

Astroway User Application

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.

Setup Astroway User Application

1:- Setup Flutter

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

Prefer video tutorials? Check out the playlist:  What is Flutter?

Flutter 3.35.2

Dart version:- 3.9.0

Gradle:-8.5.2,

jdk version:- jdk17

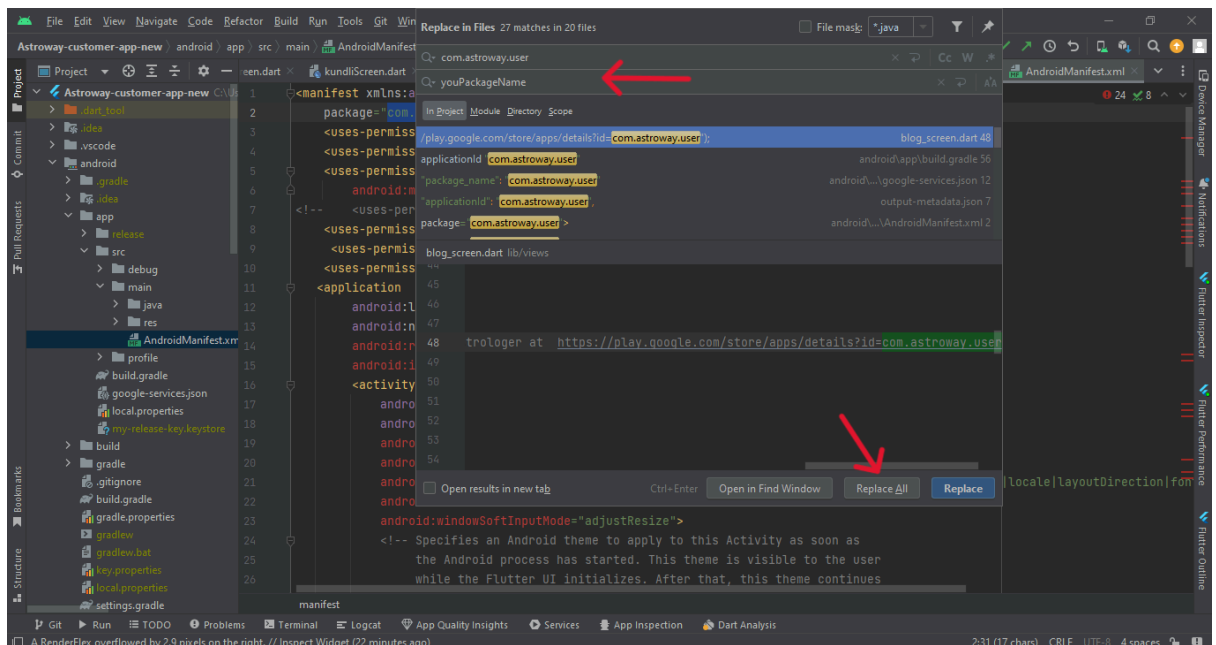
Steps to run this project successfully:

1. Begin by extracting the downloaded code, resulting in the creation of the **Astroway-User-App** - Flutter Code zip folder.
2. Next, extract the contents of the **Astroway-User-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.
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.

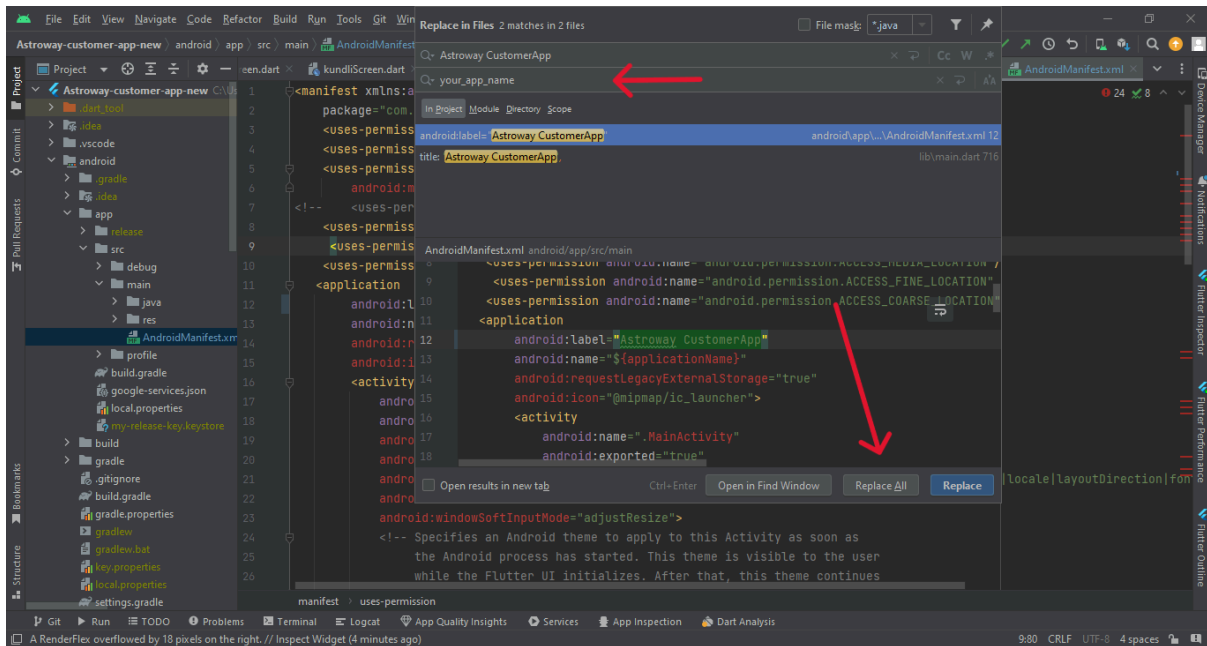
2:- Change the Package Name

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.astroway.user` and replace it with your package name in all places.



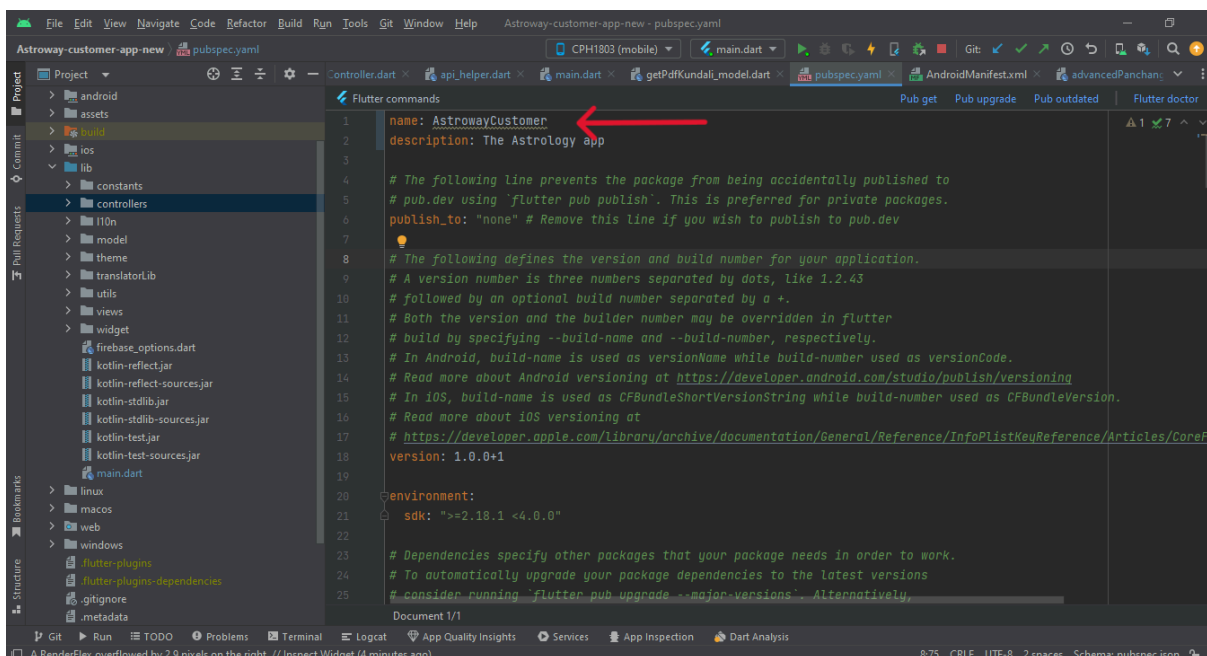
3:- Change the Application Name

1. Similarly you can change the app name, you can press **CTRL+SHIFT+R** to open Replace in Files tool.
then search our app name `Astroway CustomerApp` and replace it with your app name in all places.



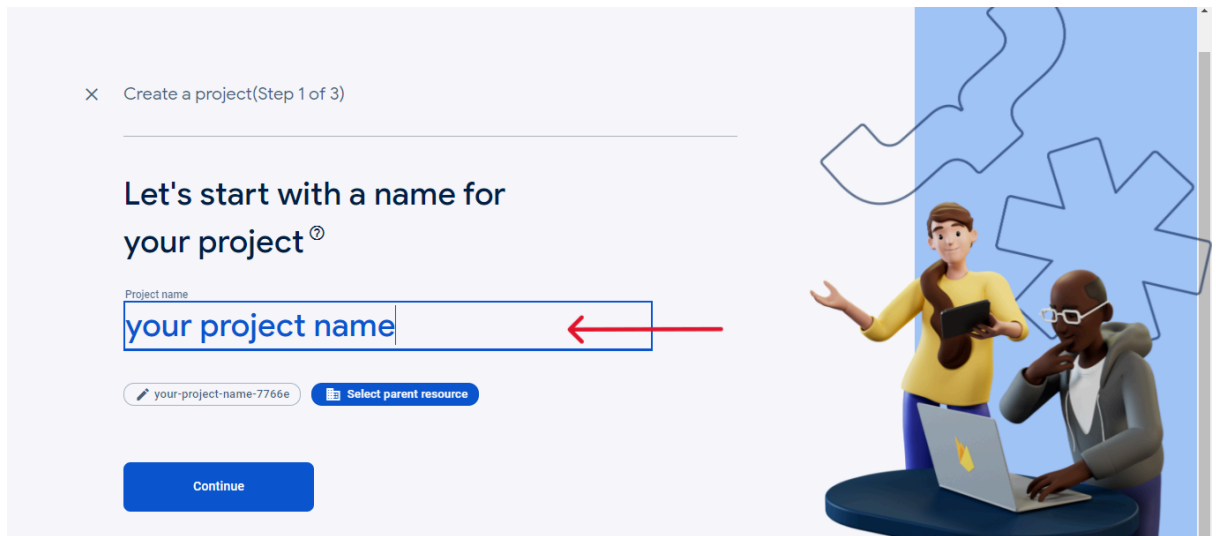
2. You have to change application name in you pubspec.yaml file

Note:- After changing the application name from pubspec.yaml file you have to import package name again.

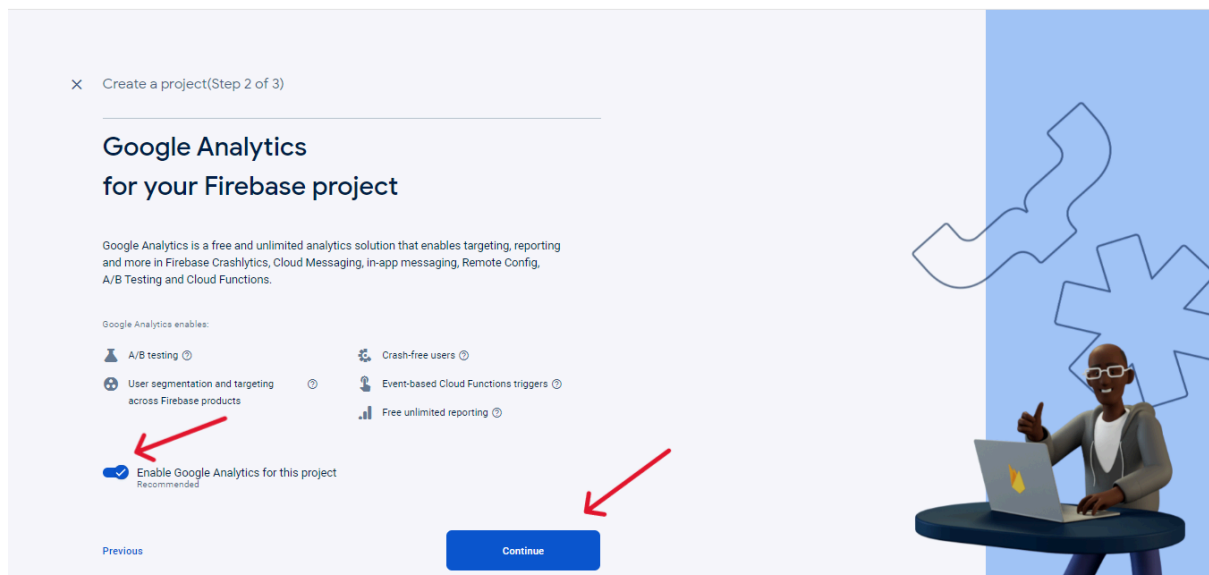


4:- Integrate firebase with your Project

1. First, Create a new Firebase Project and enter you project name



2. Enable google analytics for your project



3. Read & Accept the google analytics terms and conditions

Create a project(Step 3 of 3)

Analytics location

United States

Google Analytics is a business tool. Use it exclusively for purposes related to your trade, business, craft or profession.

