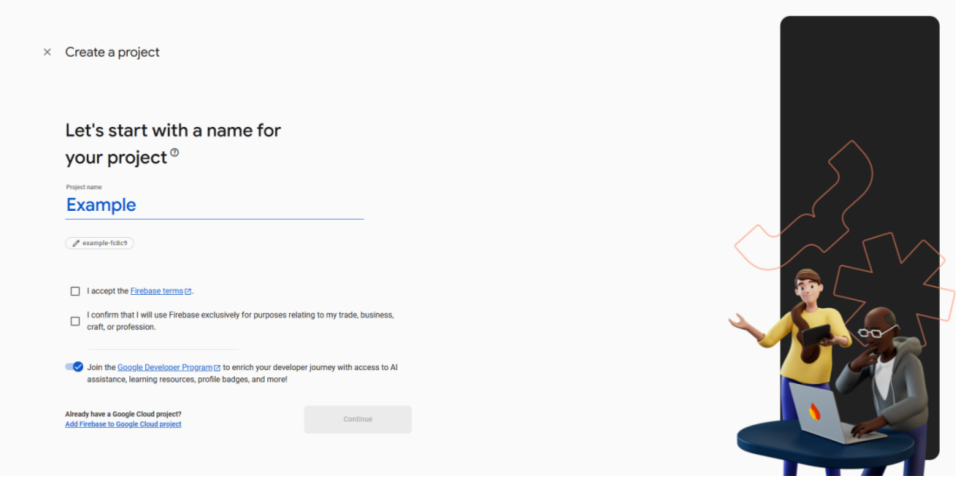
Setting up Firebase for your Nlytical App Project (It’s for social login and push notification.)

Create A New Firebase Project:

1. **We are creating the "Example" sample project.**



2. **You can add here your firebase project name.**



× Create a project

Let's start with a name for your project

Project name

Example

example-fc8c9

□ I accept the Firebase terms.

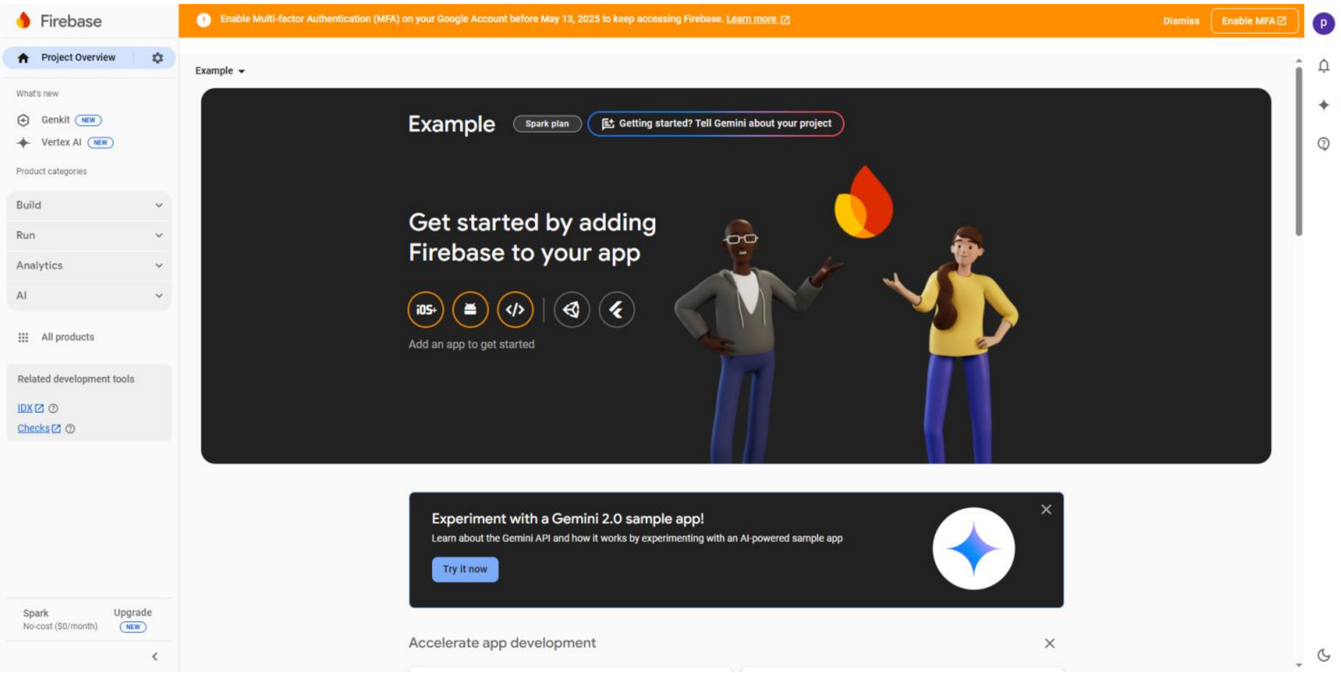
□ I confirm that I will use Firebase exclusively for purposes relating to my trade, business, craft, or profession.

● Join the Google Developer Program to enrich your developer journey with access to AI assistance, learning resources, profile badges, and more!

Already have a Google Cloud project?
Add Firebase to Google Cloud project

Continue

3. **After completing the project, you will be presented with this type of dashboard.**



Enable Multi-factor Authentication (MFA) on your Google Account before May 13, 2025 to keep accessing Firebase. Learn more

Dismiss | Enable MFA

Example

Example | Spark plan | Getting started? Tell Gemini about your project

Get started by adding Firebase to your app

Add an app to get started

Experiment with a Gemini 2.0 sample app!

Learn about the Gemini API and how it works by experimenting with an AI-powered sample app

Try it now

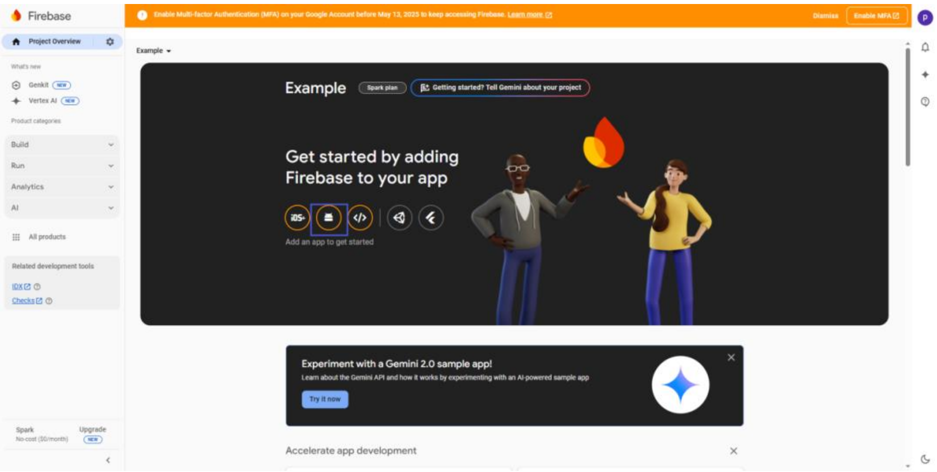
Accelerate app development

Android Firebase Configration

Setting up Firebase for your Nlytical App Project with Android.

