



Tv Show (1.0)

Last Updated: 31-10-23

The application works with the php with a control panel from the application for ease of management
With a player videos that supports dlina, cast
It is easy to follow everything

1. Introduction

Welcome to **Tv Show** app installation doc & thank you for buying our product. Hope you have successfully configured your **Live Tv** site using our **Live Tv** doc. If not then please do that first. Otherwise the app will not work.

2. Flutter Installation

So the app is developed on Flutter. So, You have to install flutter on your computer. You can use Both **Visual Studio Code** & **Android Studio** as an IDE for Flutter. The steps are the same for both IDE. We have used Visual Studio Code. So, Our Setup will be based on this IDE.

To install flutter on your computer, follow the official documentation from Google.

Flutter Official Site : <https://flutter.dev/>.

You can follow these youtube videos to install flutter also.

1. For Mac: <https://www.youtube.com/watch?v=9GuzMsZQUYs>
2. For Windows: <https://www.youtube.com/watch?v=T9LdScRVhv8>



Make sure you have installed the latest stable version of flutter. If everything is okay then you can follow the further steps below.

Note: A Mac (Apple Desktop/laptop) device and an apple developer account is required for the iOS setup. If you don't have both of them, you can ignore the iOS steps.

3. App Setup

After purchasing the app you will get a .zip file and unzip it. Inside the main folder you will get a folder named **Tv Show**. This is the source code of the app. You have to work on this to set up your app. Now open your IDE and open the **Tv Show** folder on your IDE. Wait some time to load the project.

Now go to the IDE terminal and run the following commands one after one :

```
flutter clean
```

After that, run

```
flutter pub get
```

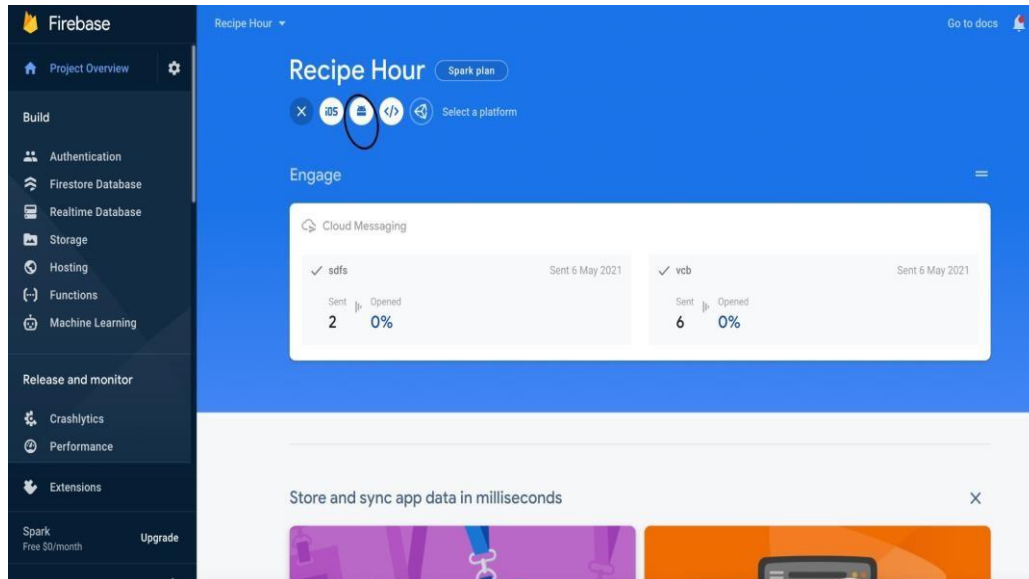
Wait some time to get all the packages.