Data-sharing settings and Google Analytics terms

☒ Use the default settings for sharing Google Analytics data. [Learn more](#)

☒ Share your Analytics data with Google to improve Google Products and Services

☒ Share your Analytics data with Google to enable Benchmarking

☒ Share your Analytics data with Google to enable Technical Support

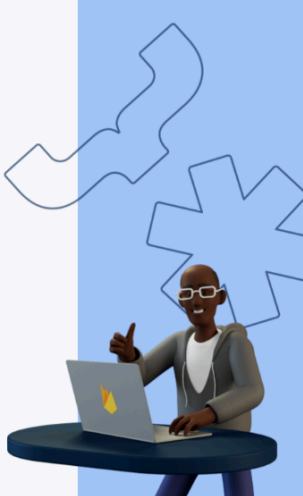
☒ Share your Analytics data with Google Account Specialists

☒ I accept the [Google Analytics terms](#)

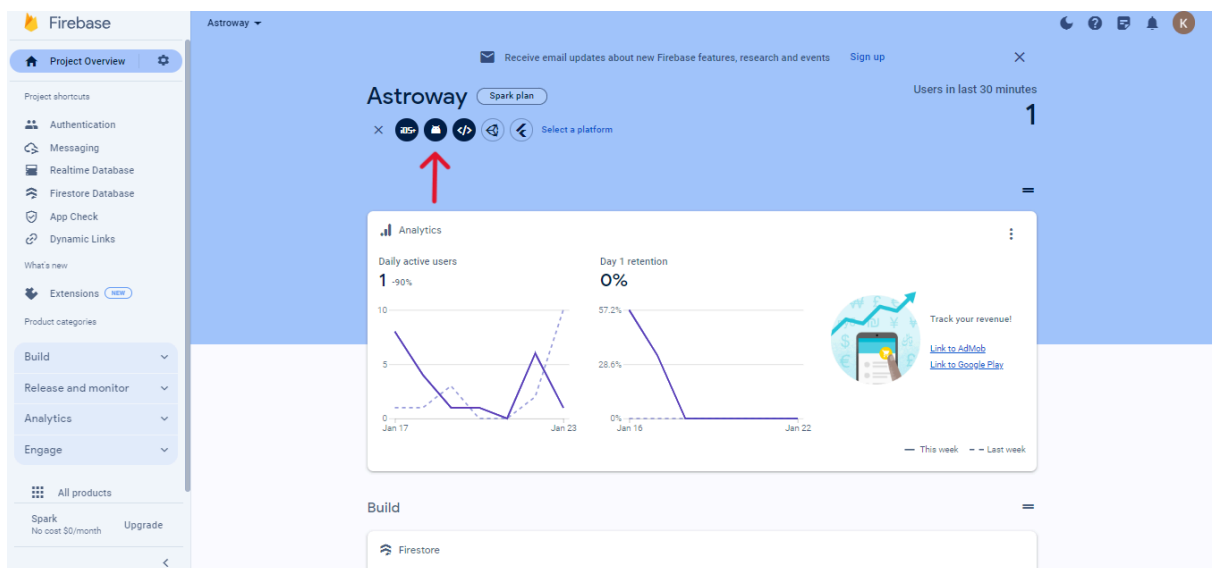
Upon project creation, a new Google Analytics property will be created and linked to your Firebase project. This link will enable data flow between the products. Data exported from your Google Analytics property into Firebase is subject to the Firebase terms of service, while Firebase data imported into Google Analytics is subject to the Google Analytics terms of service. [Learn more](#)

Previous

Create project



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



Firebase

Astroway

Receive email updates about new Firebase features, research and events Sign up

Project Overview

Analytics

Daily active users

1 -90%

10 5 0 Jan 17 Jan 23

Day 1 retention

0%

57.2% 28.6% 0% Jan 16 Jan 22

Track your revenue!

[Link to AdMob](#)

[Link to Google Play](#)

Build

Firestore

- 2 Download and then add config file
- 3 Add Firebase SDK
- 4 Next steps



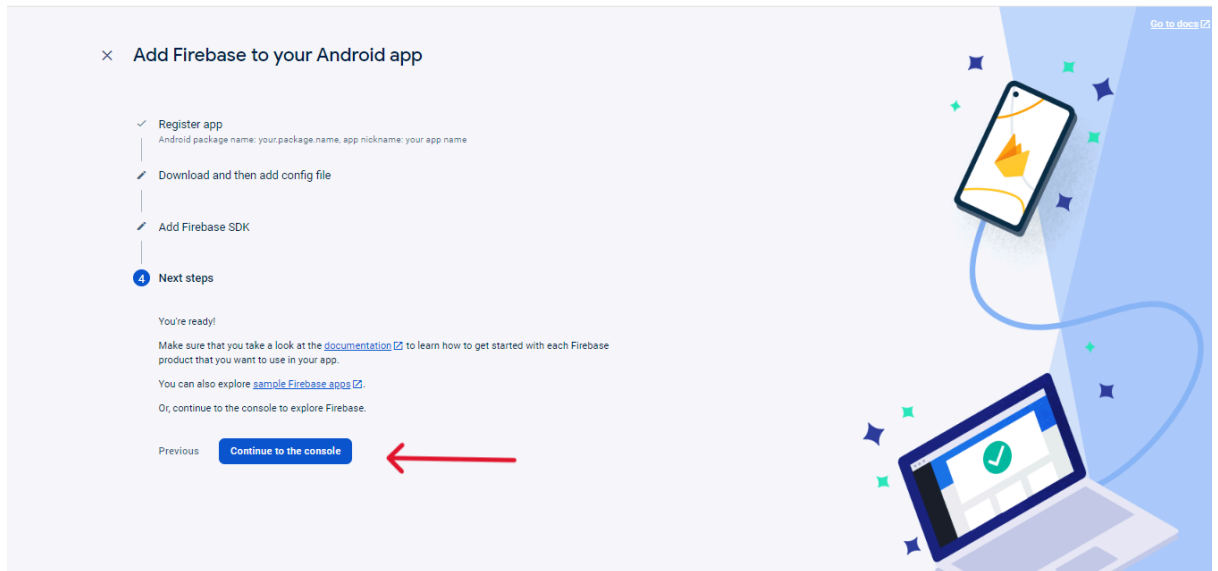
3 Add Firebase SDK

4 Next steps

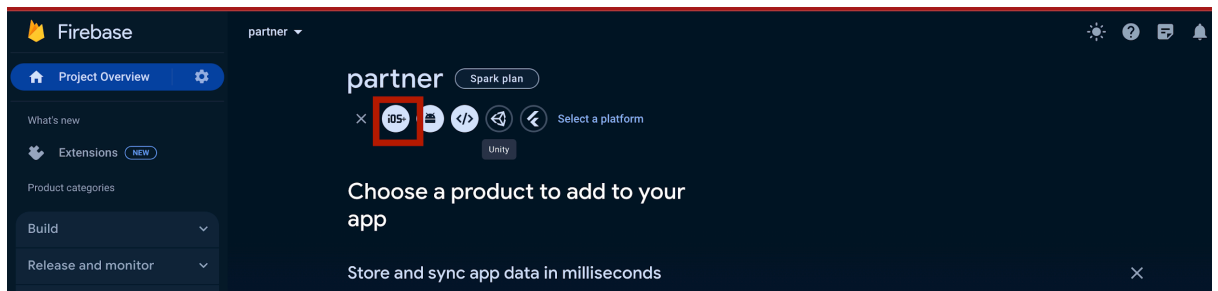


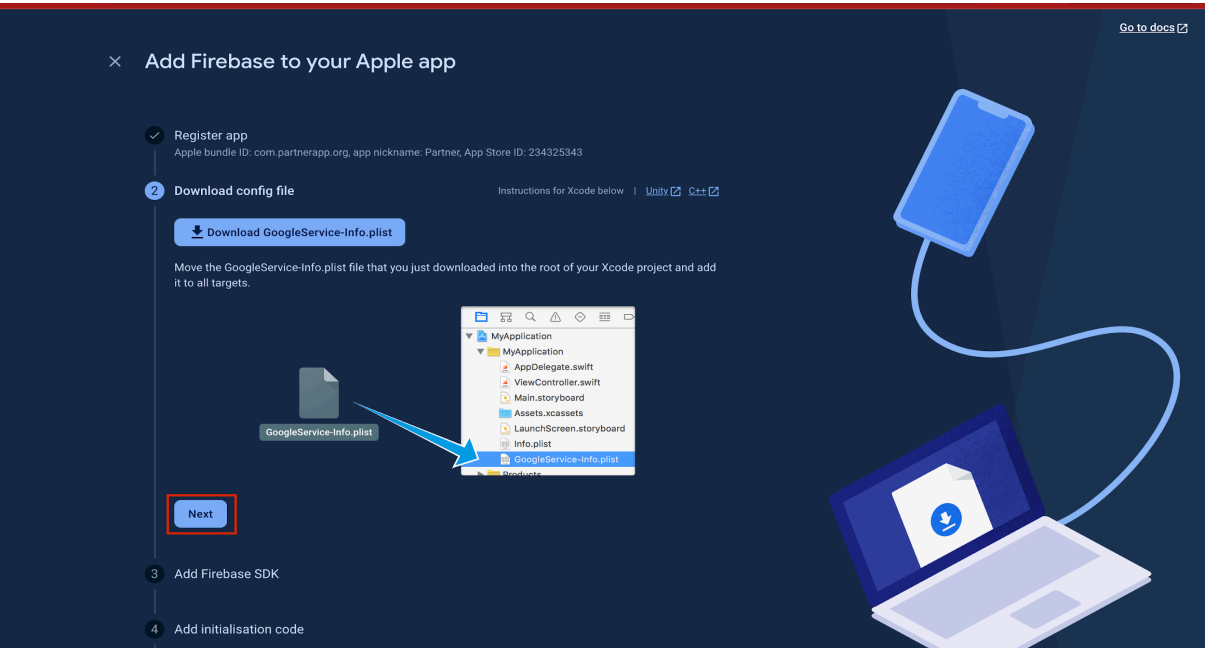
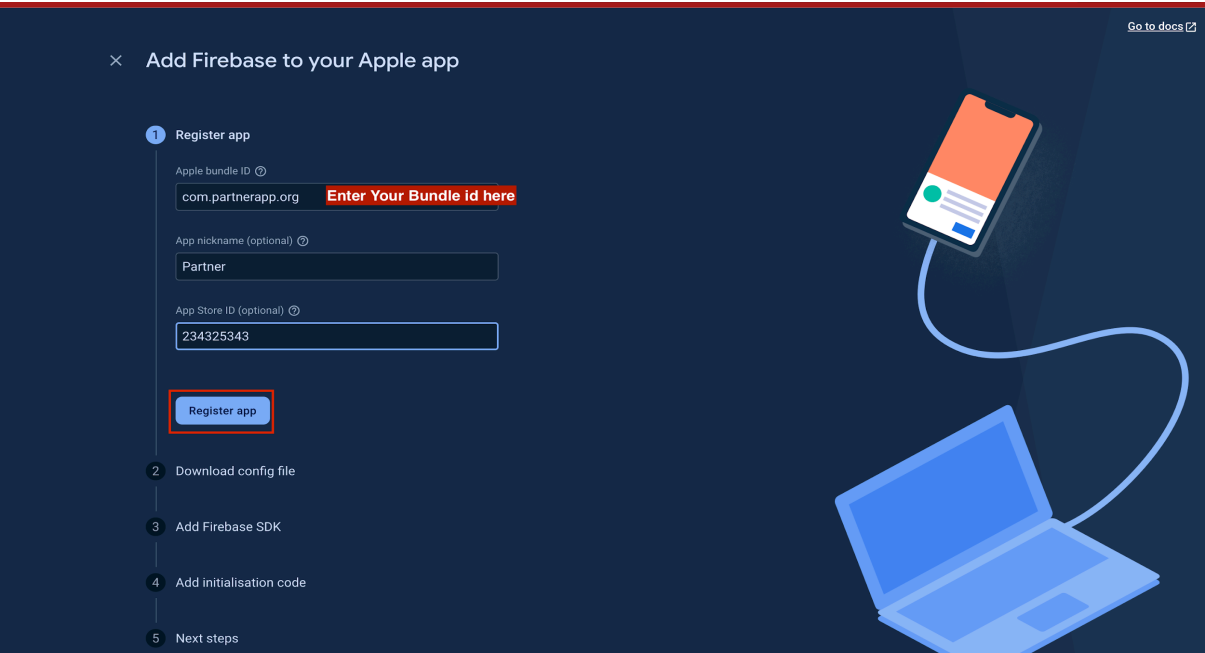
4 Next steps