Add Android App With The Package Name In Firebase Console:

1. On the Firebase console, click the Android icon.



Enable Multi-factor Authentication (MFA) on your Google Account before May 13, 2025 to keep accessing Firebase. Learn more

Example

Get started by adding Firebase to your app

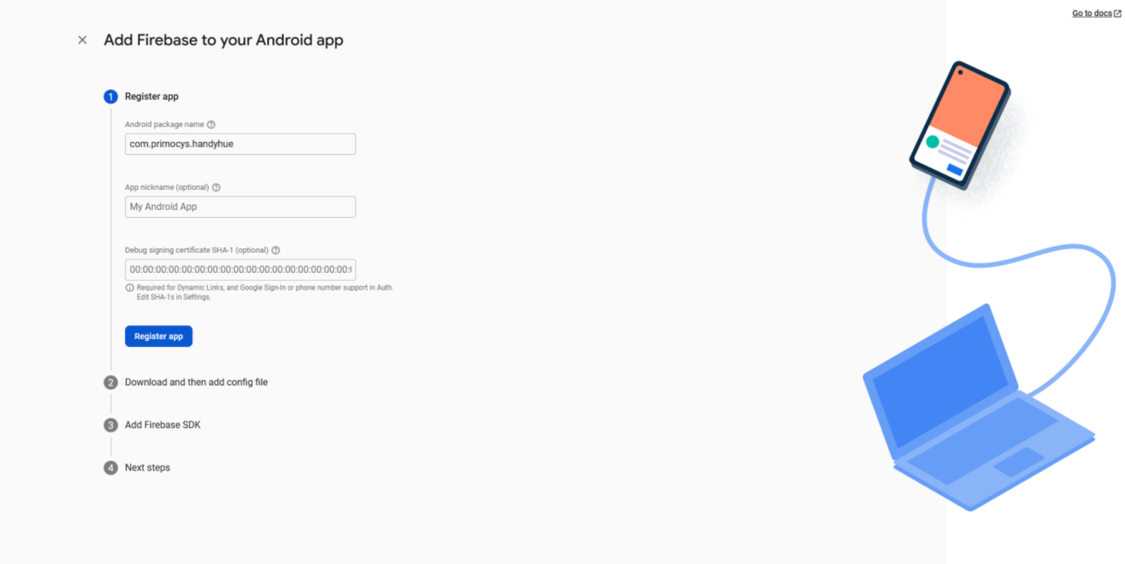
Add an app to get started

Experiment with a Gemini 2.0 sample app!

Learn about the Gemini API and how it works by experimenting with an AI-powered sample app

Accelerate app development

2. Enter Package Name (e.g., com.example.userApp) and click on register app.



Add Firebase to your Android app

1 Register app

Android package name: com.primocys.handyhue

App nickname (optional): My Android App

Debug signing certificate SHA-1 (optional): 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:0

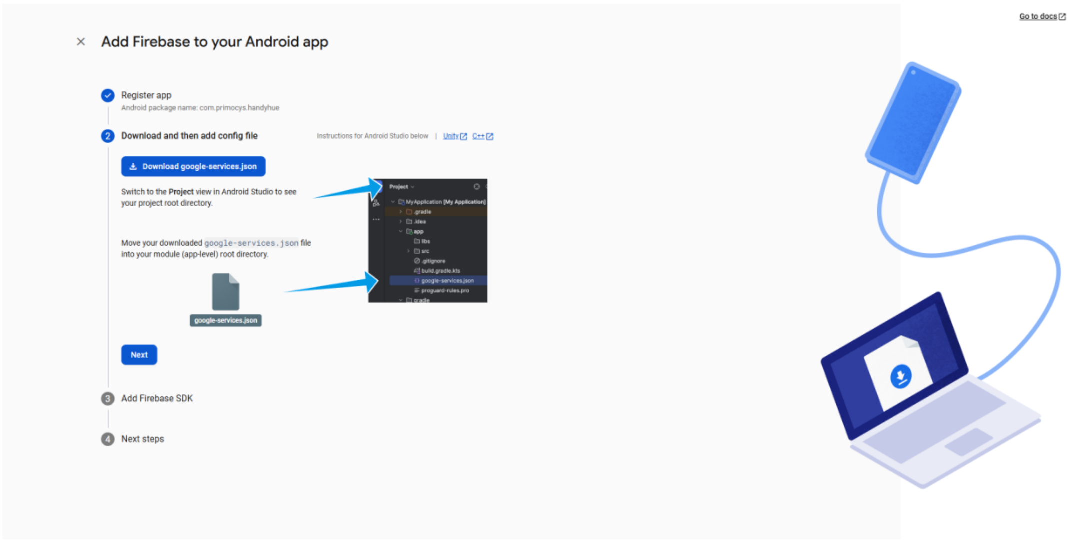
Required for Dynamic Links, and Google Sign-In or phone number support in Auth. Edit SHA-1s in Settings.

2 Download and then add config file

3 Add Firebase SDK

4 Next steps

3. After registering the app, you will receive the Google JSON file, download it and save it to the `android/app/` folder. Otherwise you can download later after add SHA fingerprint in firebase project so google login will work with your project. Check below how to add SHA in fribase



Add Firebase to your Android app

Register app

Android package name: com.primocys.handyhue

2 Download and then add config file

Instructions for Android Studio below | Unity C++

Switch to the **Project** view in Android Studio to see your project root directory.