4. Firebase Push Notification for Android Configure the application for Firebase to use the application database

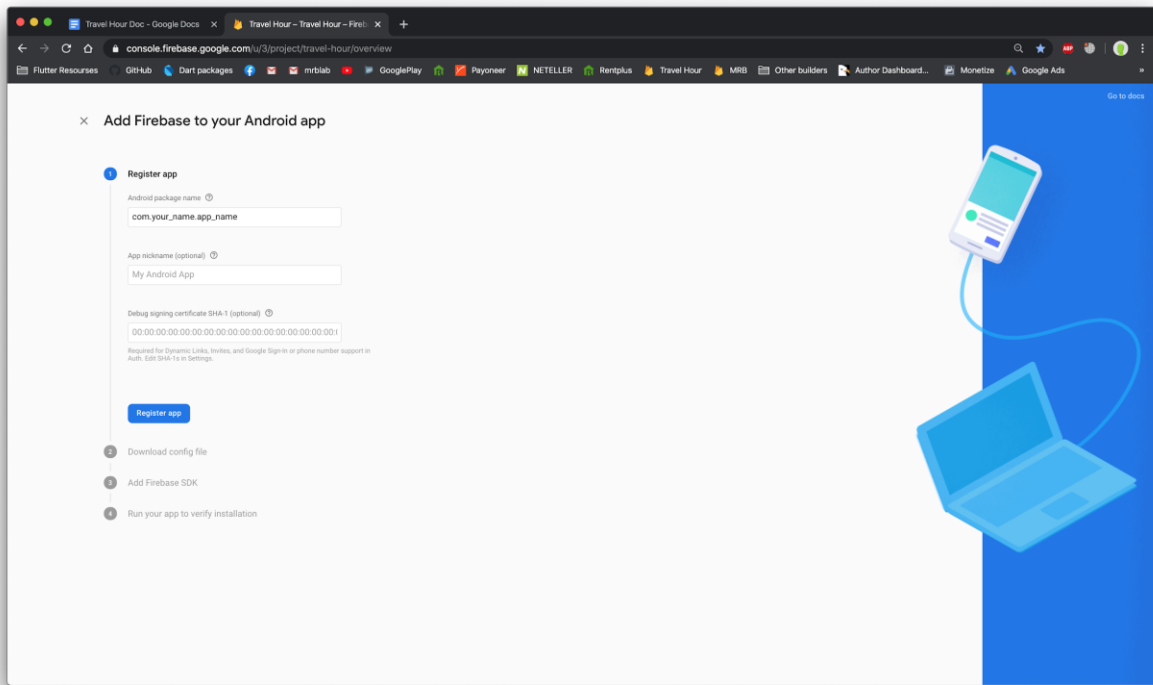
To enable push notifications, you have set up a firebase project. Please follow the steps below.

1. First go to the [Firebase Console](#) and sign in with your gmail account and go to the console.

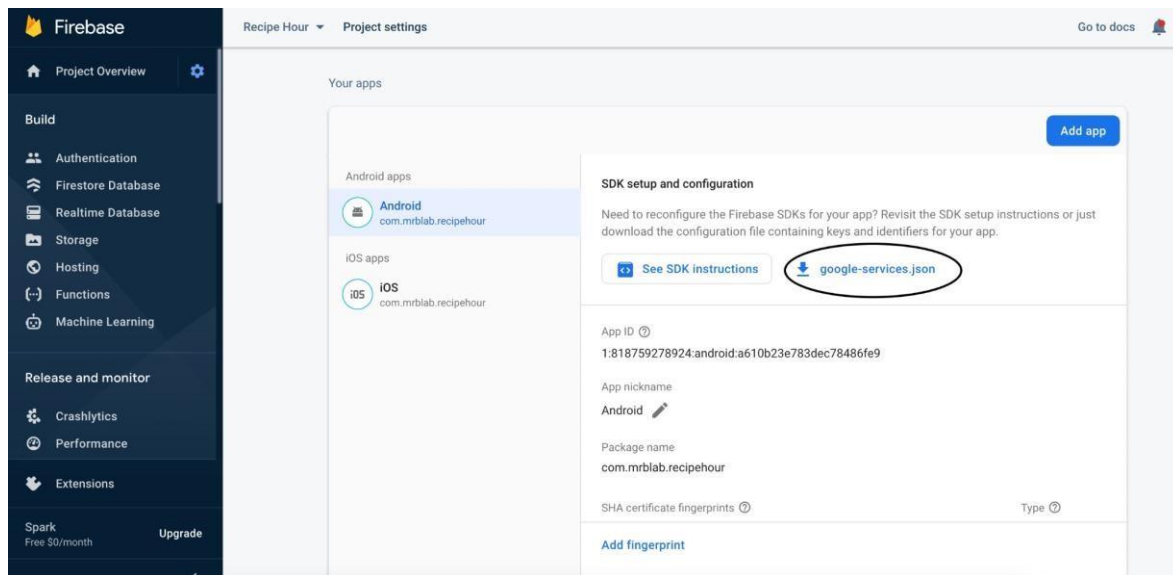
2. Create a project by your app name. And go to the project overview and click on the **plus icon** and then click on the **Android Icon**.



