

## Astroway Astrologer App

---

### Thank You for your purchase

Thank you for choosing our code. We appreciate your purchase and aim to provide exceptional service. If you need help with the code or documentation, contact our [Support Team](#).

### Latest Flutter & Dart Version used by developer

Flutter 3.35.6 • channel stable • <https://github.com/flutter/flutter.git>  
Framework • revision 9f455d2486 (4 weeks ago) • 2025-10-08 14:55:31 -0500  
Engine • hash a5f2c36e367c13f868cfe98db5806f562c52c35e (revision d2913632a4) (29 days ago) • 2025-10-07 17:26:21.000Z  
Tools • Dart 3.9.2 • DevTools 2.48.0

You Can find your flutter version using below command type in terminal of vscode or android studio

**flutter --version**

And you can upgrade your flutter version Using below command

**flutter upgrade**

### Astroway Partner App

Explore the official Flutter website at <https://docs.flutter.dev/get-started/install> for a complete installation guide.

Prefer video tutorials to Explore More About Flutter? Check out the playlist: [link](#)

### Steps to run this project successfully:

1. Begin by extracting the downloaded code, resulting in the creation of the Partner App - Flutter Code zip folder.
2. Next, extract the contents of the Partner App - Flutter Code folder and launch it in either Android Studio or Visual Studio Code.

3. Access the IDE terminal and enter the command "flutter pub get" to fetch the necessary dependencies that we have used.
4. If you want to run the app in IOS simulator (otherwise skip this step), then execute the following commands in the terminal:

```
cd ios  
pod install  
cd ..
```
5. Finally, open the terminal and input "flutter run" to run the application. This will initiate the execution of the project.

## Change app package name

1. If you are using Android Studio, you can press **CTRL+SHIFT+R** to open Replace in Files tool.  
then search our package name `com.astrowaydiploy.astrologer`
2. and replace it with your package name in all places.

<https://drive.google.com/file/d/1qewHCPg1EaWeYPxiOPM4TmuuXS6qRP2o/view?usp=sharing>

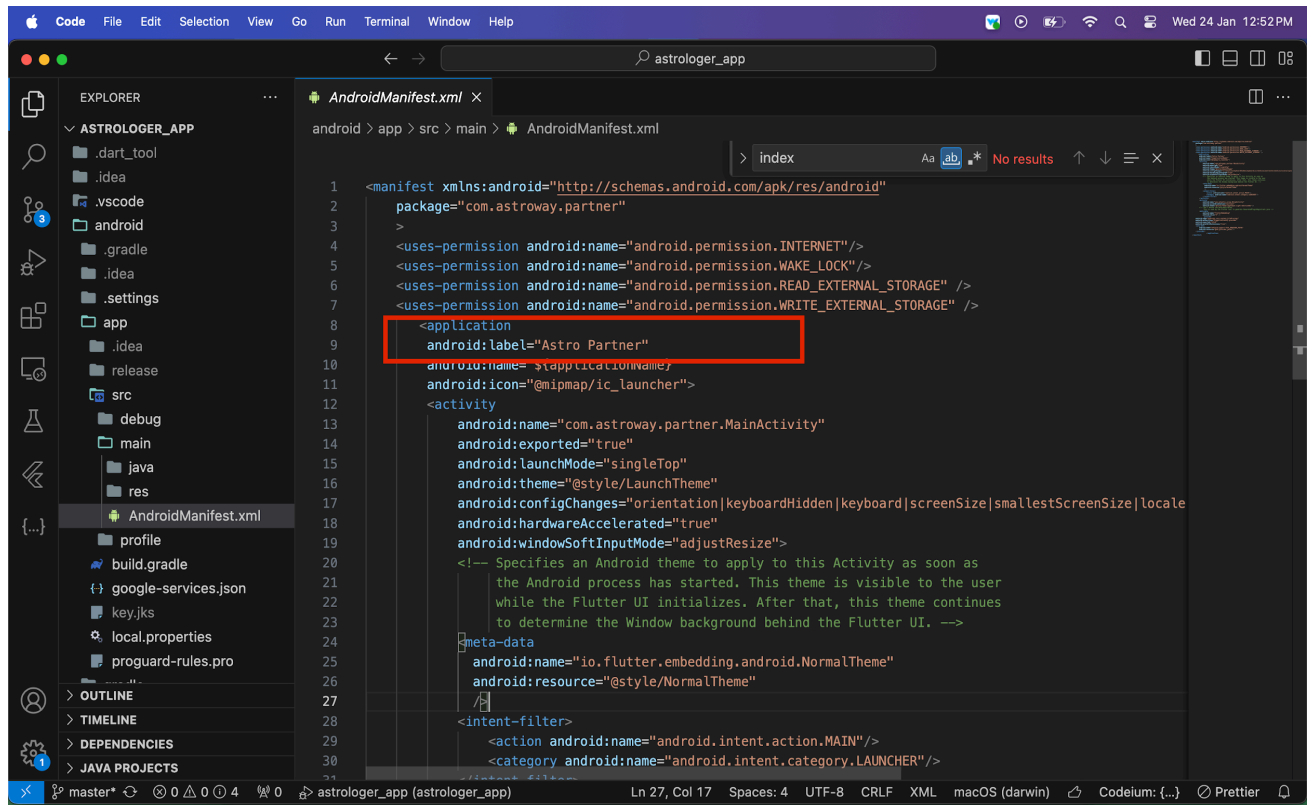
3. And if you are using VS Code, you can press **CTRL+SHIFT+F** to open Find in Files Tool. then search our package name `com.astrowaydiploy.astrologer` and replace it with your package name in all places.

<https://drive.google.com/file/d/1sMHNhmHzvYW0tMmHV8VRtmGbt9TIRjtM/view?usp=sharing>

## Change application name

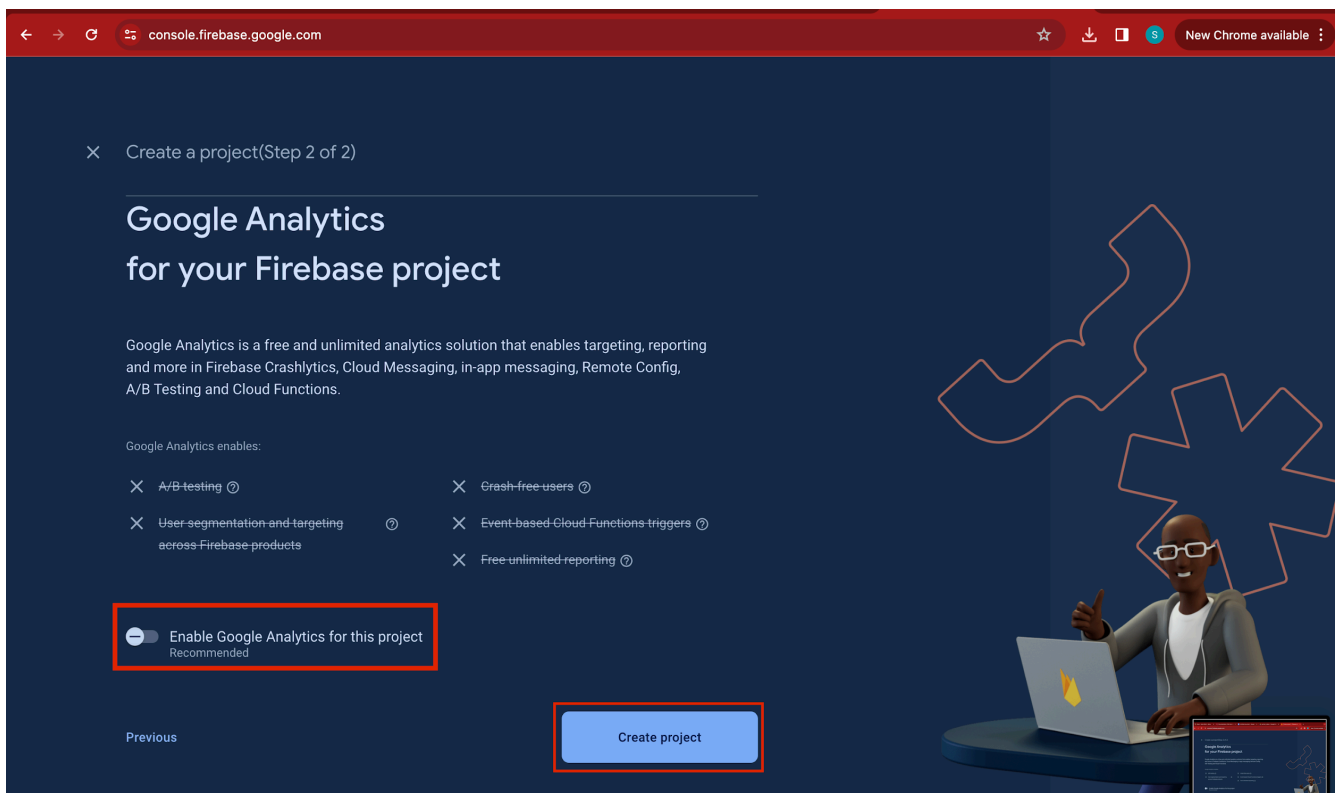
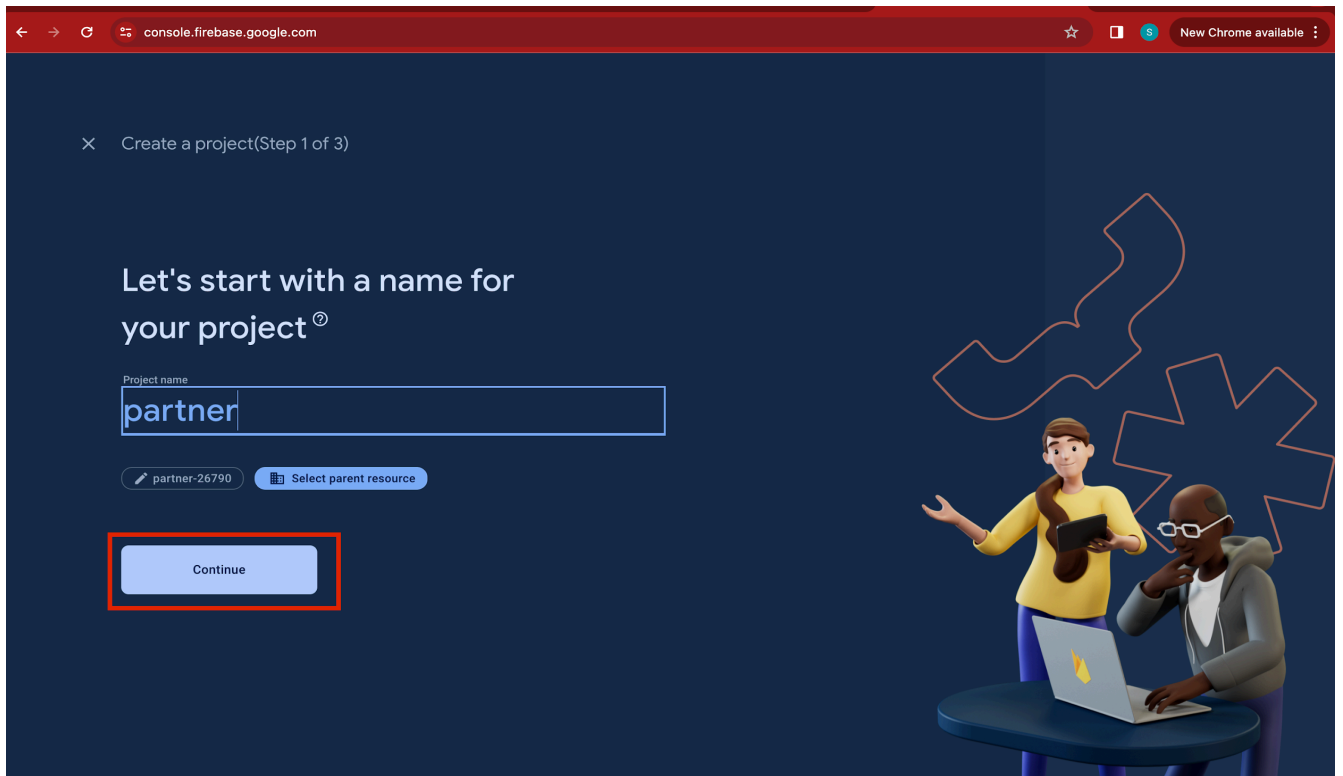
Similarly you can change the app name by finding and replacing in all the places.