5. Now it's time for connecting IOS to your Firebase Project.and follow same steps for IOS





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: [UIApplicationLaunchOptionsKey: Any]? = nil)
        -> Bool {
        FirebaseApp.configure()
        return true
    }
}


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

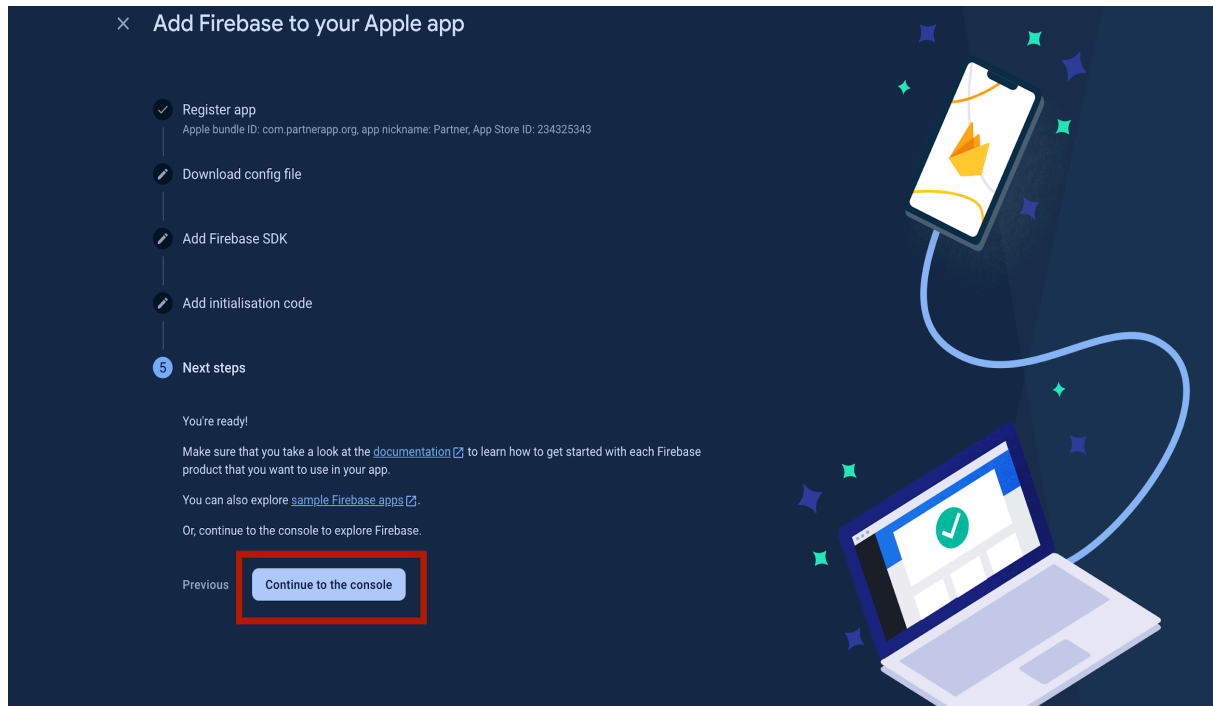
    var body: some Scene {
        WindowGroup {
            NavigationView {
                ContentView()
            }
        }
    }
}
```

Previous

Next

5 Next steps



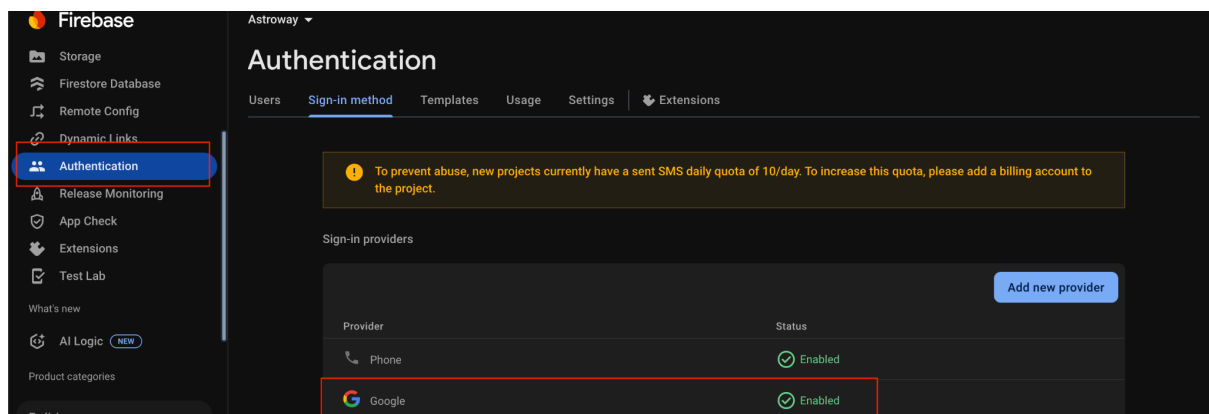
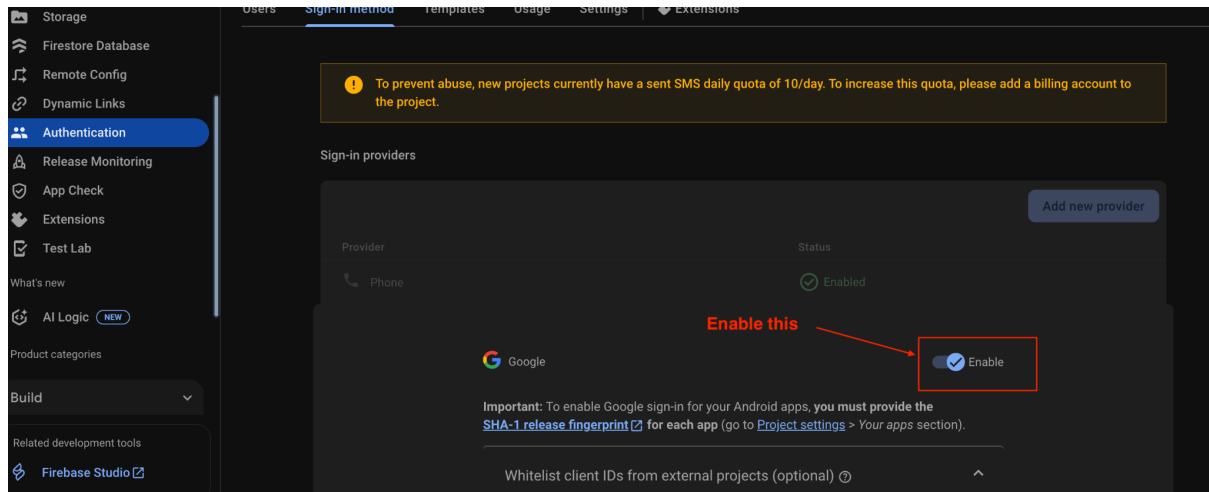


Congrats! You have successfully connected your android & IOS app to your firebase project.

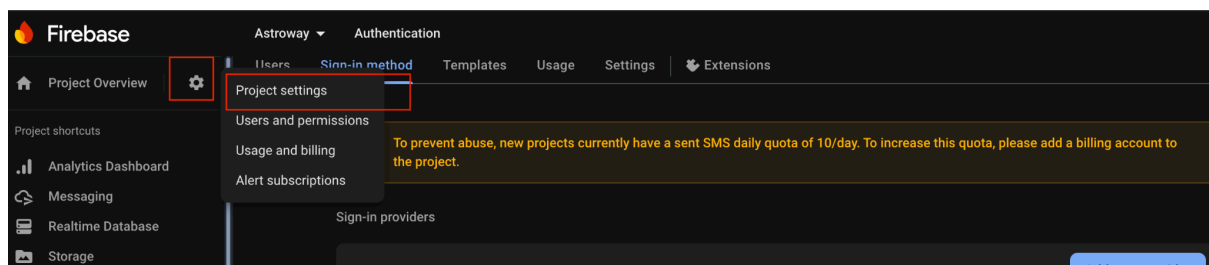
Congrats! You have successfully connected your android & IOS app to your firebase project.

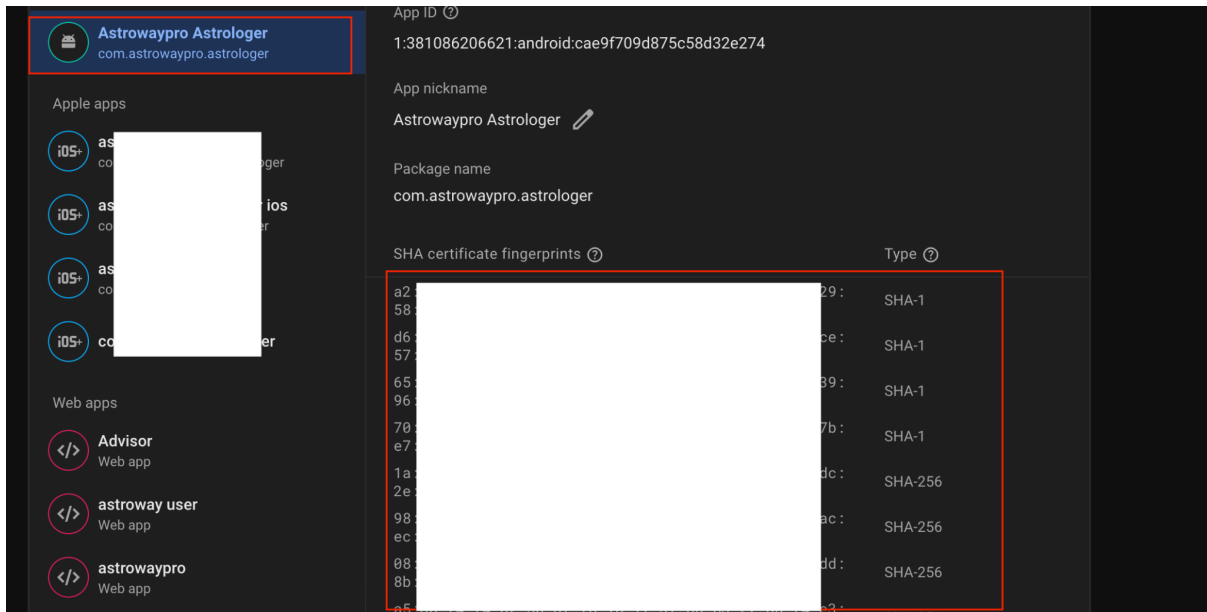
5:- Enable Firebase Authentication

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

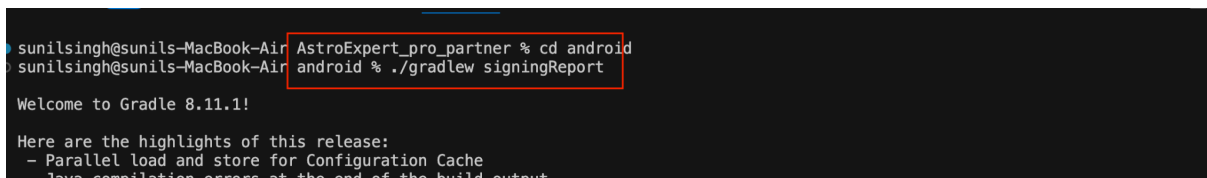


And get SHA-1 and SHA-256 keys [debug and release both] and add in below firebase setting





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



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

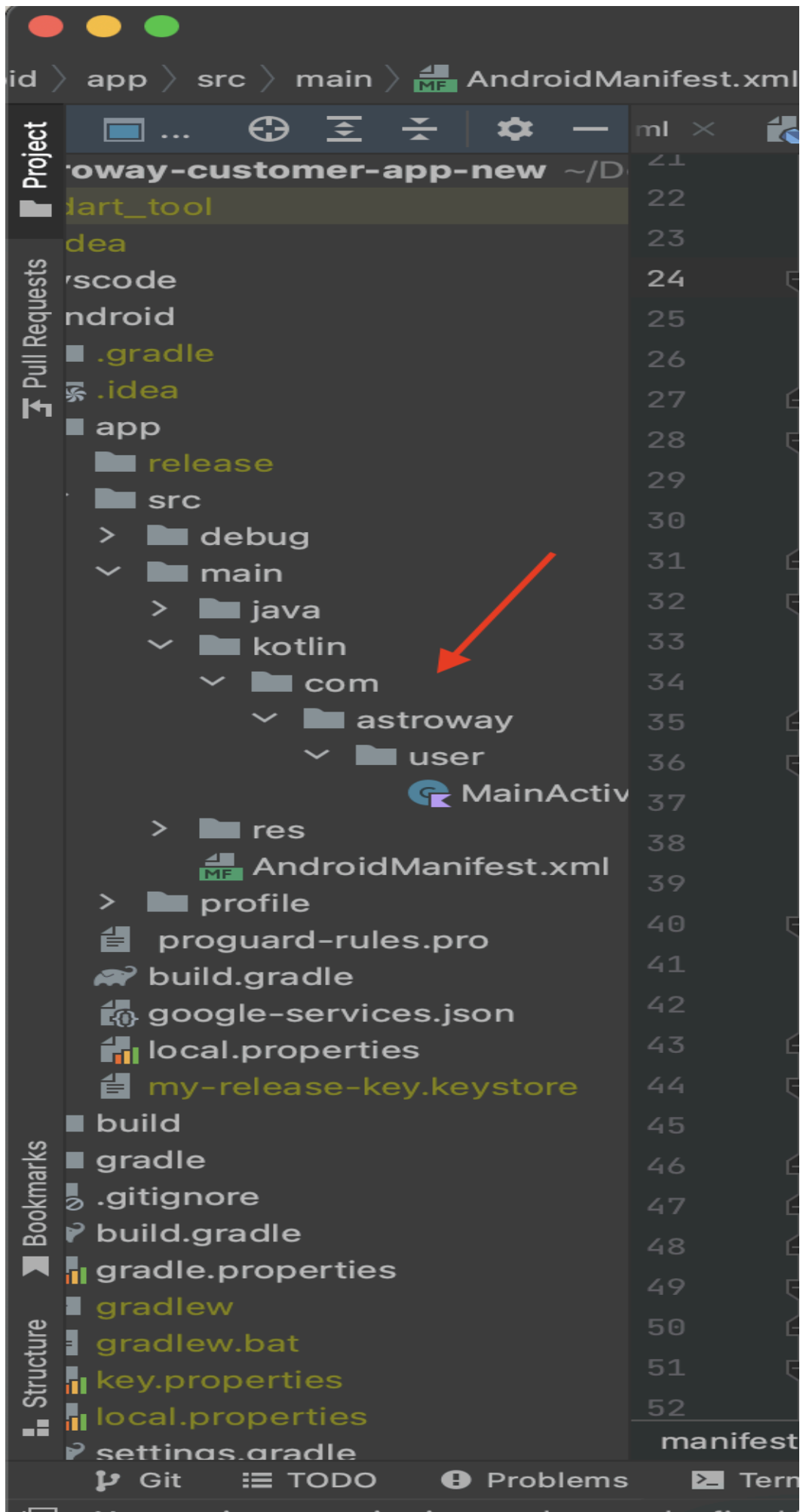