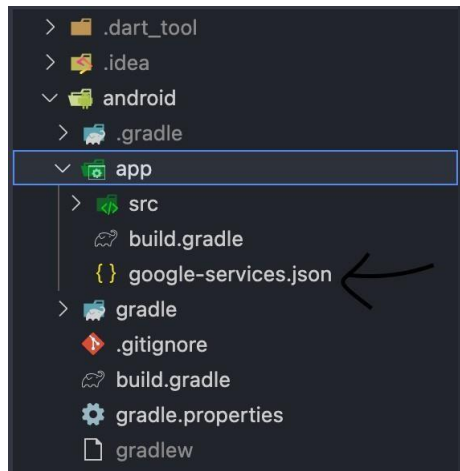
3. Enter your android package name. Your package name should be like **com.your_name.your_app_name** . Like **com.microsoft.skype**. You can use the same package name for android & iOS. iOS doesn't support **underspace** in the package name. So, keep in mind that if you want to use the same package name for both android & iOS.



4. Click on the register app and skip other steps by clicking next.
5. Now go to project settings and click on the **android icon** and download **google-service.json** file.



6. Now go to **Tv Show/android/app** directory and paste the **google-service.json** file here.



That's it. Push notification setup for android is complete.

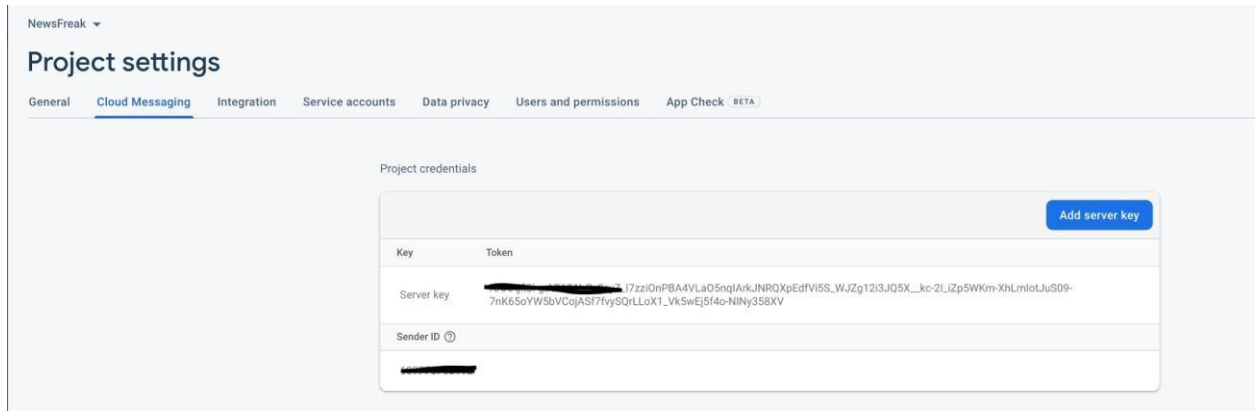
5. Firebase Push Notification for iOS

You can skip this step if you don't want to build this app for iOS.

1. Now, go to your firebase console and now click on the **iOS icon** and enter your iOS package name and click on the **Register app** and skip the other steps.
2. Now go to project settings and click on **iOS** and download the **GoogleService-info.plist** file.
3. Then go to **Tv Show/ios/Runner** directory and paste the file here.
4. Now, Open the **iOS folder** on Xcode by right clicking on the iOS folder from VSCode or Android Studio and go to the runner folder and **drag & drop** the **GoogleService-info.plist** file here. You will get a popup and click yes or confirm the popup message. That's it.
5. Now follow these additional steps from the flutter team [here](#). This is a well written doc. You can ignore the optional steps.