Only you Need to Follow this path `android\app\src\main\AndroidManifest.xml` and change your app name on the `android:label` check out the below screenshots



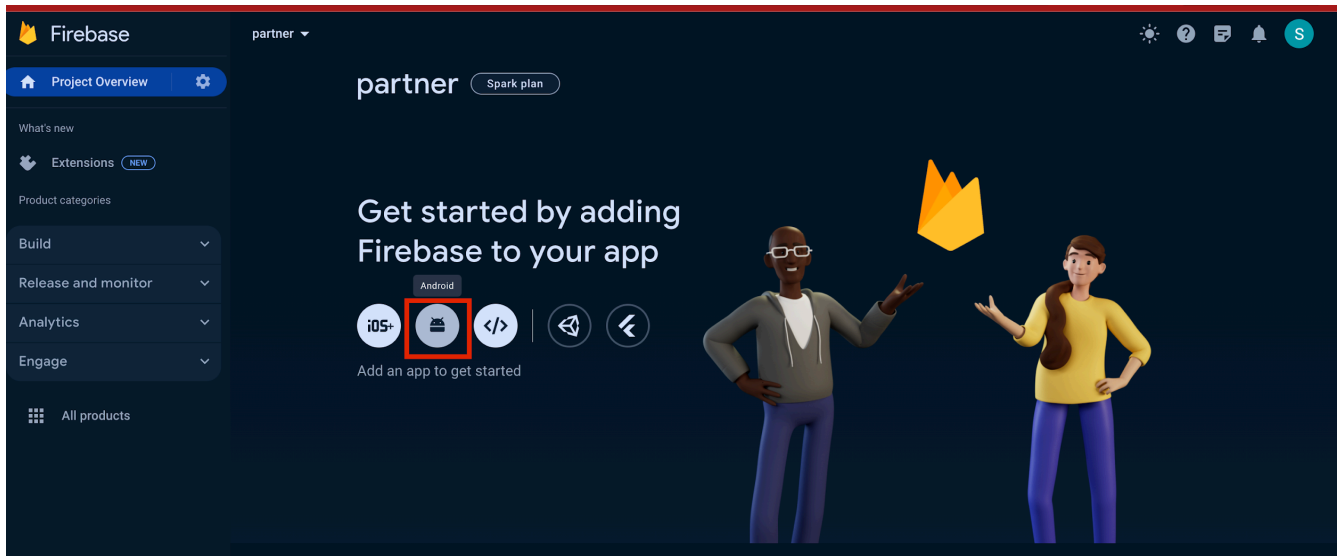
## Integrate firebase with your Project

- First, Create a new Firebase Project
- Go to Firebase Console <https://console.firebase.google.com/>

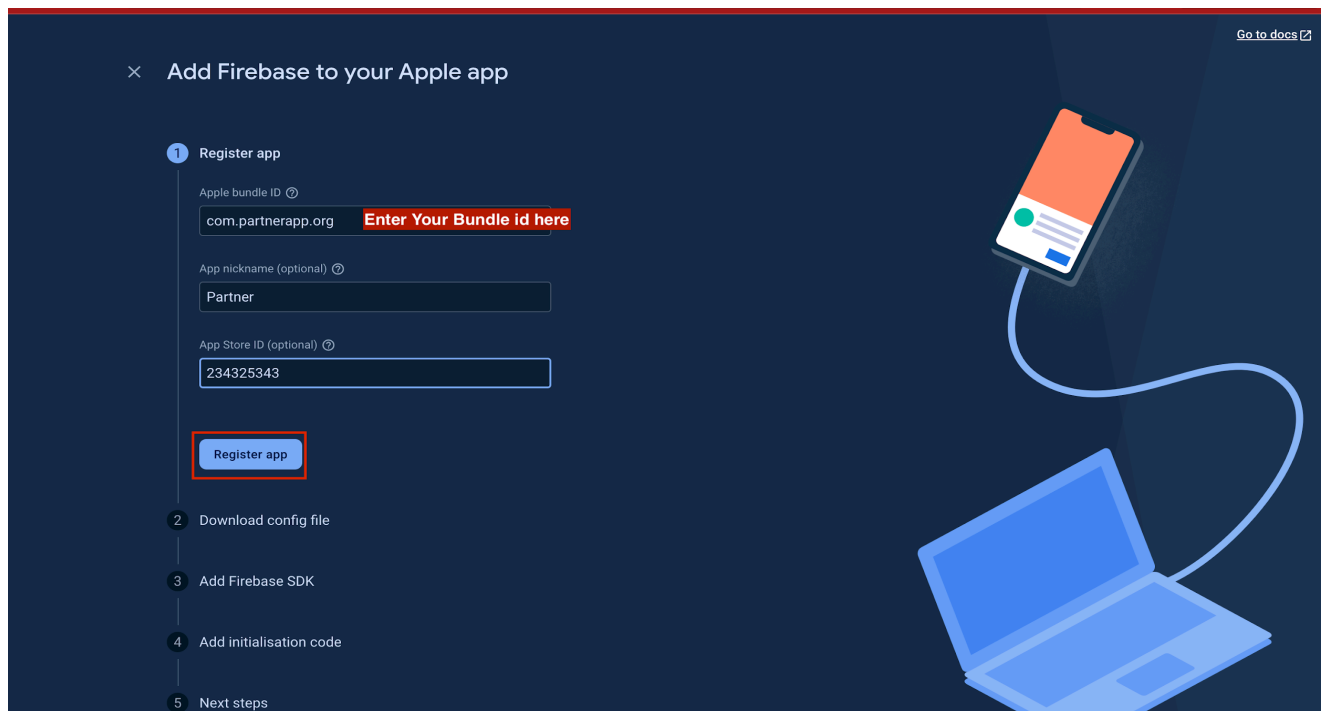
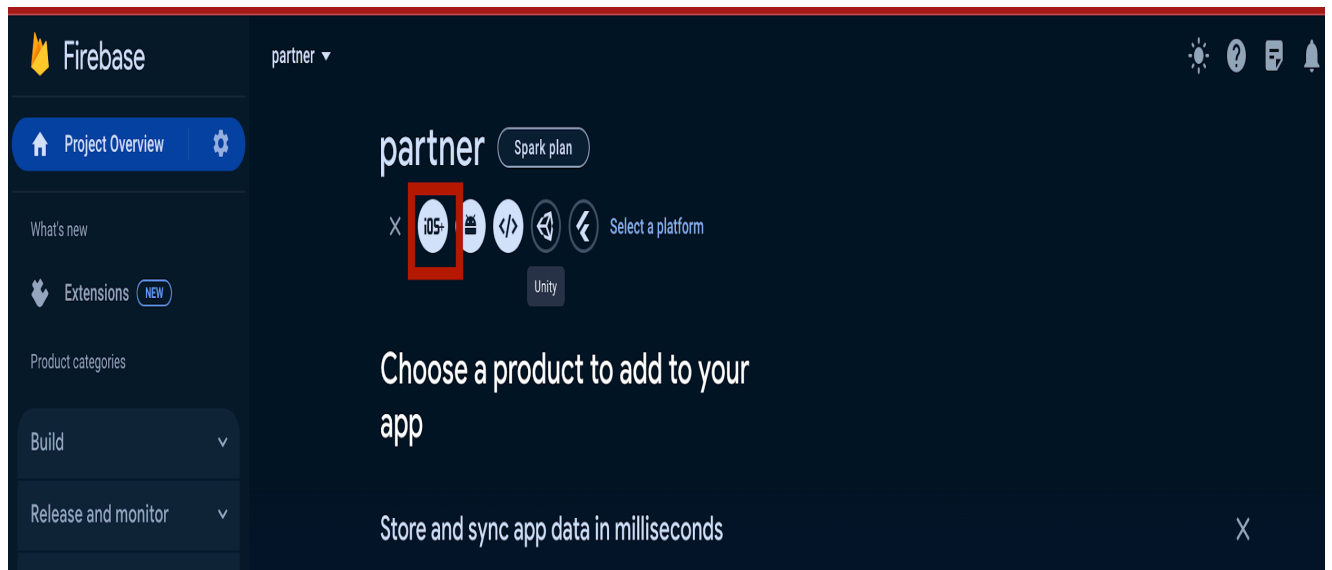




- After creating your Firebase project, you'll now connect your Android application to it

[illegible]

Now it's time for connecting IOS to your Firebase Project.



×

Add Firebase to your Apple app

Go to docs

✓

Register app

Apple bundle ID: com.partnerapp.org, app nickname: Partner, App Store ID: 234325343

2

Download config file

Instructions for Xcode below | [Unity](#) | [C++](#)

Download GoogleService-Info.plist

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

GoogleService-Info.plist

MyApplication

AppDelegate.swift

ViewController.swift

Main.storyboard

Assets.xcassets

LaunchScreen.storyboard

Info.plist

GoogleService-Info.plist

Products


Next

3

Add Firebase SDK

4

Add initialisation code



Skip this step

✓

Register app

Apple bundle ID: com.partnerapp.org, app nickname: Partner, App Store ID: 234325343

Download config file

3

Add Firebase SDK

[CocoaPods](#) | [Download ZIP](#) | [Unity](#) | [C++](#)

Use [Swift Package Manager](#) to install and manage Firebase dependencies.

1. In Xcode, with your app project open, navigate to File > Add packages

2. When prompted, enter the Firebase iOS SDK repository URL:

https://github.com/firebase/firebase-ios-sdk

3. Select the SDK version that you want to use.

We recommend using the default (latest) SDK version, but you can use an older version, if needed.

4. Choose the Firebase libraries that you want to use.

After you click Finish, Xcode will automatically begin resolving and downloading your dependencies in the background.

Previous


Next

4

Add initialisation code

5

Next steps



To connect Firebase when your app starts up, add the initialisation code below to your app's main entry point.

☒ SwiftUI ☐ Swift ☐ Objective-C

```
import SwiftUI
import FirebaseCore


class AppDelegate: NSObject, UIApplicationDelegate {
    func application(_ application: UIApplication,
        didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]? = nil)
        FirebaseApp.configure()
        return true
    }
}

@main
struct YourApp: App {
    // register app delegate for Firebase setup
    @UIApplicationDelegateAdaptor(AppDelegate.self) var delegate

    var body: some Scene {
        WindowGroup {
            NavigationView {
                ContentView()
            }
        }
    }
}
```

Previous **Next**

5 Next steps



## × Add Firebase to your Apple app

- ✓ Register app  
Apple bundle ID: com.partnerapp.org, app nickname: Partner, App Store ID: 234325343
- ✎ Download config file
- ✎ Add Firebase SDK
- ✎ Add initialisation code
- 5 Next steps**


You're ready!

Make sure that you take a look at the [documentation](#) to learn how to get started with each Firebase product that you want to use in your app.

You can also explore [sample Firebase apps](#).

Or, continue to the console to explore Firebase.

