

# Single Radio

Documentation

Thank you for purchasing my product.

Welcome to the setup guide for your Flutter application. This document contains all the necessary information to help you get started with your app. If you encounter any issues during the setup process, please do not hesitate to contact me at [support@cheebiez.com](mailto:support@cheebiez.com). For efficient troubleshooting, your email should contain:

- **Subject:** The complete title of the project, as listed on CodeCanyon.
- **Description:** Attach a screenshot of the problem and include the output of “flutter doctor -v” for a detailed analysis. Also, please provide a brief description of the issue.

Interested in exploring more of my work? Feel free to visit my portfolio at <https://codecanyon.net/user/cheebiez/portfolio>.

# Contents

- [Getting Started](#)
- [Import Project](#)
- [App Name](#)
- [Package Name](#)
- [App Icon](#)
- [Splash Screen](#)
- [Admob](#)
- [Firebase Cloud Messaging](#)
- [Config](#)
- [Color Theme](#)
- [Assets](#)
- [Build the Release](#)
- [Known Issues](#)
- [Support](#)

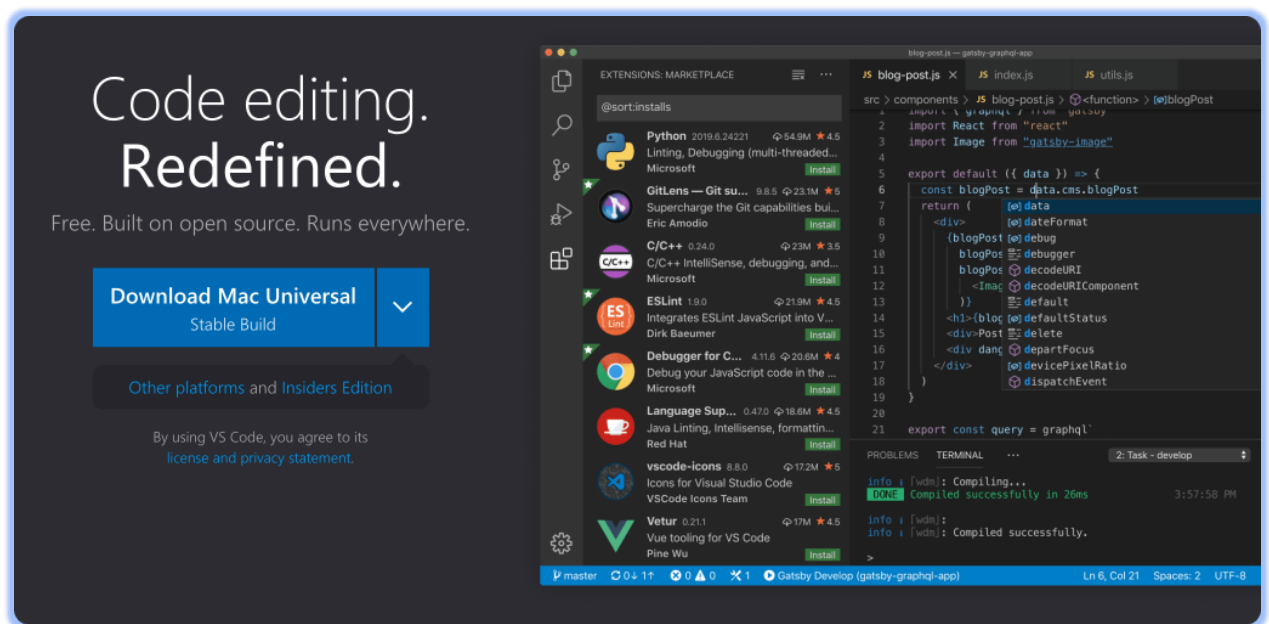
# Getting Started

This section is crucial for those setting up a Flutter development environment for the first time. However, if you have previous experience building apps in Flutter, feel free to skip this part.

## Visual Studio Code Installation

VS Code is a lightweight editor with Flutter app execution and debug support.

You can download it here: <https://code.visualstudio.com/>



Install the Flutter and Dart plugins:

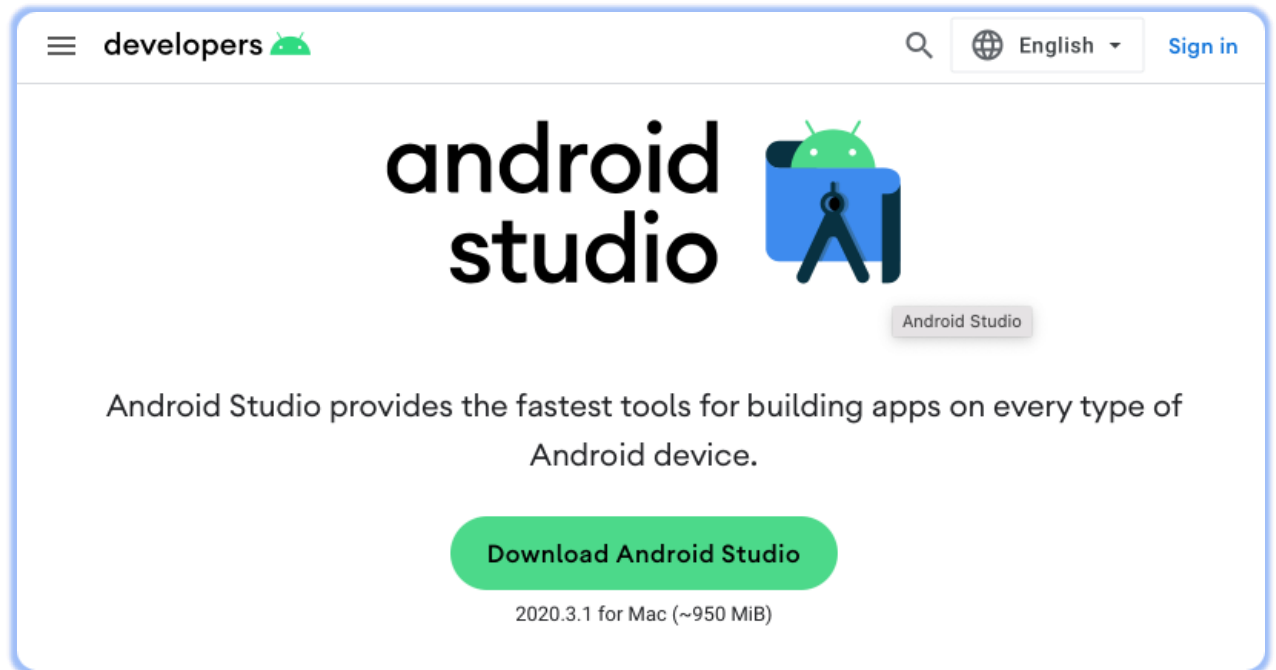
- Start VS Code.
- Invoke View > Command Palette....
- Type “install”, and select Extensions: Install Extensions.
- Type “flutter” in the extensions search field, select Flutter in the list, and click Install.

This also installs the required Dart plugin.

## Android Studio Installation

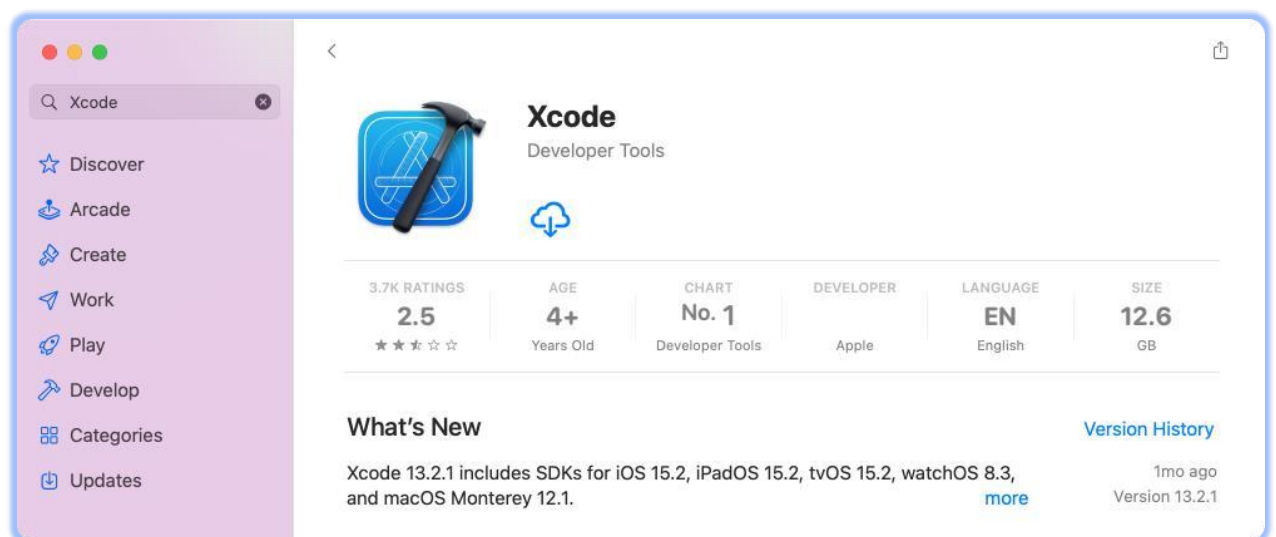
Download and install the latest Android Studio from this URL:

<https://developer.android.com/studio/>



## Xcode Installation

Launch the App Store application on your Mac, sign in with your Apple ID, and download the latest version of Xcode.