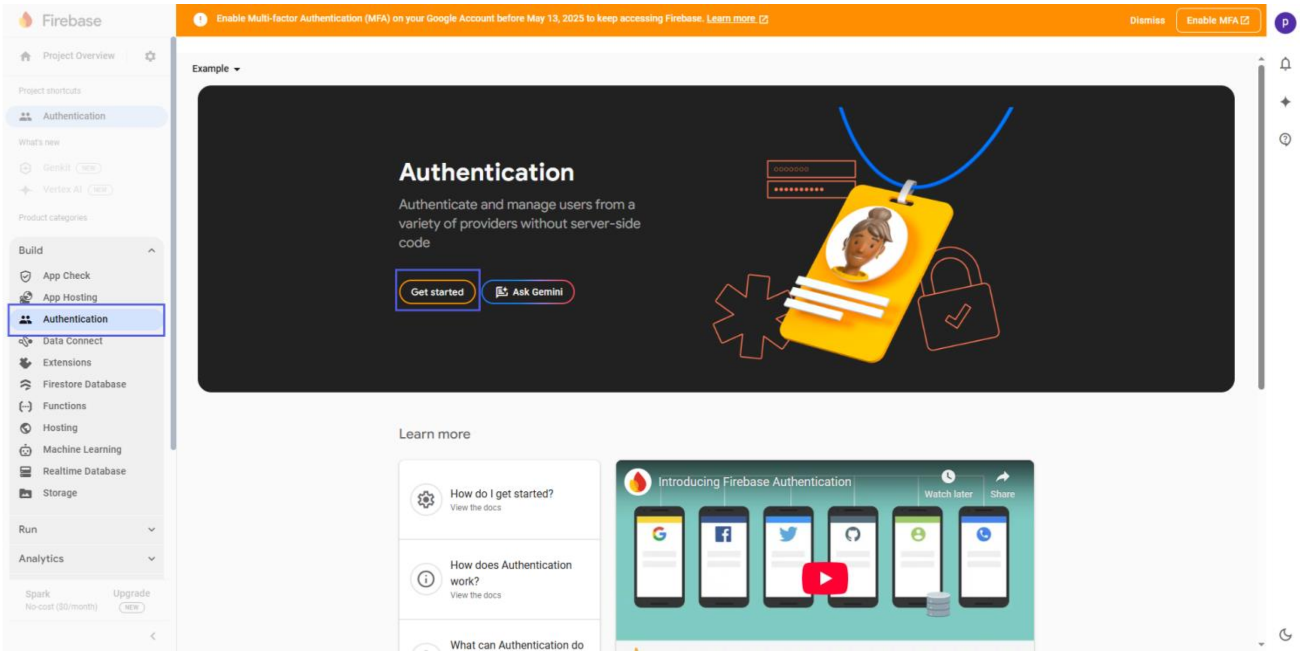
Move your downloaded `google-services.json` file into your module (app-level) root directory.

3 Add Firebase SDK

4 Next steps

Set Up Firebase Authentication

1. Go to `Build -> Authentication` tab and click on Get started.



Enable Multi-factor Authentication (MFA) on your Google Account before May 13, 2025 to keep accessing Firebase. Learn more

Authentication

Authenticate and manage users from a variety of providers without server-side code

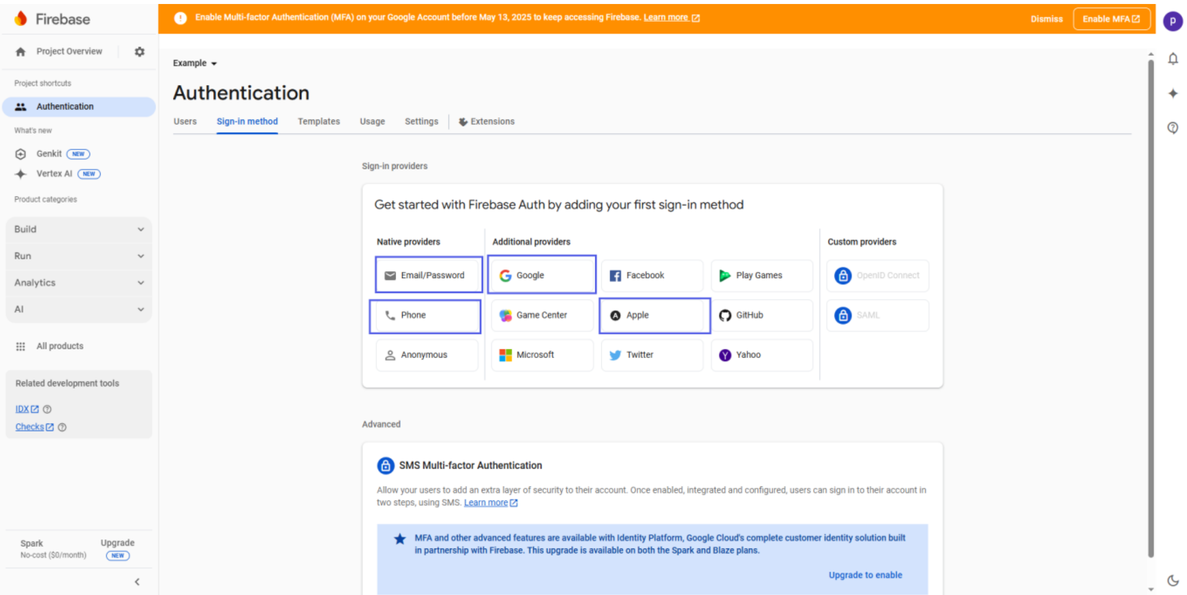
Learn more

How do I get started? View the docs

How does Authentication work? View the docs

What can Authentication do

2. Then select the Email/Password, Google, Apple, and Phone Number one by one and enable them.



Enable Multi-factor Authentication (MFA) on your Google Account before May 13, 2025 to keep accessing Firebase. Learn more

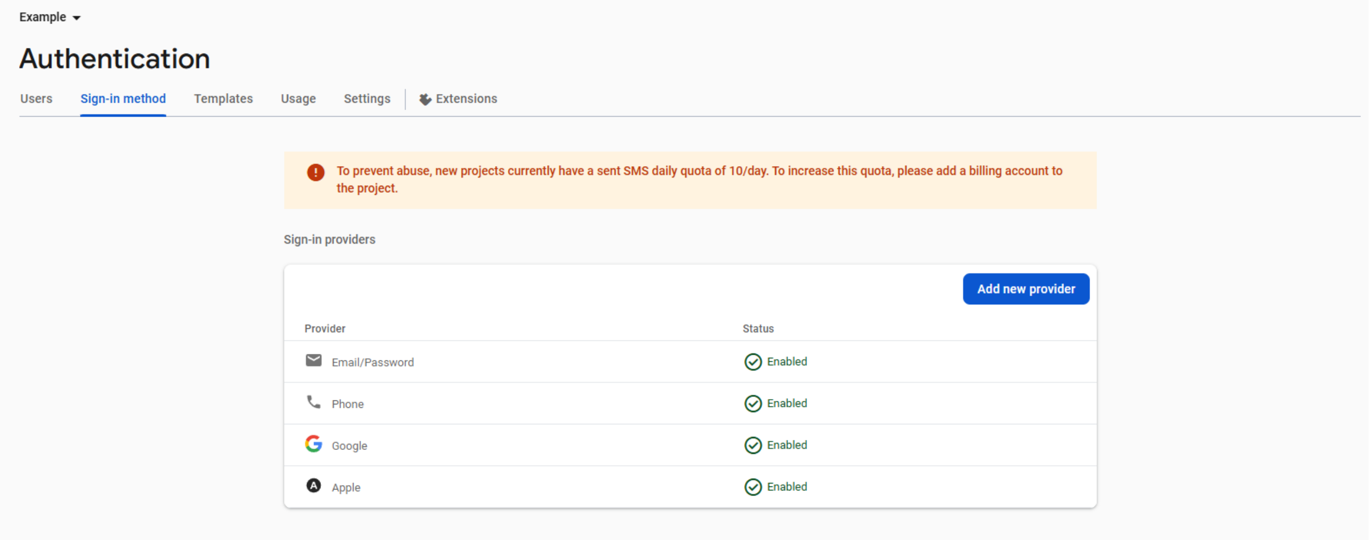
Get started with Firebase Auth by adding your first sign-in method