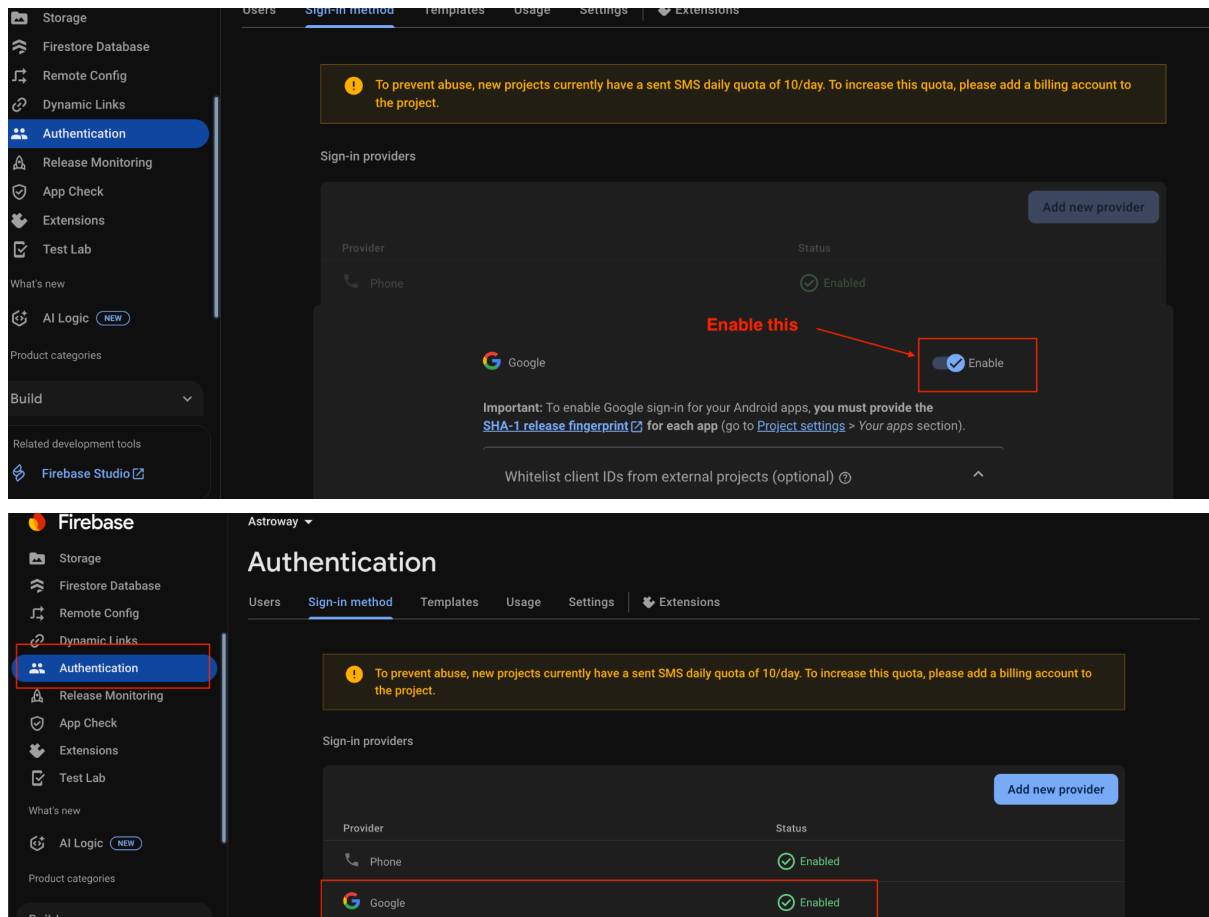
Previous **Continue to the console**



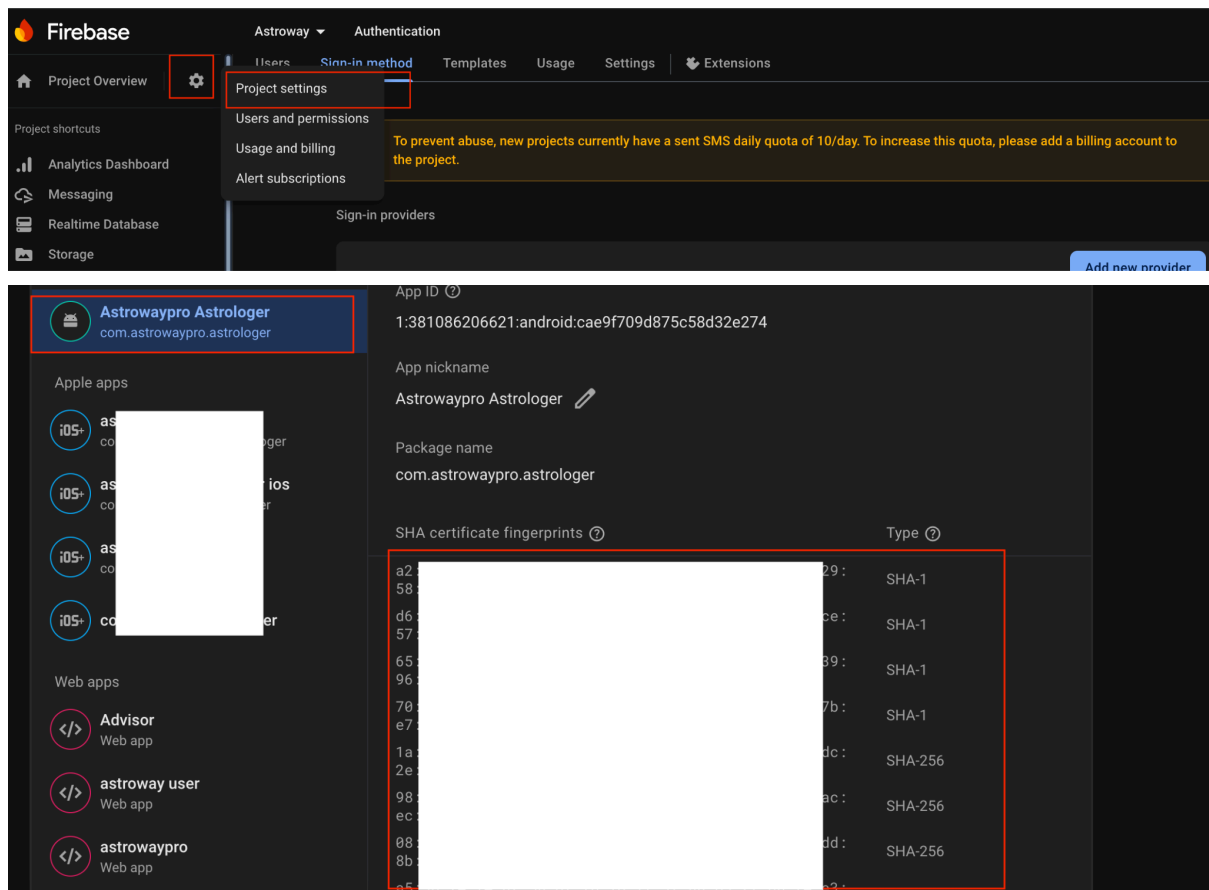
Congrats! You have successfully connected your android & IOS app to your firebase project.

## 1.Enable Firebase Authentication

Go to firebase and in left side search Authentication and enable google sign in like in below screenshot



And get SHA-1 and SHA-256 keys [debug and release both ] and add in below firebase setting



HOW TO GENERATE SHA KEYS- **cd android** in terminal

```

sunilsingh@sunils-MacBook-Air:~/AstroExpert_pro_partner % cd android
sunilsingh@sunils-MacBook-Air:~/AstroExpert_pro_partner/android % ./gradlew signingReport

Welcome to Gradle 8.11.1!

Here are the highlights of this release:
- Parallel load and store for Configuration Cache
- Java compilation errors at the end of the build output
  
```

And get both debug and release keys and paste above remember you have to add first jks file in **android** → **app** → **yourjksfile.jks**

**IMPORTANT** - you need to change your folders name inside kotlin folder goto **android/app/src/main/kotlin/com/astroway/astrologer\_app/MainActivity.kt**

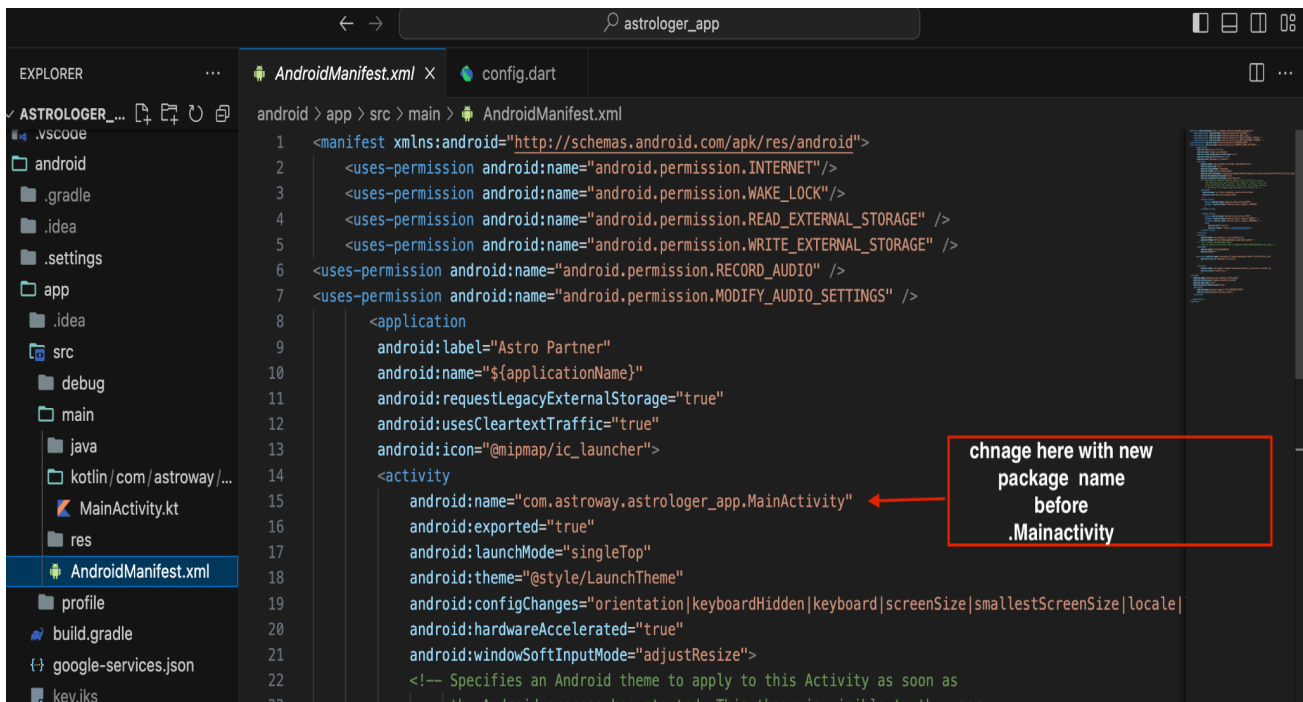
For example:-**com.astroway.astrologer\_app**

Then first folder name:-**com**

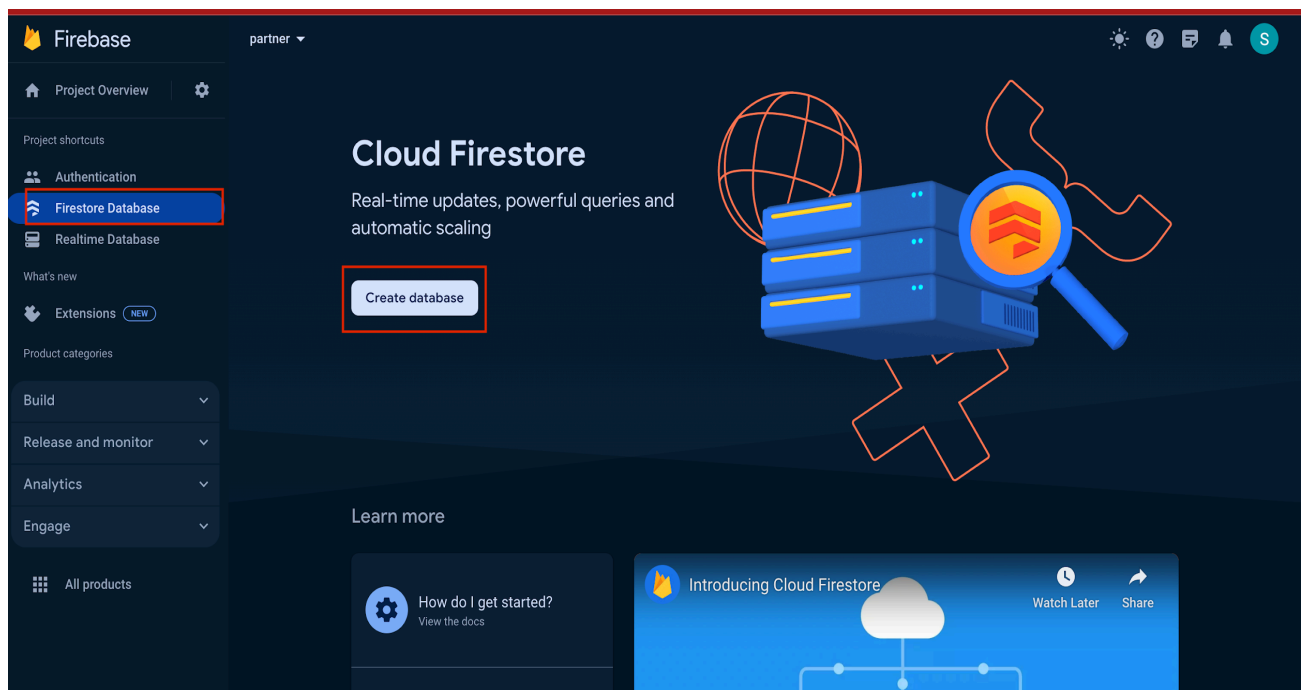
Second folder name:-**astroway**

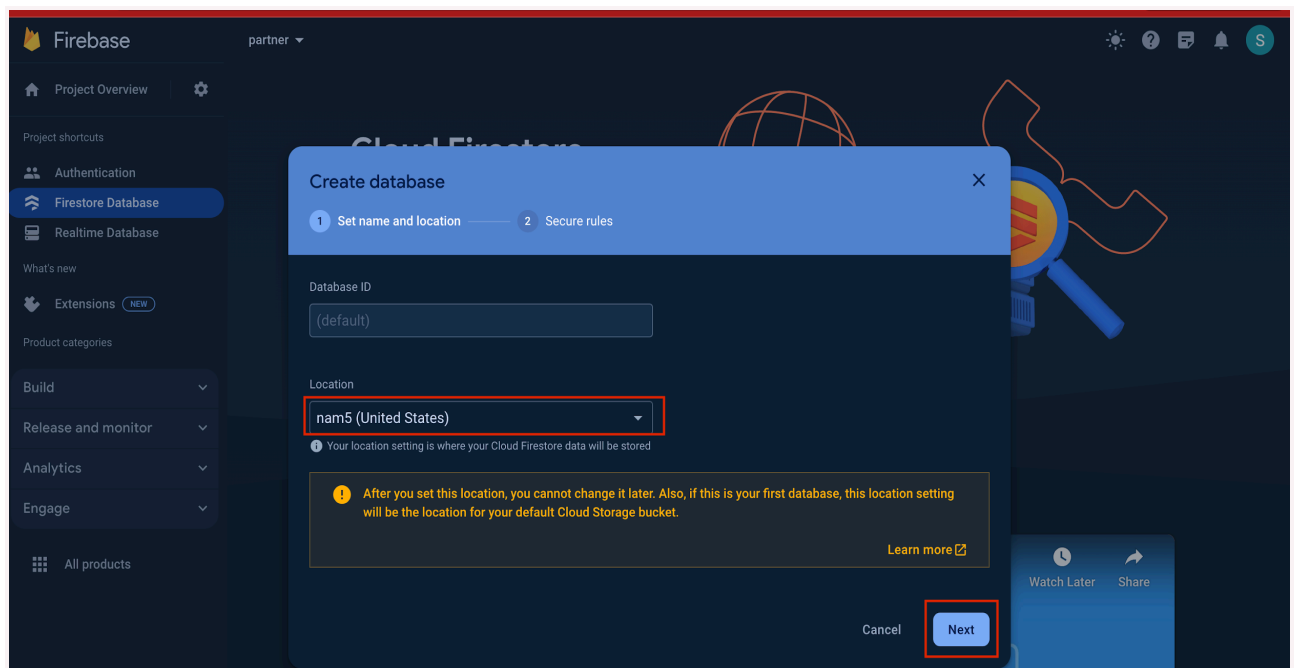
Third folder name:- **astrologer\_app**

replace this package name With your own package name  
And change that in manifest too like i have done below



### 3.Create Firestore Database



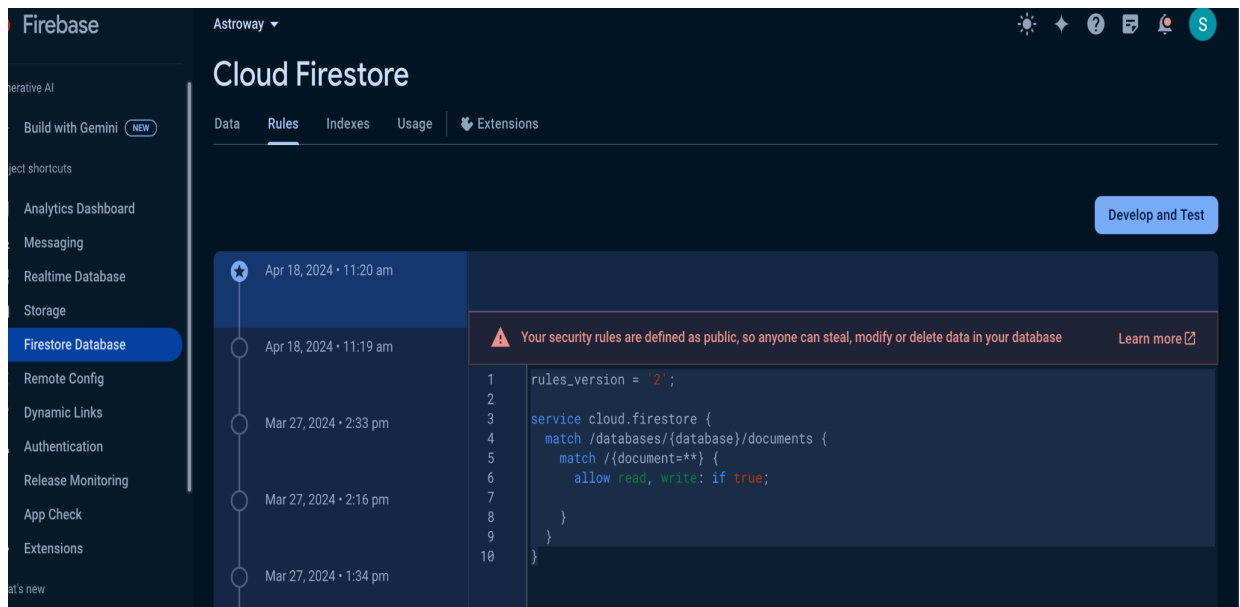


Make Sure to add these rules in your **firestore\_firestore**, otherwise it won't work.

```
rules_version = '2';
rules_version = '2';

service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if true;
    }
  }
}
```

