Documentation

Rocket LMS Flutter Mobile App

Table of contents

- Introduction

- Included Features

- Server Requirements

- Prerequisites

- Requirements

- Support & contact

- Connecting the mobile app to the web version

- Changing package name

- Translation

- Changing app name

- Changing app colors

- Changing splash screen logo

- Changing app icon

- Generating APK file

- Publishing on App Store and Play Store

Introduction

The Rocket LMS mobile app is a Flutter-based application designed for use with Rocket LMS, an online course marketplace. Rocket LMS simplifies the management of your online education business.

The mobile app utilizes the Rocket LMS web version's data, enabling users to access the platform conveniently from their smartphones. This enhances the efficiency of your online education business by providing users with access to learning materials, live classes, quizzes, and course management tools anytime, anywhere, without constraints of time or location.

Included Features

We have incorporated a comprehensive array of features into the Rocket LMS mobile app. Here's a list of the included features:

- Flutter technology for optimal performance.

- Clean design for intuitive navigation.

- Support for live classes, video courses, and text courses.

- Online quizzes with instant results.

- Meeting booking system for scheduling classes or appointments.

- Easy customization to tailor the app to your specific needs.

- Multilanguage and multi-currency support for global accessibility.

- Micro-interactions and animations for an engaging user experience.

- Financial and sales reports for tracking performance.

- Notifications and notices to keep users informed.

- Advanced learning page for easy course discovery.

- Blog posts to provide additional resources and updates.

- Gamification badges to incentivize learning progress.

- Categories and trend categories for organized content browsing.

- Integration with Google and Facebook for seamless login.

- Integration with Google firebase.

- Support for video content from YouTube and Vimeo.

- Reviews and comments to foster community interaction.

- Support and chat system for user assistance.

- Subscription options for premium access.

- Discounts and coupons for promotional campaigns.

- Special offer countdown feature to create urgency.

- Featured courses section for highlighting top content.

- Flexible course pricing plans to suit different needs.

- Special native video player for optimized viewing experience.

- IP management system for enhanced security.

- Offline mode system for seamless access to content without an internet connection.

- Ability to submit notes for course materials for personalized learning.

- Support for Firebase notifications to keep users informed and engaged.

- Report system for tracking user activity and performance.

- Certificate management to issue certificates upon course completion.

- Organizational education features for corporate training.

- Prerequisites system to ensure sequential learning.

- Grid and list layout options for customized viewing.

- Quiz review system to revisit completed quizzes.

- Quiz statistics for tracking performance.

- No network page to guide users during internet outages.

- Course wishlist for saving preferred courses.

- Comments and reviews section for user feedback.

- Separate dashboards for different user roles for personalized experiences.

- Learning progress tracking to monitor individual advancements.

- Referral system to incentivize user referrals.

- Offline payments for convenient transactions.

- Payout process management for instructors.

- Instructor profile pages for showcasing expertise.

- Ability to add courses to reminders for timely access.

- Share courses feature.

- Advanced cart functionality for smooth transactions.

- Course and instructor search for easy navigation.

- Course filters to refine search results.

- RTL & LTR Support for diverse language preferences.

- In-app PDF viewer.

- Financial approval process management.

- Account settings for user customization.

- Animated onboarding screens for user engagement.

- Error handling mechanisms for seamless user experience.

Server Requirements:

Please ensure that the server meets the following requirements:

- **PHP Version 8.1**

- **Enable and disable specific PHP extensions according to the instructions in the screenshot.**

Prerequisites:

Rocket LMS mobile app uses these prerequisites as the engine. In order to run your mobile app, you need to provide these requirements:

1-Rocket LMS v1.9.6 (Web version)

2-Rocket LMS plugins bundle v.1.9.6

Note: Rocket LMS mobile app won't work without these requirements.

Required Software and libraries:

To export a Flutter app in PC, you need the following:

Android Studio Jellyfish:

Android Studio is the official integrated development environment (IDE) for Android app development. Install Android Studio on your PC.

Flutter SDK (v3.22.0):

Install the Flutter SDK on your PC. You can download it from the official Flutter website and follow the installation instructions for your operating system.

Flutter Plugin for Android Studio:

Once you have Android Studio installed, you need to install the Flutter plugin.

This plugin provides tools for Flutter development within Android Studio.

Android SDK (API level 34):

Android Studio typically includes the Android SDK, but you may need to install

additional SDK components depending on the Android versions you're targeting.