SMS Multi-factor Authentication

Allow your users to add an extra layer of security to their account. Once enabled, integrated and configured, users can sign in to their account in two steps, using SMS. Learn more

MFA and other advanced features are available with Identity Platform, Google Cloud's complete customer identity solution built in partnership with Firebase. This upgrade is available on both the Spark and Blaze plans.

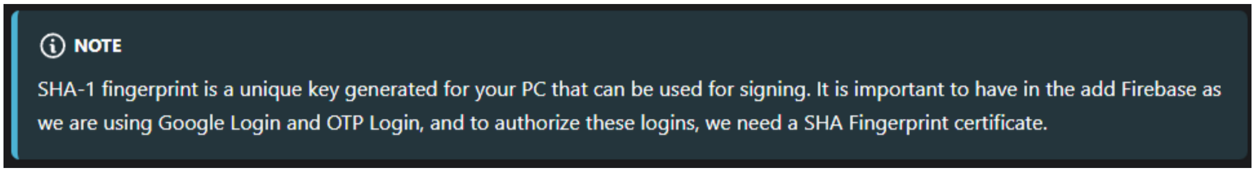
3. Re-check if all these 3 modes of authentication are enabled or not.



To prevent abuse, new projects currently have a sent SMS daily quota of 10/day. To increase this quota, please add a billing account to the project.

Provider	Status
Email/Password	Enabled
Phone	Enabled
Google	Enabled
Apple	Enabled

SHA Fingerprint (For Android)

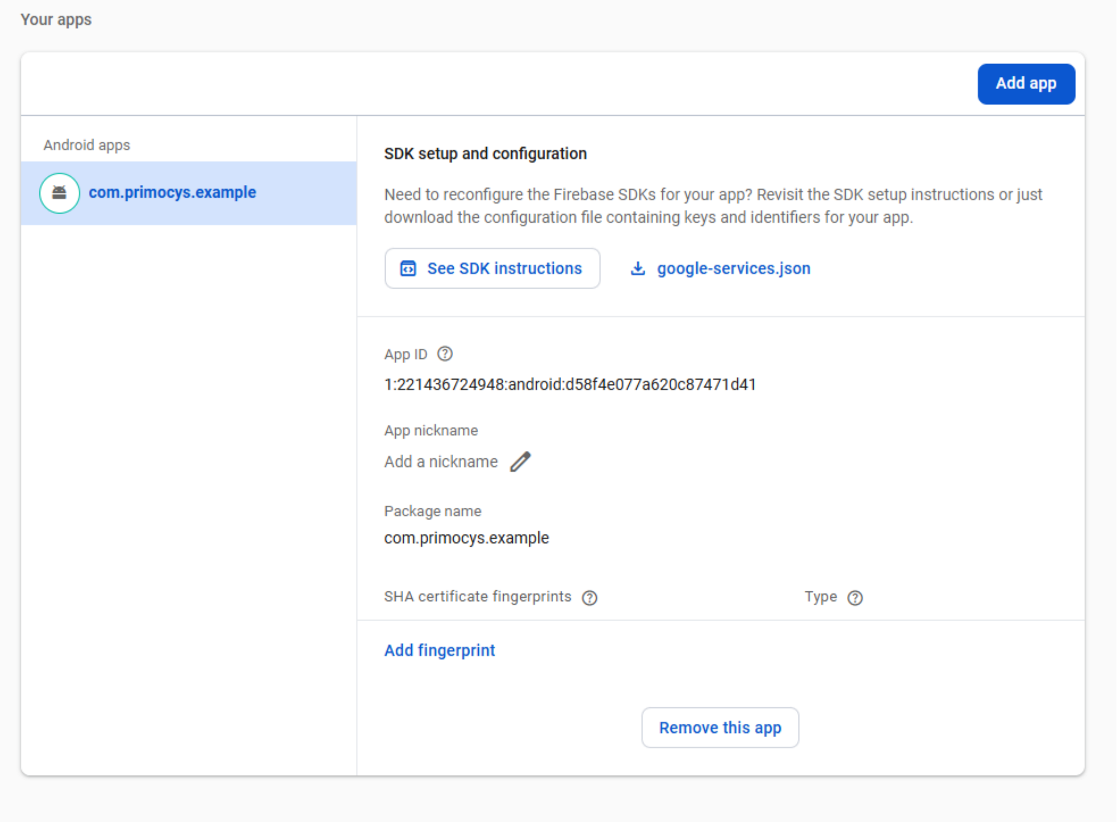


ⓘ **NOTE**

SHA-1 fingerprint is a unique key generated for your PC that can be used for signing. It is important to have in the add Firebase as we are using Google Login and OTP Login, and to authorize these logins, we need a SHA Fingerprint certificate.

Add a SHA fingerprint to Firebase:

1. Follow these steps if you didn't initially provide a SHA certificate fingerprint for your Firebase Android app or if you need to add an additional one:
   - In your Project settings, go to the Your apps card. and Select the Firebase Android app to which you want to add a SHA fingerprint.



SDK setup and configuration

Need to reconfigure the Firebase SDKs for your app? Revisit the SDK setup instructions or just download the configuration file containing keys and identifiers for your app.