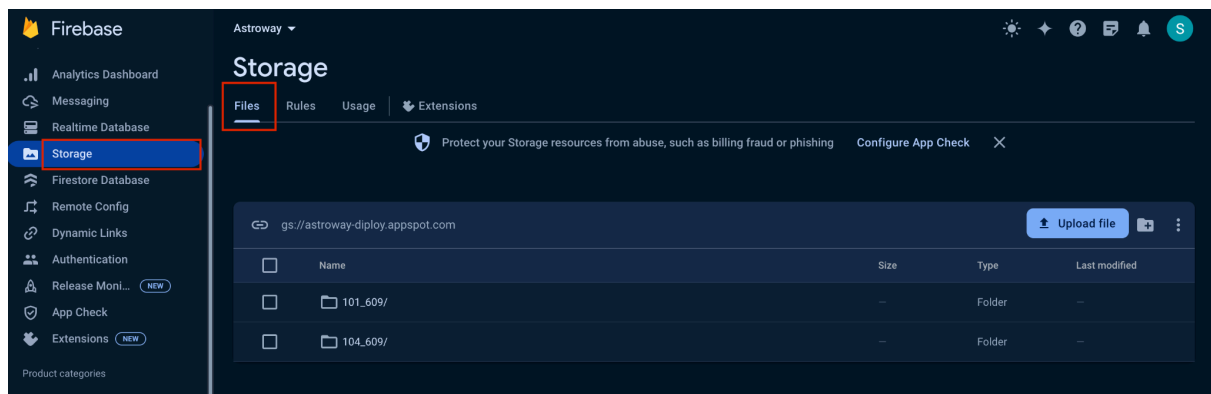


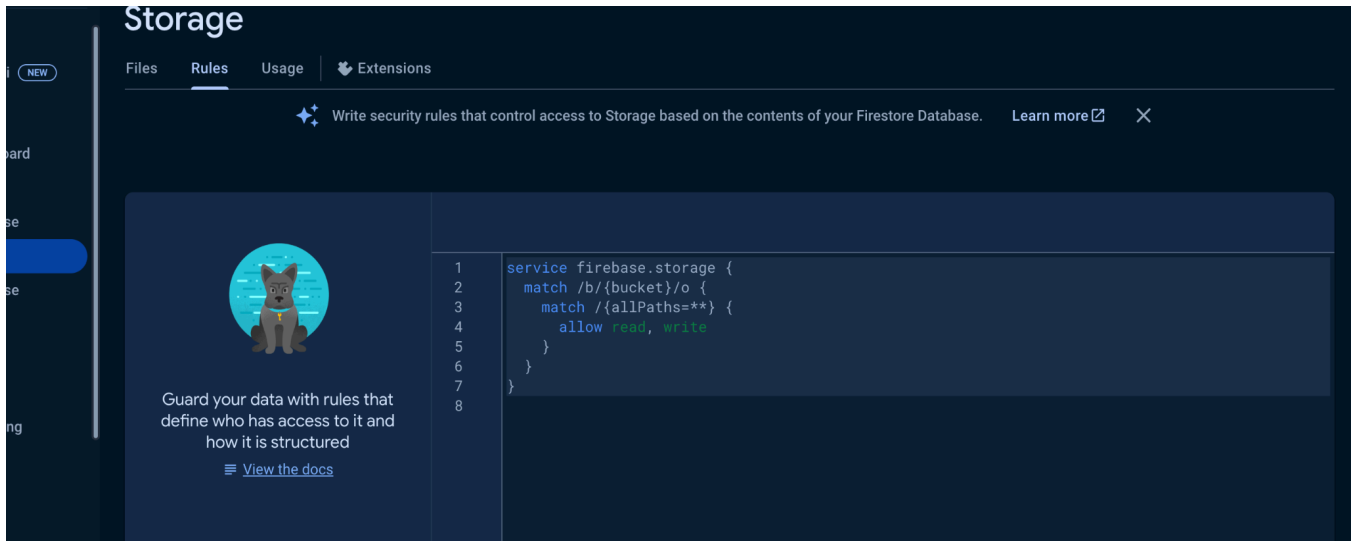


## 4.Enable Firebase storage to support chat with images/pdf

In rules section paste the below code

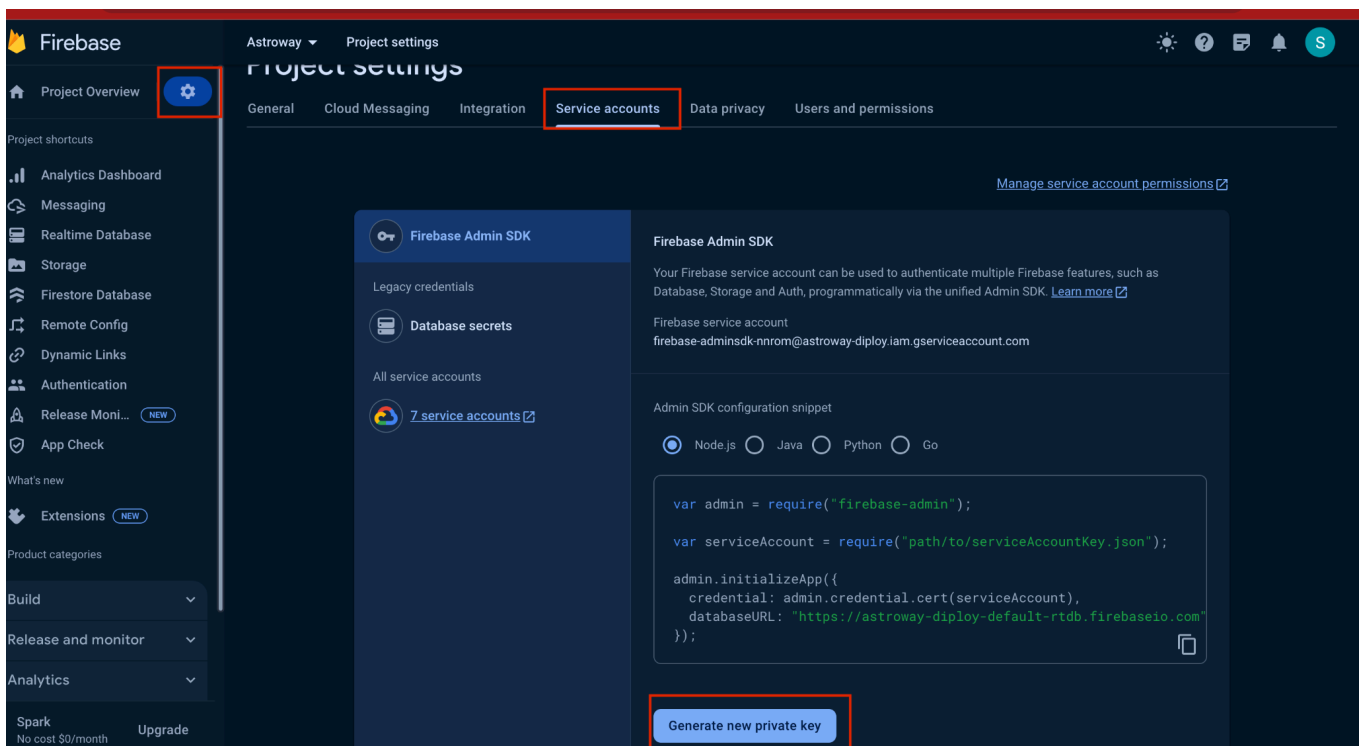
```
service firebase.storage {
  match /b/{bucket}/o {
    match /{allPaths=**} {
      allow read, write
    }
  }
}
```



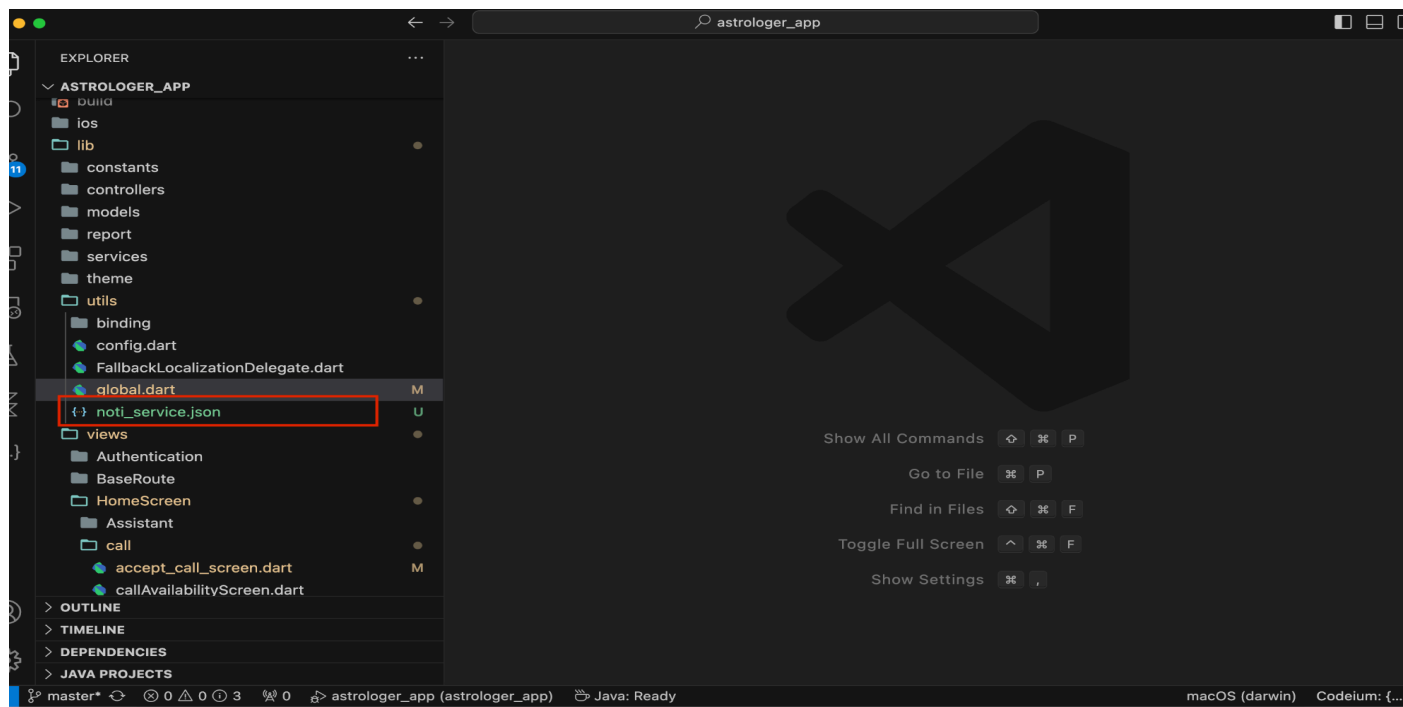


## Enable Notifications for App

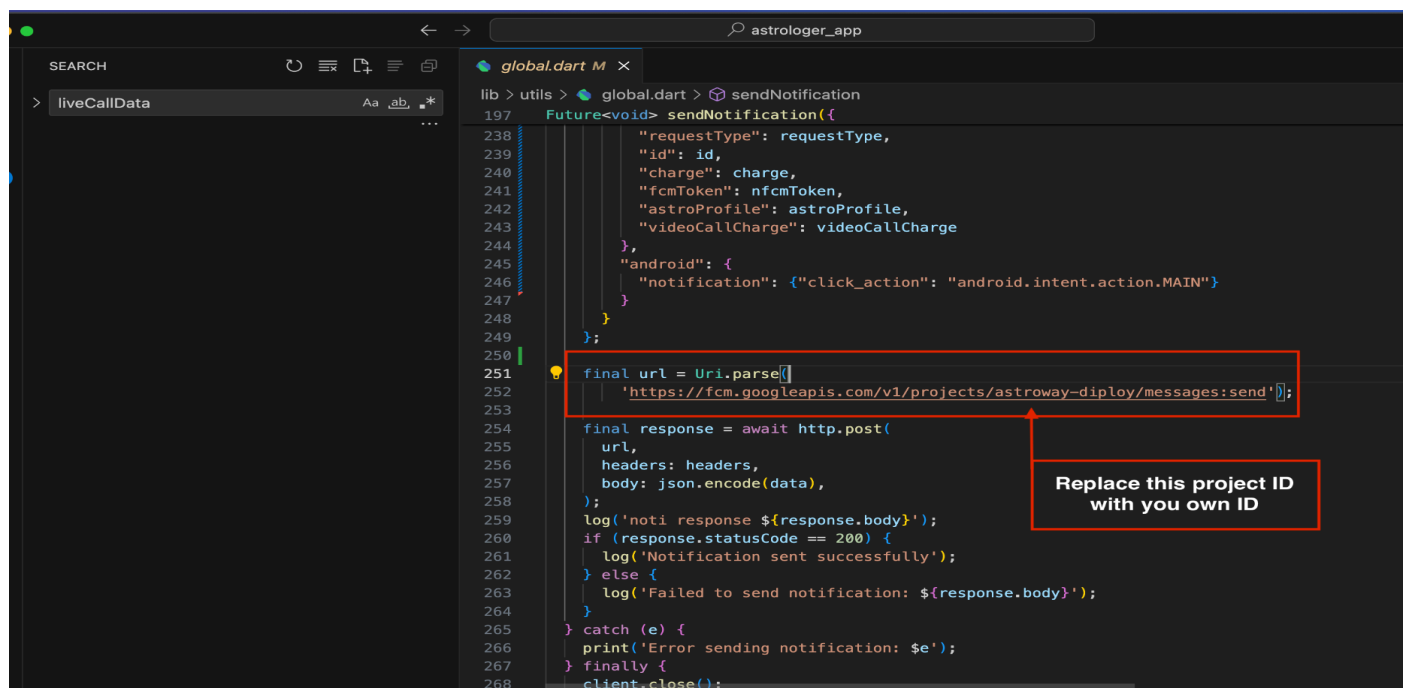
Access your Firebase project and locate "Project Settings" > "Service account TAB" and click on **Generate new Private key**