IMPORTANT - you need to change your folders name inside kotlin folder goto **android/app/src/main/kotlin/com/astroway/astrologer_app/MainActivity.kt**

For example:-**com.astroway.user**

Then first folder name:-com

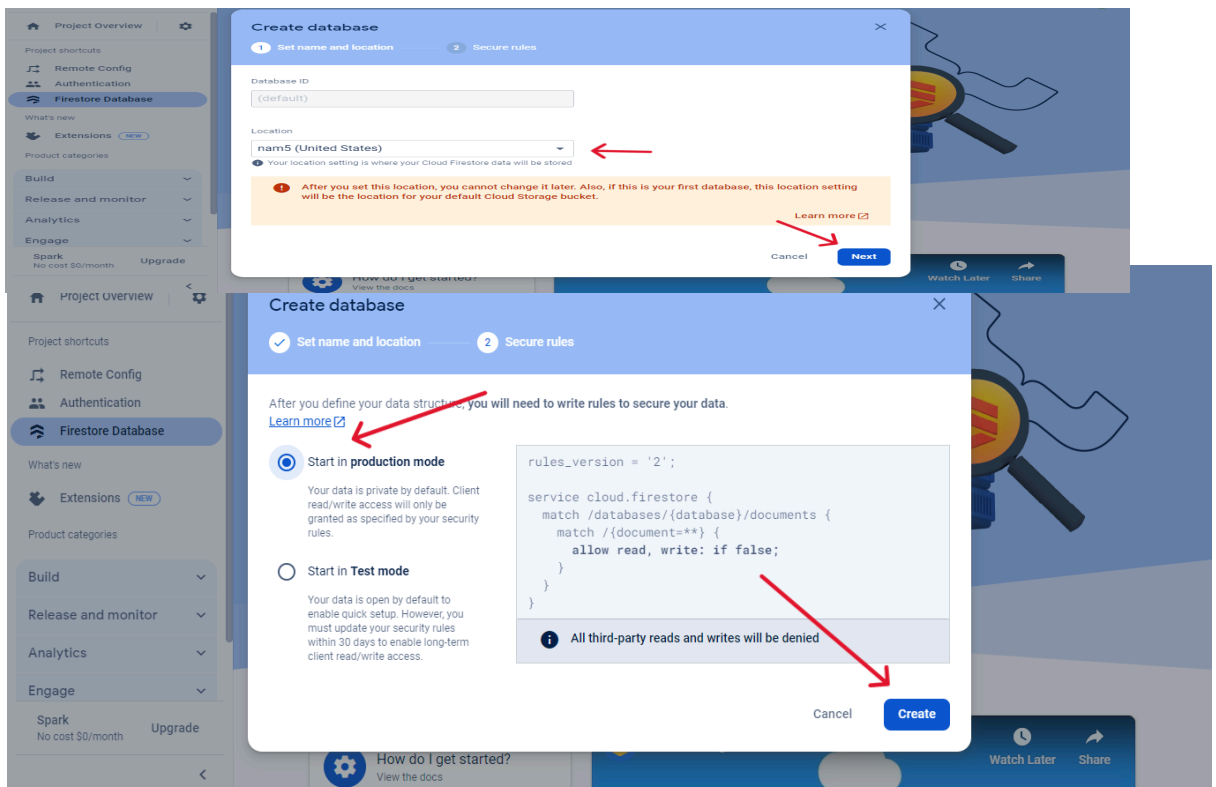
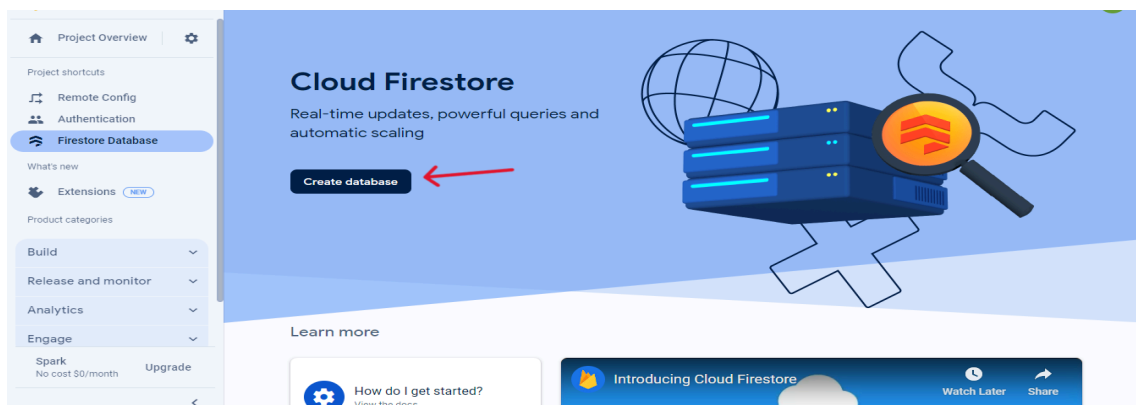
Second folder name:-astroway

Third folder name:- user

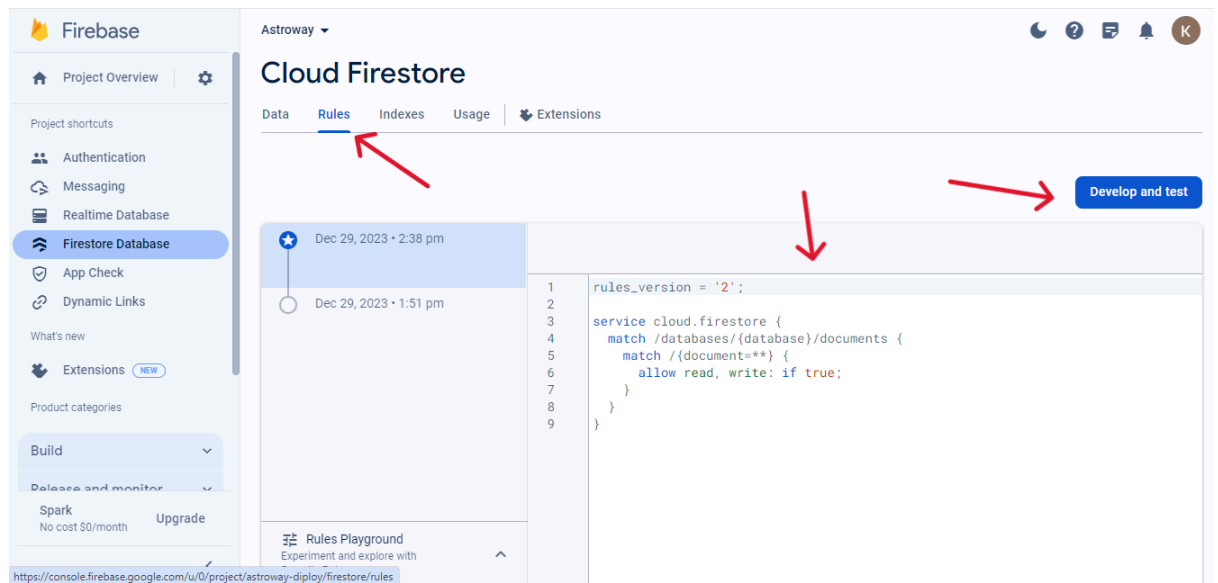


6:- Enable Cloud Firestore and storage

1. Cloud Firestore is required for Chat.
Follow these steps shown in images below.



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



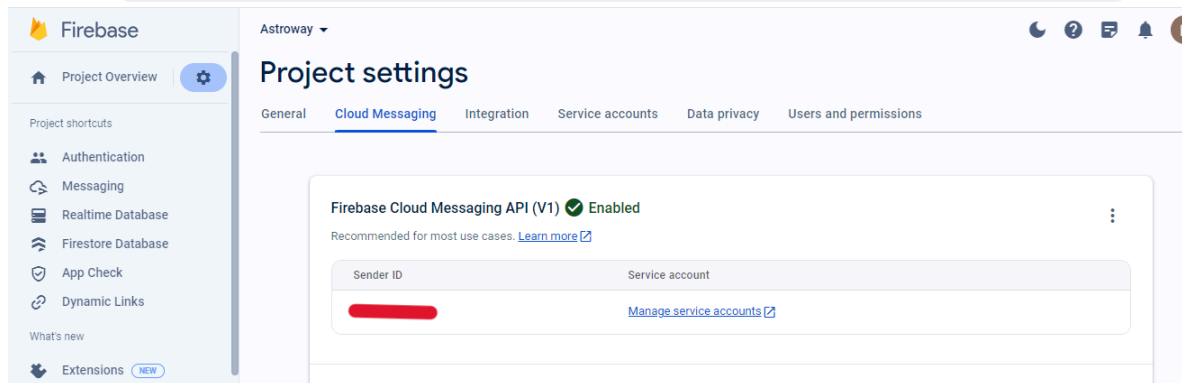
Note:- with the same process you need to enable Storage

Rules for Storage:-

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

8-A :- Enable Notification for App

1. Access your Firebase project and locate "Project Settings" > "Cloud Messaging" as shown in the image.

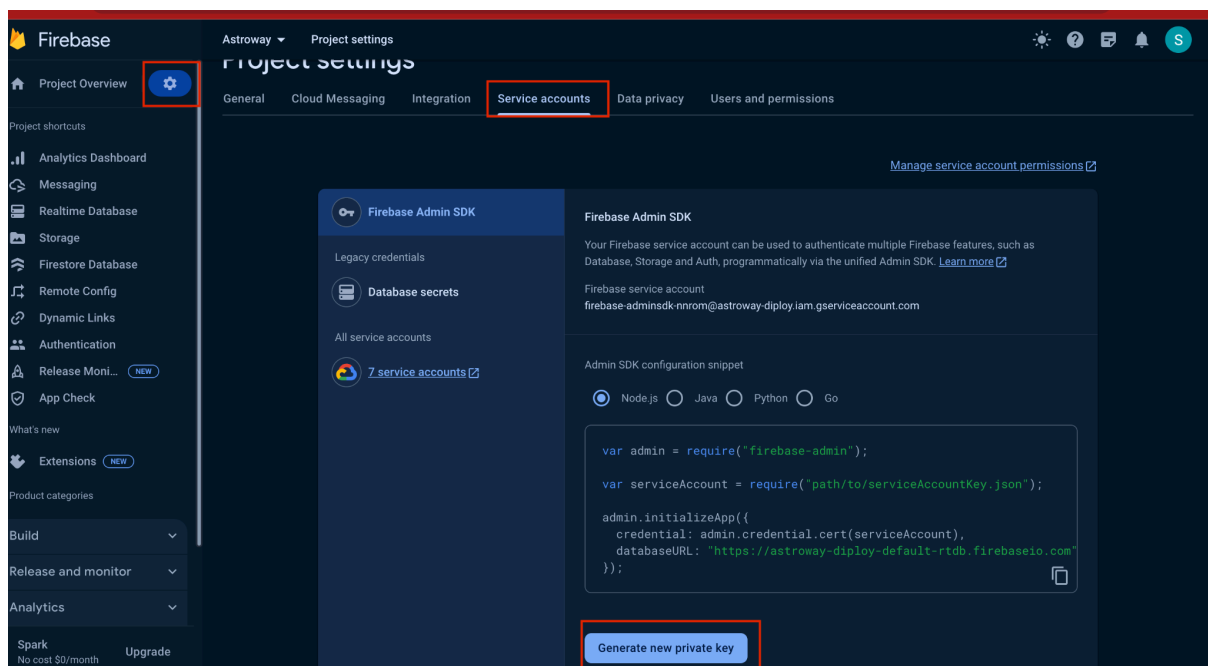


Then, enable Cloud Messaging in Google Cloud.

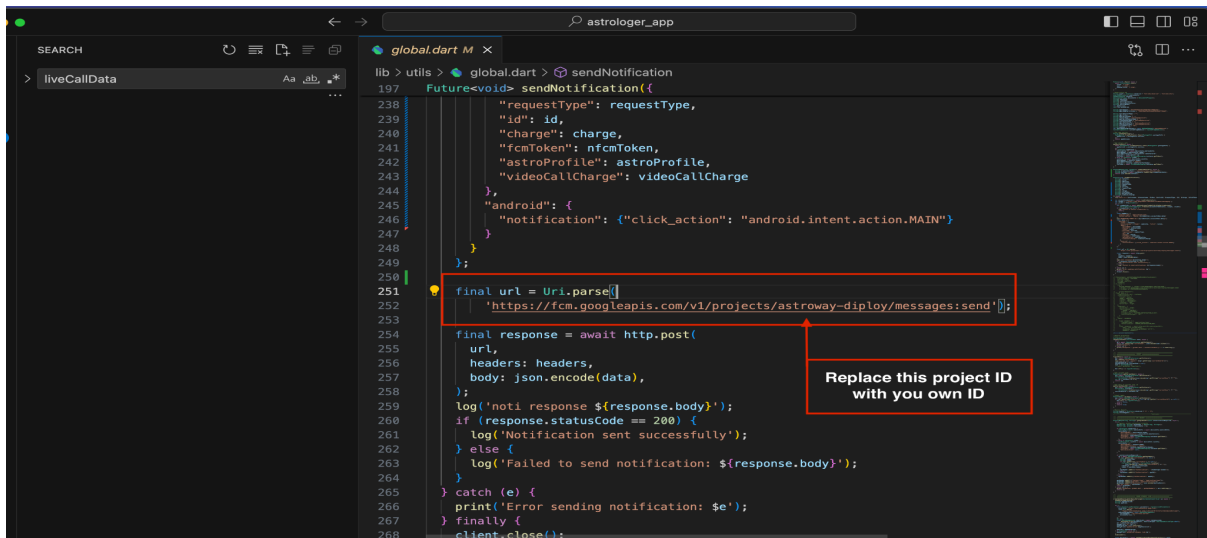
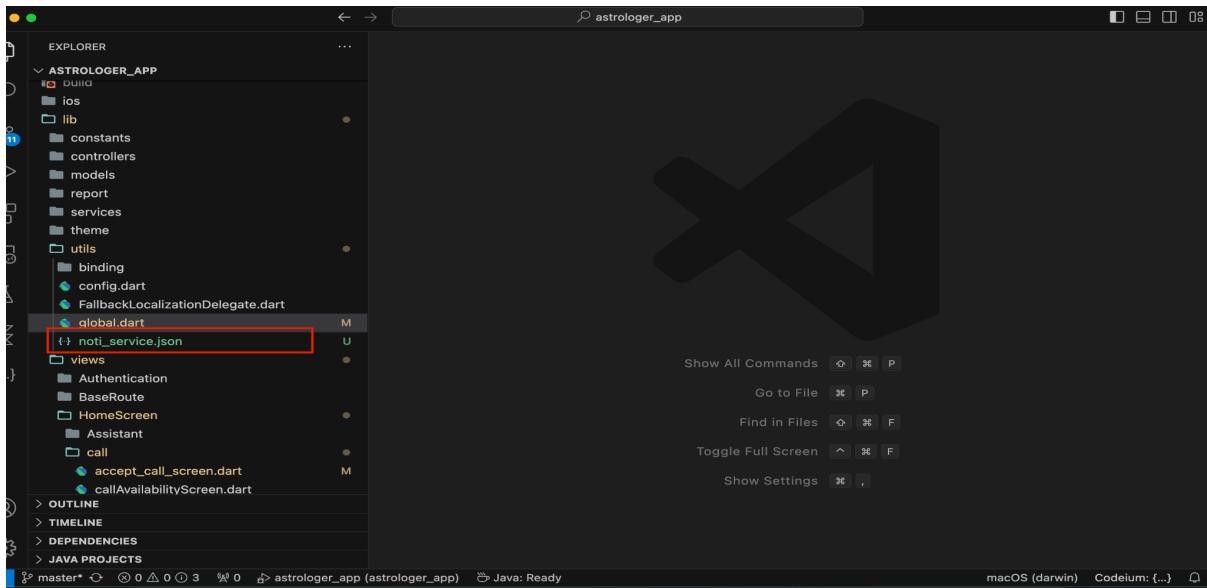
We have to add server key in admin panel

Access your Firebase project and locate "Project Settings" > "Service account TAB"

and click on **Generate new Private key**

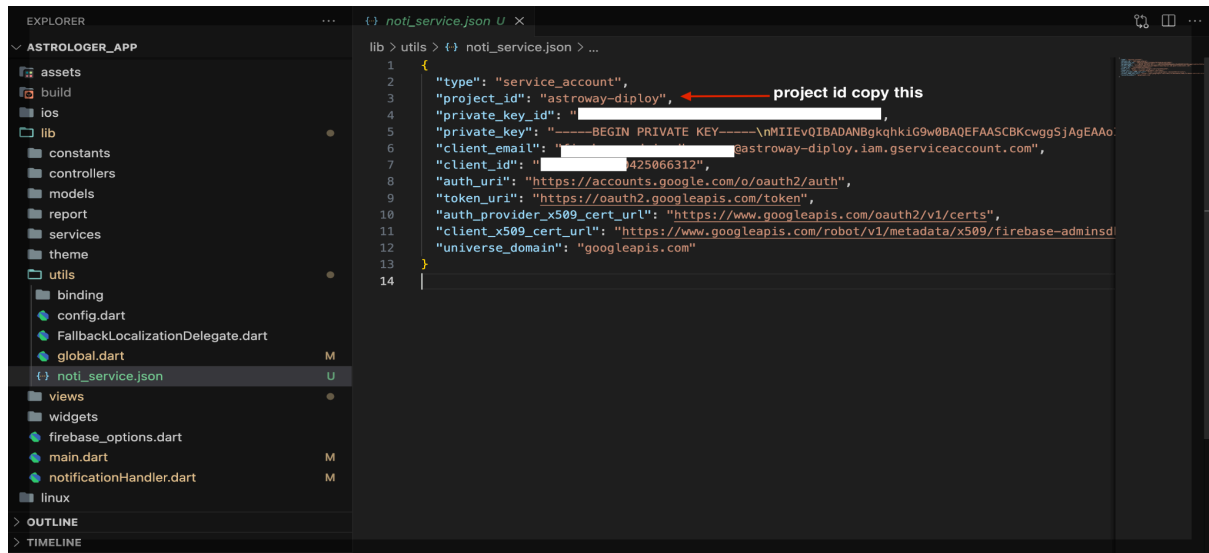


