

Introduction

We are following CodeCanyon rules and regulation, so under this licenses you can only publish single product on play store. And you can't sell this source code anywhere. Please follow Standard [Licenses](#)

Home >

Licenses

Overview **Standard** Music SFX Video Photo Specialty

Standard Licenses

The majority of our items (themes, code, graphics, 3D and flash) are covered by our Standard Licenses. If your end product including the item is going to be free to the end user then a **Regular License** is what you need. An **Extended License** is required if the end user must pay to use the end product.

	Regular view full license	Extended view full license
Number of end products	1	1
Use in a single end product	✓	✓
Use in a free end product (more than one end user allowed)	✓	✓
Use in an end product that's sold	✗	✓
On-demand products/services (e.g. "made to order" or "create your own" apps and sites)	One license per each customized end product	✗
Use in stock items/templates (See FAQ for more details)	✗	✗

Installation

Introduction to Flutter

Flutter is Google's UI toolkit for building beautiful, natively compiled applications for [mobile](#), [web](#), and [desktop](#) from a single codebase. It is very easy to learn and currently it is getting more and more popular. With this blog post, you will learn some basic stuff about Flutter and after reading it, you will be able to create a simple application using this technology.

[Click here](#) to check out more details about flutter.

Tools & Setup

Prerequisite

- Flutter & Dart [SDK](#) (We have used Flutter version 1.20)
- Any IDE [Android Studio](#) (Recommended), [Visual Studio Code](#) or [IntelliJ IDEA](#)
- To edit this project you must have Flutter and Dart installed and configured successfully on your computer.
- Set up your editor – Install the [Flutter and Dart plugins](#).
- If you have got Android SDK installed and configured, to install Flutter you only need to:
 - Download Flutter SDK from official website and extract it.
 - Add path to previously extracted SDK to your PATH variable
 - Run flutter doctor tool to check if everything is configured correctly.
 - All above steps are mentioned here: <https://flutter.dev/docs/get-started/install/>

Basic Installation

Important:

Choose your Platform and follow steps to build and run application.

How to Setup Flutter in Android Studio – Windows

- Download Android Studio – <https://developer.android.com/studio/>
- Get the Flutter SDK – <https://flutter.dev/docs/get-started/install>
- Learn more about Android Studio – <https://developer.android.com/studio/intro/>

Step 1 : Get the Flutter SDK

1 Download the following installation bundle to get the latest stable release of the Flutter SDK:

2 Extract the zip file and place the contained flutter in the desired installation location for the Flutter SDK (for example, C:\src\flutter; do not install Flutter in a directory like C:\Program Files\ that requires elevated privileges).

Step 2 : Update your path

If you wish to run Flutter commands in the regular Windows console, take these steps to add Flutter to the PATH environment variable:

From the Start search bar, enter 'env' and select Edit environment variables for your

account.

Under User variables check if there is an entry called Path:

- If the entry exists, append the full path to flutter\bin using ; as a separator from existing values.
- If the entry doesn't exist, create a new user variable named Path with the full path to flutter\bin as its value.

Important:

Note that you have to close and reopen any existing console windows for these changes to take effect.

You are now ready to run Flutter commands in the Flutter Console!

Step 3 : Run flutter doctor

From a console window that has the Flutter directory in the path (see above), run the following command to see if there are any platform dependencies you need to complete the setup:

```
C:\src\flutter>flutter doctor
```

If you find any issue during environment setup, please go online [Click here](#)

How to Setup Flutter in Android Studio – macOS

- Download Android Studio – <https://developer.android.com/studio/>
- Download Xcode – <https://apps.apple.com/us/app/xcode/id497799835?mt=12>
- Get the Flutter SDK – <https://flutter.dev/docs/get-started/install>
- Learn more about Android Studio – <https://developer.android.com/studio/intro/>

Step 1 : Get the Flutter SDK

1 Download the following installation bundle to get the latest stable release of the Flutter SDK:

2 Download SDK and extract downloaded file, just double click on that. and just copy extracted folder and paste it to your desired location (for example, Documents\flutter).

Step 2 : Update your path

Important:

Path variable needs to be updated to access “flutter” command from terminal. you can just update path variable for current terminal window only. and if you want to access flutter commands from anywhere in terminal, we need to update SDK path permanently.

To update PATH variable, we need to open terminal.

To update PATH variable for current terminal window only, then enter this command “export PATH=\"\$PATH:`pwd`/flutter/bin” and hit enter key.

To update PATH variable permanently, then Open or create .bash_profile file. to open or create that file, then enter “sudo open -e \$HOME/.bash_profile” and hit enter key.

Append below line to bash_profile file at bottom of all other content. “export PATH=\"\$PATH:[PATH_TO_FLUTTER_GIT_DIRECTORY]/flutter/bin” as [PATH_TO_FLUTTER_GIT_DIRECTORY] is actual path of SDK folder.

Run this command on terminal “source \$HOME/.bash_profile” to refresh PATH variables.

Then check whether our SDK is successfully installed or not.

You are now ready to run Flutter commands in the Flutter Console!

Run “flutter doctor” into terminal, If you are getting check list of flutter sdk requirements, it means SDK is successfully installed on your machine. and you can start building flutter apps on your machine.

If you find any issue during environment setup in macos, please go online [Click here](#)

How to Setup Flutter in Android Studio – Linux

- Download Android Studio – <https://developer.android.com/studio>
- Get the Flutter SDK – <https://flutter.dev/docs/get-started/install/linux>
- Learn more about Android Studio – <https://developer.android.com/studio/intro/>

Step 1 : Get the Flutter SDK

1 Download the following installation bundle to get the latest stable release of the Flutter SDK:

2 Download SDK and extract downloaded file, just double click on that. and just copy extracted folder and paste it to your desired location (for example, Documents\flutter).