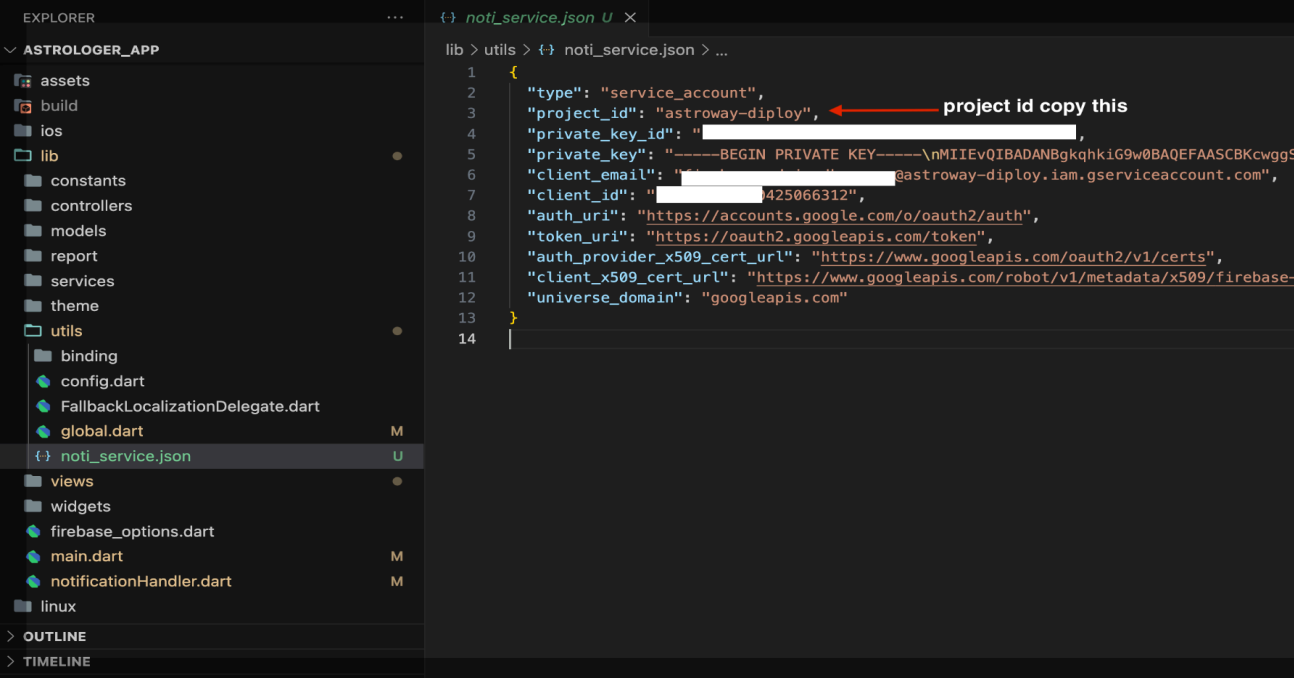
1. Download the file.json and save it into [lib/Utils/noti\\_service.json](#)



2. Now go to [lib/Utils/global.dart](#)



And replace the projectID from your project id which is in the .json file you downloaded earlier



The screenshot shows an IDE interface. On the left, the 'EXPLORER' panel displays the project structure for 'ASTROLOGER\_APP'. The 'lib' directory is expanded, showing subdirectories like 'constants', 'controllers', 'models', 'report', 'services', 'theme', and 'utils'. The 'utils' directory is selected, and the file 'noti\_service.json' is highlighted. On the right, the editor shows the content of 'noti\_service.json'. The JSON object contains fields for 'type', 'project\_id', 'private\_key\_id', 'private\_key', 'client\_email', 'client\_id', 'auth\_uri', 'token\_uri', 'auth\_provider\_x509\_cert\_url', 'client\_x509\_cert\_url', and 'universe\_domain'. A red arrow points to the 'project\_id' field, which is 'astroway-diploy', with the text 'project id copy this' next to it. The 'private\_key' field is redacted with a black box.

```
1  {
2    "type": "service_account",
3    "project_id": "astroway-diploy",
4    "private_key_id": " ",
5    "private_key": "-----BEGIN PRIVATE KEY-----\nMIIIEvQIBADANBgkqhkiG9w0BAQEFAASCCKwggS
6    "client_email": " " @astroway-diploy.iam.gserviceaccount.com",
7    "client_id": "425066312",
8    "auth_uri": "https://accounts.google.com/o/oauth2/auth",
9    "token_uri": "https://oauth2.googleapis.com/token",
10   "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
11   "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/firebase-
12   "universe_domain": "googleapis.com"
13 }
14
```

**Download the google-services.json file and replace it in code**

You will get the google-services.json file from **project setting in firebase>general** and on **your app section**

**Url:- <https://console.firebase.google.com/project>**

### 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.



See SDK instructions




google-services.json

App ID ⓘ

1:381086206621:android:f82a6af18b824fbd32e274

App nickname

Astroway User 

Package name

com.astrowaydiploy.user

**Note:- This image is only for reference**

And you need to replace it from **google-services.json** file in your code

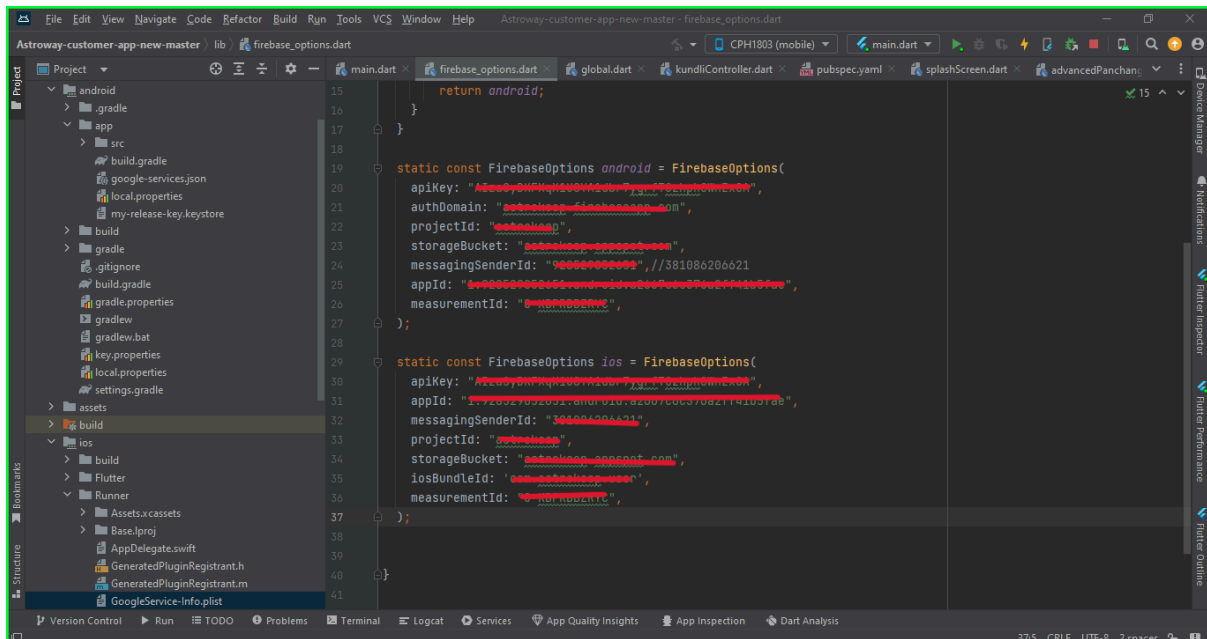
**Path:- project>android>app>google-services.json** file

### Changes in firebase\_options.dart file

Now you need to make changes in firebase\_options.dart file

File path:- **project>lib>firebase\_options.dart**

For this file you can get data from the noti\_service.json and google-services.json please make sure all the data that you are changing in **firebase\_options.dart** is same as in google-service.json

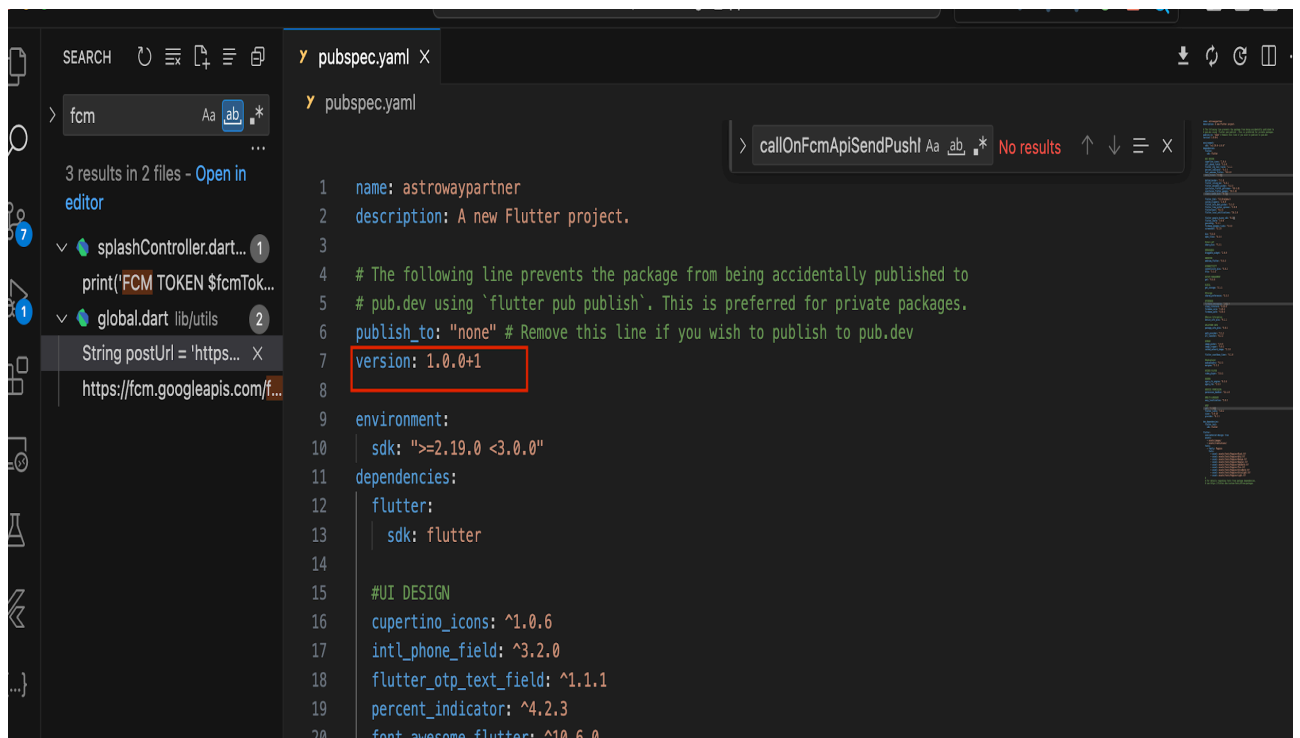


**Note:-** This image is only for reference

## Change application version and app launcher icon

Go to `pubspec.yaml` Update the `version:A.B.C+X` as shown in image below.

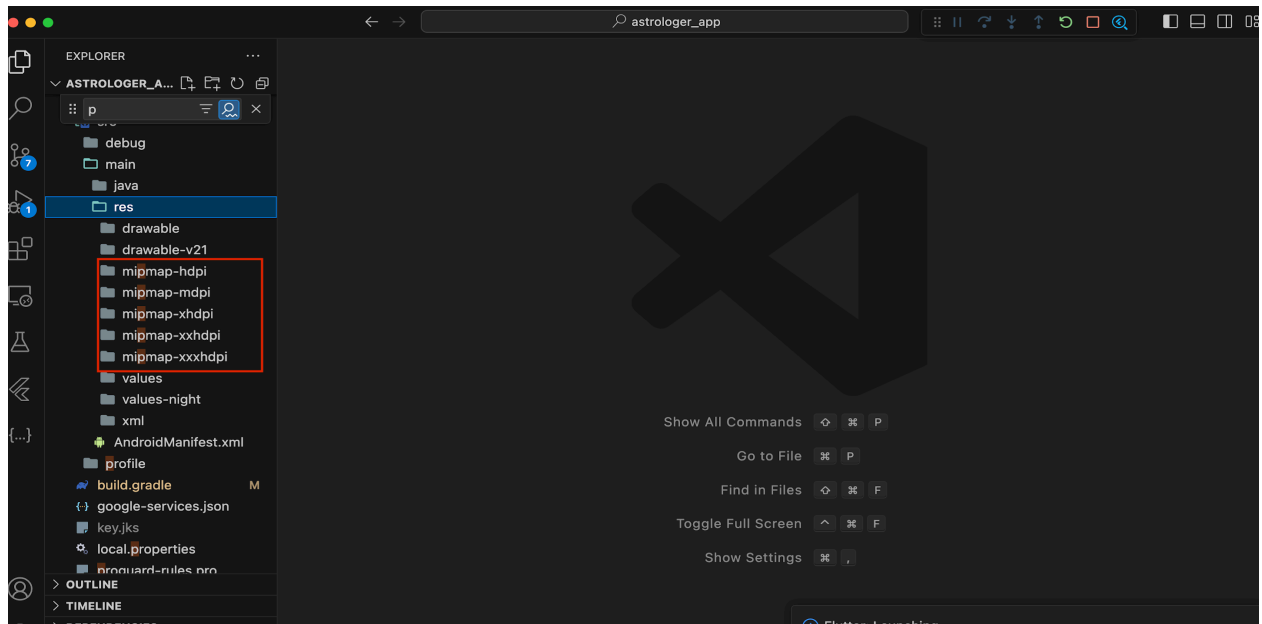
**Note :-** Make sure execute `flutter packages get`, `flutter build` or `flutter run` after changing the version.