6. Firebase Push Notification for Live Tv

- Go to your **firebase console** > **your app** > **Project Settings** > **Cloud Messaging Tab** and copy the **server key**. Look at the picture below:



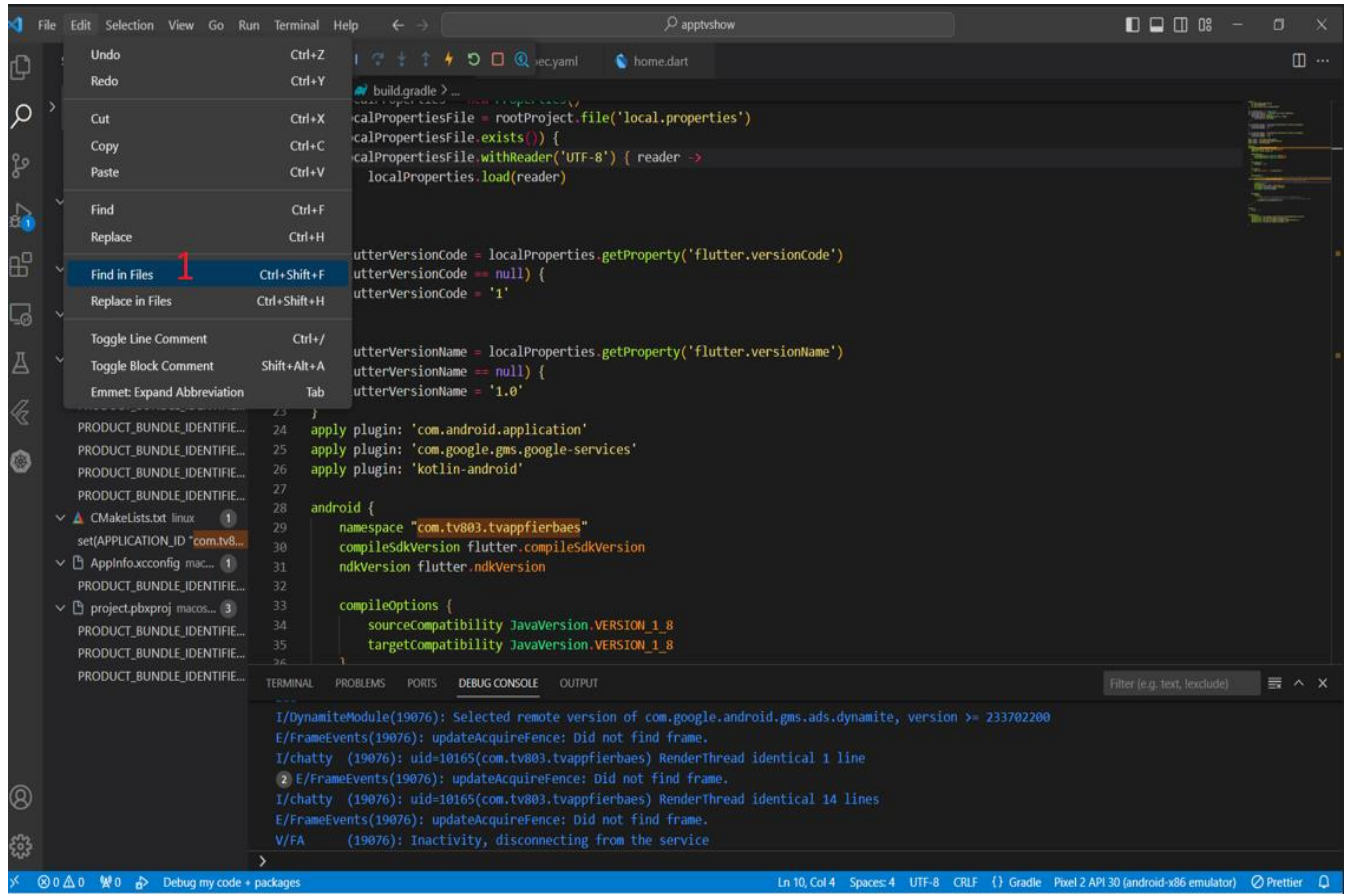
- Save the server key for later use
- In order to be used later in the application's control panel