1. Download the file.json and save it into **lib/Utils/noti_service.json**



2. Now go to `lib/utils/global.dart`

And replace the projectID from your project id which is in the .json file you downloaded while generating private key



8-B :- 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

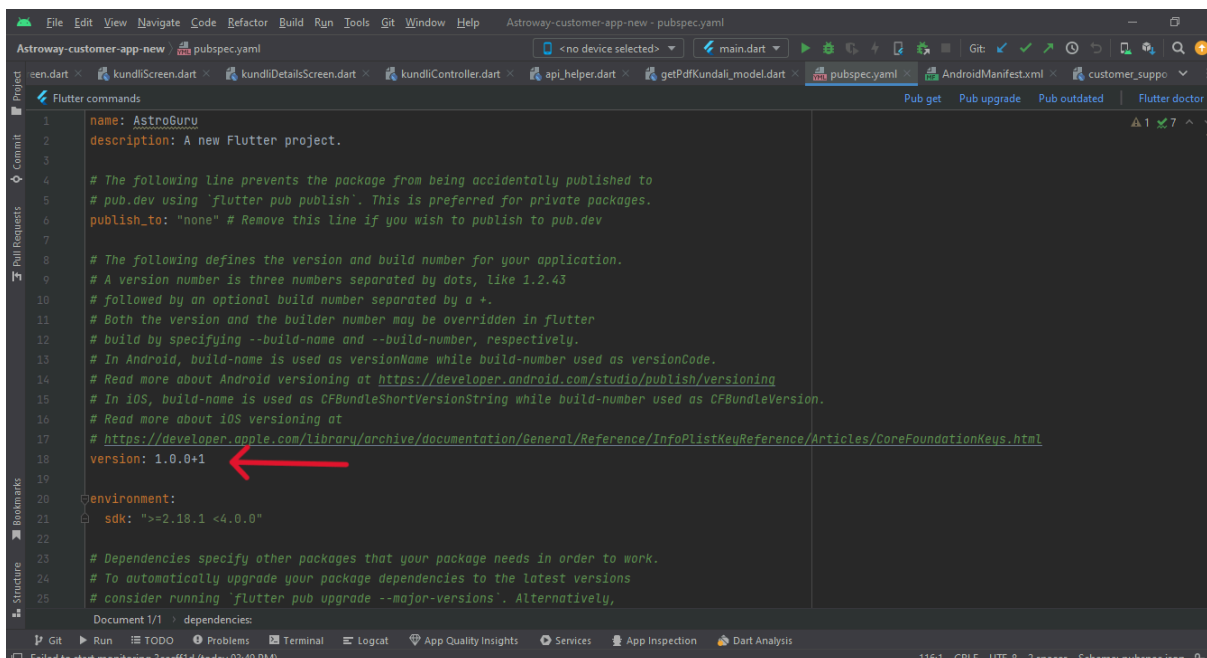
Astroway User 

Package name

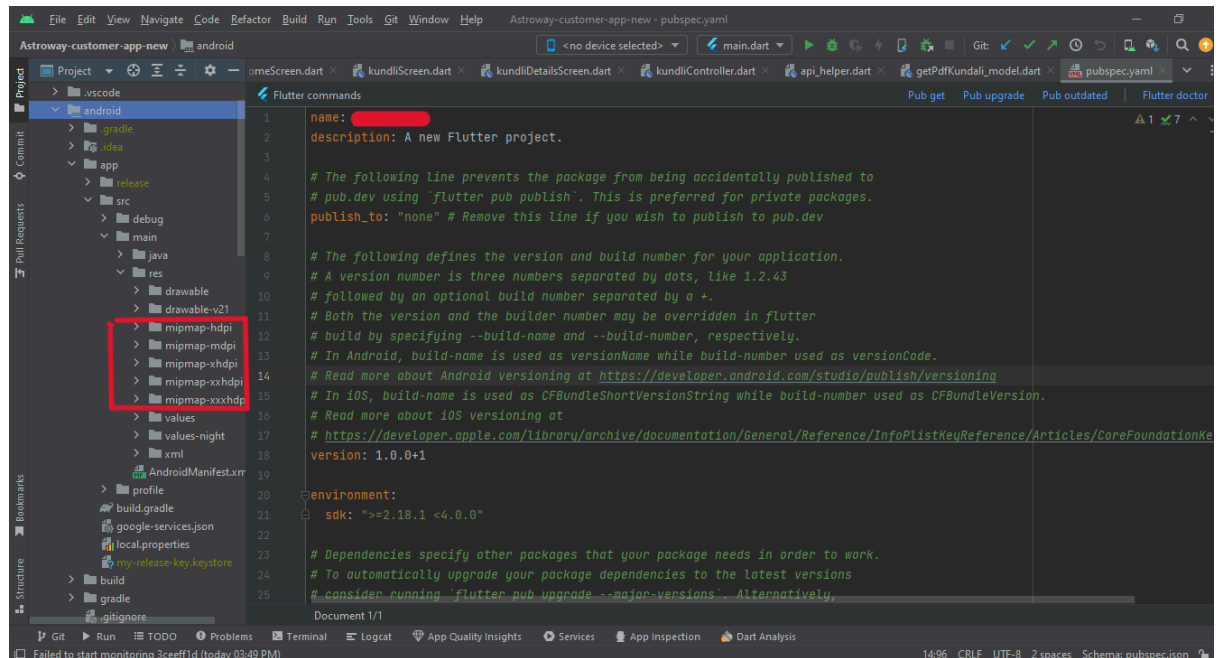
com.astrowaydiploy.user

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

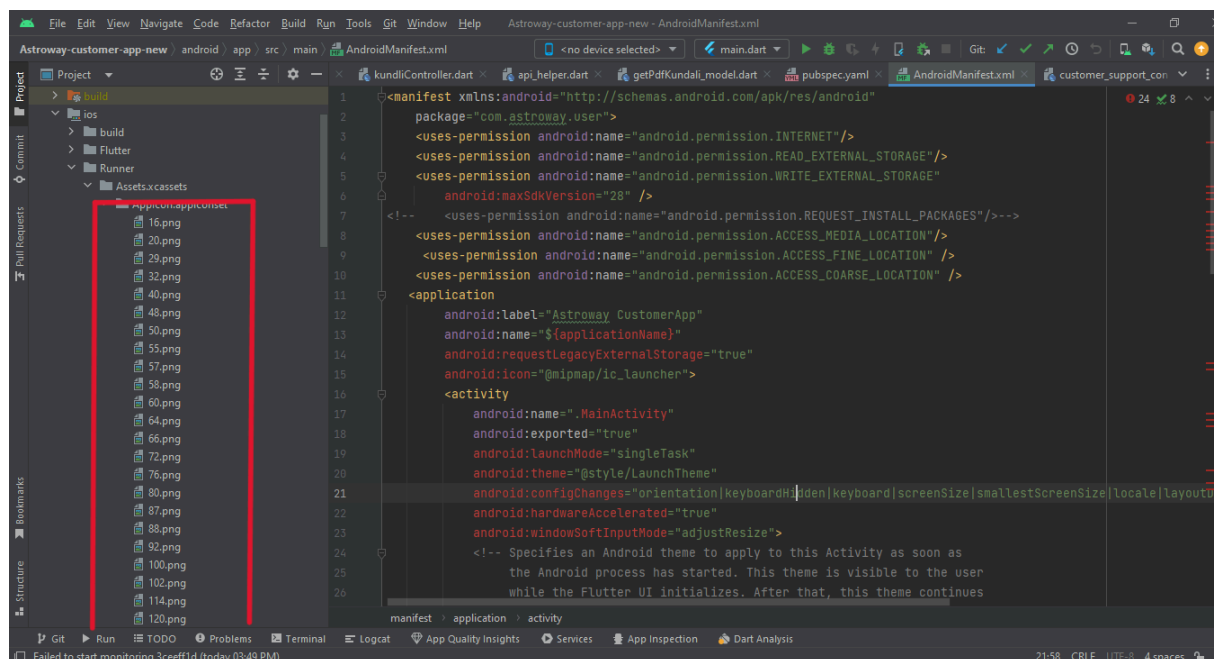
Path:- project>android>app>google-services.json file



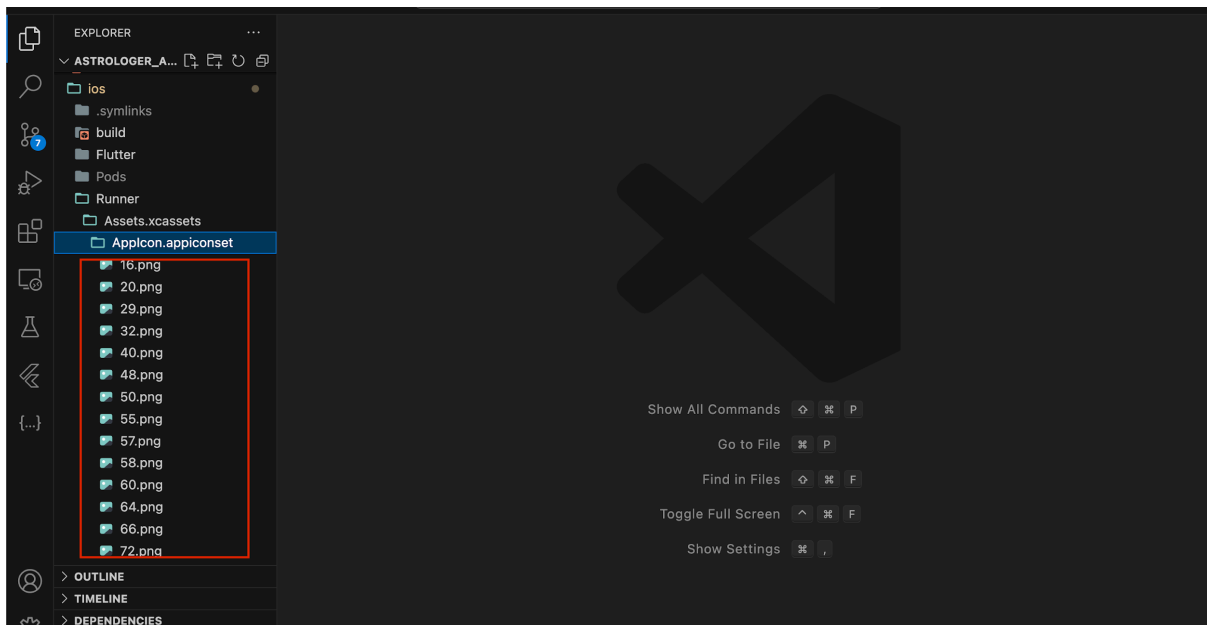
2. Change App Logo
3. For Android, open android > app > src > main > res and add here your logo according to device screen size



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

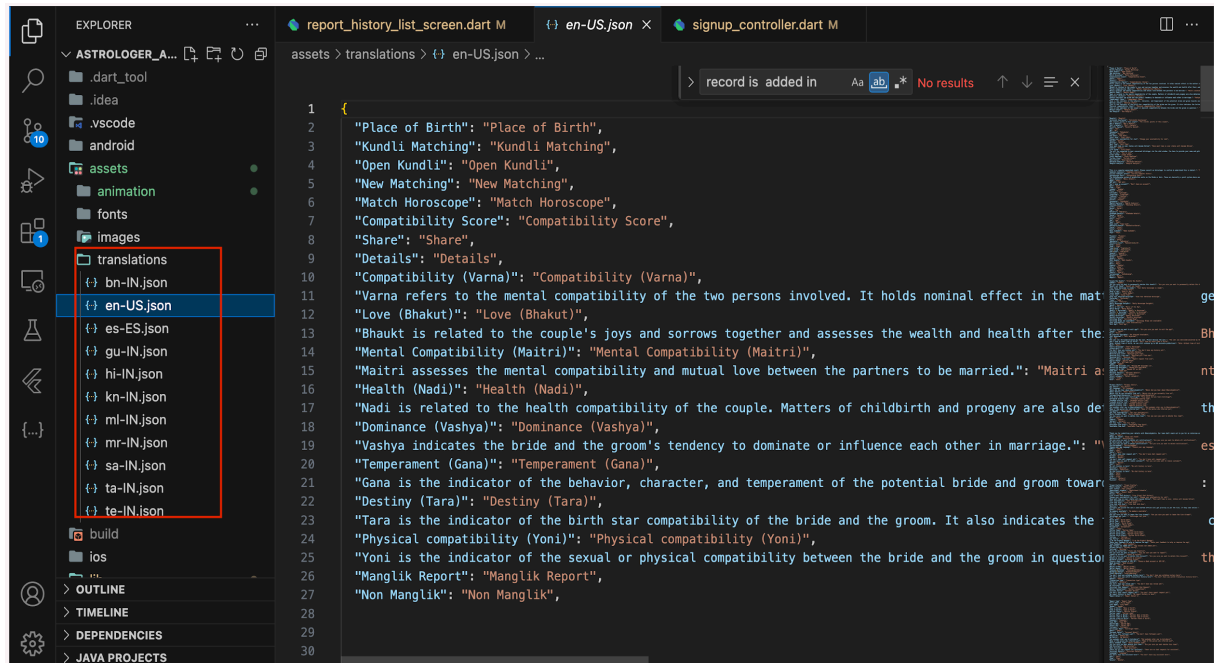
The App's default language is English, but you can easily modify it. To do so, navigate to the `lib/main.dart` file and locate the section displaying the language codes, as illustrated in the image below.

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

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

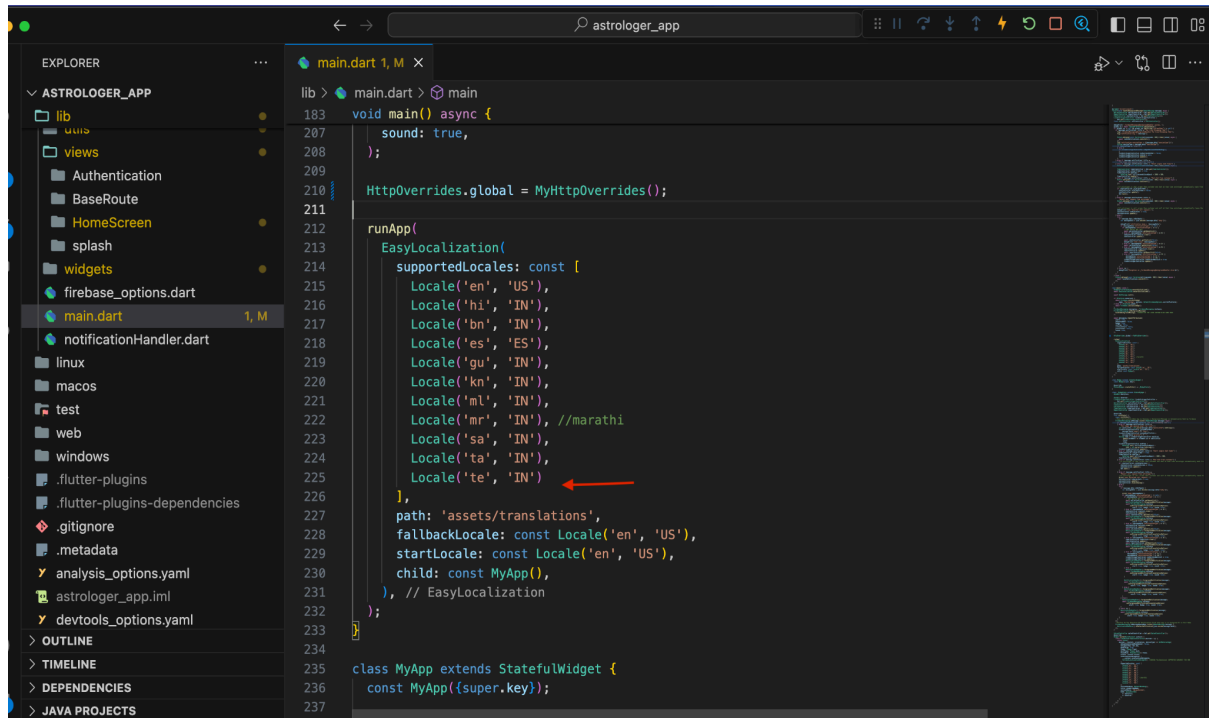
Just ensure that the language code you select is available in the aforementioned link for proper functionality.