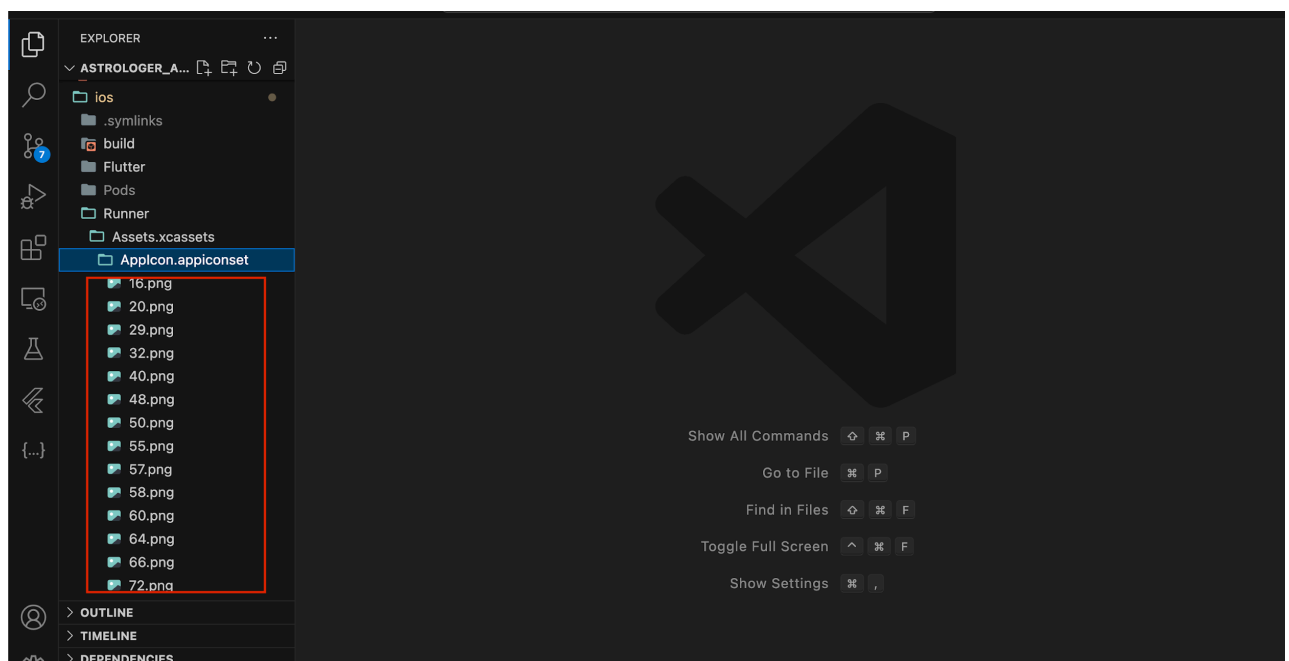
## Change App Logo

For Creating the logo Go to the web site <https://www.appicon.co/> paste your app logo here and tick the android and ios boxes only, now click on the Generate button it will generate a zip folder for you, now extract this folder and follow the below instructions for changing the app logo

1.For Android, open android > app > src > main > res and add here your logo according to device screen size



2.For IOS open, ios > Runner > Assets.xcassets > AppIcon.appiconset here and add your logo according to different size.



## Set app's default language

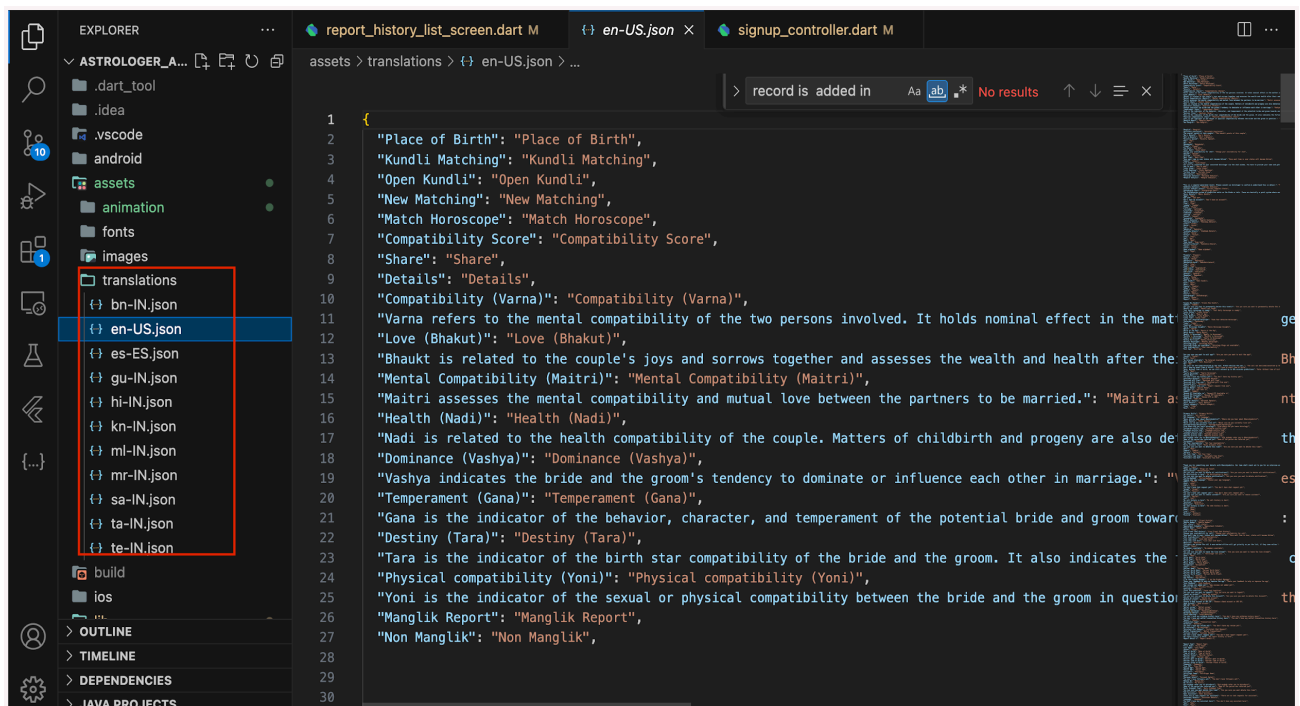
The App's default language is English, but you can easily modify it. To do so, navigate to the `lib/main.dart` file and locate the section displaying the language codes, as illustrated in the image below.

You can obtain your desired language code from the following link:

<https://developers.google.com/admin-sdk/directory/v1/languages>.

Just ensure that the language code you select is available in the aforementioned link for proper functionality.

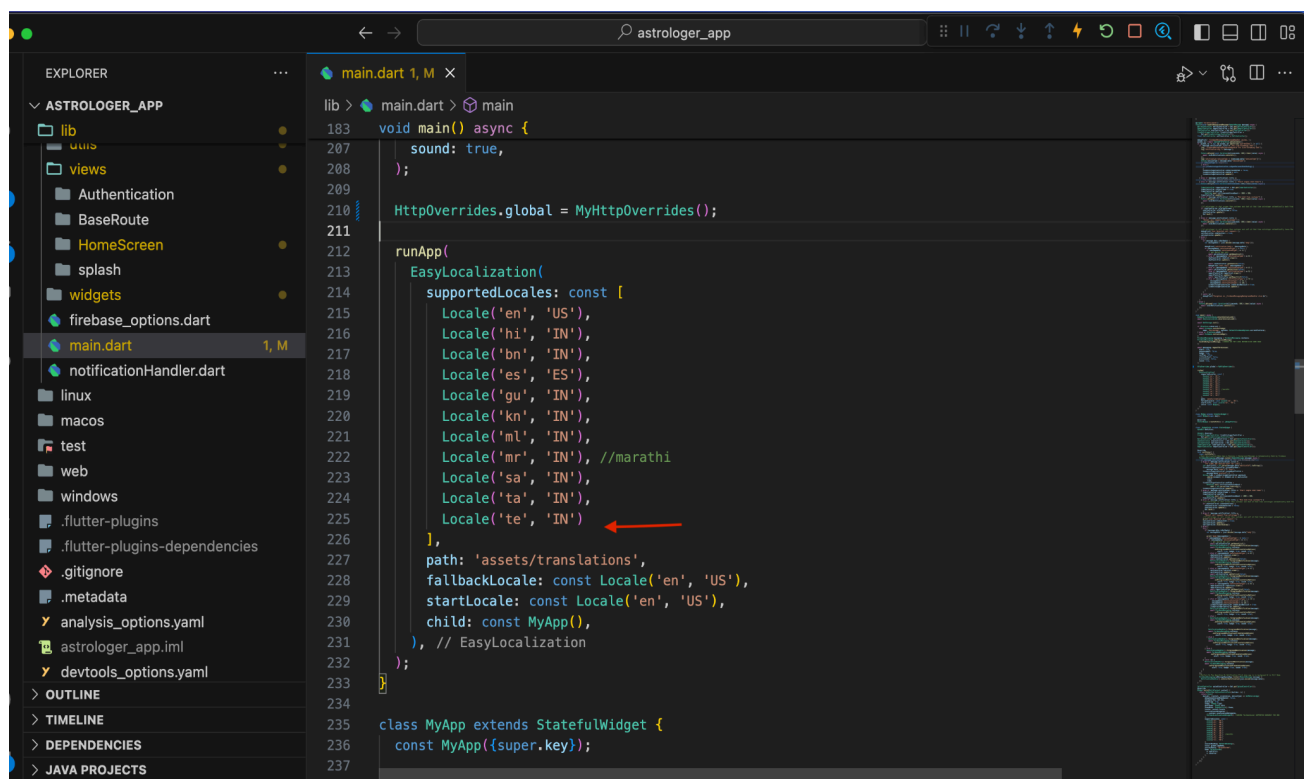
```
runApp(  
  EasyLocalization(  
    supportedLocales: const [  
      Locale('en', 'US'),  
      Locale('hi', 'IN'),  
    ],  
    path: 'assets/translations',  
    fallbackLocale: const Locale('en', 'US'), //Fallback language  
    startLocale: const Locale('en', 'US'), //Default language set  
    child: const MyApp(),  
  ),  
);  
}
```



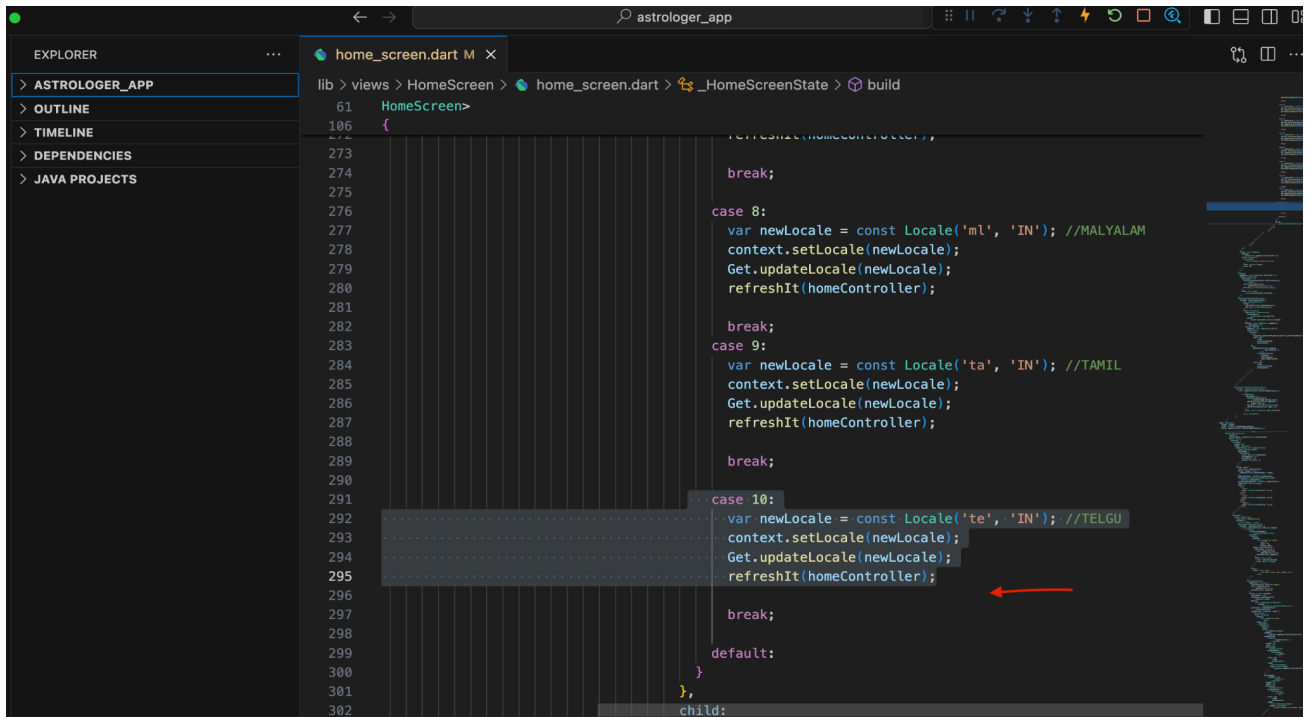


If you are adding a new language to the application , make sure to create a separate file inside **assets/translations/** with the appropriate language code (e.g., **en-US.json** for English with the United States country code). if your desired language is not available. Copy all labels from en-US.json and translate the values accordingly.