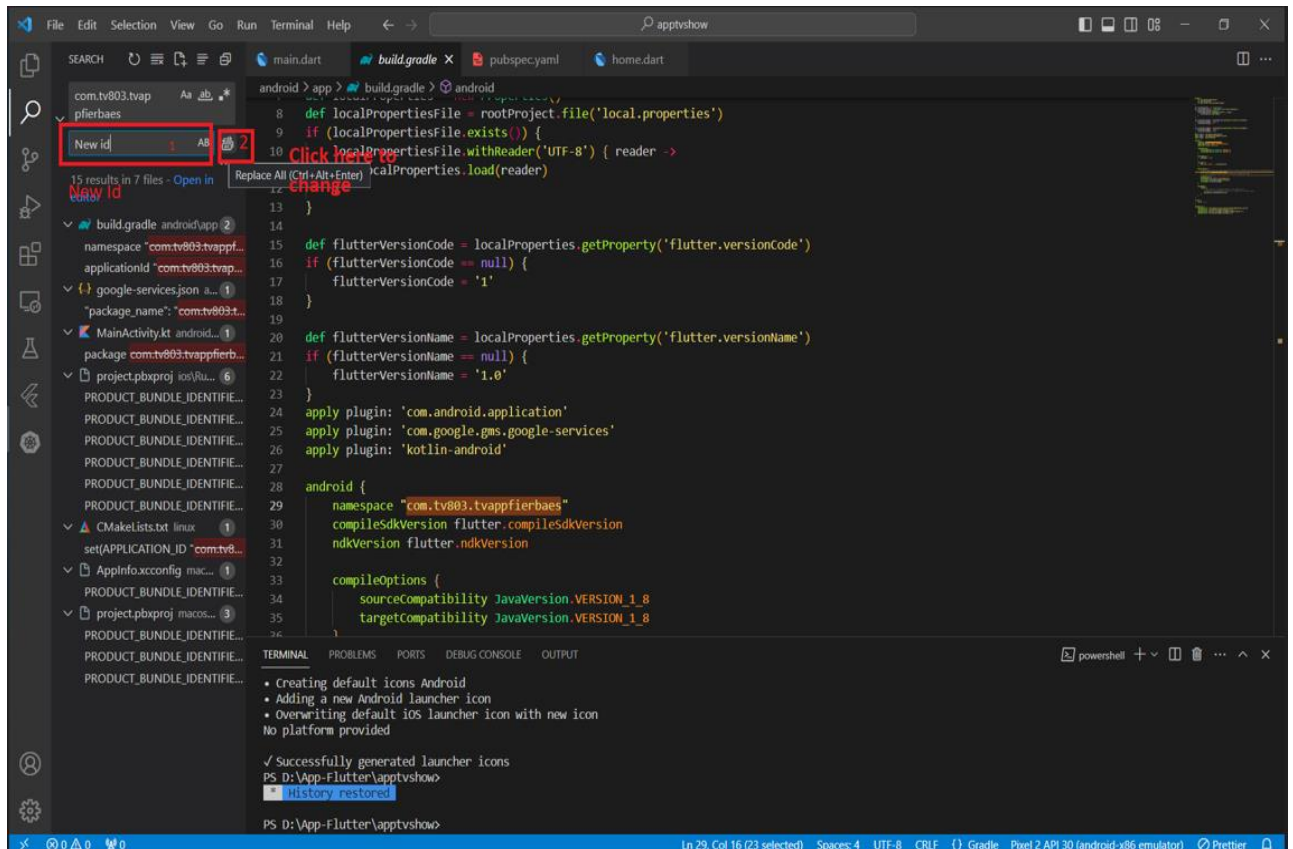