# Flutter Installation

You can download the stable release of the Flutter SDK from this URL:

<https://docs.flutter.dev/get-started/install>

## Get the Flutter SDK

- Download the following installation bundle to get the latest stable release of the Flutter SDK:  
`flutter_macos_2.8.1-stable.zip`  
For other release channels, and older builds, see the [SDK releases](#) page.
- Extract the file in the desired location, for example:

```
$ cd ~/development
$ unzip ~/Downloads/flutter_macos_2.8.1-stable.zip
```
- Add the `flutter` tool to your path:

```
$ export PATH="$PATH: `pwd` /flutter/bin"
```

This command sets your `PATH` variable for the *current* terminal window only. To permanently add Flutter to your path, see [Update your path](#).

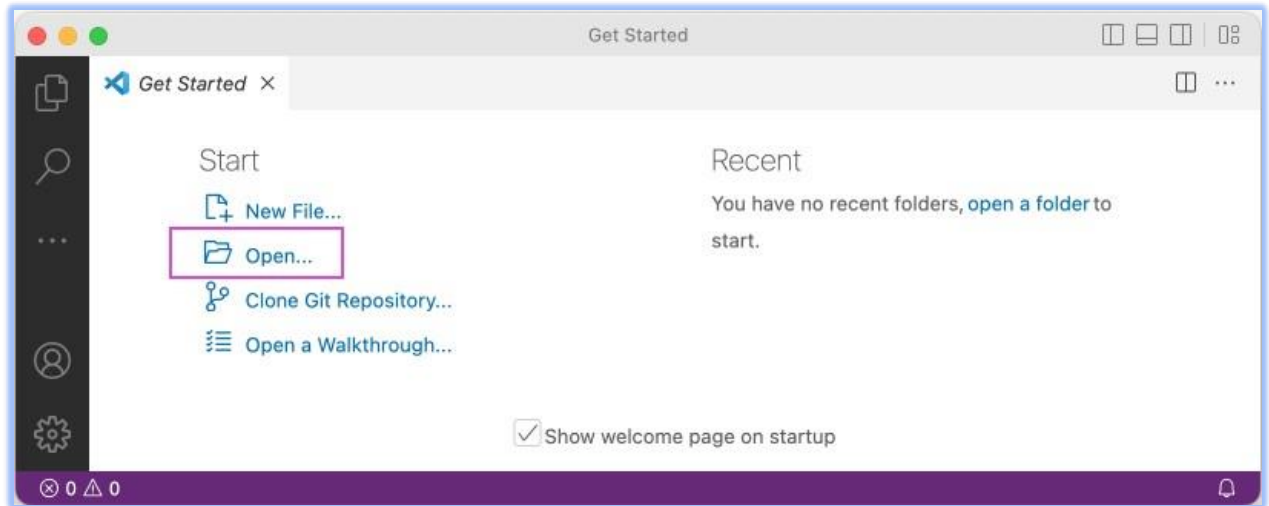
### Contents

- [System requirements](#)
- [Get the Flutter SDK](#)
  - [Run flutter doctor](#)
  - [Downloading straight from GitHub instead of using an archive](#)
  - [Update your path](#)
- [Platform setup](#)
- [iOS setup](#)
  - [Install Xcode](#)
  - [Set up the iOS simulator](#)
  - [Create and run a simple Flutter app](#)
  - [Deploy to iOS devices](#)
- [Android setup](#)
  - [Install Android Studio](#)
  - [Set up your Android device](#)

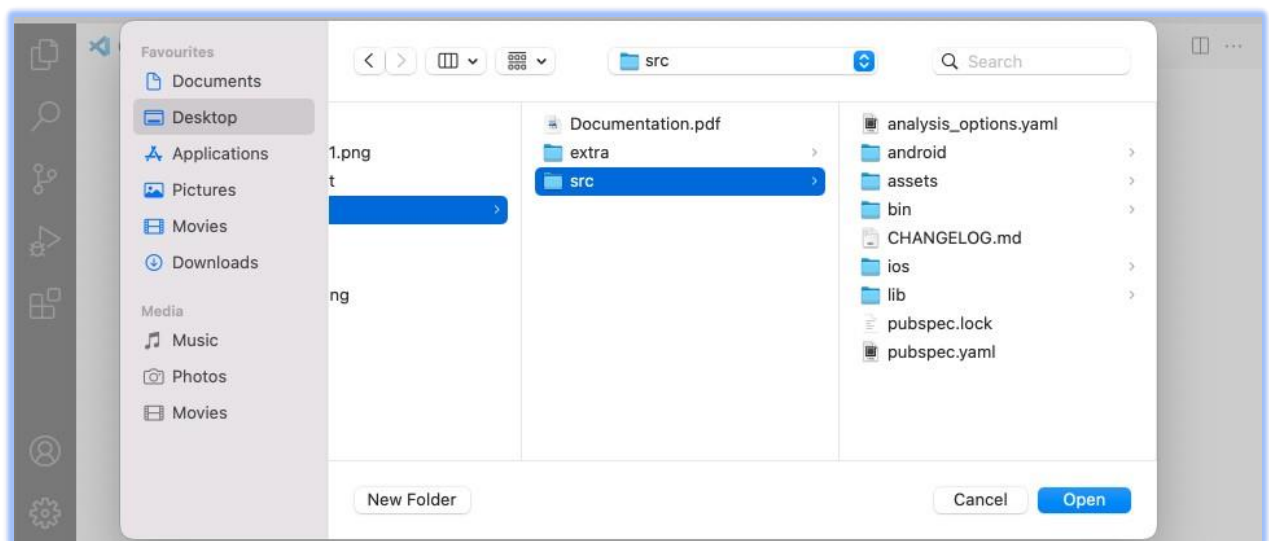
Complete all installation steps.

# Import Project

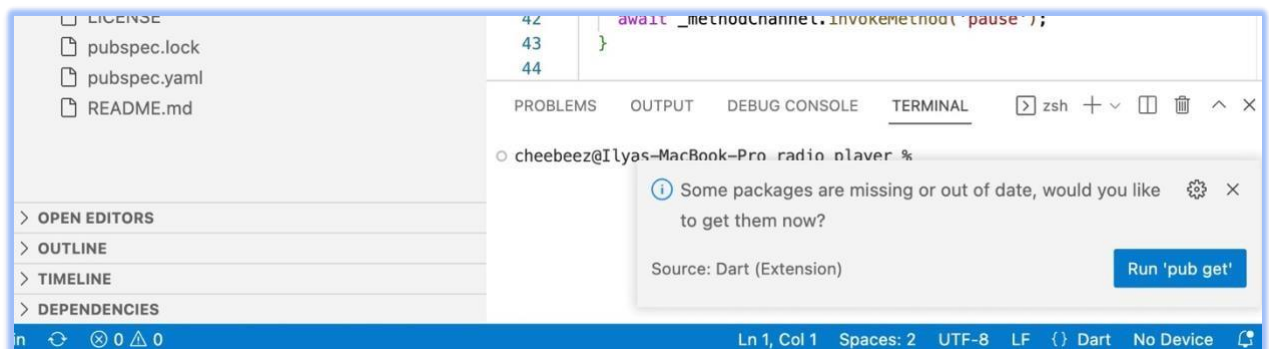
Launch Visual Studio Code and click "Open..."



Select the location of the `src` directory.



Click on "Run pub get" in the dialog box and wait for the packages to be downloaded.

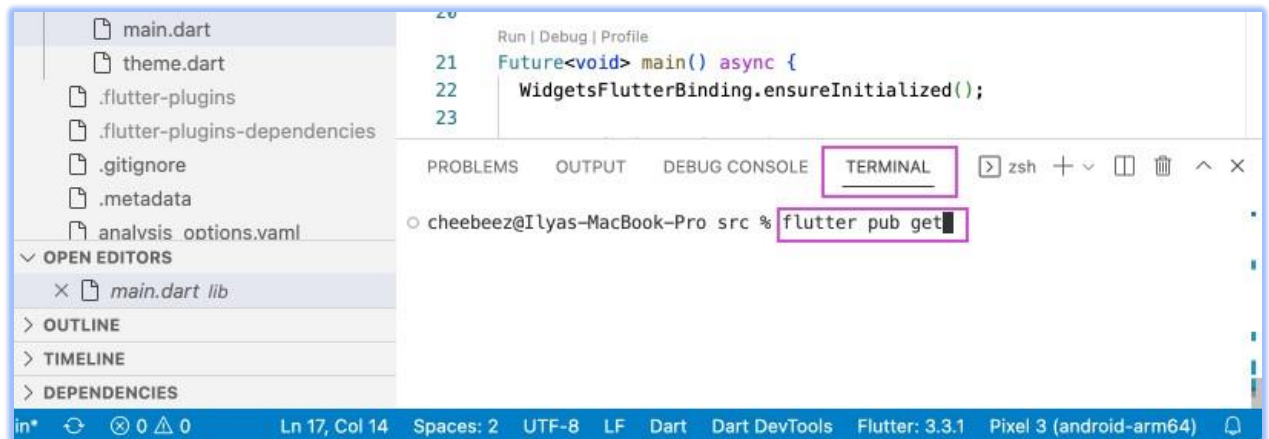


On the first build, VS Code automatically installs dependencies, including the Android SDK and Pods for Xcode.