And add Your language code in main.dart file

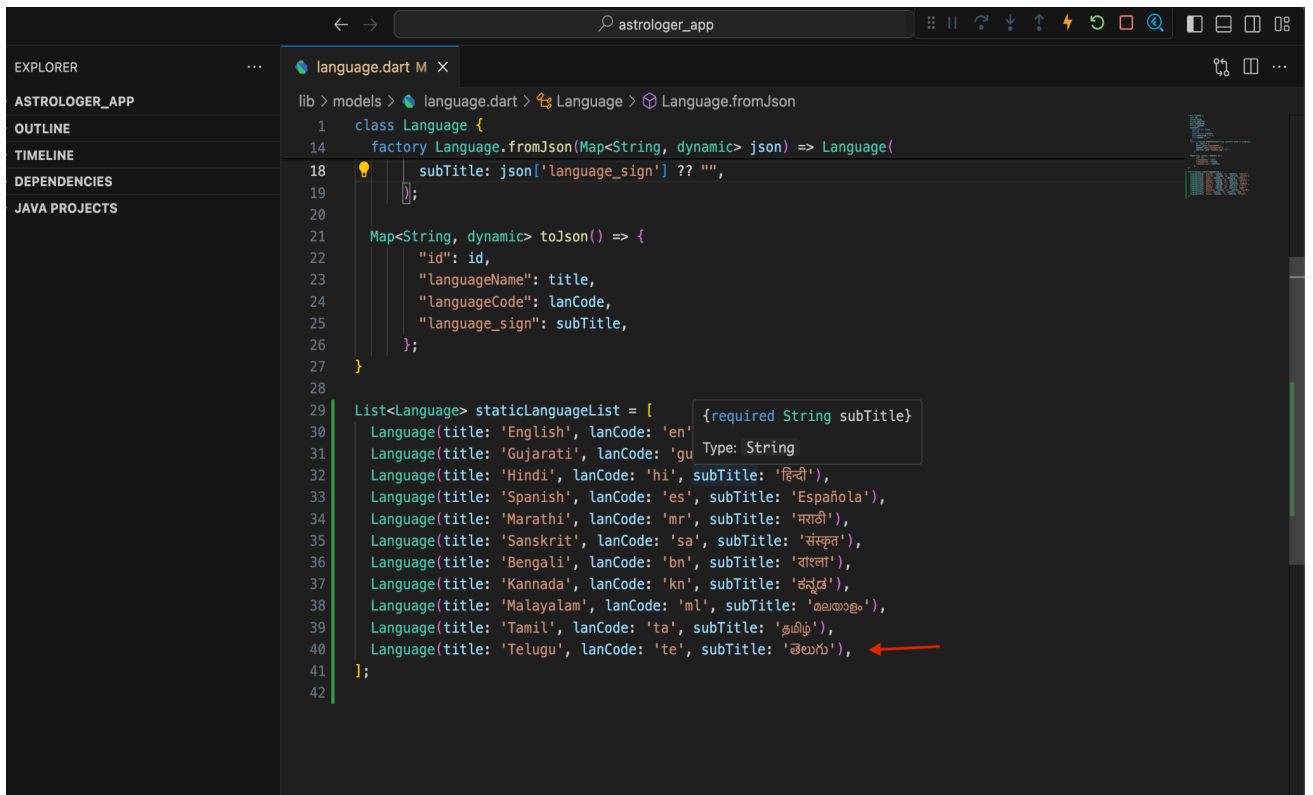


Update Homescreen.dart with new language code case 11



```
lib > views > HomeScreen > home_screen.dart > _HomeScreenState > build
61   HomeScreen>
106   {
272   //refresh(homeController);
273
274   break;
275
276   case 8:
277     var newLocale = const Locale('ml', 'IN'); //MALYALAM
278     context.setLocale(newLocale);
279     Get.updateLocale(newLocale);
280     refreshIt(homeController);
281
282     break;
283   case 9:
284     var newLocale = const Locale('ta', 'IN'); //TAMIL
285     context.setLocale(newLocale);
286     Get.updateLocale(newLocale);
287     refreshIt(homeController);
288
289     break;
290
291     .. case 10:
292     var newLocale = const Locale('te', 'IN'); //TELUGU
293     context.setLocale(newLocale);
294     Get.updateLocale(newLocale);
295     refreshIt(homeController);
296
297     break;
298
299   default:
300   }
301 },
302 child:
```

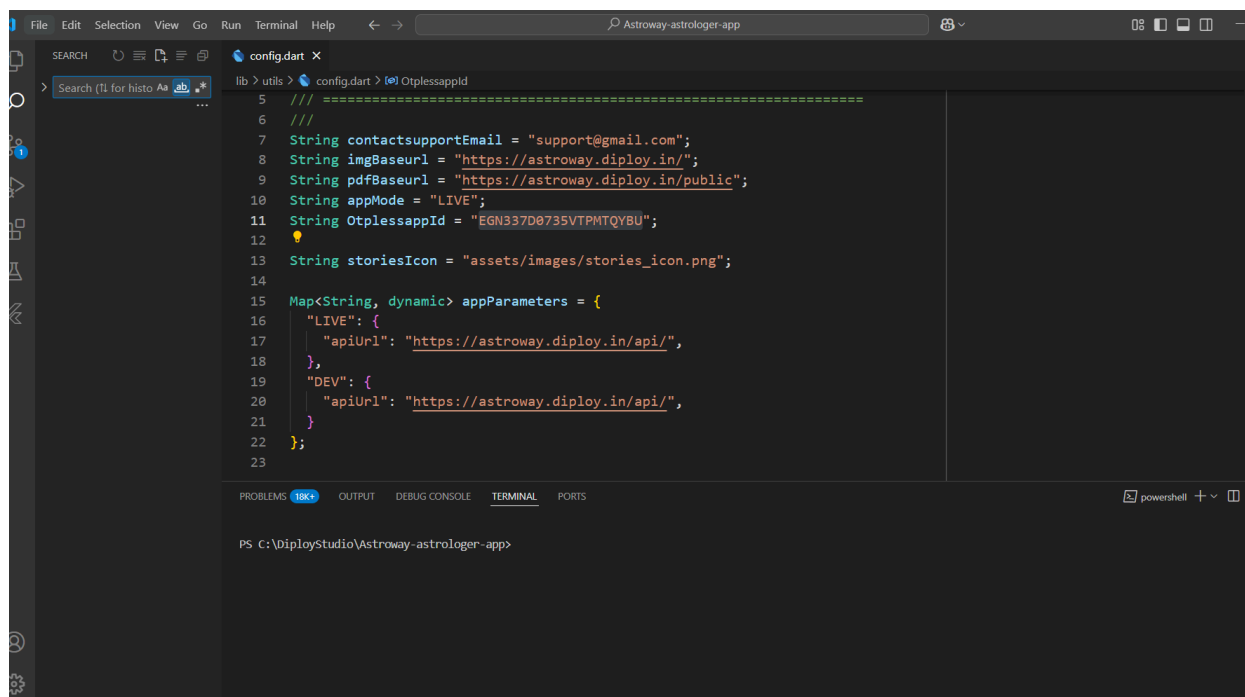
Now Update Language.dart file add new language in list



```
lib > models > language.dart > Language > Language.fromJson
1   class Language {
14   factory Language.fromJson(Map<String, dynamic> json) => Language(
18     subtitle: json['language_sign'] ?? "",
19   );
20
21   Map<String, dynamic> toJson() => {
22     "id": id,
23     "languageName": title,
24     "languageCode": lanCode,
25     "language_sign": subtitle,
26   };
27 }
28
29 List<Language> staticLanguageList = [
30   Language(title: 'English', lanCode: 'en', subtitle: 'English'),
31   Language(title: 'Gujarati', lanCode: 'gu', subtitle: 'ગુજરાતી'),
32   Language(title: 'Hindi', lanCode: 'hi', subtitle: 'हिन्दी'),
33   Language(title: 'Spanish', lanCode: 'es', subtitle: 'Española'),
34   Language(title: 'Marathi', lanCode: 'mr', subtitle: 'मराठी'),
35   Language(title: 'Sanskrit', lanCode: 'sa', subtitle: 'संस्कृत'),
36   Language(title: 'Bengali', lanCode: 'bn', subtitle: 'বাংলা'),
37   Language(title: 'Kannada', lanCode: 'kn', subtitle: 'ಕನ್ನಡ'),
38   Language(title: 'Malayalam', lanCode: 'ml', subtitle: 'മലയാളം'),
39   Language(title: 'Tamil', lanCode: 'ta', subtitle: 'தமிழ்'),
40   Language(title: 'Telugu', lanCode: 'te', subtitle: 'తెలుగు'),
41 ];
42
```

## Change App Base Url

You need to update keys such as **CONTACT\_SUPPORT** , **IMAGE\_BASE\_URL**, **PDF\_BASE\_URL**, or **API\_URL**, simply open the **config.dart** file and paste your keys there.



```
5 // =====
6 //
7 String contactsupportEmail = "support@gmail.com";
8 String imgBaseUrl = "https://astroway.diploy.in/";
9 String pdfBaseUrl = "https://astroway.diploy.in/public";
10 String appMode = "LIVE";
11 String OtplessappId = "EGN337D0735VTPMTQYBU";
12
13 String storiesIcon = "assets/images/stories_icon.png";
14
15 Map<String, dynamic> appParameters = {
16   "LIVE": {
17     "apiUrl": "https://astroway.diploy.in/api/",
18   },
19   "DEV": {
20     "apiUrl": "https://astroway.diploy.in/api/",
21   }
22 };
23
```

## Change Google Places API Key

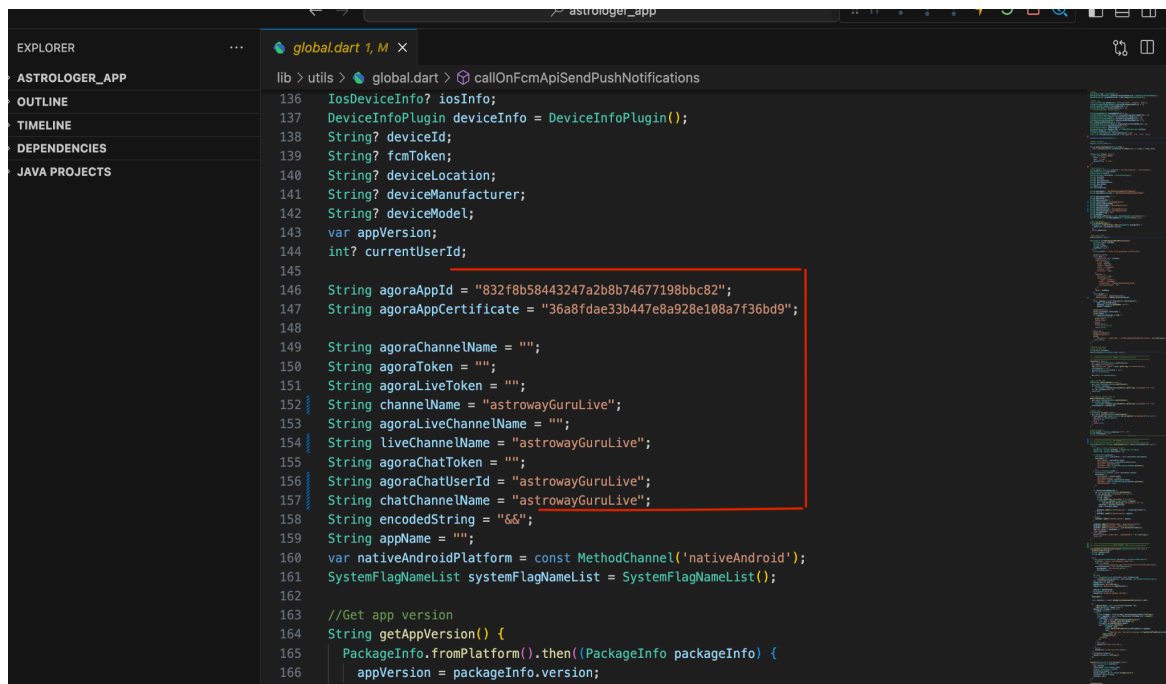
To change the API key for Google Places, please follow these steps:

1. Visit the following link to create an API key: [places key generated](#)
2. Once you have obtained the API key, paste it into the **config.dart** file above the existing configuration.