App ID
1:221436724948:android:d58f4e077a620c87471d41

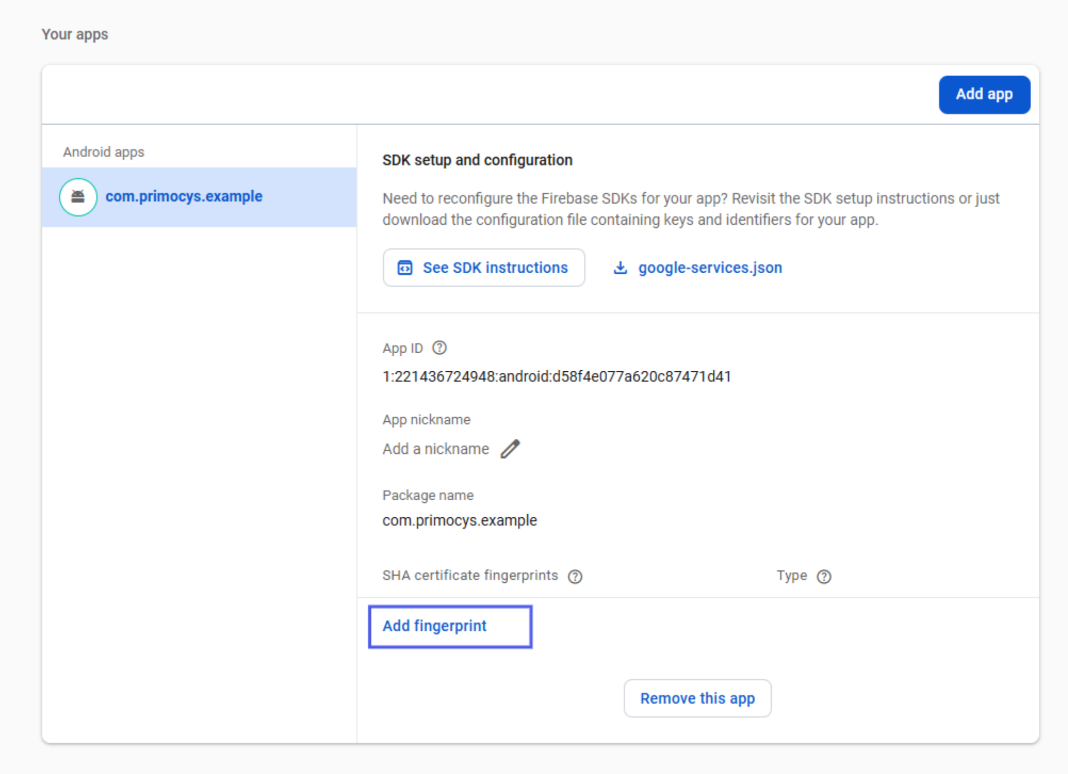
App nickname
Add a nickname

Package name
com.primocys.example

SHA certificate fingerprints | Type

- Click Add fingerprint to add SHA.
  Notes: Open your terminal go below android path of project and enter following command to check your system SHA1 and SHA256, Which you want to add after click Add fingreprint.
  Path: cd android

```
./gradlew signinReport
```



SDK setup and configuration

Need to reconfigure the Firebase SDKs for your app? Revisit the SDK setup instructions or just download the configuration file containing keys and identifiers for your app.

App ID
1:221436724948:android:d58f4e077a620c87471d41

App nickname
Add a nickname

Package name
com.primocys.example

SHA certificate fingerprints　Type

- Enter or paste the SHA fingerprint, then click Save.

How to Build APK for Android

1. Create a file in android folder of your project name `key.properties` if not exists.
2. Add properties and it's value below

```
storePassword=<Your JKS password>
keyPassword=<Your JKS Key Password>
keyAlias=<You JKS Key Alias>
storeFile=<Your jks file location path>
```

3. Add following line to your `Android/app/build.gradle` file above the all `android` block (Check if this line already exist or not)

```
def keystoreProperties = new Properties()
def keystorePropertiesFile = rootProject.file('key.properties')
if (keystorePropertiesFile.exists()){
        keystoreProperties.load(new FileInputStream(keystorePropertiesFile))
}
```

4. Now in app:build.gradle android block check for below lines if not exist Add into your Android/app/build.gradle

```
  signingConfigs {
      release {
          keyAlias keystoreProperties['keyAlias']
          keyPassword keystoreProperties['keyPassword']
          storeFile keystoreProperties['storeFile'] ? file(keystoreProperties['storeFile']) : null
         storePassword keystoreProperties['storePassword']
         }
}

buildTypes {
  release {
      signingConfig signingConfigs.release}
  }
```

5. The aforementioned steps have been completed for all modifications to build.gradle. Therefore, in the subsequent steps following the iOS Firebase configuration, you will find instructions on how to set up the base URL, colors, and other related configurations.

SUCCESSFUL !!

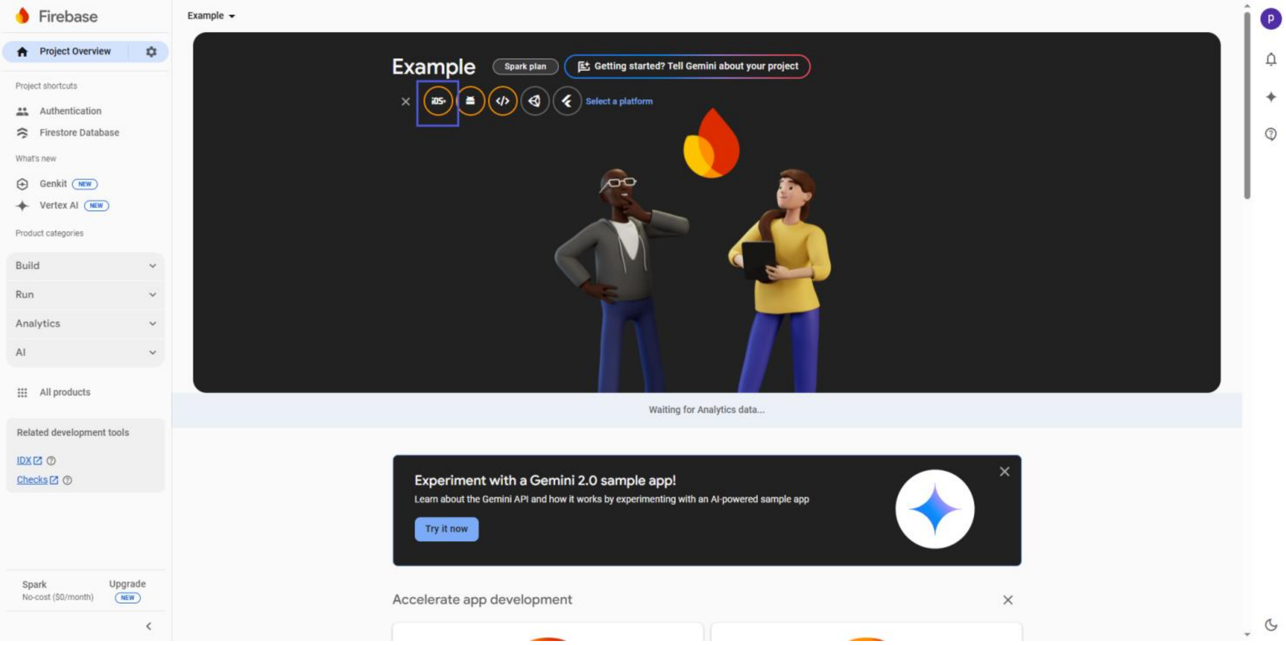
Great! You have successfully configured the Android App Firebase for Push Notificationa and Google Social Login!