Step 2 : Update your path

Important:

Path variable needs to be updated to access “flutter” command from terminal. you can just update path variable for current terminal window only. and if you want to access flutter commands from anywhere in terminal, we need to update SDK path permanently.

You’ll probably want to update this variable permanently, so you can run **flutter** commands in any terminal session. To update PATH variable, we need to open terminal.

```
export PATH="$PATH:[PATH_TO_FLUTTER_GIT_DIRECTORY]/flutter/bin"
```

1. Run **source \$HOME/** to refresh the current window, or open a new terminal window to automatically source the file.
2. Verify that the **flutter/bin** directory is now in your PATH by running:

```
echo $PATH
```

Verify that the **flutter** command is available by running:

```
which flutter
```

You are now ready to run Flutter commands in the Flutter Console!

Getting Started (Build & Run)

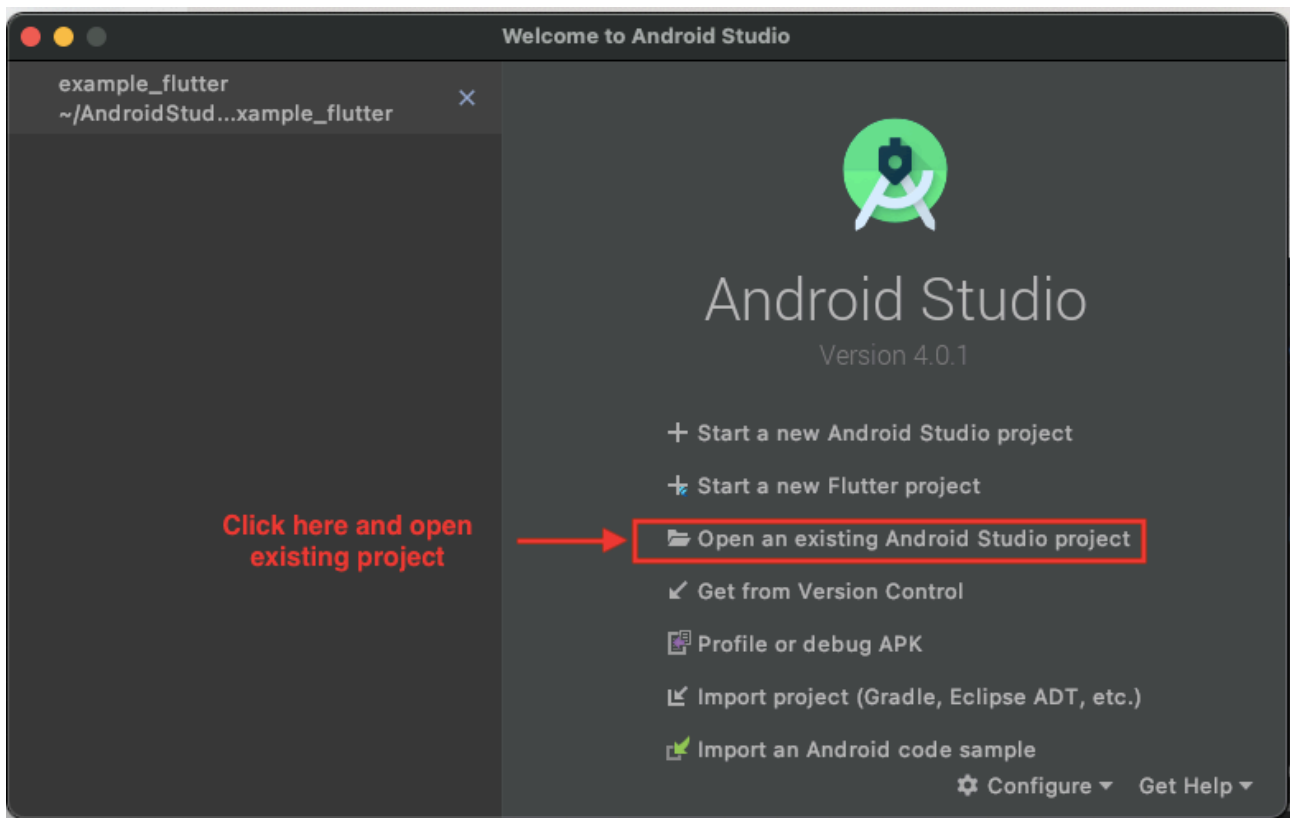
Follow below steps to build and run your application.

Important:

All below steps are must be followed to build and run application

Download Project

Download and find the your project folder, use your preferred IDE (Android Studio / Visual Studio Code / IntelliJ IDEA) to run the project.



Get Dependencies

After you loaded project successfully, run the following command in the terminal to install all the dependencies listed in the pubspec.yaml file in the project's root directory or just click on Pub get in pubspec.yaml file if you don't want to use command.