```
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



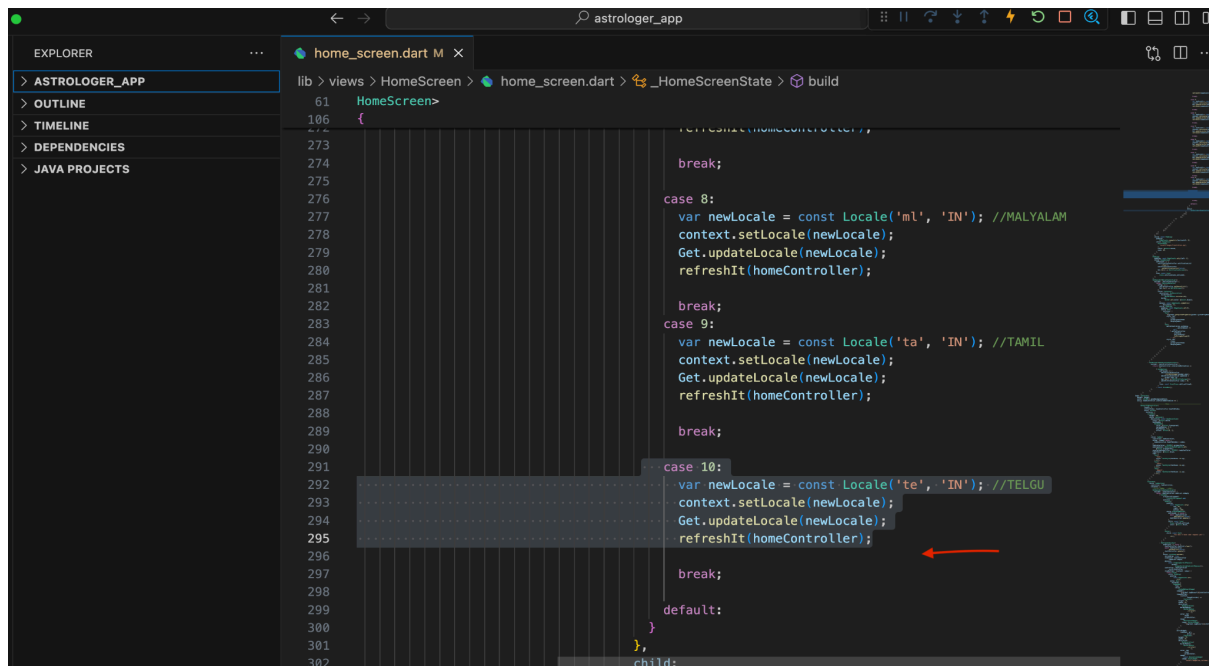
```
lib > main.dart > main
void main() async {
  sound: true,
};

HttpOverrides.global = MyHttpOverrides();

runApp(
  EasyLocalization(
    supportedLocales: const [
      Locale('en', 'US'),
      Locale('hi', 'IN'),
      Locale('bn', 'IN'),
      Locale('es', 'ES'),
      Locale('gu', 'IN'),
      Locale('kn', 'IN'),
      Locale('ml', 'IN'),
      Locale('mr', 'IN'), //marathi
      Locale('sa', 'IN'),
      Locale('ta', 'IN'),
      Locale('te', 'IN')
    ],
    path: 'assets/translations',
    fallbackLocale: const Locale('en', 'US'),
    startLocale: const Locale('en', 'US'),
    child: const MyApp(),
  ), // EasyLocalization
);

class MyApp extends StatefulWidget {
  const MyApp({super.key});
```