iOS Firebase Configration

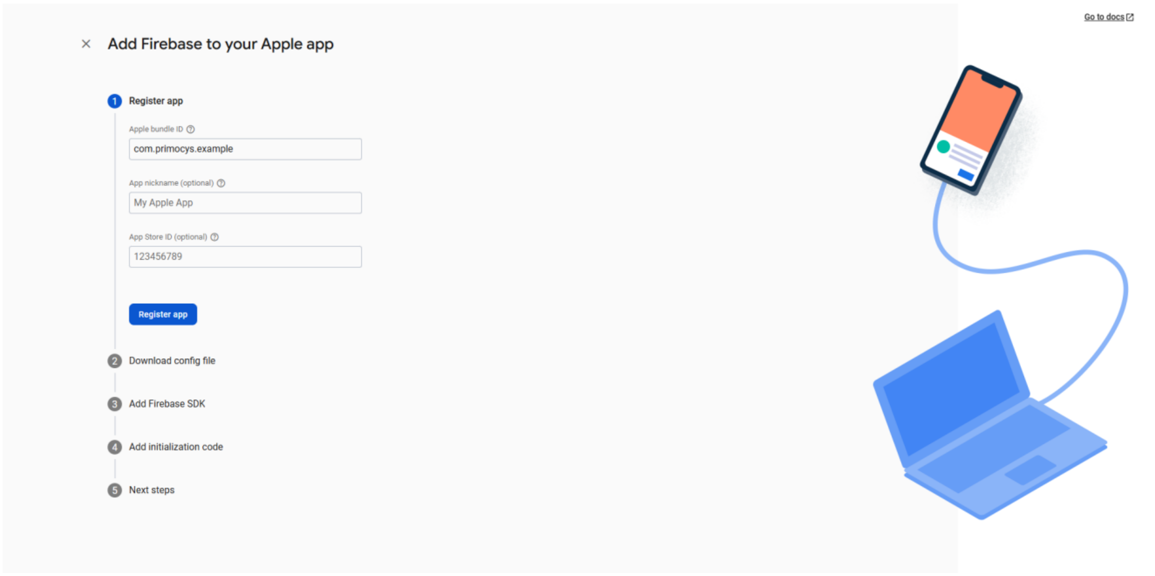
Setting up Firebase for your Nlytical App Project with iOS.

Add iOS App With The Package Name In Firebase Console:

1. On the Firebase console, click the iOS icon.


Firebase
Project Overview
Project shortcuts
Authentication
Firestore Database
What's new
Genkit NEW
Vertex AI NEW
Product categories
Build
Run
Analytics
AI
All products
Related development tools
IDX
Checks
Spark
No-cost ($0/month)
Upgrade NEW
Example
Example
Spark plan
Getting started? Tell Gemini about your project
iOS+
Select a platform
Waiting for Analytics data...
Experiment with a Gemini 2.0 sample app!
Learn about the Gemini API and how it works by experimenting with an AI-powered sample app
Try it now
Accelerate app development
P

2. Enter Package Name (e.g., com.example.userApp) and click on register app.



Go to docs

× Add Firebase to your Apple app

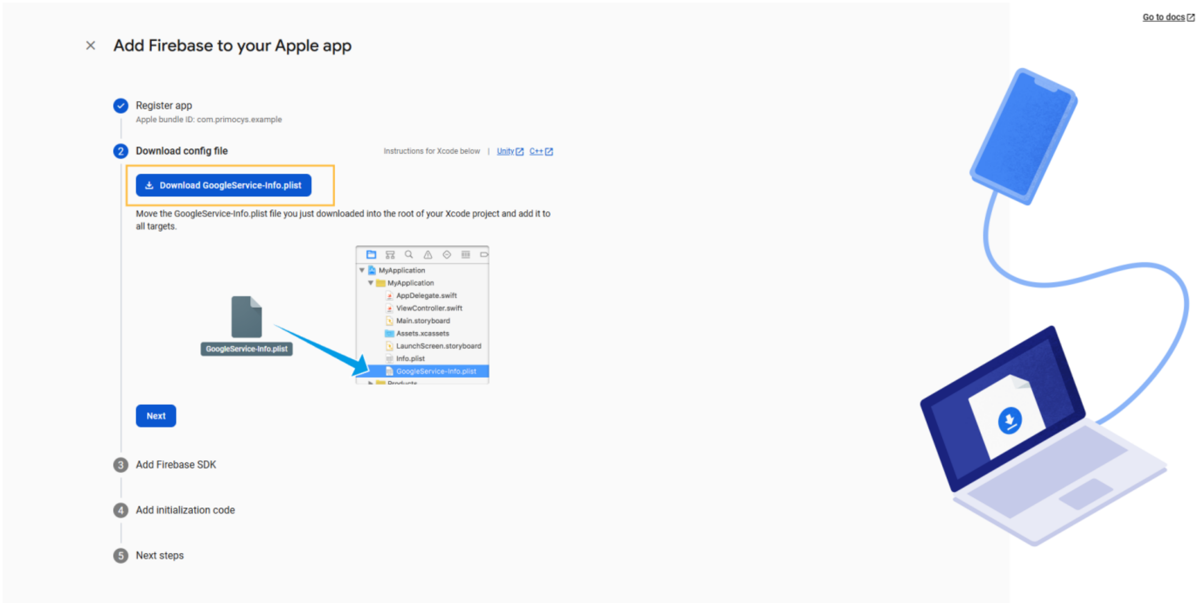
1 Register app

Apple bundle ID: com.primocys.example

App nickname (optional): My Apple App

App Store ID (optional): 123456789

3. After registering the app, you will receive the `GoogleService-Info.plist`, download it. Otherwise you can download later after add Gogole login and push notification configration in firebase project so google login and push notification will work with your project. [Check below how to add SHA in fribase]