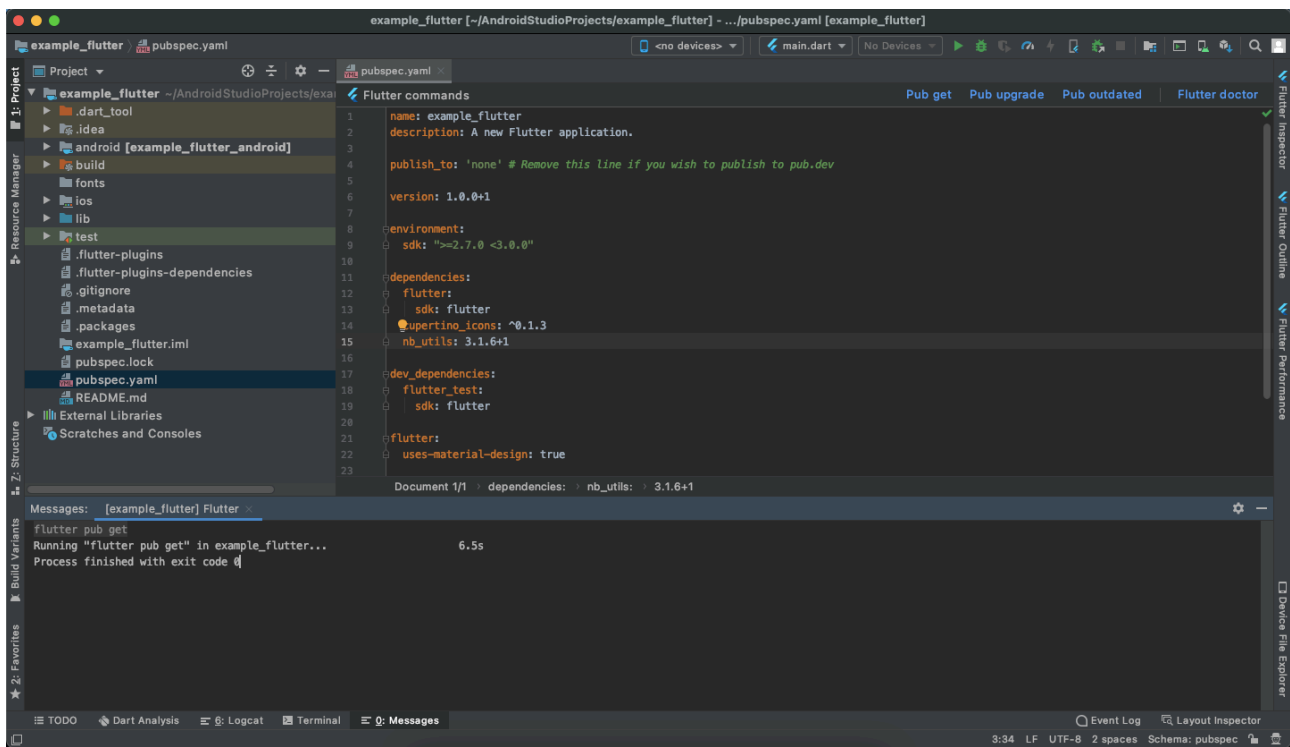
```
flutter pub get
```

DartCopy

Important:

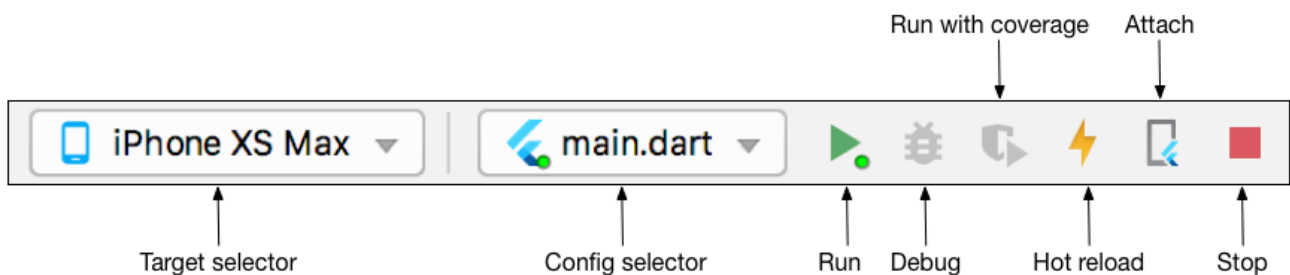
Restart Android Studio if the errors are still persists.

You will see below like screen after you have build your app successfully



Build and Run App

1. Locate the main Android Studio toolbar.
2. In the target selector, select an Android device for running the app. If none are listed as available, select Tools > Android > AVD Manager and create one there. For details, see [Managing AVDs](#).
3. Click the run icon in the toolbar, or invoke the menu item Run > Run.



After the app build completes, you'll see the app on your device.