Java Development Kit (JDK):

Flutter apps are developed using the Dart programming language, but the

Android platform requires Java for certain tasks. Install the JDK on your PC.

Once you have all these components installed and set up, you can create and

export Flutter apps in Android Studio. Remember to configure your Flutter

project correctly, including specifying the target Android SDK version,

permissions, and any other necessary configurations before exporting the app.

How to get support and license?

Before proceeding, please obtain your item license from our [CRM](#). To do this, create a ticket in the [CRM](#) from the side menu and send your item purchase code. We will then send the license file to you.

The mobile app doesn't include support.

If you want support services for the mobile app, you need to purchase one of the service packages from the following link:

https://crm.rocket-soft.org/index.php/items/grid_view

Once you have chosen a package, please let us know so that we can send you an invoice.

How to connect the mobile app to the web version?

The Rocket LMS mobile app uses the web version as its engine. This means that all users, courses, and any other data will be stored in the web version. As a result, all data will be accessible through both the mobile app and the web version simultaneously.
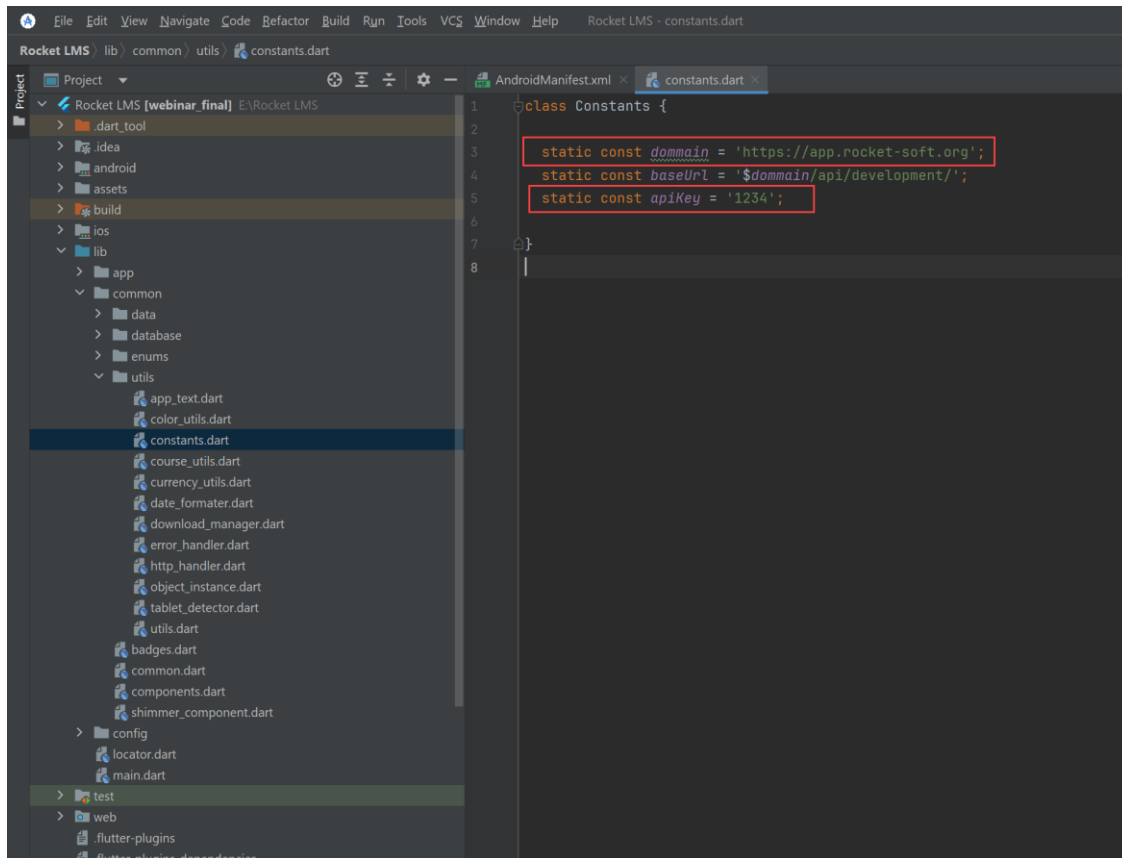
To connect your mobile app to the Rocket LMS web version, follow these steps:

1. Open the `lib\common\utils\constants.dart` file using an IDE like Android

Studio.

2. Edit the `dommain` and `apiKey` parameters.

Note: The red part might be different in your package.