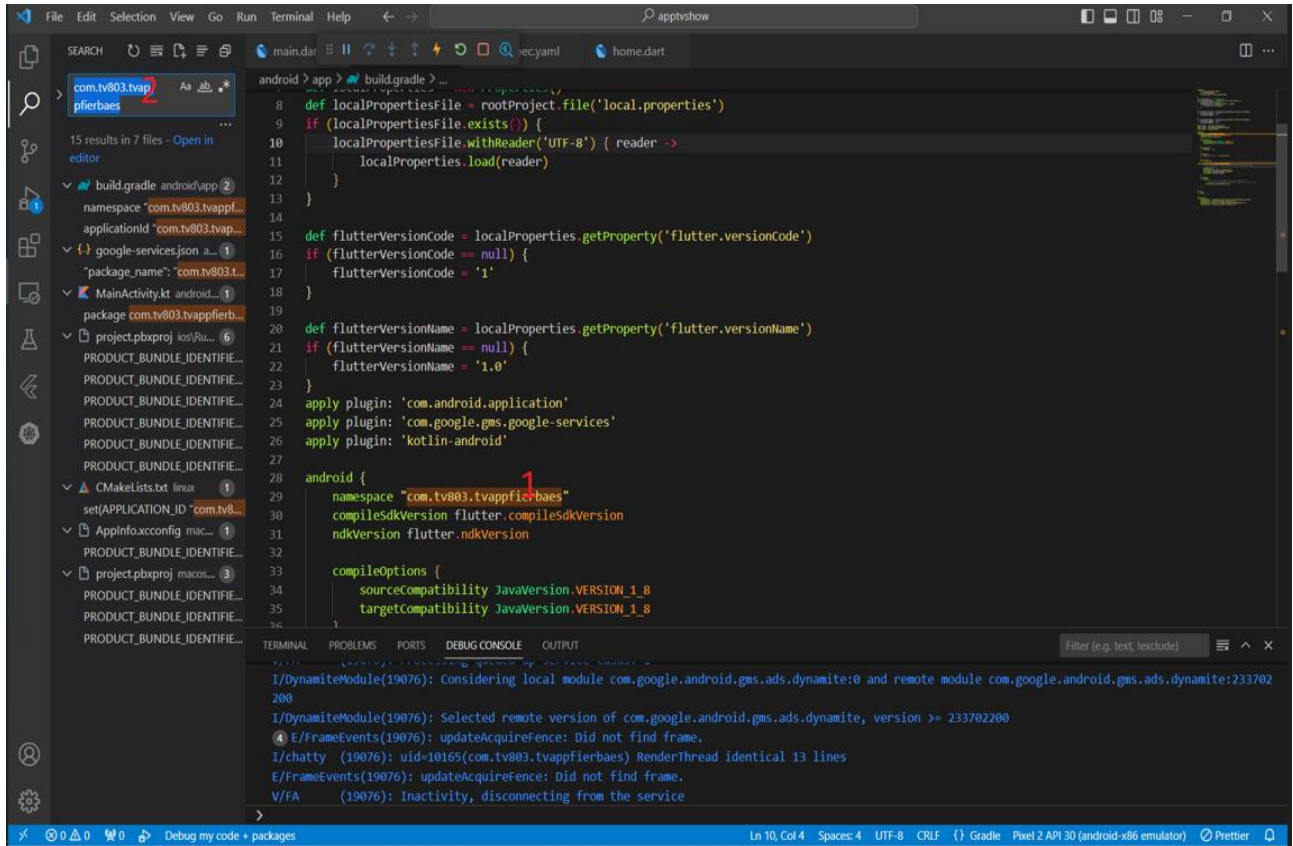
7. Change Package Name for Android