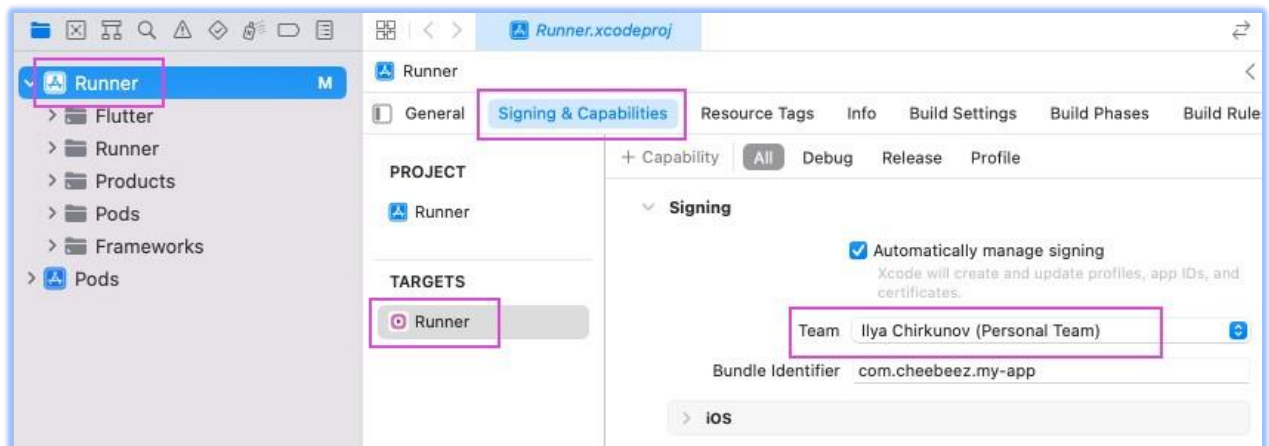
To install them manually, open the terminal (Ctrl+`) and run the commands below.

**flutter pub get**

**pod install --project-directory=ios**



To develop iOS apps, you must be a member of the Apple Developer Program. Open the `ios/Runner.xcworkspace` file in Xcode and select your Apple Developer account.



Now you can run the app.

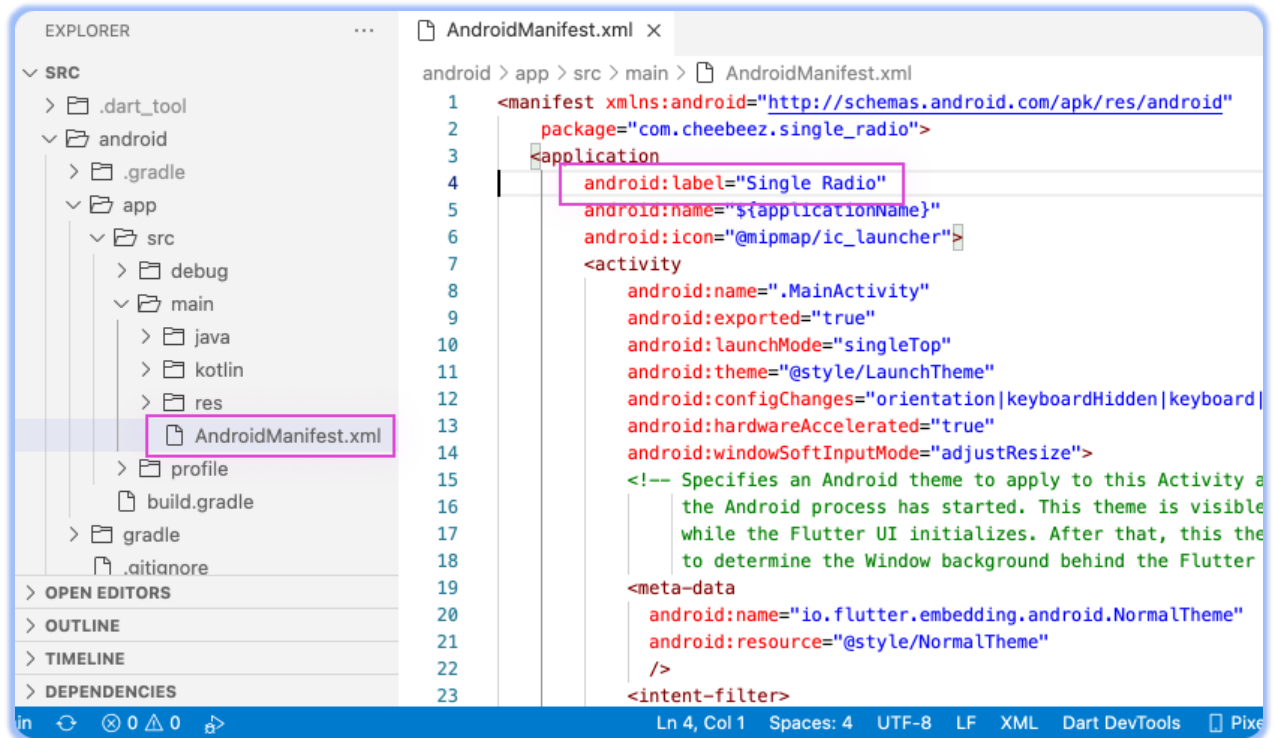




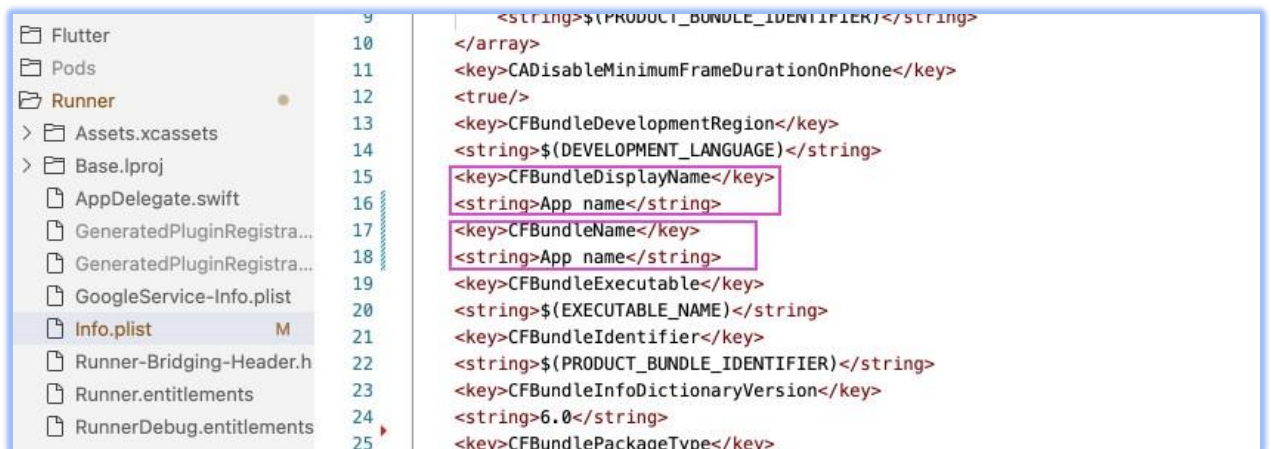
# App Name

Users will see this name under the application icon.

Change the “label” value in the `android/app/src/main/AndroidManifest.xml` file.

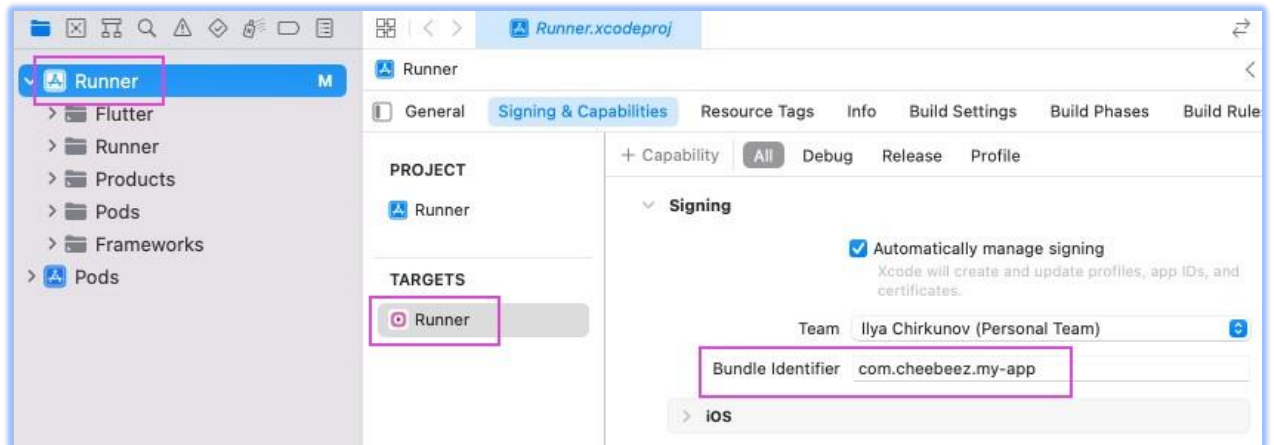


For iOS, change the strings under “CFBundleDisplayName” and “CFBundleName” in the `ios/Runner/Info.plist` file.