Update Homescreen.dart with new language code case 11



```
lib > views > HomeScreen > home_screen.dart > _HomeScreenState > build
61 HomeScreen>
106 {
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302

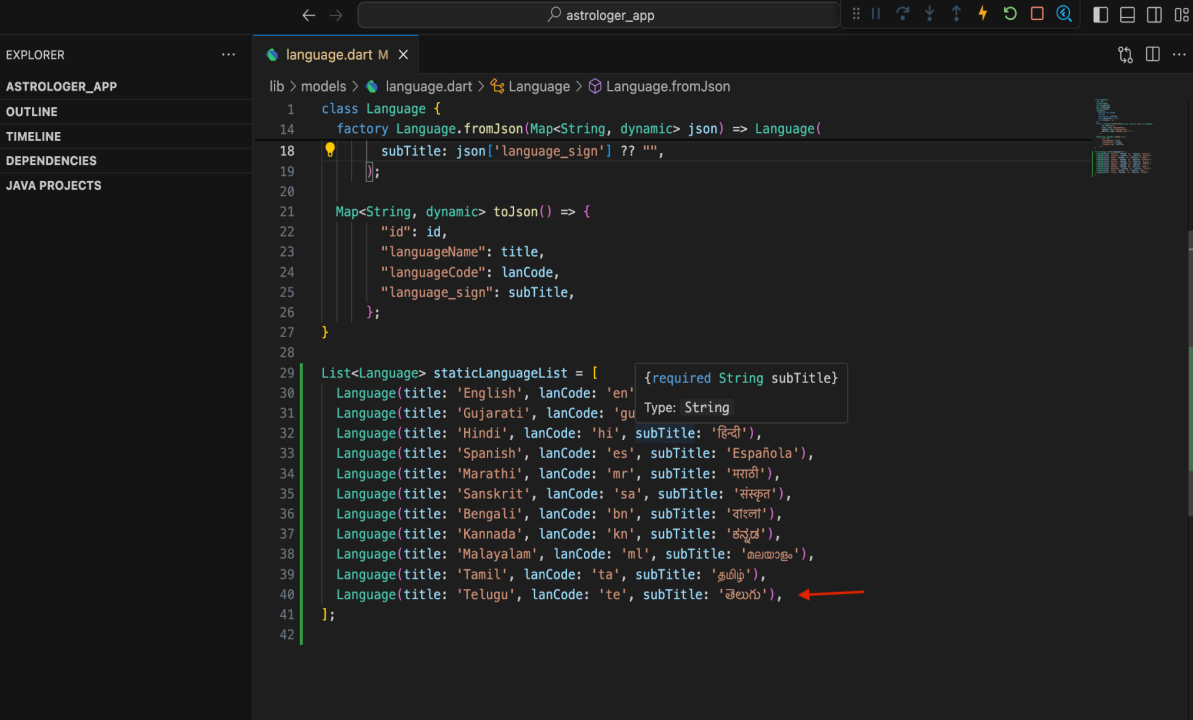
case 8:
  var newLocale = const Locale('ml', 'IN'); //MALYALAM
  context.setLocale(newLocale);
  Get.updateLocale(newLocale);
  refreshIt(homeController);
  break;

case 9:
  var newLocale = const Locale('ta', 'IN'); //TAMIL
  context.setLocale(newLocale);
  Get.updateLocale(newLocale);
  refreshIt(homeController);
  break;

case 10:
  var newLocale = const Locale('te', 'IN'); //TELUGU
  context.setLocale(newLocale);
  Get.updateLocale(newLocale);
  refreshIt(homeController);
  break;

default:
}
```

Now Update Language.dart file add new language in list



The screenshot shows an IDE window for a file named `language.dart`. The left sidebar contains a navigation pane with sections: EXPLORER, ASTROLOGER_APP, OUTLINE, TIMELINE, DEPENDENCIES, and JAVA PROJECTS. The main editor area displays the following Dart code:

```
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',  
31     {required String subTitle} Type: String  
32   Language(title: 'Gujarati', lanCode: 'gu', subTitle: 'ગુજરાતી'),  
33   Language(title: 'Hindi', lanCode: 'hi', subTitle: 'हिन्दी'),  
34   Language(title: 'Spanish', lanCode: 'es', subTitle: 'Española'),  
35   Language(title: 'Marathi', lanCode: 'mr', subTitle: 'मराठी'),  
36   Language(title: 'Sanskrit', lanCode: 'sa', subTitle: 'संस्कृत'),  
37   Language(title: 'Bengali', lanCode: 'bn', subTitle: 'বাংলা'),  
38   Language(title: 'Kannada', lanCode: 'kn', subTitle: 'ಕನ್ನಡ'),  
39   Language(title: 'Malayalam', lanCode: 'ml', subTitle: 'മലയാളം'),  
40   Language(title: 'Tamil', lanCode: 'ta', subTitle: 'தமிழ்'),  
41   Language(title: 'Telugu', lanCode: 'te', subTitle: 'తెలుగు'),  
42 ];
```

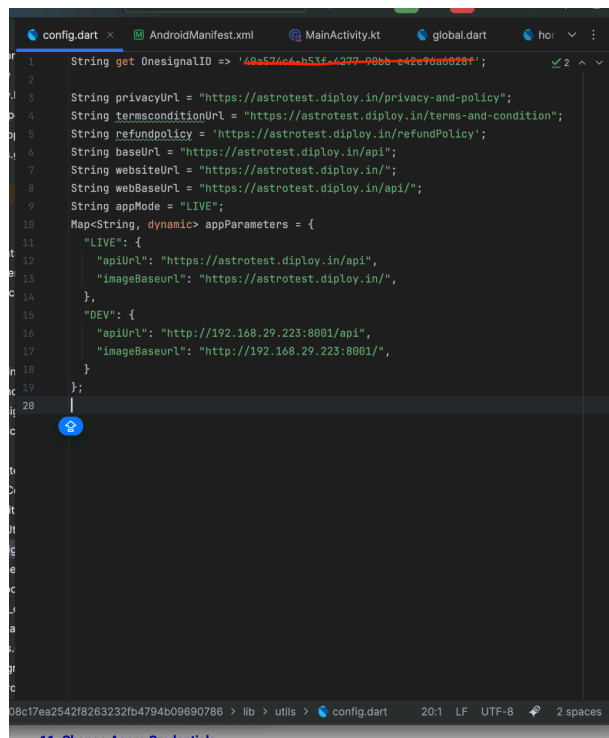
A red arrow points to the new `Language` entry for Telugu on line 41.

10:- Change App Base Url & add onesignalID

lib>utils>config.dart.

From here you can change

- **privacyUrl**
- **termsconditionUrl**
- **refundpolicy**
- **baseUrl**
- **websiteUrl**
- **webBaseUrl**
- **LIVE > apiUrl**



```
1 String get OnesignalID => '60a526c4-b53f-4297-90b6-e42e9ad028f';
2
3 String privacyUrl = "https://astrotest.diploy.in/privacy-and-policy";
4 String termsconditionUrl = "https://astrotest.diploy.in/terms-and-condition";
5 String refundpolicy = 'https://astrotest.diploy.in/refundPolicy';
6 String baseUrl = "https://astrotest.diploy.in/api";
7 String websiteUrl = "https://astrotest.diploy.in/";
8 String webBaseUrl = "https://astrotest.diploy.in/api/";
9 String appMode = "LIVE";
10 Map<String, dynamic> appParameters = {
11   "LIVE": {
12     "apiUrl": "https://astrotest.diploy.in/api",
13     "imageBaseUrl": "https://astrotest.diploy.in/",
14   },
15   "DEV": {
16     "apiUrl": "http://192.168.29.223:8001/api",
17     "imageBaseUrl": "http://192.168.29.223:8001/",
18   }
19 };
20
21
22
23
24
25
26
27
28
```

10B. Setup Process in OneSignal Account

Follow the steps below to configure the Push Platform in OneSignal using your Firebase credentials.

Step 1: Open Push Platform Settings

- Go to your OneSignal project dashboard.


- **Navigate to:**
Settings → Push Platforms → Click on “Settings” under the Push section.
-

Step 2: Upload Firebase Service Account File

- You will be prompted to upload your **Firebase Service Account JSON file**.
- Download this file from your **Firebase Console** (under:
Project Settings → Service Accounts → Generate New Private Key).
- Upload the downloaded **.json** file to OneSignal.

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`

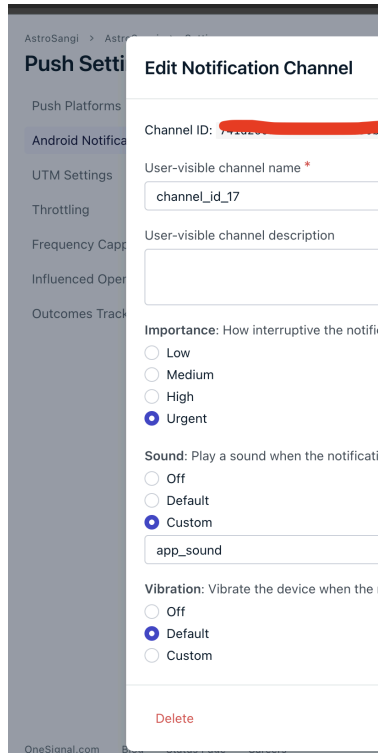
- Save the group.
-

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:

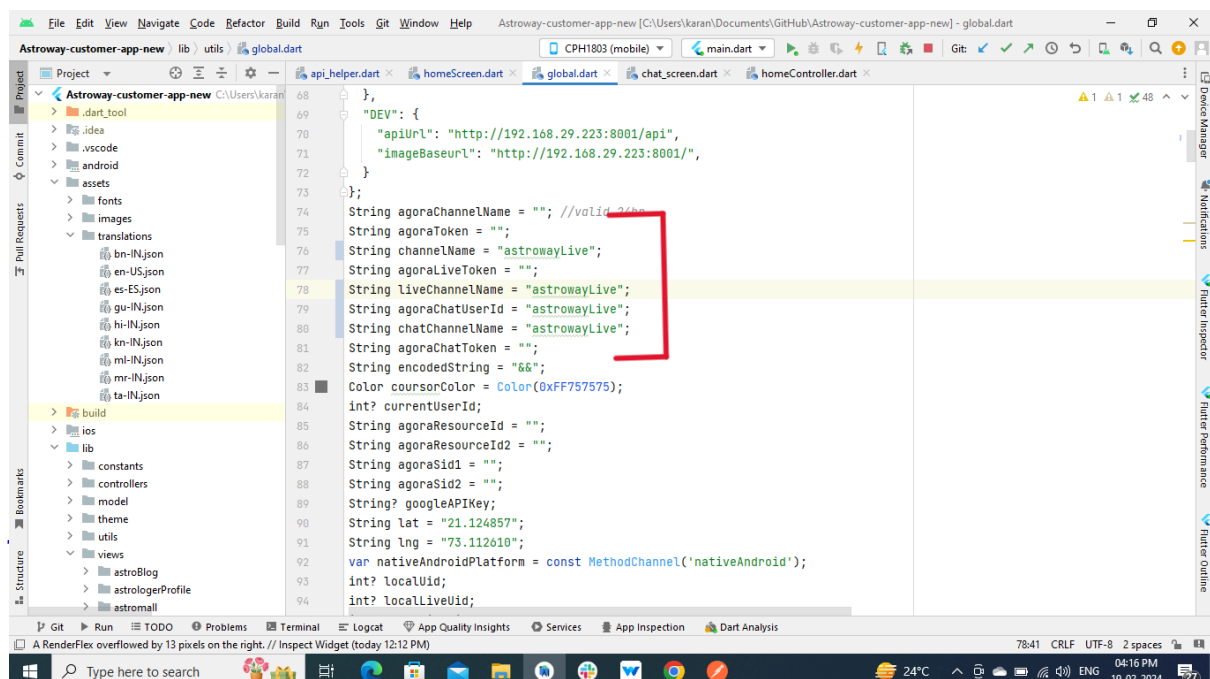
Setting	Value
Name	<code>channel_id_17</code>
Importance	<code>Urgent</code>
Sound	Custom → Type <code>app_sound</code>
Vibration	Default
LED	Default
Badges	Enabled
Lockscreen	Public