Go to docs

× Add Firebase to your Apple app

✓ Register app

Apple bundle ID: com.primocys.example

2 Download config file

Instructions for Xcode below | Unity C++

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

How to add Google SignIn in iOS Flutter Project:

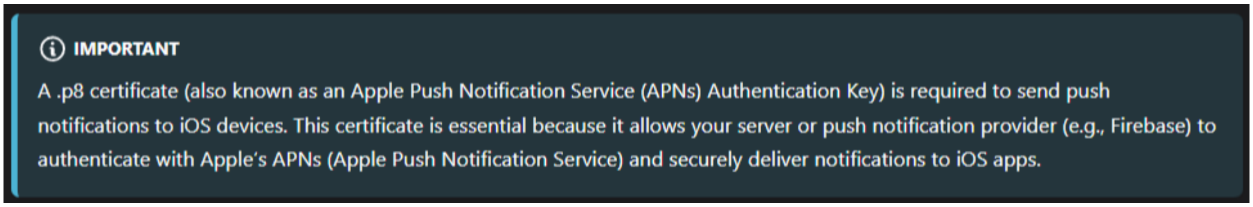
To enable Google SignIn in your app:

- Make Sure that your app on Firebase like above steps.
- Make sure that GoogleService-Info.plist is the name of the file you downloaded from Firebase.

Add the following CFBundleURLTypes attributes into the [My_project]/ios/Runner/Info.plist file.

```
  <key>CFBundleURLTypes</key>
  <array>
      <dict>
        <key>CFBundleTypeRole</key>
        <string>Editor</string>
       <key>CFBundleURLSchemes</key>
<array>
<!-- Replace this value with the REVERSED_CLIENT_ID copied from GoogleService-Info.plist->
<string>Enter your REVERSED_CLIENT_ID</string>
    </array>
    </dict>
</array>
```

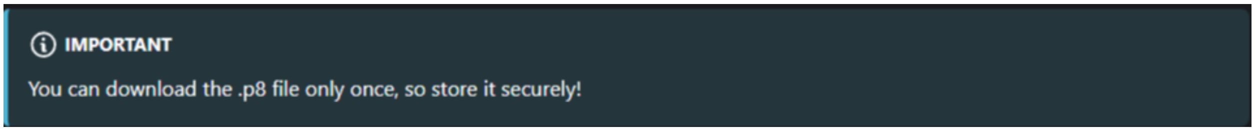
Generate .p8 APNs Authentication Key from Apple Developer Account



ⓘ IMPORTANT

A .p8 certificate (also known as an Apple Push Notification Service (APNs) Authentication Key) is required to send push notifications to iOS devices. This certificate is essential because it allows your server or push notification provider (e.g., Firebase) to authenticate with Apple's APNs (Apple Push Notification Service) and securely deliver notifications to iOS apps.

1. Log in to App Store Connect.
2. Click Certificates, Identifiers & Profiles.
3. Go to Keys → Click the "+" (Create a new key) button..
4. Enter a key name (e.g., FCM Push Key).
5. Enable "Apple Push Notifications service (APNs)" and click Continue.
6. Click Register and Download the .p8 file (It will contain a Key ID).



ⓘ IMPORTANT

You can download the .p8 file only once, so store it securely!

7. Note your Key ID and Team ID (found in Apple Developer → Membership section).

Now Let's Add .p8 Certificate in Firebase

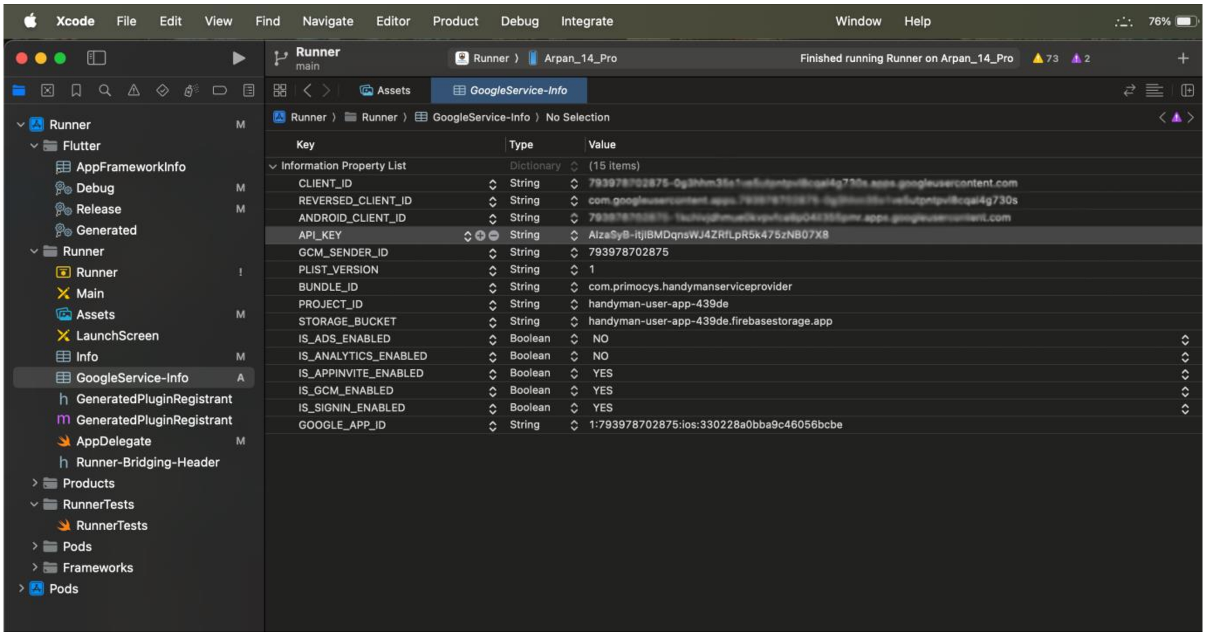
1. Go to Firebase Console
2. Select your iOS app.
3. In the left sidebar, go to Project Settings (⍰) → Cloud Messaging.
4. Under iOS app configuration, find APNs Authentication Key and click Upload.
5. Upload your .p8 file and enter the required details:
    - Key ID (from the Apple Developer account).

- Team ID (from your Apple Developer Membership).
- Bundle ID (of your iOS app).