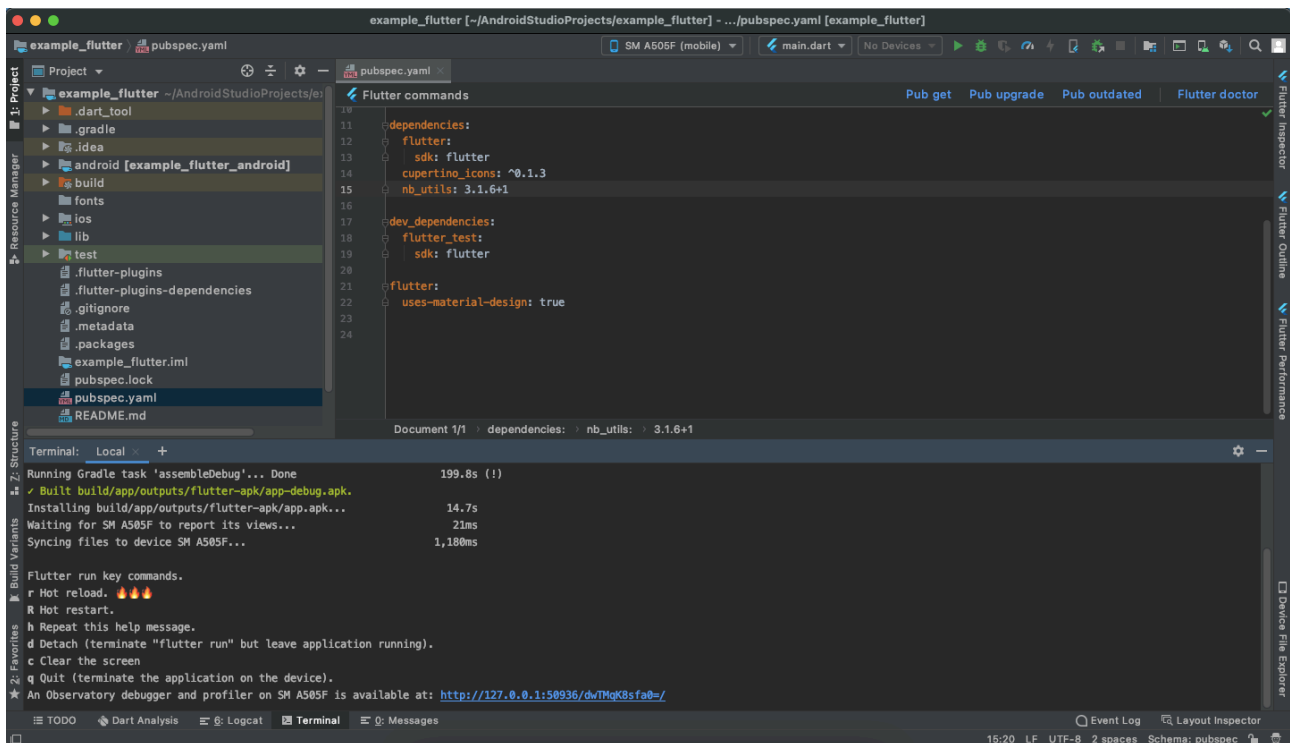
If you don't use Android Studio or IntelliJ you can use the command line to run your application using the following command

Important:

Below step requires flutter path to be set in your Environment variables. See <https://flutter.dev/docs/get-started/install/windows>

flutter run

DartCopy



Try hot reload

Flutter offers a fast development cycle with *Stateful Hot Reload*, the ability to reload the code of a live running app without restarting or losing app state. Make a change to app source, tell your IDE or command-line tool that you want to hot reload, and see the change in your simulator, emulator, or device.

Important:

Do *not* stop your app. let your app run.

Download and configuring Flutter and building the Themes Flutter

Configuration

Android Configuration

Open Android module in Android Studio

1. Open Android Studio.
2. Select Open an existing Android Studio Project.
3. Open the android directory within your app.
4. Wait until the project has been synced successfully. (This happens automatically once you open the project, but if it doesn't, select Sync Project with Gradle Files from the File menu).
5. Now, click on Run button.

Change Application Name

1. You must want to change your application name. This is how you can do. Follow the below step.
2. Open `/android/app/src/main/AndroidManifest.xml` and specify your application name.

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE" />

    <application
        android:name="io.flutter.app.FlutterApplication"
        android:label="YOUR_APPLICATION_NAME"
        android:icon="@mipmap/ic_launcher">
        <activity
```

Change Application Icon

1. See [How to generate an application icon?](#)
2. Browse your image and click on Download icon. After successfully generated, replace all icons in respective folders:
 - `/mipmap-hdpi` in `/android/app/src/main/res/` folder
 - `/mipmap-mdpi` in `/android/app/src/main/res/` folder

- `/mipmap-xhdpi` in `/android/app/src/main/res/` folder
- `/mipmap-xxhdpi` in `/android/app/src/main/res/` folder
- `/mipmap-xxxhdpi` in `/android/app/src/main/res/` folder

Important:

Application icon name must be `ic_launcher`

Change Application ID

1. Follow the below steps to change you Application ID.
2. Open `/android/app/build.gradle`