Now, you have to change the package name in the source code. You have to use the same package name that you have registered in the firebase console.

1. Go to your IDE and now you **have to change the package name** of your app. Run the following command from your IDE terminal by changing the package name with yours.

Don't forget to replace **com.new.package.name** with your package name. That's it. Your package name has been changed now.





Don't forget to replace **com.new.package.name** with your package name. That's it. Your package name has been changed now.

8. Change Package Name for iOS

10. Admob Setup

We have used admob and 2 types of admob ads. **Banner, Interstitial** ads. So, you have to create an admob account if you don't have one. You can use two types of ads from them. Banner & Interstitial.

10.2 Admob for Android

1. Create an App in your admob account for Android. Create one banner ad, one interstitial ad
2. Go to **android/app/src/main/AndroidManifest.xml** file and replace with your **admob app id** (Android) of yours which you will get from your admob account.

```
78      <!--admob section-->
79
80      <meta-data
81          android:name="com.google.android.gms.ads.APPLICATION_ID"
82          android:value="ca-app-pub-3940256099942544~3347511713"/>
83
84
```

10.3 Admob for iOS

1. Create an App in your admob account for iOS. Create one banner ad, one interstitial ad & one rewarded video ad.

2. Go to **ios/Runner/Info.plist** file and replace with your **admob app id** (iOS) of yours which you will get from your admob account.

```
70  
71     <key>GADApplicationIdentifier</key>  
72     <string>ca-app-pub-3940256099942544~1458002511</string>  
73  
74     <key>SKAdNetworkItems</key>  
75     <array>
```

For the first time to run the application, enter the control panel and add the ad units to the Google experience and ignore any errors when running the application for the first time without adding the ad units. Any error will end after adding the trial ads for Google

13. Change App Name for Android

Go to `android/app/src/main/AndroidManifest.xml` file and Change your app name.

```
6
7     <application
8         android:label="Recipe Hour"
9         android:icon="@mipmap/ic_launcher"
10        android:usesCleartextTraffic="true">
11        <activity
```

14. Change App name for iOS

Go to `ios/Runner/Info.plist` file and Change your app name.

```
19     </array>
20     <key>CFBundleName</key>
21     <string>Recipe Hour</string>
22     <key>CFBundlePackageType</key>
23     <string>APPL</string>
24     <key>CFBundleShortVersionString</key>
25     <string>$(FLUTTER_BUILD_NAME)</string>
26     <key>CFBundleSignature</key>
27     <string>????</string>
28     <key>CFBundleVersion</key>
29     <string>$(FLUTTER_BUILD_NUMBER)</string>
30     <key>LSApplicationQueriesSchemes</key>
```

15. Change App Icon

1. Go to the `/images` /folder and delete the default icon (**icon.png**).
2. Now upload your app icon as png in the `/images` folder and rename it to **icon.png**

- 
3. Now run the following command on the terminal,

```
flutter pub get flutter pub run flutter_launcher_icons:main
```

That's it.. For more info, visit this [site](#).