## Change Agora Credentials

For changing the agora credentials you need to change to channel name inside the global.dart file

**lib>utils>global.dart**

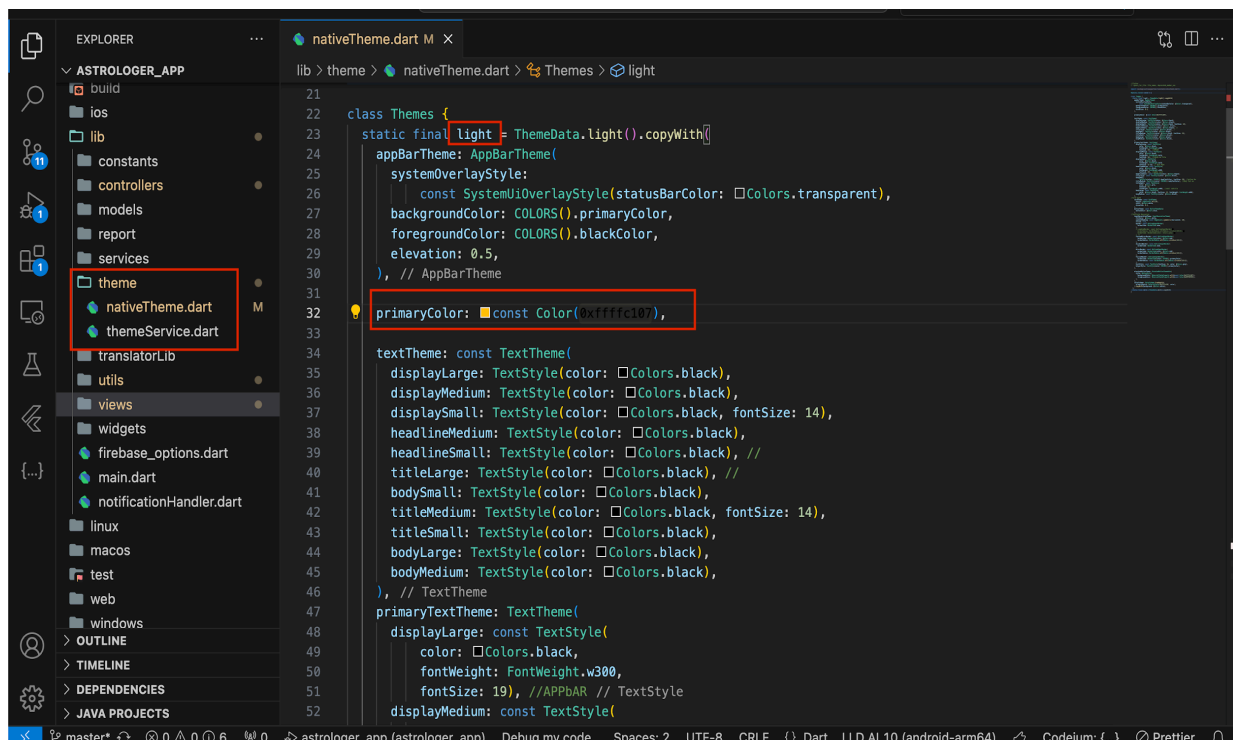


```
lib > utils > global.dart > callOnFcmApiSendPushNotifications
136 IosDeviceInfo? iosInfo;
137 DeviceInfoPlugin deviceInfo = DeviceInfoPlugin();
138 String? deviceId;
139 String? fcmToken;
140 String? deviceLocation;
141 String? deviceManufacturer;
142 String? deviceModel;
143 var appVersion;
144 int? currentUserId;
145
146 String agoraAppId = "832f8b58443247a2b8b74677198bbc82";
147 String agoraAppCertificate = "36a8fdae33b447e8a928e108a7f36bd9";
148
149 String agoraChannelName = "";
150 String agoraToken = "";
151 String agoraLiveToken = "";
152 String channelName = "astrowayGuruLive";
153 String agoraLiveChannelName = "";
154 String liveChannelName = "astrowayGuruLive";
155 String agoraChatToken = "";
156 String agoraChatUserId = "astrowayGuruLive";
157 String chatChannelName = "astrowayGuruLive";
158 String encodedString = "&";
159 String appName = "";
160 var nativeAndroidPlatform = const MethodChannel('nativeAndroid');
161 SystemFlagNameList systemFlagNameList = SystemFlagNameList();
162
163 //Get app version
164 String getAppVersion() {
165   PackageInfo.fromPlatform().then((PackageInfo packageInfo) {
166     appVersion = packageInfo.version;
167   });
168 }
```

## Change app colors

can change colors of the app from **lib/theme/nativeTheme.dart**.

Replace the old color code with you new color code to set as primary color




```
lib > theme > nativeTheme.dart > Themes > light
21
22 class Themes {
23   static final light = ThemeData.light().copyWith(
24     appBarTheme: AppBarTheme(
25       systemOverlayStyle:
26         const SystemUiOverlayStyle(statusBarColor: Colors.transparent),
27       backgroundColor: Colors().primaryColor,
28       foregroundColor: Colors().blackColor,
29       elevation: 0.5,
30     ), // AppBarTheme
31
32     primaryColor: const Color(0xffffe107),
33
34     textTheme: const TextTheme(
35       displayLarge: TextStyle(color: Colors.black),
36       displayMedium: TextStyle(color: Colors.black),
37       displaySmall: TextStyle(color: Colors.black, fontSize: 14),
38       headlineMedium: TextStyle(color: Colors.black),
39       headlineSmall: TextStyle(color: Colors.black), //
40       titleLarge: TextStyle(color: Colors.black), //
41       bodySmall: TextStyle(color: Colors.black),
42       titleMedium: TextStyle(color: Colors.black, fontSize: 14),
43       titleSmall: TextStyle(color: Colors.black),
44       bodyLarge: TextStyle(color: Colors.black),
45       bodyMedium: TextStyle(color: Colors.black),
46     ), // TextTheme
47     primaryTextTheme: TextTheme(
48       displayLarge: const TextStyle(
49         color: Colors.black,
50         fontWeight: FontWeight.w300,
51         fontSize: 19), // APPBAR // TextStyle
52       displayMedium: const TextStyle(
```

Follow the steps below to configure a **High Priority Notification Channel** in OneSignal for your Android project.

---

### Step 1: Create a OneSignal Account

- Open the following URL:  
 <https://onesignal.com/>
  - Create a new account or log in if you already have one.
- 

### Step 2: Navigate to Android Notification Settings

- Go to your project dashboard in OneSignal.
  - Navigate to:  
**Settings** → **Android Notification Channels**
  - Click on “**Add Group**” to create a new notification channel group.
- 

### Step 3: Create a Notification Channel Group

In the “Add Group” form, set the **Group Name** as:

`high_priority_channel`

- Save the group.
- 

### Step 4: Configure the Notification Channel

- After saving, the newly created group will appear in the list.
- Click on the **group name** (`high_priority_channel`) to open its settings.

Now configure the channel as follows:

| Setting    | Value                      |
|------------|----------------------------|
| Name       | channel_id_17              |
| Importance | Urgent                     |
| Sound      | Custom → Type<br>app_sound |
| Vibration  | Default                    |
| LED        | Default                    |
| Badges     | Enabled                    |
| Lockscreen | Public                     |

The screenshot shows the 'Edit Notification Channel' dialog in the OneSignal mobile app. The background shows the 'Push Settings' menu with options like 'Push Platforms', 'Android Notifications', 'UTM Settings', 'Throttling', 'Frequency Cap', 'Influenced Oper', and 'Outcomes Track'. The dialog box contains the following fields and options:

- Channel ID:** 7420244 (redacted)
- User-visible channel name \***: channel\_id\_17
- User-visible channel description**: (empty text box)
- Importance:** How interruptive the notification is.
  - ☐ Low
  - ☐ Medium
  - ☐ High
  - ☒ Urgent
- Sound:** Play a sound when the notification arrives.
  - ☐ Off
  - ☐ Default
  - ☒ Custom

app\_sound
- Vibration:** Vibrate the device when the notification arrives.
  - ☐ Off
  - ☒ Default
  - ☐ Custom
- Delete** button at the bottom.

**Your Onesignal Push Setting should look like above Screenshot and add channel id in admin panel**

## **Path of the Screens**

- loginScreen:- lib>views>Authentication>login\_screen.dart
- signupScreen:- lib>views>Authentication>signup\_screen.dart
- successRegistration:- lib>views>Authentication>success\_registration\_screen.dart
- login\_otp\_screen:- lib>views>Authentication>OtpScreen> login\_otp\_screen.dart
- signup\_otp\_screen:- lib>views>Authentication>OtpScreen> signup\_otp\_screen.dart
- splashScreen:- lib>views>splash>splash\_screen.dart
- AddAssistanScreen:- lib>views>HomeScreen>Assistant> add\_or\_edit\_assistant\_screen.dart
- assistantChatRequestScreen:- lib>views>HomeScreen> Assistant> assistant\_chat\_request\_screen.dart
- assistantChatScreen:- lib>views>HomeScreen> Assistant> assistant\_chat\_screen.dart
- assistantScreen:- lib>views>HomeScreen> Assistant> assistant\_screen.dart
- AcceptCallScreen:- lib>views>HomeScreen> call> accept\_call\_screen.dart
- callAvailabilityScreen:- lib>views>HomeScreen> call> callAvailabilityScreen.dart
- app\_review\_Screen:-lib/views/HomeScreen/Drawer/AppReview/app\_review\_screen.dart
- privacy\_policy\_screen:-lib/views/HomeScreen/Drawer/Setting/privacy\_policy\_screen.dart
- terms\_and\_condition\_screen:-lib/views/HomeScreen/Drawer/Setting/term\_and\_condition\_screen.dart
- setting\_list\_screen:-lib/views/HomeScreen/Drawer/Setting/setting\_list\_screen.dart
- add\_amout\_screen:-lib/views/HomeScreen/Drawer/Wallet/add\_amount\_screen.dart
- wallet\_screen:-lib/views/HomeScreen/Drawer/Wallet/Wallet\_screen.dart
- customer\_review\_Screen:- lib/views/HomeScreen/Drawer/customer\_review\_screen.dart
- drawer\_screen:- lib/views/HomeScreen/Drawer/drawer\_screen.dart
- astrology\_blog\_detil\_screen:-lib/views/HomeScreen/FloatingButton/AstroBlog/astrology\_blog\_detil\_screen.dart
- astrology\_blog\_screen:-lib/views/HomeScreen/FloatingButton/AstroBlog/astrology\_blog\_screen.dart
- dailyHoroScopeDetailScreen:-lib/views/HomeScreen/FloatingButton/DailyHoroscope/dailyHoroScopeDetailScreen.dart
- dailyHoroscopeVedic:- lib/views/HomeScreen/FloatingButton/DailyHoroscope/dailyHoroscopeVedic.dart
- chatAvailabilityScreen:-lib/views/HomeScreen/FloatingButton/FreeKundli/Tabs/report Tabs/chatAvailabilityScreen.dart
- basicKundliScreen:-lib/views/HomeScreen/FloatingButton/FreeKundli/Tabs/basicKundliScreen.dart
- Pdf\_Screen:-lib/views/HomeScreen/FloatingButton/FreeKundli/Tabs/Pdf\_Screen.dart

- createNewKundli:-lib/views/HomeScreen/FloatingButton/FreeKundli/createNewKundli.dart
- editKundliScreen:-lib/views/HomeScreen/FloatingButton/FreeKundli/editKundliScreen.dart
- kundliDetailsScreen:-lib/views/HomeScreen/FloatingButton/FreeKundli/kundliDetailsScreen.dart
- kundliScreen:-lib/views/HomeScreen/FloatingButton/FreeKundli/kundliScreen.dart
- New\_matching\_screen:-  
lib/views/HomeScreen/FloatingButton/KundliMatching/Tabs/new\_matching\_screen.dart
- open\_kundli\_screen:-lib/views/HomeScreen/FloatingButton/KundliMatching/Tabs/open\_kundli\_screen.dart
- kundli\_matching\_screen:-lib/views/HomeScreen/FloatingButton/KundliMatching/kundli\_matching\_screen.dart
- place\_of\_birth\_screen:-lib/views/HomeScreen/FloatingButton/KundliMatching/place\_of\_birth\_screen.dart
- north\_kundli\_match\_result\_screen:-lib/views/HomeScreen/FloatingButton/KundliMatching/north\_kundli\_match\_result\_screen.dart
- south\_kundali\_matching\_screen:-lib/views/HomeScreen/FloatingButton/KundliMatching/south\_kundali\_matching\_screen.dart
- onetooneVideo:-lib/views/HomeScreen/live/onetoone\_video/onetooneVideo.dart
- live\_screen:- lib/views/HomeScreen/live/live\_screen.dart
- assignment\_detail\_screen:-  
lib/views/HomeScreen/Profile/ProfileDetailScreen/assignment\_detail\_screen.dart
- availability\_Detail\_screen:-  
lib/views/HomeScreen/Profile/ProfileDetailScreen/availability\_Detail\_screed.dart
- other\_detail\_screen:-  
lib/views/HomeScreen/Profile/ProfileDetailScreen/other\_detail\_screen.dart
- personal\_detail\_screen :-  
lib/views/HomeScreen/Profile/ProfileDetailScreen/personal\_detail\_screen.dart
- skill\_detail\_screen :-  
lib/views/HomeScreen/Profile/ProfileDetailScreen/skill\_detail\_screen.dart
- edit\_profile\_screen:- lib/views/HomeScreen/Profile/edit\_profile\_screen.dart
- follower\_list\_screen:- lib/views/HomeScreen/Profile/follower\_list\_screen.dart
- Profile\_screen:- lib/views/HomeScreen/Profile/profile\_screen.dart
- report\_detail\_screen :-  
lib/views/HomeScreen/Report\_Module/report\_detail\_screen.dart
- report\_history\_details\_screen: -  
lib/views/HomeScreen/Report\_Module/report\_history\_details\_screen.dart
- Report\_history\_list\_screen:-  
lib/views/HomeScreen/Report\_Module/report\_history\_list\_screen.dart
- report\_request\_screen:-  
lib/views/HomeScreen/Report\_Module/report\_request\_screen.dart
- view\_report\_pdf\_screed:-  
lib/views/HomeScreen/Report\_Module/view\_report\_pdf\_screed.dart
- call\_detail\_screen:- lib/views/HomeScreen/call\_detail\_screen.dart
- chat\_screen:- lib/views/HomeScreen/chat\_screen.dart
- home\_screen:- lib/views/HomeScreen/home\_screen.dart



- notification\_screen:- lib/views/HomeScreen/notification\_screen.dart
- search\_blog\_screen:- lib/views/HomeScreen/search\_blog\_screen.dart

## Contact Support

Delighted to serve and support you! Contact our support team. We're available Mon-Fri, 9:00 am - 6:00 pm IST (GMT +5.30),

India - Asia. Expect prompt responses within 24 hours via comments, forum, or email.

---

## Reach Us

Unlock the potential of diploy, your go-to hub for top-notch Flutter app development and web applications. Reach out to us today for quality solutions and exciting freelance opportunities.

Visit us at [diploy.in](https://diploy.in) or drop an email to [start@diploy.in](mailto:start@diploy.in)

Thank you for considering us!