# Package Name

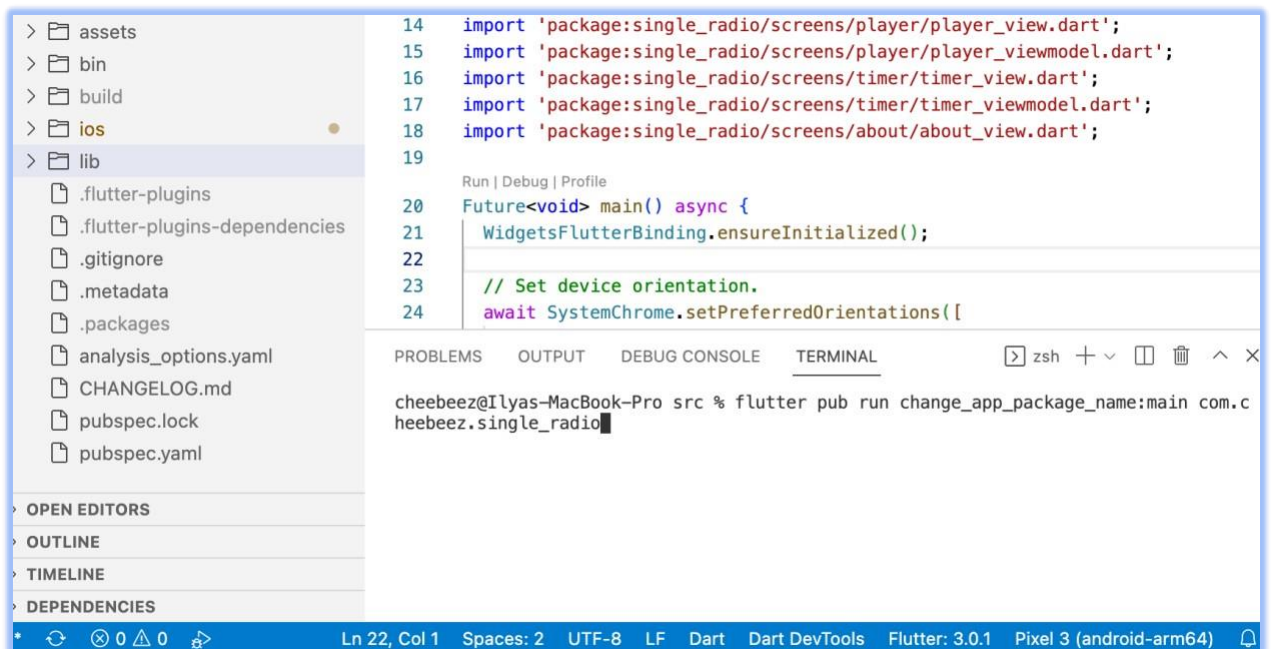
Open the `ios/Runner.xcworkspace` file in Xcode and change the “Bundle Identifier” on something unique. Standard naming procedure: `com.yourdomainname.appname`



For Android, open the terminal (Ctrl+`) and run this command:

**`dart run change_app_package_name:main com.cheebiez.my_app --android`**

Where `com.cheebiez.my_app` is the new package name that you want for your app.

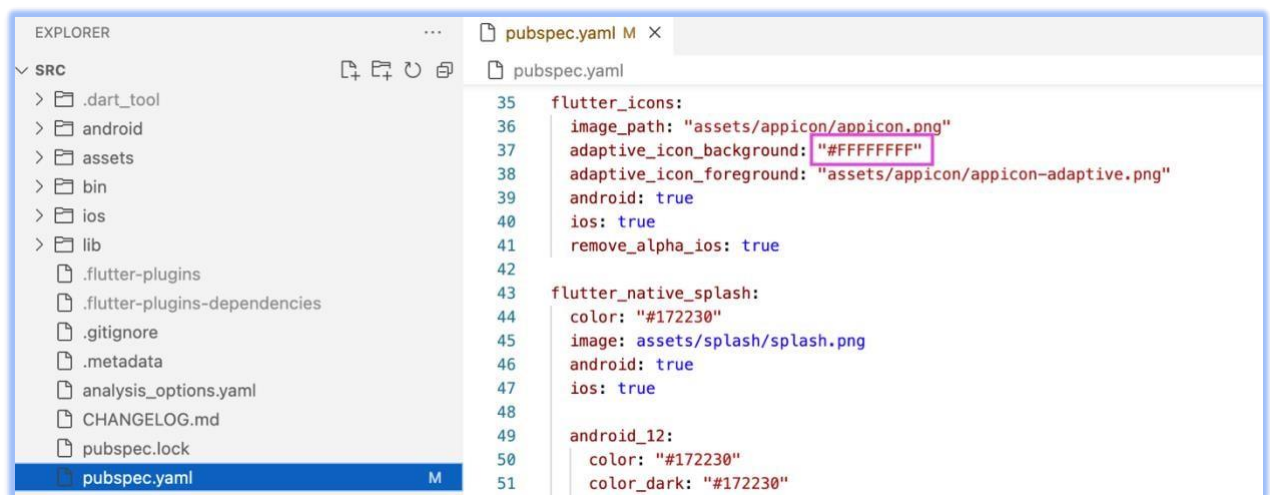


# App Icon

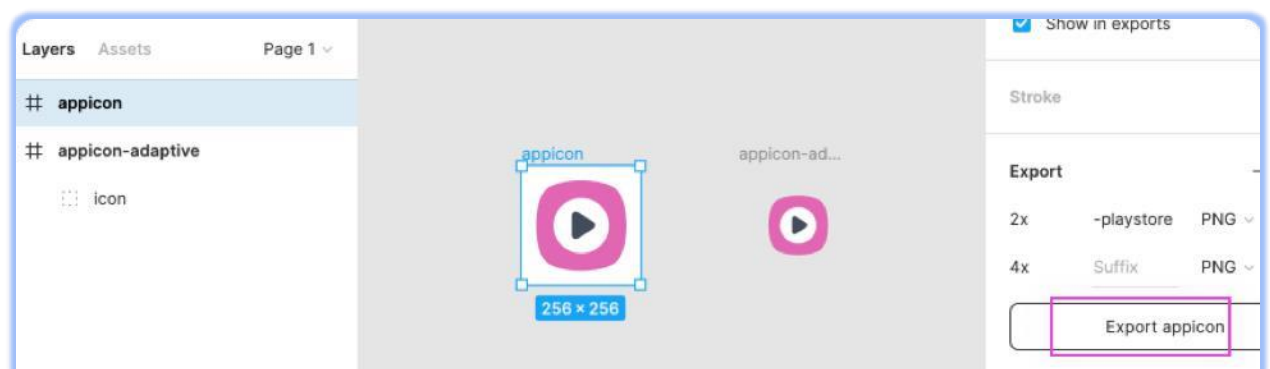
Icons are located in the `assets/appicon/` directory.

- `appicon.png` is a regular 1024x1024 pixel image.
- `appicon-adaptive.png` is a 1024x1024 pixel image with transparent background for the latest Android versions.

The background color of the adaptive icon is set separately in the `pubspec.yaml` file in [ARGB](#) format.



To create icons, you can use the `extra/templates/appicon.fig` which can be edited in [Figma](#).

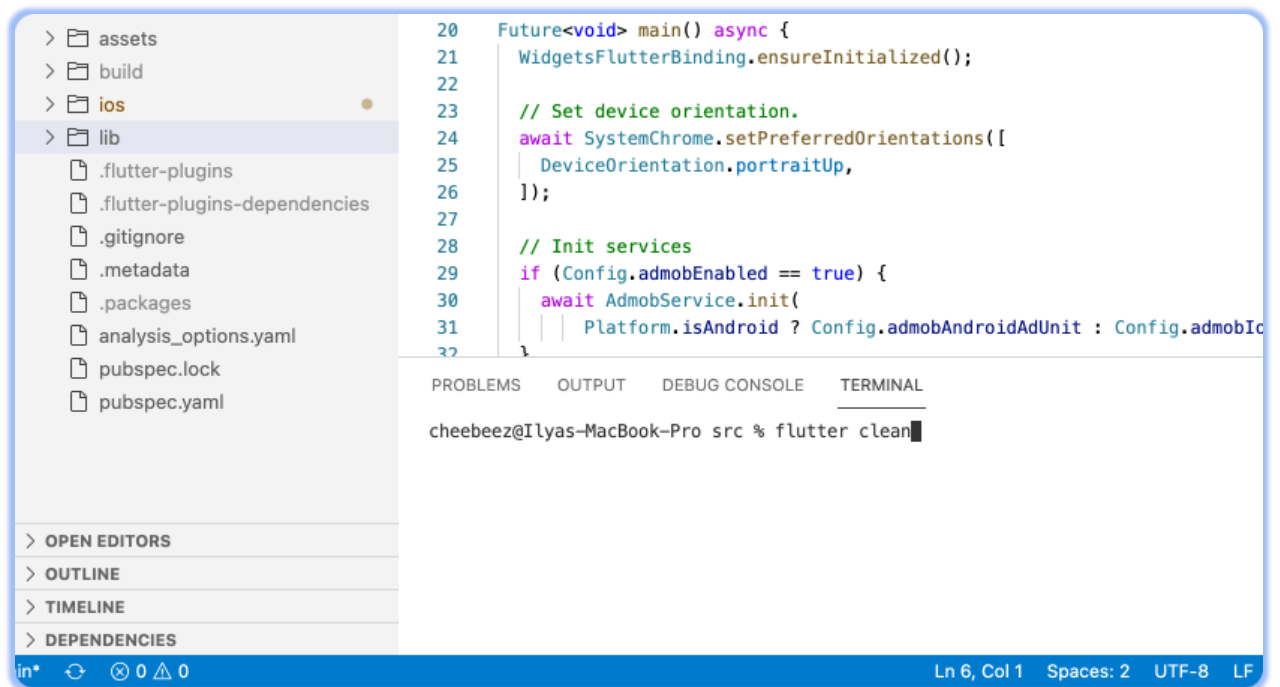


After changing the icons, all that is left to do is open the terminal (Ctrl+`) and run the commands below.

flutter clean

flutter pub get

dart run flutter\_launcher\_icons:main



**Beta:** To enable monochrome icons in Android 13+, add the line

```
<monochrome android:drawable="@drawable/ic_launcher_monochrome"/>
```

to the `android/app/src/main/res/mipmap-anydpi-v26/ic_launcher.xml` file as shown in the screenshot below.



Then you can replace the `android/app/src/main/res/drawable/ic_launcher_monochrome.png` icon with yours.