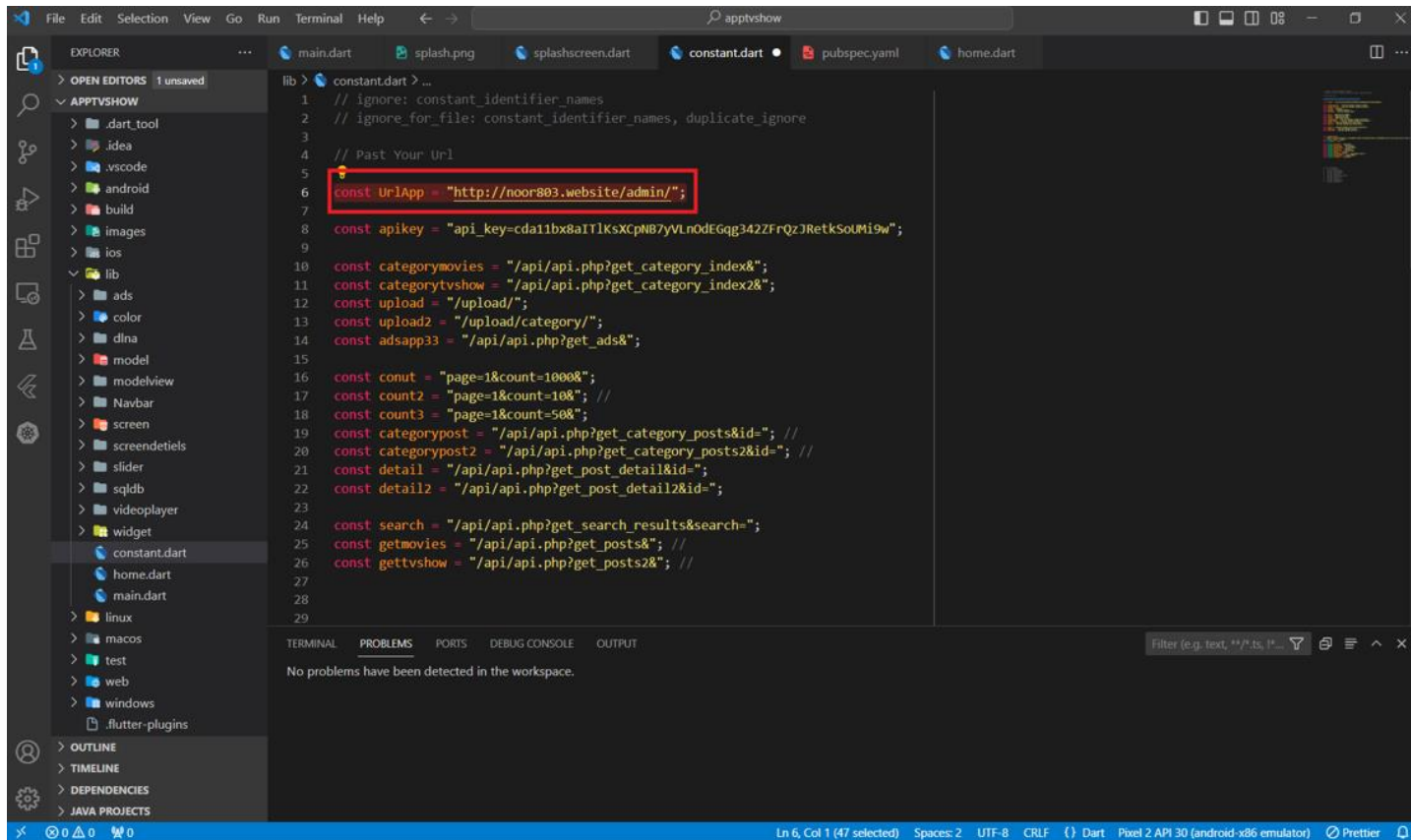
16. Change Splash Icon

To change the **splash icon**, you have to upload your own splash icon. The icon should be in the **.png** format and make sure you have renamed it to **splash** . Go to the **lib /images** folder and drop the icon here and replace it with our icon.

17. Add a link to your control panel after uploading and installing everything

Paste your link here

1- Go to `\lib\constant`



Change the color of the application once and control it go to this path
\\lib\\color\\colors.dart\\

18. Android Release Key Setup

To generate a release certificate, You have to generate a keystore file. To generate a keystore file, run this command below from the root of your project directory on the terminal.

For Mac users, run

```
keytool -genkey -v -keystore ~/upload-keystore.jks -keyalg RSA -keysize 2048 -validity 10000 -alias upload
```

For Windows users, run

```
keytool -genkey -v -keystore c:\Users\USER_NAME\upload-keystore.jks -storetype JKS -keyalg RSA -keysize 2048 -validity 10000 -alias upload
```

1. Enter your details and remember the **password**. After this, you will get an **upload-keystore.jks** keystore file.
2. Locate this file and move the file into the **android/app** folder and copy the path by right clicking on the **upload-keystore.jks** file
3. Then go to **android/key.properties** file and replace the path of the keystore file of yours. Then also replace the **password** which you have inputted to generate the keystore file.

```
android > key.properties
1 storePassword=
2 keyPassword=
3 keyAlias=upload
4 storeFile=/Users/rakibbhuiyan/Desktop/imac/final_projects/recipe_hour/source/wordpress_app/android/app/upload-keystore.jks
```

That's it. Android & iOS App Setup is complete.

19. Run The App

So Your Setup is 100% complete now. Now run this following command to clean the project.

```
flutter clean
```

After that, run this command,

```
flutter run
```

And After that run the following command to run this app on your physical or emulator devices. Make sure you have connected an android/ios emulator or connected an real device via USB.



```
flutter run
```

Test if everything is okay or not.

20. Release The Android App on Google Play Store:

You have done all the things that are required for android release. To Test the release android app, run the following command on the terminal.

```
flutter build apk --split-per-abi
```

You will get 3 apk files from the **build/app/output/apk/release** folder. You can test the **v7** version of the apk file. If you want to publish the app in the google play store, don't upload any of the following files. Use an **appbundle** file which is recommended by Google. To generate an appbundle, run the following command on terminal :

```
flutter build appbundle
```

After that, you will get an **.aab** file in the **build/app/output/appbundle/release** folder. Now you can upload this .aab file to the google play store.

21. Release the iOS app on App Store :

Follow the official doc from flutter team [here](#).



22. Conclusion

That's it. We know that you are so tired right now. Take some rest. Everything is complete now.

If you love our work then don't forget to submit a review on the codecanyon market.

Thanks

Contact: noor80312@gmail.com