```
defaultConfig {  
    applicationId "YOUR_APPLICATION_ID"  
    minSdkVersion 21  
    targetSdkVersion 29  
    versionCode 1  
    versionName "1.0.0"  
    testInstrumentationRunner  
"androidx.test.runner.AndroidJUnitRunner"  
}
```

How to build Themes Flutter in Android

Video Content:-

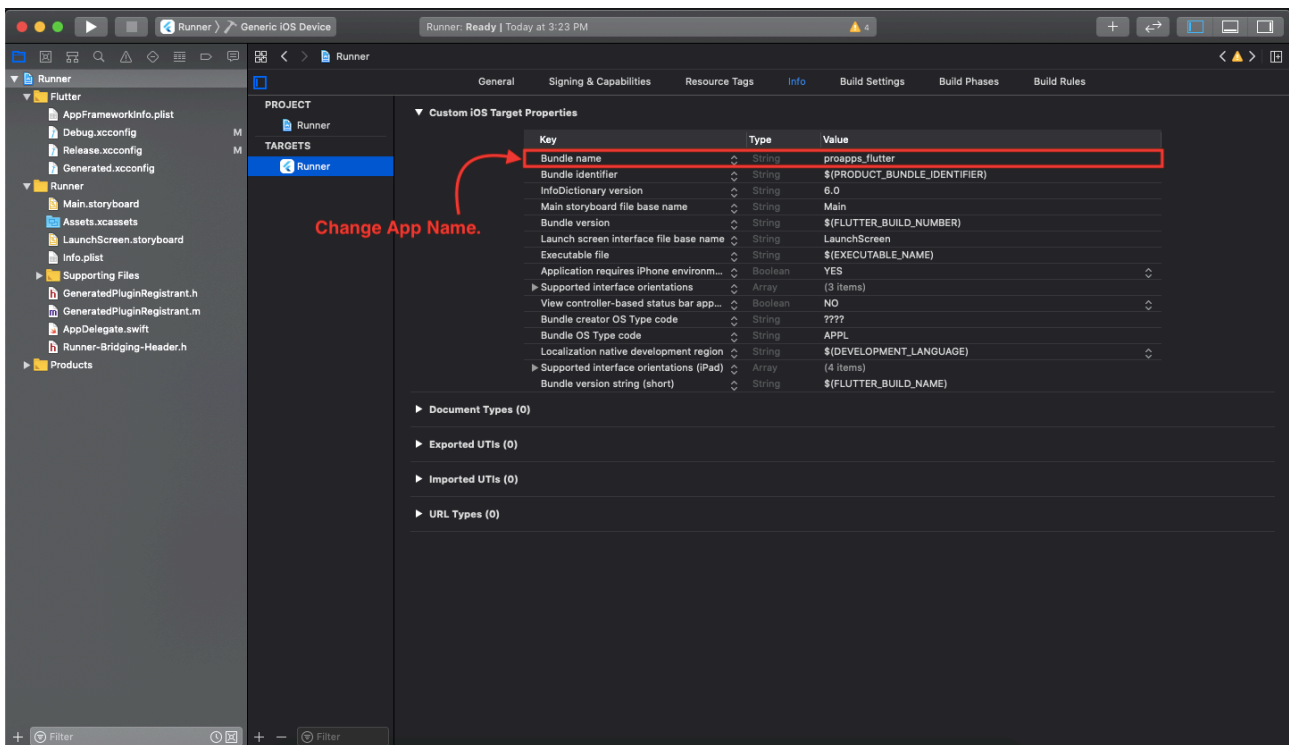
- Introduction
- Open the Android module from the Flutter app
- Changing Application ID
- Changing the Application Version Code and Version Name
- Changing Application Name
- Generating Sign APK

iOS Configuration

Change App Name

Open your project in Xcode.

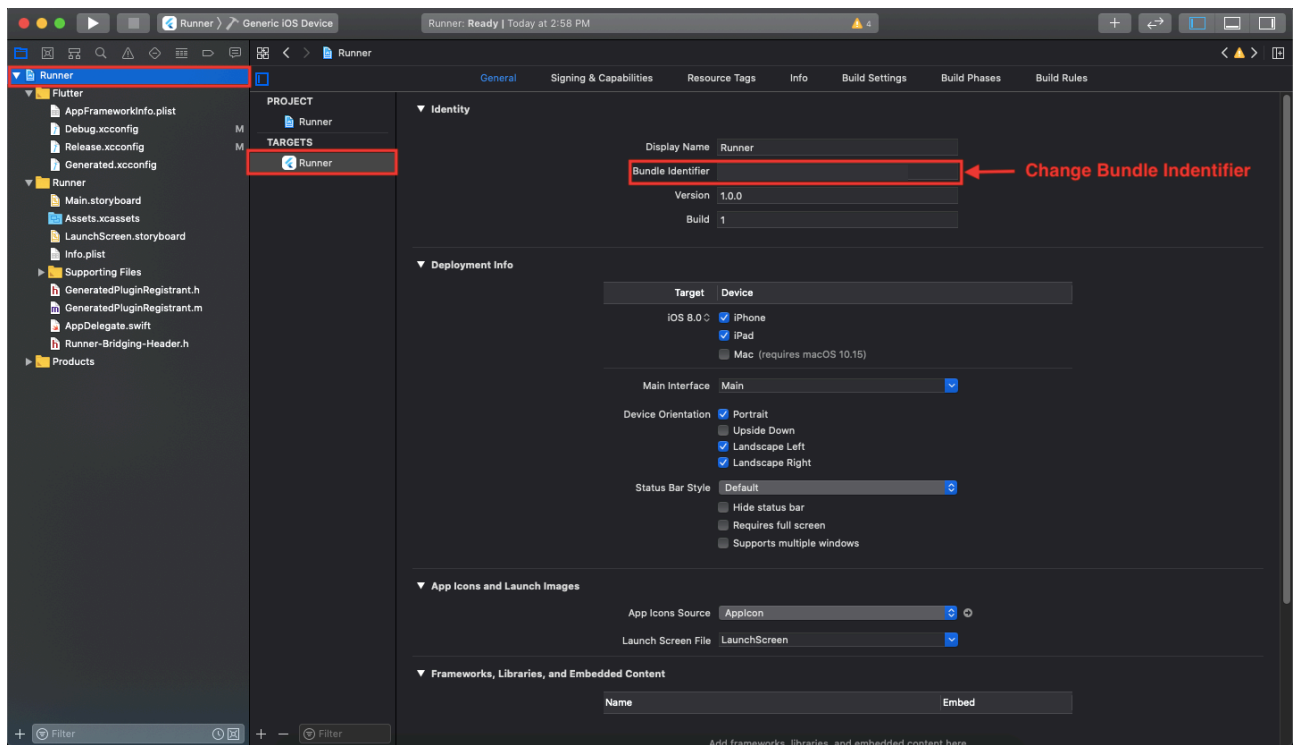
1. Select your project file icon in Group and files panel.
2. Then Select Target -> Info Tab.
3. At last change Bundle Name.



Change Bundle Identifier.

Bundle Id is a unique Identifier of your of app on iOS and MacOS. iOS and MacOS use it to recognise updates to your app. The identifier must be unique for your app.

1. Select your project file icon in Group and files panel.
2. Select General Tab.
3. After Select General tab you can see the details of your application.
4. In Identity section, rename your Bundle identifier.



Change App Icons

1. see [How to Generate App Icons?](#)
2. In Group and files panel find “Assets.xcassets” folder.
3. In Assets.xcassets folder replace AppIcon.

How to build Themes Flutter in iOS

Video Content:-

- Introduction
- Set path of iOS folder and pod install to get dependencies
- Open project in Xcode
- Changing Bundle Identifier
- Changing the Application Version Code
- Changing Application Name in info.plist file
- Changing the Application Icon
- Run Application

Web Configuration

Setup

Run the commands to use the latest version of the SDK from the beta channel and enable web support (hit command one by one).