6. Click **Save**.

Now Let's add firebase generated info.plist in iOS Flutter Project:

1. Open Xcode on your Mac computer.
2. Navigate to the directory of your Flutter project on your computer.
3. Inside your Flutter project directory, locate the ios folder and open it.
4. Look for a file with the extension `.xcworkspace` within the ios folder. This file represents your Xcode project for the Flutter app.
5. Select the `.xcworkspace` or `.xcodeproj` file and click on the "Open" button.
6. Drag and drop `GoogleService-Info.plist` under `info.plist`.



Key	Type	Value
Information Property List	Dictionary	(15 items)
CLIENT_ID	String	[redacted]
REVERSED_CLIENT_ID	String	[redacted]
ANDROID_CLIENT_ID	String	[redacted]
API_KEY	String	AIzaSyB-itjlBMDqnsWJ4ZRfLpR5k475zNB07X8
GCM_SENDER_ID	String	793978702875
PLIST_VERSION	String	1
BUNDLE_ID	String	com.primocys.handymanserviceprovider
PROJECT_ID	String	handyman-user-app-439de
STORAGE_BUCKET	String	handyman-user-app-439de.firebasestorage.app
IS_ADS_ENABLED	Boolean	NO
IS_ANALYTICS_ENABLED	Boolean	NO
IS_APPINVITE_ENABLED	Boolean	YES
IS_GCM_ENABLED	Boolean	YES
IS_SIGNIN_ENABLED	Boolean	YES
GOOGLE_APP_ID	String	1:793978702875:ios:330228a0bba9c46056bcbe

Enable Push Notifications in Xcode

1. Open your **Xcode project**.
2. Go to **Signing & Capabilities** tab.
3. Click "**+ Capability**" and add **Push Notifications** and **Background Modes** (Enable "Remote Notifications").
4. Ensure your app's **Bundle Identifier** matches the one in Firebase & Apple Developer.

Test Push Notifications

1. Run your app on a **real iOS device** (Push notifications don't work on simulators).
2. Send a test notification from **Firebase Console**:
   - Go to **Cloud Messaging** → **New Notification** → Select your iOS app.
   - Click **Send Test Message** and enter a valid **FCM token** (from logs or app).

Done! Now Firebase can send push notifications to your iOS app using APNs.

If you face issues, check Firebase logs and **APNs authentication errors** in Firebase Cloud Messaging settings.

SUCCESSFUL !!

Great! You have successfully configured the iOS App Firebase for Push Notificationa and Google Social Login!

Run an existing flutter project on IDE

To begin, it’s essential to verify that your Flutter and Integrated Development Environment (IDE) setup has been configured accurately

- Execute the command flutter doctor in your terminal. If any issues arise during this process, make sure to address and resolve them promptly.
- Then open your project. Once the project is open, Android Studio may prompt you to install the dependencies. If not, you can run the flutter pub get from the terminal in the project directory to fetch the dependencies.
- After the dependencies are installed, you should be able to run the app.

Change App Icon

1. Generate the App Icon

- Use an Online Tool to Generate your app icon using a reliable icon generation website.
- For example, visit App Icon Generator.
- Download the Generated Icons
- After generating the icons, download the folder containing the assets for both Android and iOS.

2. Update the App Icon for Android

- Navigate to the following directory in your Flutter project:

```
/android/app/src/main/res
```

- Replace all `mipmap` folders in this directory with the `mipmap` folders from your generated Android icons.

3. Update the App Icon for iOS

- Navigate to the following directory in your Flutter project:

- Replace the existing `Assets.xcassets` folder with the `Assets.xcassets` folder from your generated iOS icons.

```
/ios/Runner/Assets.xcassets
```

Change App Name in Android

Also you need to set the app name from your Android project, follow these steps:

2.1 Open the `AndroidManifest.xml` file

- Open your project folder.
- Navigate to `<project>/android/app/src/main/AndroidManifest.xml`.

2.2 Modify the `android:label`

- In the `AndroidManifest.xml` file, find the following line and set your desired app name:

```
android:label="YOUR_APP_NAME"
```

Change App Name in Ios

Also you need to set the app name from your Ios project, follow these steps:

3.1 Open the `info.plist` file

- Open your project folder.

- Navigate to `<project>/iOS/Runner/info.plist`.

3.2 Modify the `CFBundleName`

- In the `info.plist` file, find the following line and set your desired app name:

```
<key>CFBundleDisplayName</key>

<string>YOUR_APP_NAME</string>

<key>CFBundleName</key>

<string>YOUR_APP_NAME</string>
```

Change base URL

- Open <project>/lib/apo helper/api_helper.dart and replace baseUrl variable value with your own URL

```
static const baseUrl = 'https://your_domain_name/api/';
```

Change App Package

First, you have to find out the existing package name. You can find it out from the top of the android/app/src/main/AndroidManifest.xml file. Now right-click on the project folder from Visual Studio and click on replace in the path. You will get a popup window with two input boxes. In the first box, you have to put the existing package name that you saw in the AndroidManifest.xml file previously and write down your preferred package name in the second box and then click on the Replace All button.

Add Google Map API Key

You need to generate the google API key. Visit this link – https://developers.google.com/maps/documentation/embed/get-api-key

You need to enable the mentioned APIs: Direction API, Distance Matrix API, Geocoding API, Maps SDK for Android, Maps SDK for iOS, Place API.

You have to enable a billing account. Visit this url for activating: https://support.google.com/googleapi/answer/6158867?hl=en

After generating the API key, you have to put it in 2 different places for Android, And iOS.

For android: open <project>/android/app/src/main/AndroidManifest.xml and place the value of com.google.android.geo.API_KEY

```
<meta-data
android:name="com.google.android.geo.API_KEY"android:value="your_project_google_a
pi_key" />
```

For iOS: open <project>/iOS/Runner/AppDelegate.swift and place the value of GMSServices.provideAPIKey

- /iOS/Runner/AppDelegate.swift

```
GMSServices.provideAPIKey(“YOUR_MAP_API_KEY_HERE")
```

Build for Android

Run the following command to build the APK.

```
flutter pub get
flutter build apk
flutter build appbundle
```

This command will compile your Flutter code and generate the APK file. You can find the APK in the build/app/outputs/flutter-apk directory inside your project folder.

Build for iOS

There is no general way to generate apps for iOS. Apple doesn’t allow you to install apps like this. If you want to install it on your iOS device, then you have to deploy it on TestFlight or AppStore. For deploying it, please follow this documentation: https://docs.flutter.dev/deployment/ios

---

Need Help with Setup?

If you'd like to set up **only the website** without the app, or if you want a **complete setup with the app**, we're here to help with **end-to-end support**.

Feel free to reach out to us:

- **Email**: primocys@gmail.com
- **WhatsApp**: wa.me/919601814016

Thanks