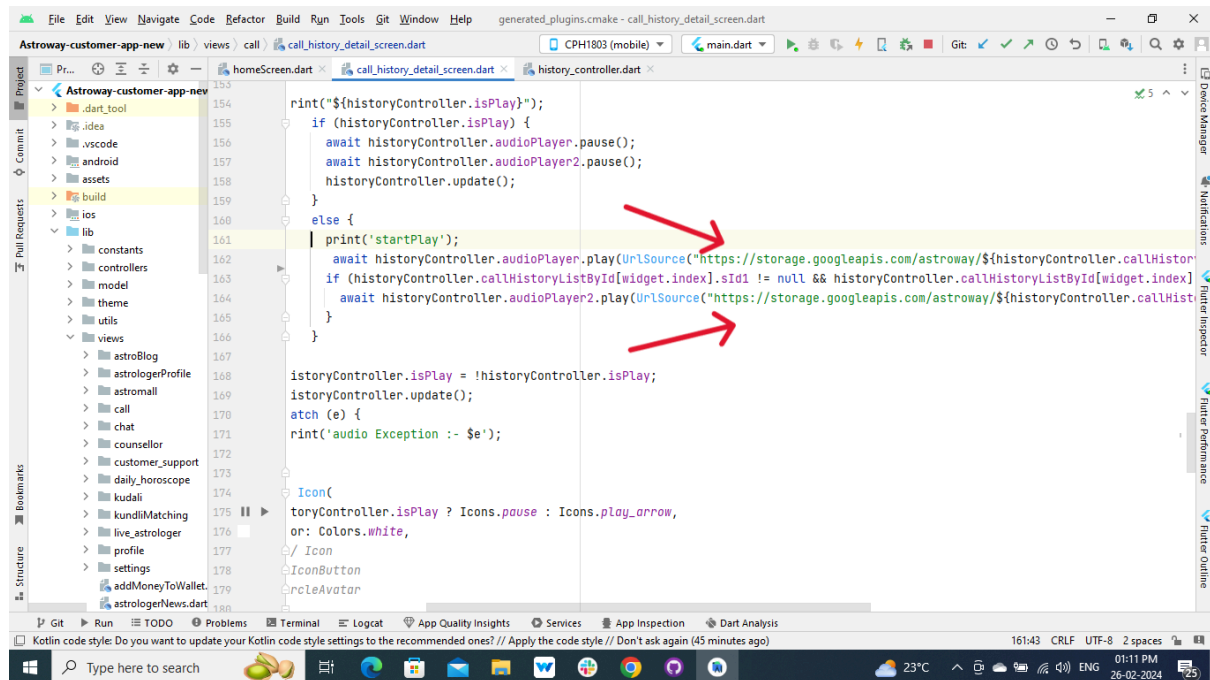
Your OneSignal Push Setting should look like above Screenshot and add channel id in admin panel

11. Change Agora Credentials

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

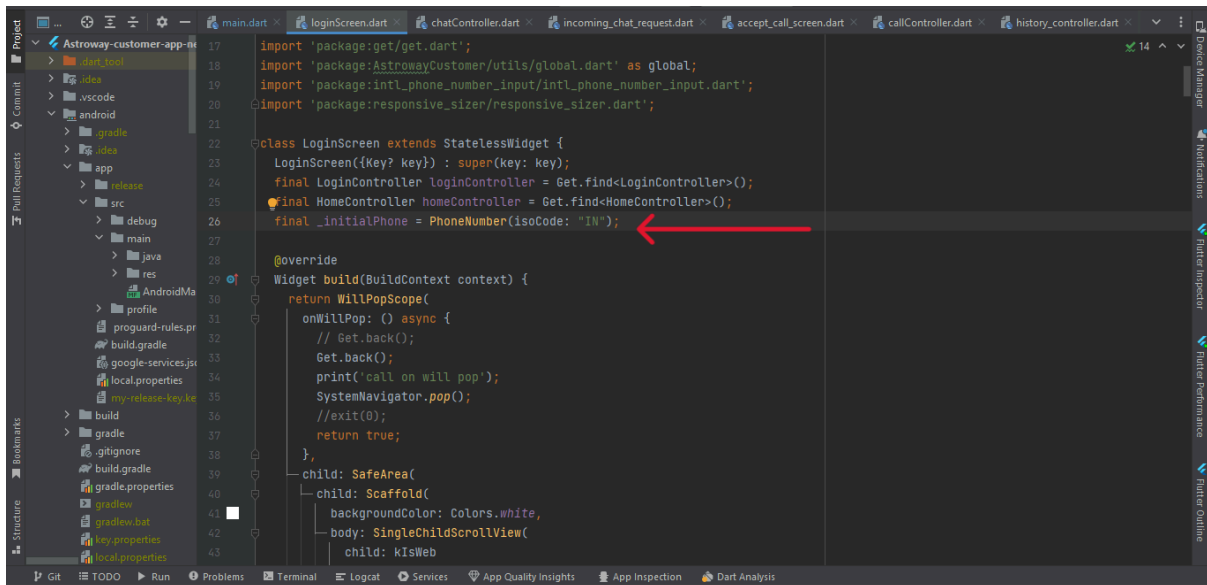


Change the base url from lib>views>call>call_history_detail_screen.dart
Which will help to show the history between customer and astrologer



13. How to change the default country in login screen

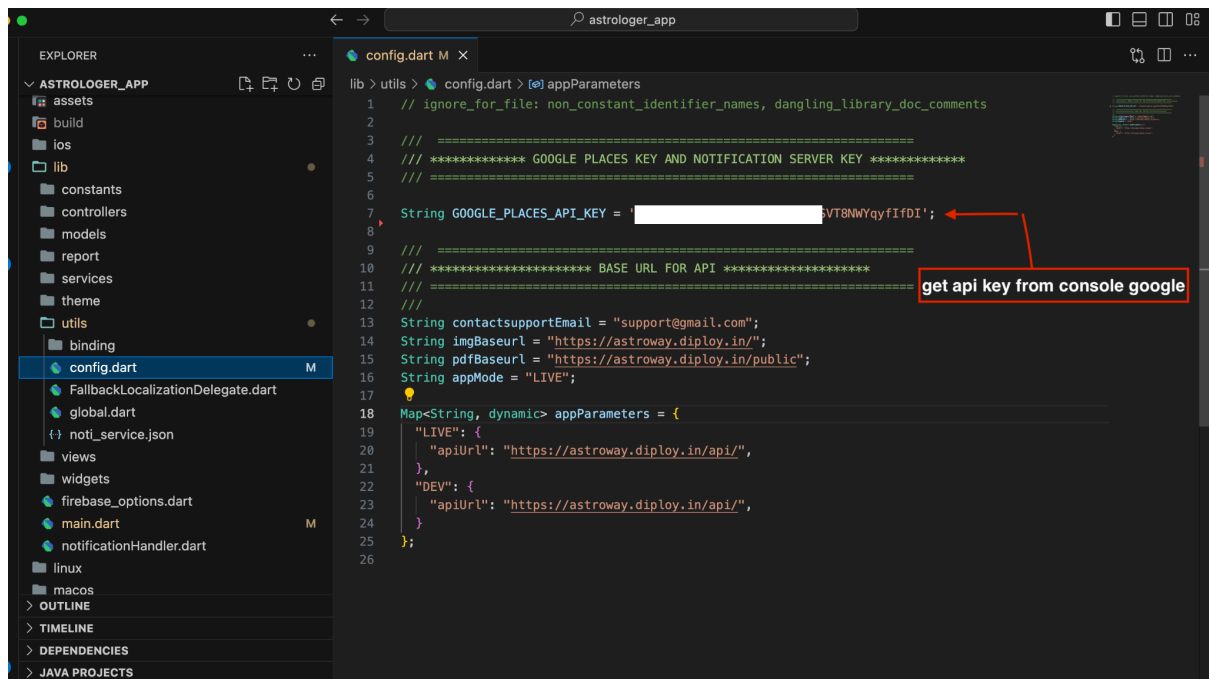
lib>views>loginScreen.dart



Change the country code.

14:-Change App Base Url

You need to update keys such as **CONTACT_SUPPORT** , **IMAGE_BASE_URL**, **PDF_BASE_URL**, **GOOGLE_PLACES_API_KEY**, **SERVER_NOTIFICATION_KEY**, or **API_URL**, simply open the **config.dart** file and paste your keys there.



A:- Change Google Places API Key

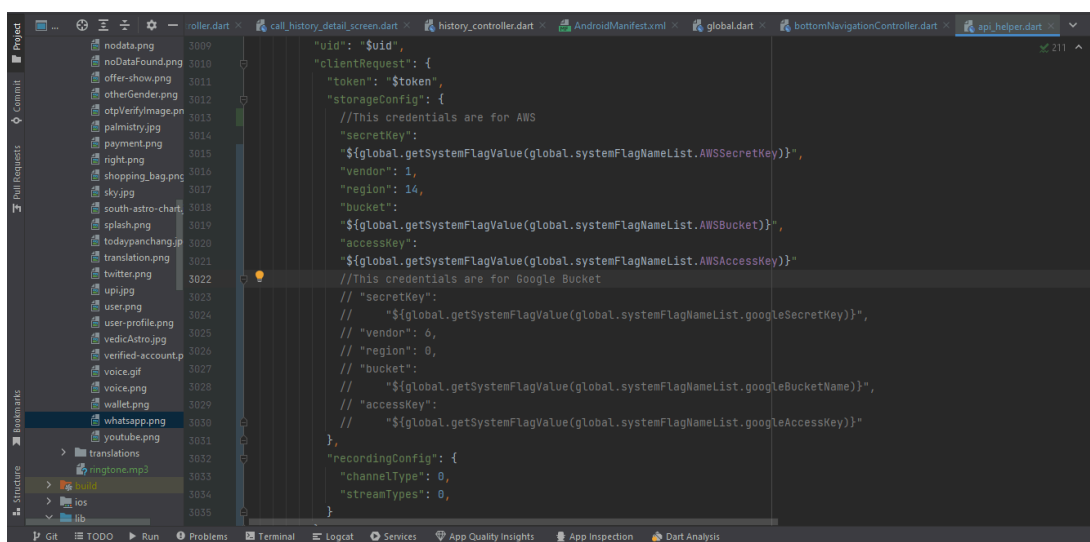
To change the API key for Google Places, please follow these steps:

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.

15. How to change googleBucket Storage to AWS and vice versa

You have to `api_helper.dart` file

Path:- **Project>lib>utils>services>>api_helper.dart**



16. Path of the Important Screens

- loginScreen:- lib>views>loginScreen.dart
- otpScreen:- lib>controllers>intake_verifyotp_screen.dart
- Homescreeen:- lib>views>homeScreen.dart
- daily Horoscope Screen:- lib>views>daily_horoscope>dailyHoroscopeScreen.dart
- Free Kundali Screen:- lib>views>kundali>kundliScreen.dart
- Kundali Matching Screen:- lib>views>kundliMatching>kundliMatchingScreen.dart
- Shopping Screen:- lib>views>astromall>astromallScreen.dart
- CatgoriesScreen:- lib>views>categoryScreen.dart
- Blog Screen:- lib>views>astroBlog>astrologyBlogListScreen.dart
- Panchang Screen:- lib>views>panchangScreen.dart
- Chatl Screen:- lib>views>chatScreen.dart
- Call Screen:- lib>views>callScreen.dart
- Live Astro Screen:- lib>views>liveAstrologerList.dart
- Histroy Screen:- lib>views>historyScreen.dart
- Edit Profile Screen:- lib>views>profile>editUserProfileScreen.dart
- Project Report Screen:- lib>views>getReportScreen.dart
- My Following Screen:- lib>views>myFollowingScreen.dart
- Free Services Screen:- lib>views>freeServicesScreen.dart
- Settings Screen:- lib>views>setting>settingsScreen.dart
- Add Money Screen:- lib>views>addMoneyToWallet.dart
- Customer Support Screen:-
lib>views>customer_support>customerSupportChatScreen.dart
- Search Screen:- lib>views>setting>searchAstrologerScreen.dart
- Astrologer Profile Screen:- lib>views>astrologerProfile>astrologerProfile.dart
- Astrologer Product Screen:- lib>views>astromall>astroProductsScreen.dart
- Payment Detail Screen:- lib>views>paymentInformationScreen.dart
- Call In Take Form:- lib>views>callIntakeFormScreen.dart

Still Have a Question?

Please mail us on nb@diploy.in

Contact Support

We're available Mon-Fri, 10:00 am - 7:00 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 Team, your go-to hub for top-notch Flutter app development and web applications.

Visit us at <https://diploy.in/> . Thank you for considering us!