```
$ flutter channel beta  
$ flutter upgrade  
$ flutter config --enable-web
```

Warning:

Running flutter channel beta replaces your current version of Flutter with the beta version and after this, running the flutter upgrade upgrades your install to the latest beta. Returning to the stable channel (or any other) requires calling flutter channel explicitly

Adding Web support in existing app

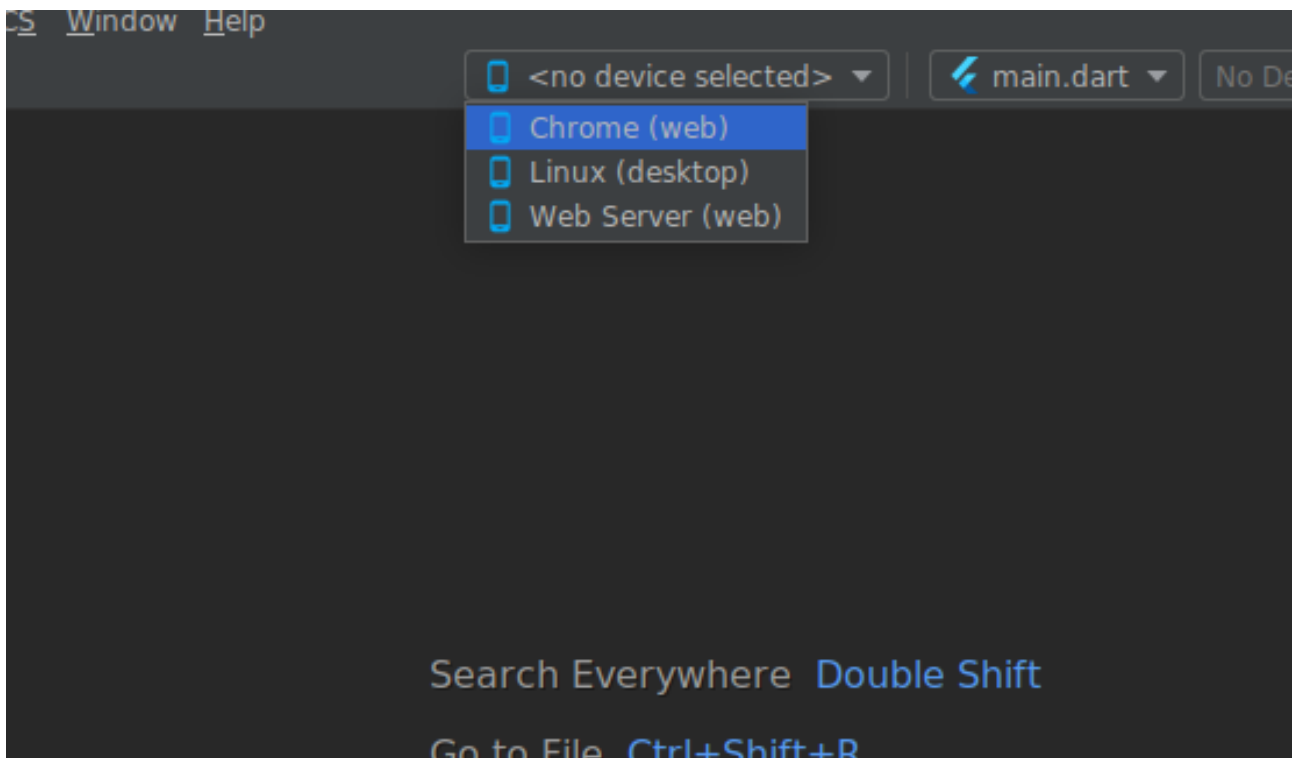
To Web support to the existing app project, run the command in a terminal from the root directory.