The `apiKey` value should match the `API_KEY` in the `.env` file for the web

version.

You can define any value for this parameter, but it must be the same in both the

`.env` file and the `constants.dart` file.

Define your website URL as the `dommain`.

For example, if you want to configure the mobile app for `https://sample-

site.com`, you should set `https://sample-site.com` as the `dommain`.

How to Change the package name?

To change the package name in a Flutter project, follow these steps. These steps include changing the package name for both Android and iOS platforms.

Changing Package Name for Android:

Change Package Name in build.gradle:

- The build.gradle file is located at android/app/.

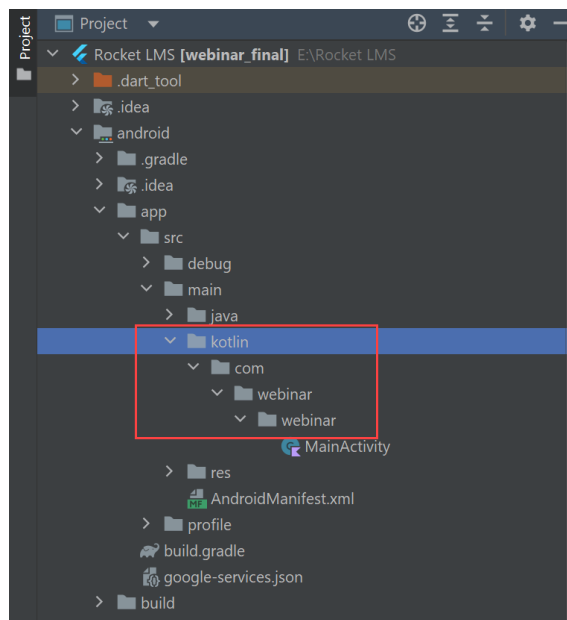- Change the "namespace" and "applicationId" values:
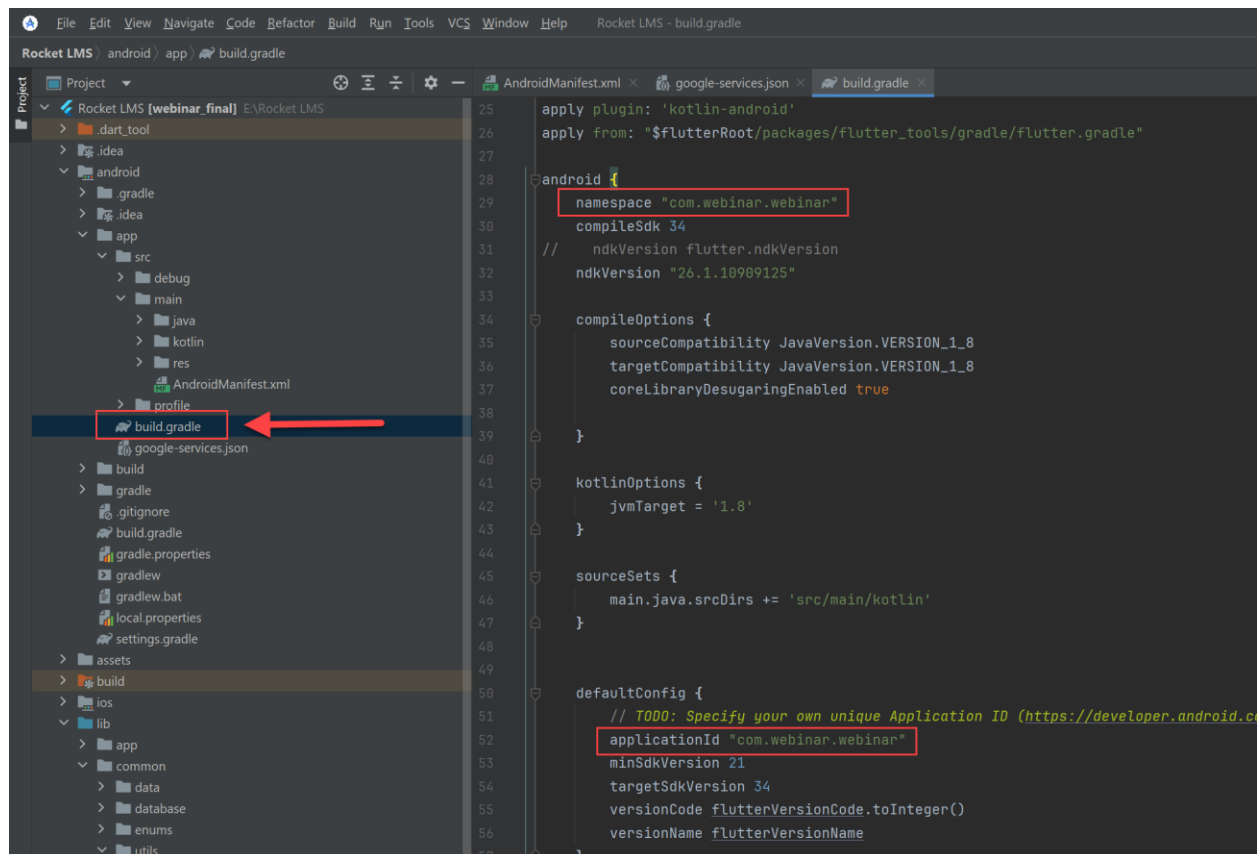
Change Directory Structure:

- Rename the existing package directories at:

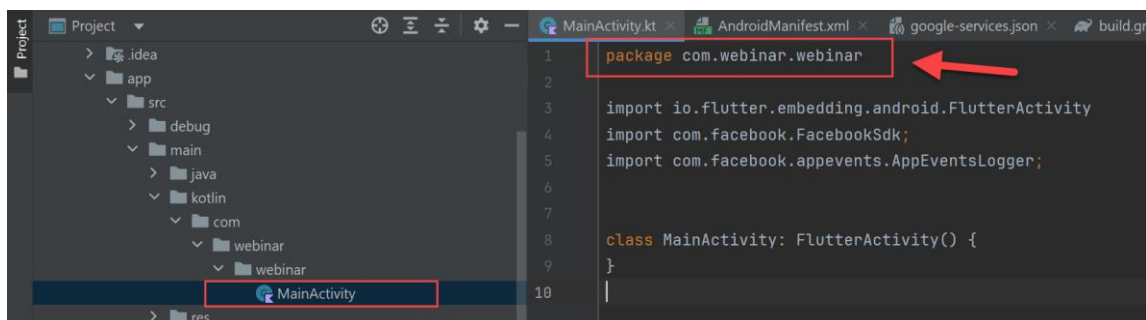**android/app/src/main/kotlin/com/oldpackage/name**

- Create new directories with the new package structure:

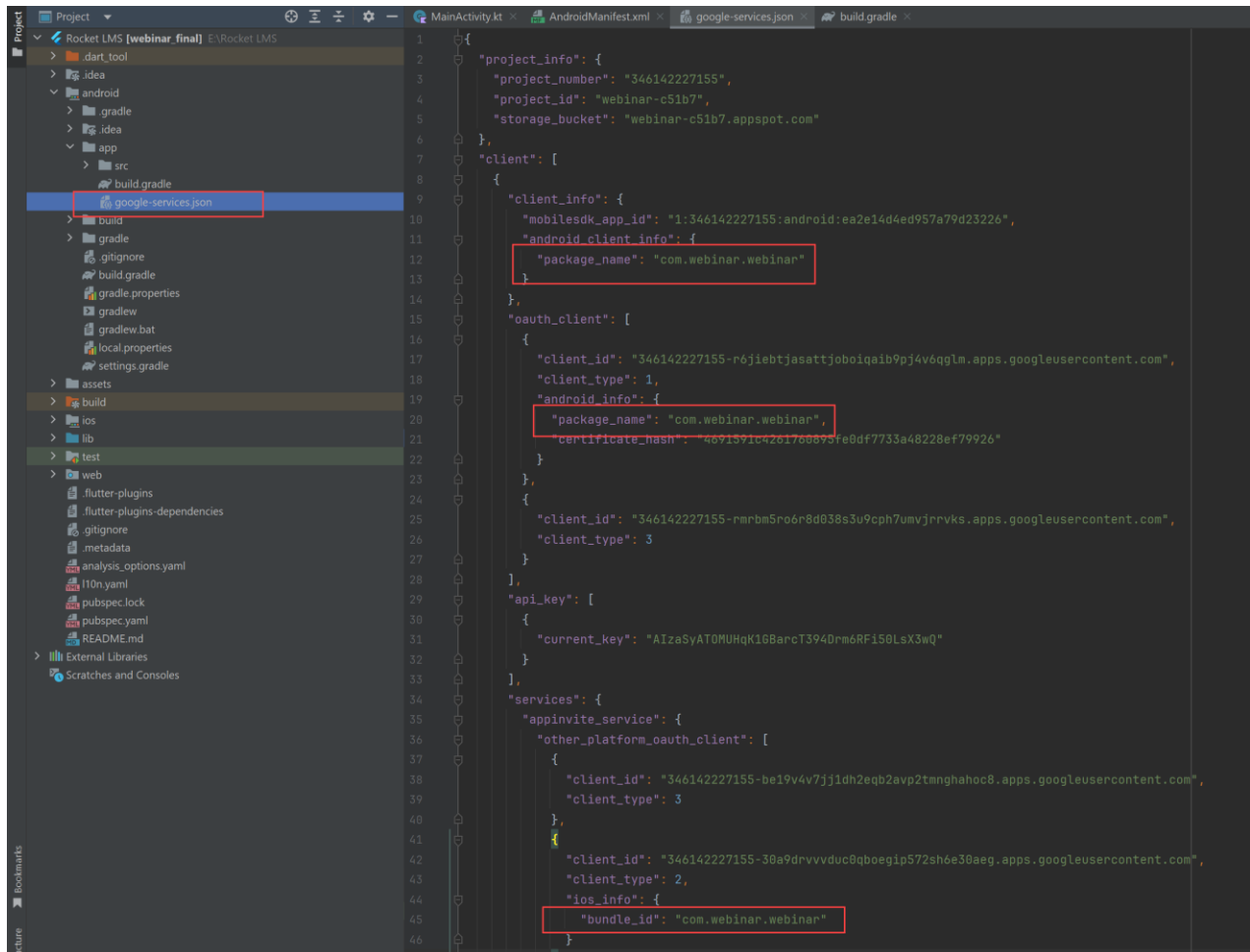**android/app/src/main/kotlin/com/newpackage/name**