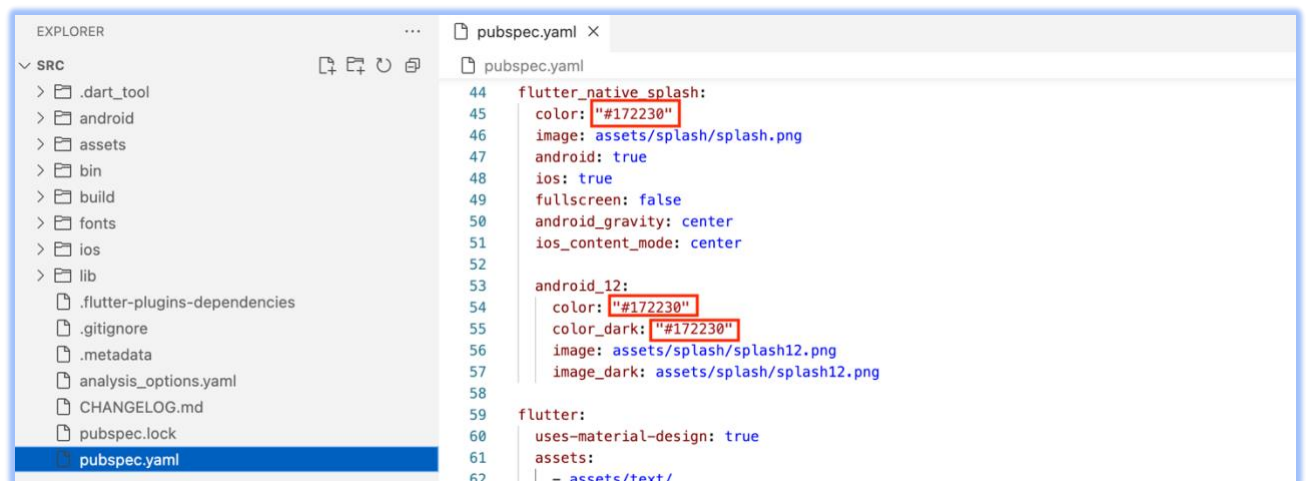
# Splash Screen

Splash screen provide a simple initial experience while your mobile app loads.

As with icons, you need to change images located in the `assets/splash/` directory.

- `splash.png` is the default 728x671 pixel image. This size is chosen for the best compatibility with most devices.
- `splash12.png` is an 1152x1152 pixel image with transparent background for the latest Android versions (12+). The visible part should fit within a circle of 768 pixels in diameter.

The background is set separately in the `pubspec.yaml` file in [RGB](#) format.



To create splash screen, you can also use the `extra/templates/splash.fig` template which can be edited in [Figma](#).

If you want the `splash.png` image to stretch fullscreen, you will need a higher-resolution image (e.g., 2732x2732 pixels) and to update the settings in `pubspec.yaml`.



The `splash12.png` image cannot be stretched fullscreen. This is a system requirement from Google for all apps on Android 12 and newer.

After changing the images, open a terminal (Ctrl+`) and run the commands below.

**flutter clean**

**flutter pub get**













**dart run flutter\_native\_splash:create**

Note: The splash screen may not appear when you launch the app from Visual Studio Code.

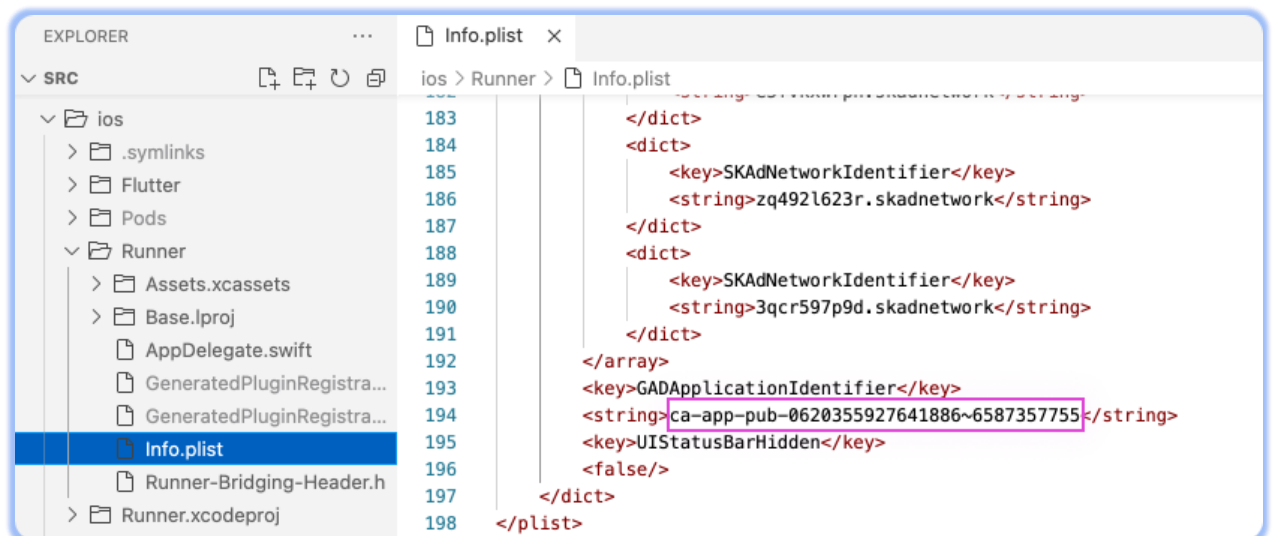
However, it should appear when you launch by clicking on the launch icon in Android.

# Admob

Login to your AdMob account and create apps for your platforms.

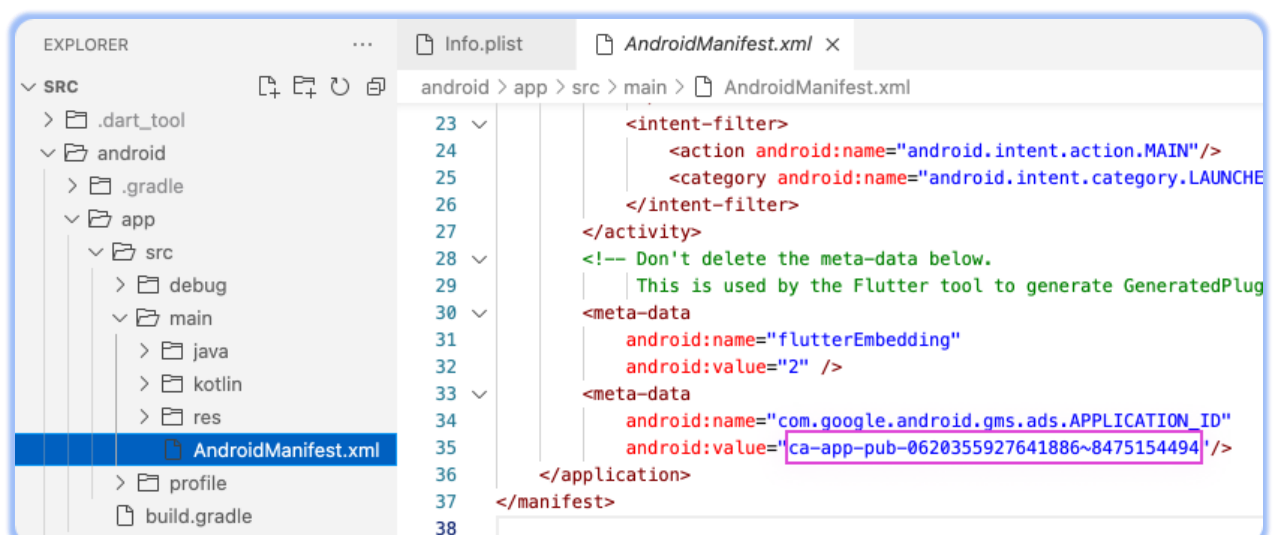
<input type="checkbox"/>	App  	App ID 	Approval status 	Status details 	Shops 	Pac sto
<input type="checkbox"/>	 <b>Soundao</b> Android	 ca-app... 8475154494	 Requires review	Ad serving enabled Add store to prevent interruption	<a href="#">Add shop</a>	—
<input type="checkbox"/>	 <b>Soundao</b> iOS	 ca-app... 6587357755	 Requires review	Ad serving enabled Add store to prevent interruption	<a href="#">Add shop</a>	—

Replace App ID in `ios/Runner/Info.plist` for iOS.



```
183 <dict>
184 <key>CFBundleIdentifier</key>
185 <string>ca-app-pub-0620355927641886~8475154494</string>
186 </dict>
187 <dict>
188 <key>SKAdNetworkIdentifier</key>
189 <string>zq492l623r.skadnetwork</string>
190 </dict>
191 <dict>
192 <key>SKAdNetworkIdentifier</key>
193 <string>3qcr597p9d.skadnetwork</string>
194 </dict>
195 </array>
196 <key>GADApplicationIdentifier</key>
197 <string>ca-app-pub-0620355927641886~6587357755</string>
198 <key>UIStatusBarHidden</key>
199 <false/>
200 </dict>
201 </plist>
```

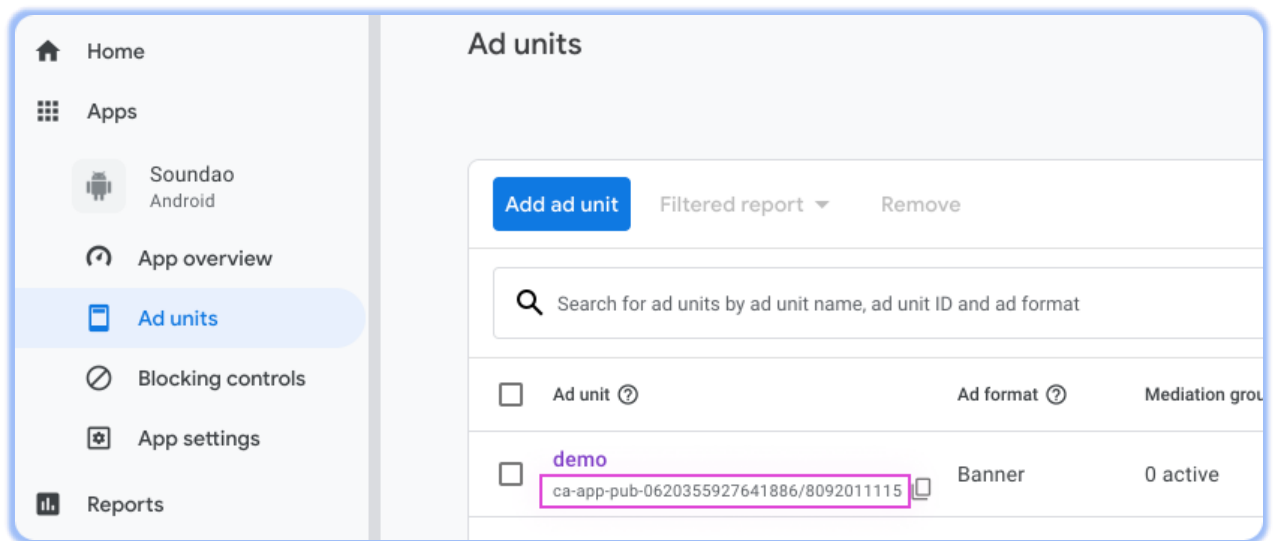
And in `android/app/src/main/AndroidManifest.xml` for Android.