```
$ flutter create .
```

Important:

After forming this command restarts the Android studio.

Run Project



Select Chrome or any preferred browser and run the project.

Firebase Configuration

Create a Firebase project

Before you can add Firebase to your Flutter app, you need to create a Firebase project to connect to your app. Visit [Understand Firebase Projects](#) to learn more about Firebase projects.

Important:

If you're releasing your Flutter app on both iOS and Android, register both the iOS and Android versions of your app with the same Firebase project.

Visit for more information on [how to setup for Android](#)

Visit for more information on [how to setup for iOS](#)

Register your app with Firebase

- In the center of the [Firebase console's project overview page](#), click the Android icon to launch the setup workflow.
- Enter your app's package name in the Android package name field.

Important:

Make sure that you enter the ID that your app is actually using. You cannot add or modify this value after you register your app with your Firebase project. Both applicationId should be matched.

- Click Register app.

Add a Firebase configuration file

- Click Download google-services.json to obtain your Firebase Android config file (google-services.json).
- Move your config file into the android/app directory of your Flutter app.

Follow the instruction for enabling Firebase services in your Android app [here](#)

Upload iOS App on AppStore

Assemble App Store Information

- It is very important to collect all of the information you need to submit before you start the process.
- Like, Screenshots, Name, Description, Keywords, Support URL, Marketing URL, Privacy Policy URL, App Icon, Copyright, Demo Account etc...
-

Create the Bundle Identifier

- In your browser, navigate to Apple's [Developer Portal](#).
- Login
- Click Identifiers
- Click the "+" in the top right of the screen
- Name the App ID the same as your app
- Make sure the Bundle ID follows the standard naming convention: com.yourcompanyname.yourappname
- Check any App Services the app needs, click Continue
- Verify the services are correct, then click Submit

Create a Signing Certificate Request

- Certificate Signing Requests are used to link your computer to your Apple developer account.
- Open the program called Keychain Access

- In the top left menu, click Certificate Assistant
- Click Request a Certificate From a Certificate Authority...
- User email: Enter your email address
- Common Name: We recommend using the name of the app
- CA email is not actually required
- Check Saved to Disk, click Continue
- Save the Certificate Signing Request for later

Create an App Store Production Certificate

- Code Signing Certificates are used to link iOS apps to your Apple developer account.
- In your browser, navigate to Apple's Developer Portal
- Click Certificates
- Click the "+" in the top right of the screen
- Click the App Store Production
 - Currently named "App Store and Ad Hoc" under Production
- Click Continue
- Upload the Certificate Signing Request created earlier
- Download the Certificate
- Install the certificate on your computer by double-clicking
- Keep this file somewhere safe

Create a Production Provisioning Profile

- In your browser, navigate to Apple's [Developer Portal](#)
- Log in
- Click Provisioning Profiles
- Click the "+" in the top right of the screen
- Click the App Store Distribution option, click Continue
- Select the Bundle ID created earlier, click Continue
- Select the Certificate created earlier, click Continue
- Make sure the Profile Name follows the standard naming convention: App Name App Store Distribution
- Click Generate
- Download it
- Install the provision profile on your computer by double-clicking
- Keep this file somewhere safe

Create a App Store Listing

- Start in your browser, navigate to [iTunes Connect](#)
- Log in
- Click My Apps
- Click the "+" in the top left of the screen
- Click "New iOS App"
- Bundle ID: choose the one created on Developer Portal

- For convenience, make the SKU match the Bundle ID created earlier
- Click Create to create the first version listing

Make The Release Build

- Start Xcode
- Open the project or workspace
- Update the version and build numbers
- Open Build Settings
- Make sure All settings is selected
- Scroll to Code Signing
- Use the provisioning profile created earlier
- Use the code signing identity created earlier
- In the top menu, select Generic iOS Device as the build destination if no actual device is connected
- Menu, Project, Archive
- Click Distribute
- Sign in as your apple developer account
- Submit to app Store
- Wait for the confirmation

Fill in Version Information

- Enter all of the information assembled in step 1
- Select the build you wish to tie to the version (Note: This usually appears 10-15 minutes after sending from Xcode)
- Fill in pricing information
- Click Save
- Fix any errors based on the messages that appear and save changes

Submit Version For Review

- Select the release type
Manual release – after the app is accepted, a Release button will appear
Automatic release – as soon as the app is accepted, it will be released
- Wait for approval
This process usually takes 2-3 weeks for a new submission
- If your app was not approved, review the notes in the Resolution Center and make any necessary changes to the app or version information then resubmit

Release

- If you selected Manual release, click Release your app when ready
- Wait for the green “Ready for Sale” version status (This might take from 1 hour to 1 day after release)
- Search for your app on the App Store
- Celebrate!

Common errors

List of common issues

Finished with non-zero exit value 1

In case, you are getting this error, there could multiple reasons for this. To solve this error, we have covered all the possible solutions here.

Clean & Re-build project

1. Open the terminal in Android studio.
2. Enter flutter clean command in terminal.
3. After finish processing, hit second command flutter pub get.
4. Run your project.

Removing Some local file and folder in your project

1. Open your project folder and show all hidden files and folder.
2. Then removing .dart_tool folder, .idea folder, build folder, .packages, .pubspec.lock files in project root directory.
3. Open android folder and removing .gradle folder, .idea folder, local.properties files.
4. At the end, restart your project and then open pubspec.yaml file click in to pub get and run.

Recheck your full project's path

- Make sure that your full project path is less than 80 characters.

Removing “dart.js”

- If you are using ‘import dart:js’ in any .dart file, kindly remove it.

Disable antivirus

- Turn off if anti virus is on, before build.
- Anti virus disable in Windows.
 1. Select Start > Settings > Update & Security > Windows Security > Virus & threat protection > Manage settings (or Virus & threat protection settings in previous versions of Windows 10).
 2. Switch Real-time protection to Off. Note that scheduled scans will continue to run.
- Anti virus disable in mac.

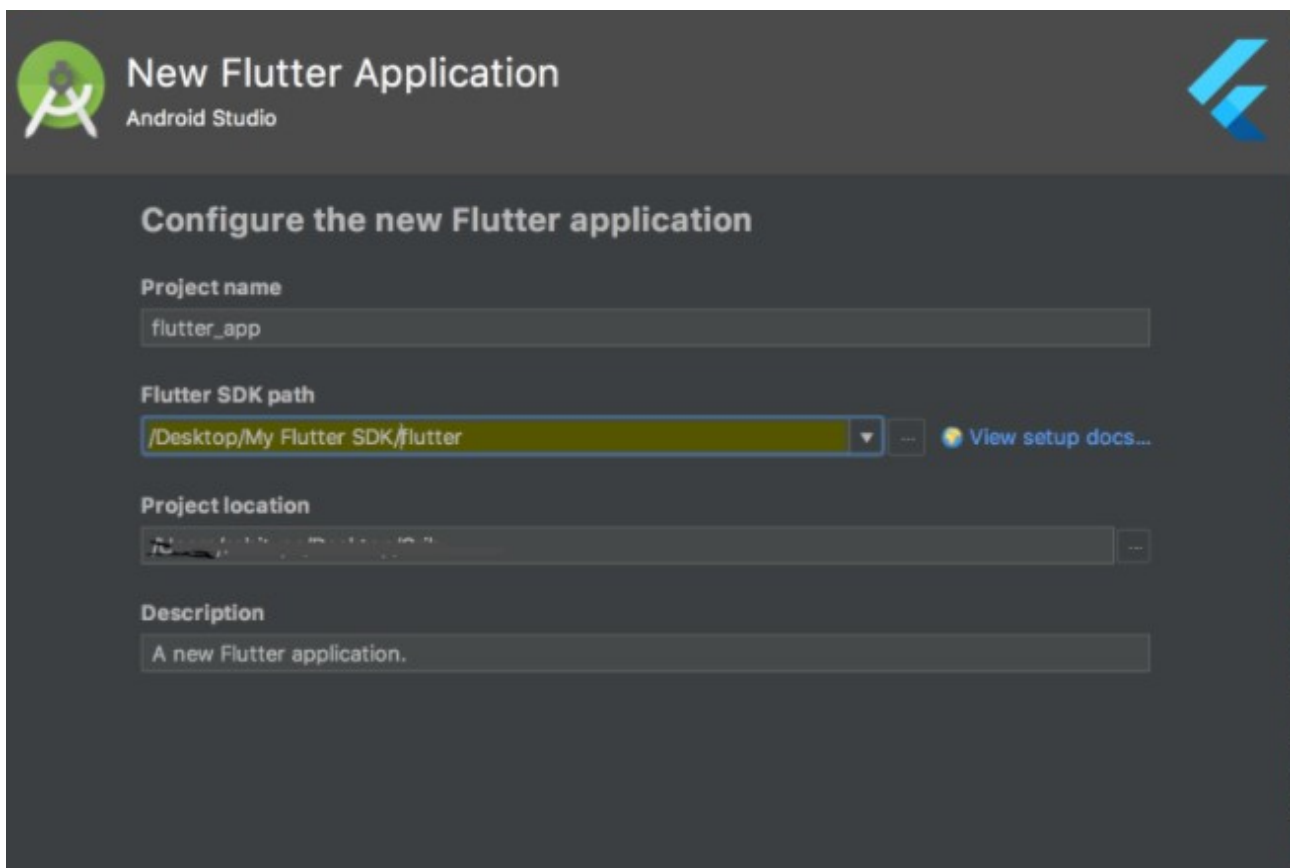
- Access the Mac menu bar, and there you should see the antivirus icon.
- Right-click it or just left click and a drop-down menu appears.
- Click the ideal function such as Disable or Stop and if it asks for any confirmation, hit “Yes” or “Confirm.”.

Flutter SDK Not Available

Download the SDK and point the SDK folder path in your future projects. There are different sources you can try

1. You can clone it from the [Github Repository](#)
2. [Download SDK zip](#) file + extract it after downloading
3. You can also Download [any version](#)(including older) from here (For Mac, Windows, Linux)

Use the SDK path in your future projects



Important:

Restart Android Studio if the errors are still persists.

Dependency Error

When you run the application and you see error like this

```
Running "flutter pub upgrade" in YOUR PROJECT NAME Themes Flutter...  
The current Flutter SDK version is 1.17.5.  
  
Because YOUR PROJECT NAME Themes Flutter depends on flutter_svg  
>=0.18.0 which requires Flutter SDK version >=1.18.0-6.0.pre <2.0.0,  
version solving failed. pub upgrade failed (1; Because YOUR PROJECT  
NAME Themes Flutter depends on flutter_svg >=0.18.0 which requires  
Flutter SDK version >=1.18.0-6.0.pre <2.0.0, version solving failed.)  
Process finished with exit code 1
```

Error Possibilities:

Generally, this error occurs when

1. one of your third party package version is not supported.
2. your Flutter SDK version not matched with one of your third party packages. for ex, your current Flutter SDK version is 1.20.0 and third party package targets Flutter SDK version 1.17.0.

Solution:

- look for the latest version of your package in https://pub.dev/packages/YOUR_PACKAGE_NAME
- or upgrade your flutter version if you don't have the latest with flutter upgrade command.

After implementing the above solution, run the below command in the terminal.

IPHONEOS_DEPLOYMENT_TARGET Out Of Range

Important:

Flutter does not support iOS versions lower then to 8.0.

Change deployment target version.

1. You want to change the target version then In the project Editor. select your app target and go to the General tab.
2. Under the Deployment Info change the target version whatever you versions want to your support.
3. after changing the target version removing the pod folder and podlock.file and reinstall the pod and clean your project and then run your project.

Important:

Flutter does not support iOS versions lower then to 8.0.

Change deployment target version.

1. You want to change the target version then In the project Editor. select your app target and go to the General tab.
2. Under the Deployment Info change the target version whatever you versions want to your support.
3. after changing the target version removing the pod folder and podlock.file and reinstall the pod and clean your project and then run your project.

No matching client found for package name

- The error is “package_name” in google-services.json is not matching with your “applicationId” in app gradle.
- Just make sure the **package_name** and **applicationId** both are same.

Unsupported gradle version 3.*.*

Solution:

1. Open project_root_directory/android in Android Studio
2. Wait for indexing
3. Now run your application from android module

Help and Support

Faced a problem? Need assistance with the product? No worries – our customer support team is always ready to help you.

- Support requests are being processed on business days from 9:00 to 18:00 (GMT +05.30) [generally] within 24h to 48h in the order they were received.
- We suggest, while our team reviews your support request, please read the documentation that comes in the zip file of Codecanyon. You can download it from ThemeForest: <https://codecanyon.net/downloads>
- We are in GMT+5:30 timezone. We address all the support queries 6 days weekly (Sunday off).
- If any support ticket has no response from the item owner for 7 days, the ticket will be considered closed. If you need further assistance you can create another ticket or drop us an email asking to re-open the ticket for you.
- If you like our product and support then please drop a rate and write a review at: <https://codecanyon.net/downloads>