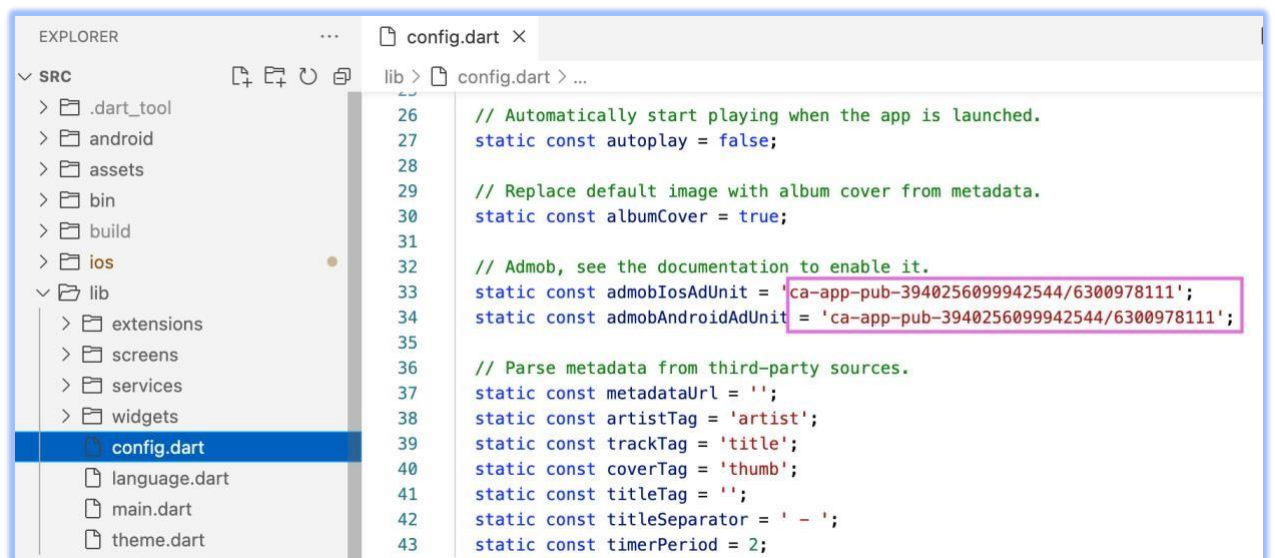
```
23 <intent-filter>
24 <action android:name="android.intent.action.MAIN"/>
25 <category android:name="android.intent.category.LAUNCHER"/>
26 </intent-filter>
27 </activity>
28 <!-- Don't delete the meta-data below.
29 This is used by the Flutter tool to generate GeneratedPluginRegistrant.java -->
30 <meta-data
31 android:name="flutterEmbedding"
32 android:value="2" />
33 <meta-data
34 android:name="com.google.android.gms.ads.APPLICATION_ID"
35 android:value="ca-app-pub-0620355927641886~8475154494"/>
36 </application>
37 </manifest>
38
```



Now you have to create banner Ad units.



Copy these Ad unit IDs to the `lib/config.dart` file.



Open a terminal (Ctrl+`) and run the commands below to activate Admob.

**dart run single\_radio:admob\_enable**

**flutter clean**

**flutter pub get**

To disable it, replace `admob_enable` with `admob_disable`.

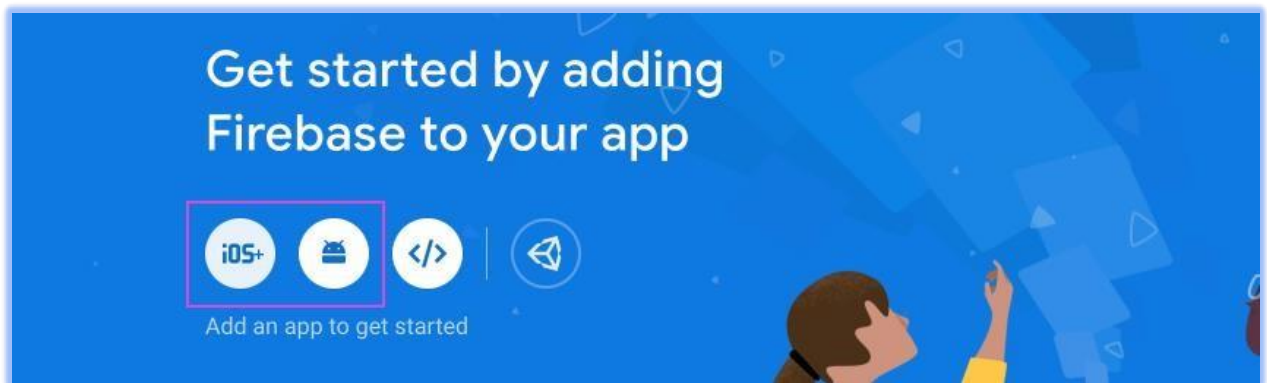


# Firebase Cloud Messaging

Login to your [Firebase Console](#) and create a new project.



Register app for your platforms.



You don't need to make any changes to the source code, these steps are already done. Just enter the correct [package name](#), download and replace the `google-services.json` file in the `src/android/app/` directory and the `GoogleService-Info.plist` file in the `src/ios/Runner` directory.

**4 Next steps**

You're ready!

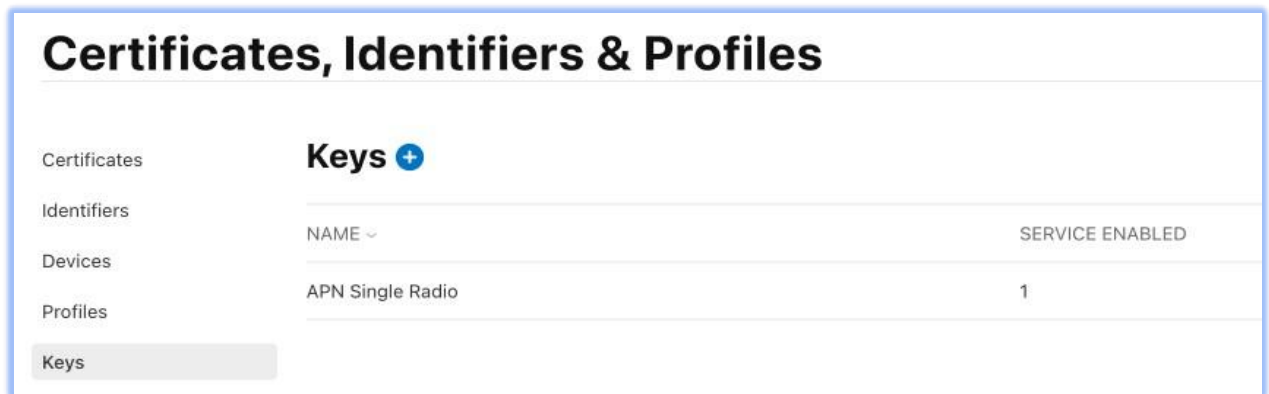
Make sure that you take a look at the [documentation](#) to learn how to get started with each Firebase product that you want to use in your app.

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

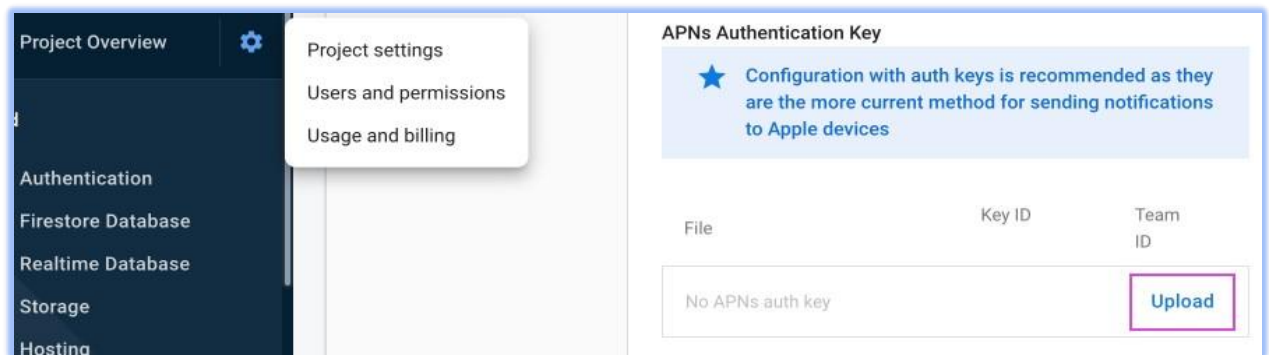
Or, continue to the console to explore Firebase.

[Previous](#) [Continue to the console](#)

For iOS, you will also need to login to your [Apple Developer Account](#) and create a Key file with Apple Push Notifications service (APNs) permission.



Upload this AuthKey file to your firebase project settings under Cloud Messaging.



Open a terminal (Ctrl+`) and run the commands below to activate push notifications.

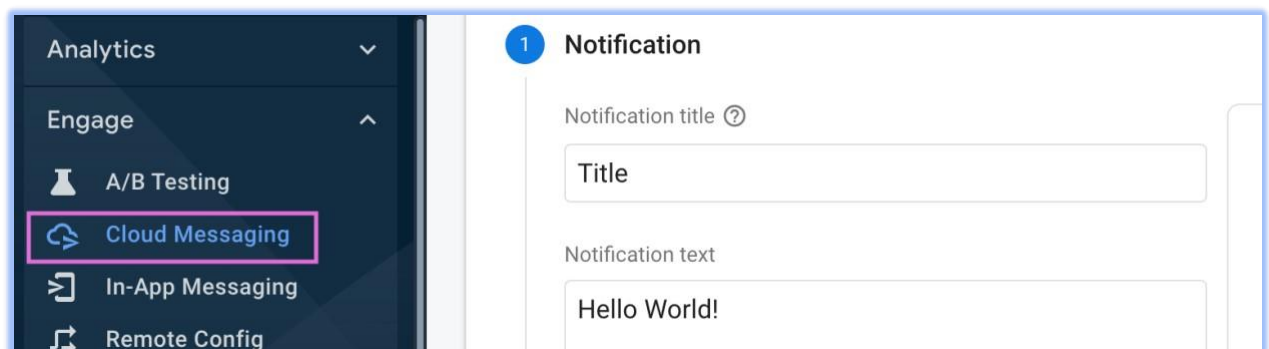
```
dart run single_radio:fcmm_enable
```

```
flutter clean
```

```
flutter pub get
```

To disable it, replace fcm\_enable with fcm\_disable.

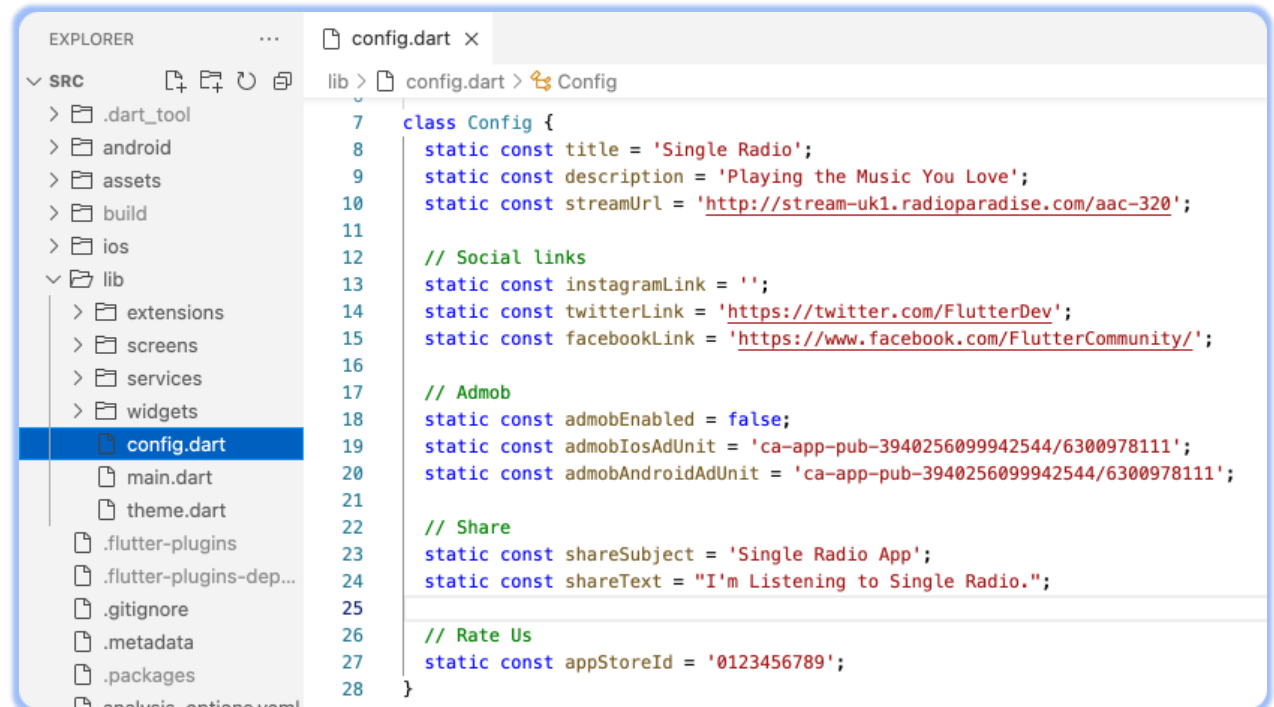
Well done, now you can send your first push notification.



Note: Simulator does not support remote notifications. Please test on a real device.

# Config

In the config file, you can set up your radio station, social links, and more.

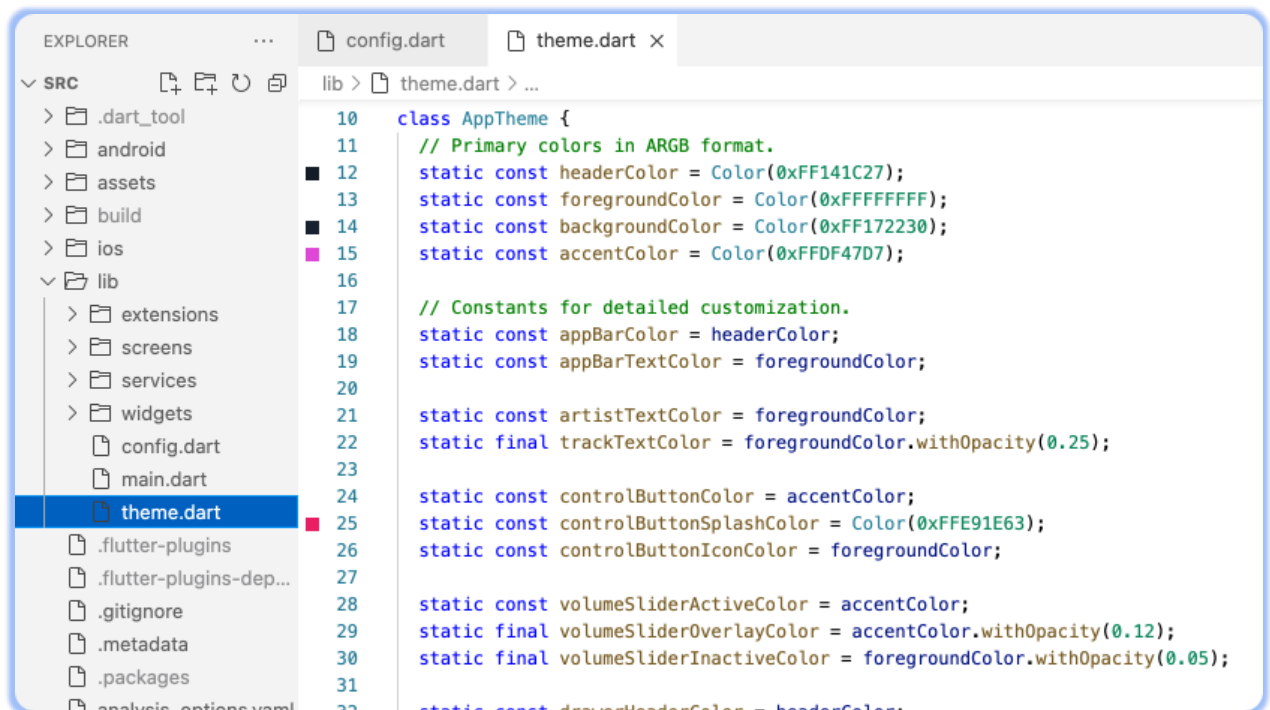


Setting "Rate Us" for Android is not required.

If you don't want to set up social links, leave these fields blank.

# Color Theme

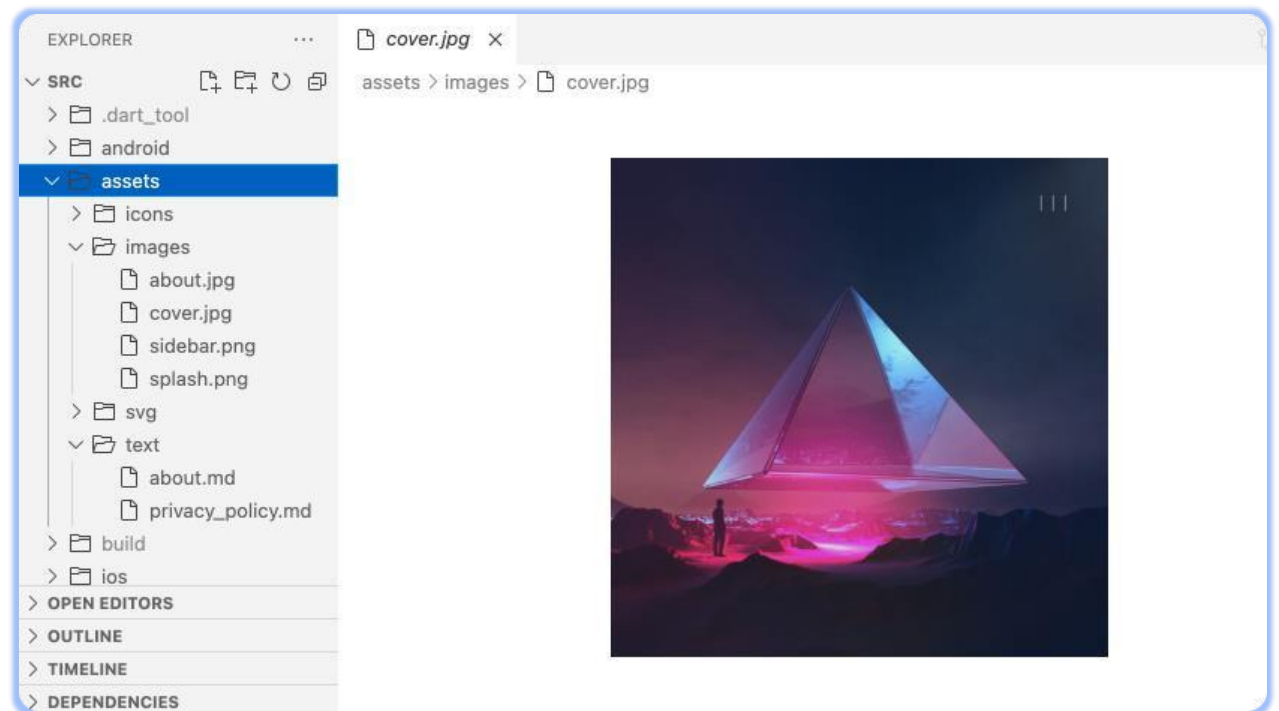
You have the option to customize the color theme. To do this, edit the `lib/theme.dart` file.



In most cases, just change the four primary colors. If you want a more detailed customization, find and edit the constant you need.

# Assets

The images are located in the `assets/images` directory. There, you can change the background image `assets/images/background.jpg`, the default cover image `assets/images/cover.jpg`, the top icon for the sidebar `assets/images/sidebar.png`, and the "About Us" page image `assets/images/about.jpg`.



The assets also contain text files in [Markdown](#) format such as Privacy Policy `assets/text/privacy_policy.md` and About Us `assets/text/about.md`.

# Build the Release

To publish your app on the Google Play, you need to give your app a digital signature. Use the following instructions to sign your app.

On Mac/Linux, run the following command:

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

On Windows, run the following command:

```
keytool -genkey -v -keystore %userprofile%\upload-keystore.jks -storetype JKS -keyalg RSA  
keysize 2048 -validity 10000 -alias upload
```

This command stores the `upload-keystore.jks` file in your home directory. If you want to store it elsewhere, change the argument you pass to the `-keystore` parameter. However, keep the keystore file private.

```
cheebiez@Ilyas-MacBook-Pro src % keytool -genkey -v -keystore ~/upload-keystore.jks -keyalg RSA  
-keysize 2048 -validity 10000 -alias upload  
Enter keystore password:  
Re-enter new password:  
What is your first and last name?  
[Unknown]: Ilya Chirkunov
```

Create a file named `src/android/key.properties` that contains a reference to your keystore:

```
key.properties X  
android > key.properties  
1 storePassword=<password from previous step>  
2 keyPassword=<password from previous step>  
3 keyAlias=upload  
4 storeFile=<location of the key store file, such as /Users/<user name>/upload-keystore.jks>
```

After that you can build the release app:

**flutter clean**

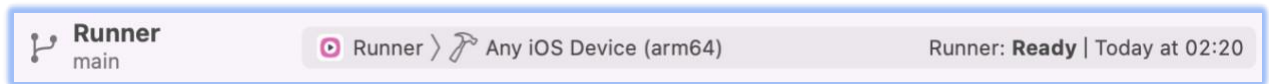
**flutter build appbundle** (or `flutter build apk --split-per-abi`)

**flutter install** (Install an APK on a device)

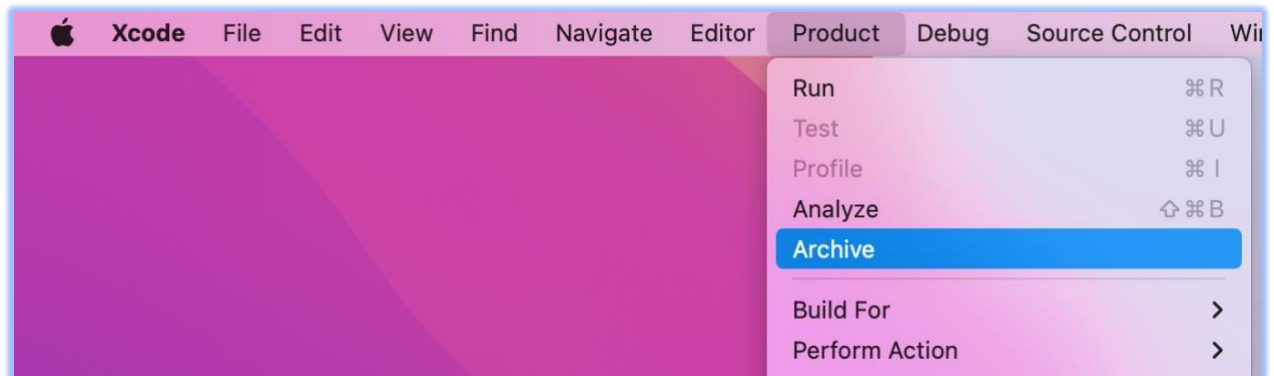
The resulting app bundle or apk files are located in `build/app/outputs` within your app's folder.

Xcode is required to release your iOS app.

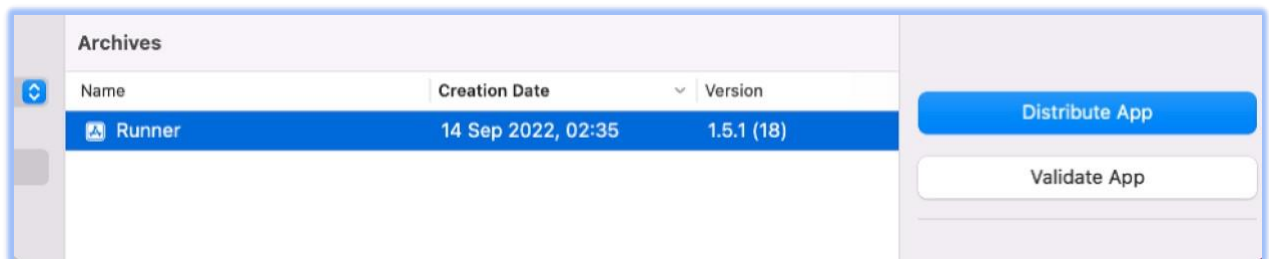
Open the `src/ios/Runner.xcworkspace` file and select "Any iOS Device" at the top.



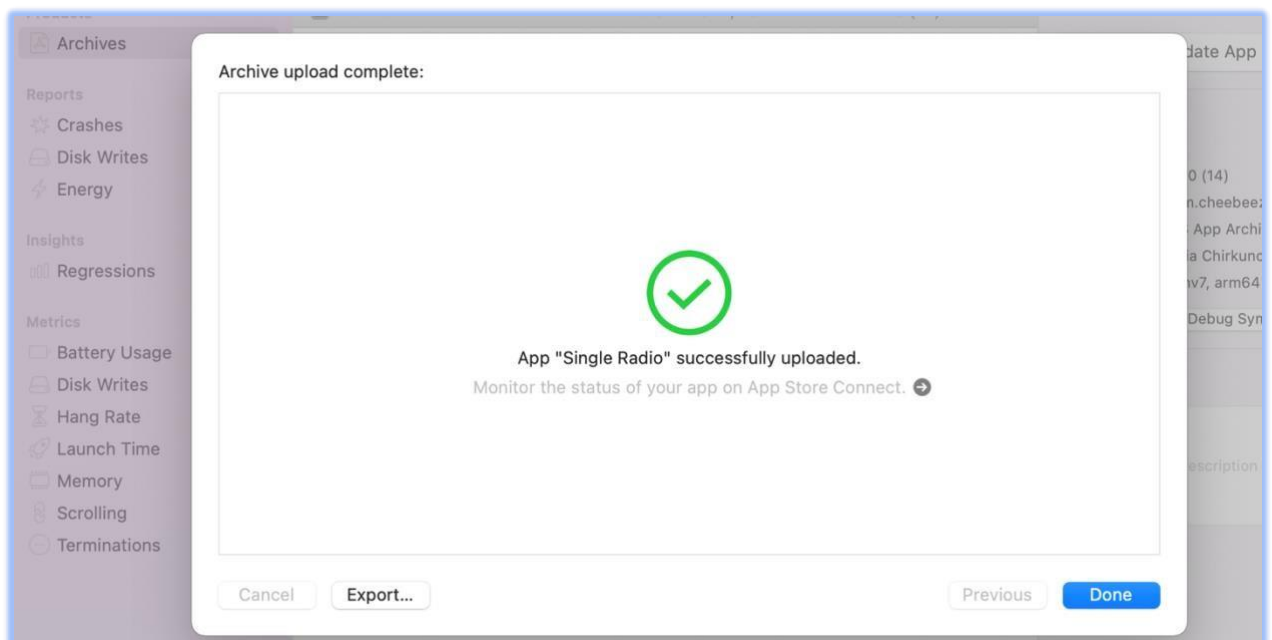
Then in the "Product" tab, select "Archive" and wait for the compilation to finish.



In the window that appears, click "Distribute App". This process can take a long time.



In the end, if you configure the app correctly, it will be successfully uploaded to App Store.



# Known Issues

When building the app on macOS, the following error occurs: “Operation not permitted”

**Cause:** macOS does not trust files downloaded from the internet.

**Solution:** Execute the command to mark the files as trusted: `sudo xattr -c -r <project_folder>`

Can't move the slider in the iOS simulator.

**Cause:** Simulator does not process all events, unlike a real device.

**Solution:** To debug the slider use a real iOS device.



# Support

Once again, I appreciate your decision to purchase this app. If you have any general questions about the apps on CodeCanyon, please feel free to contact me either through the [profile](#) contact form or directly at [support@cheebiez.com](mailto:support@cheebiez.com)