Change Package Name in MainActivity.java and Other Java Files:

Open your Java files and change the package declaration at the top of the file:

Change Package Name in google-services.json:

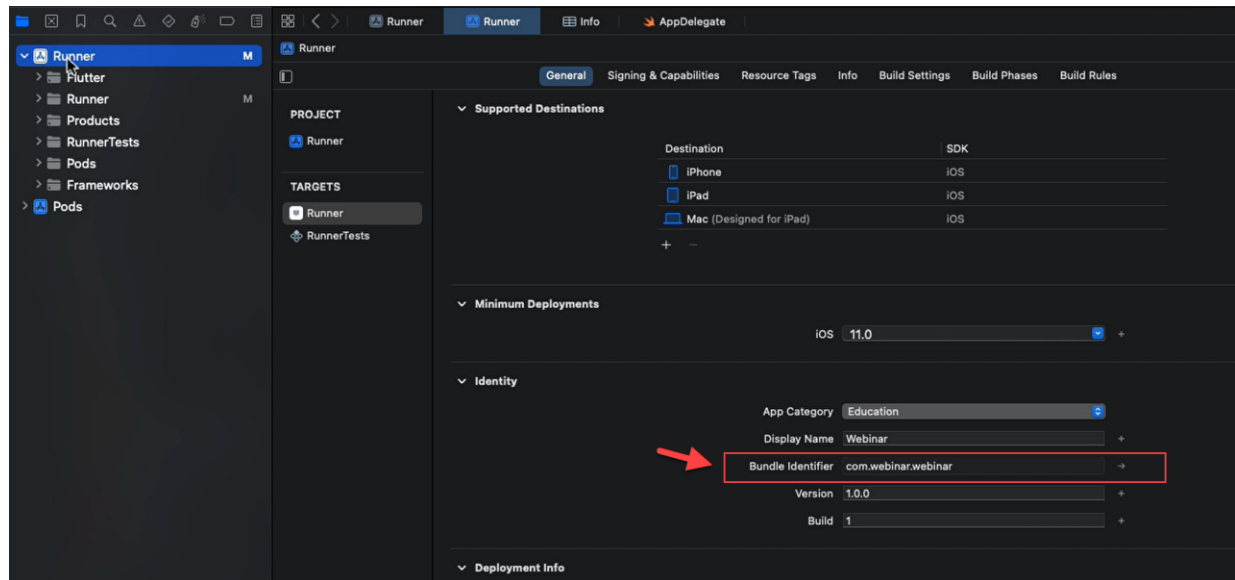The google-services.json file is located at android/app/.

Changing Package Name for iOS:
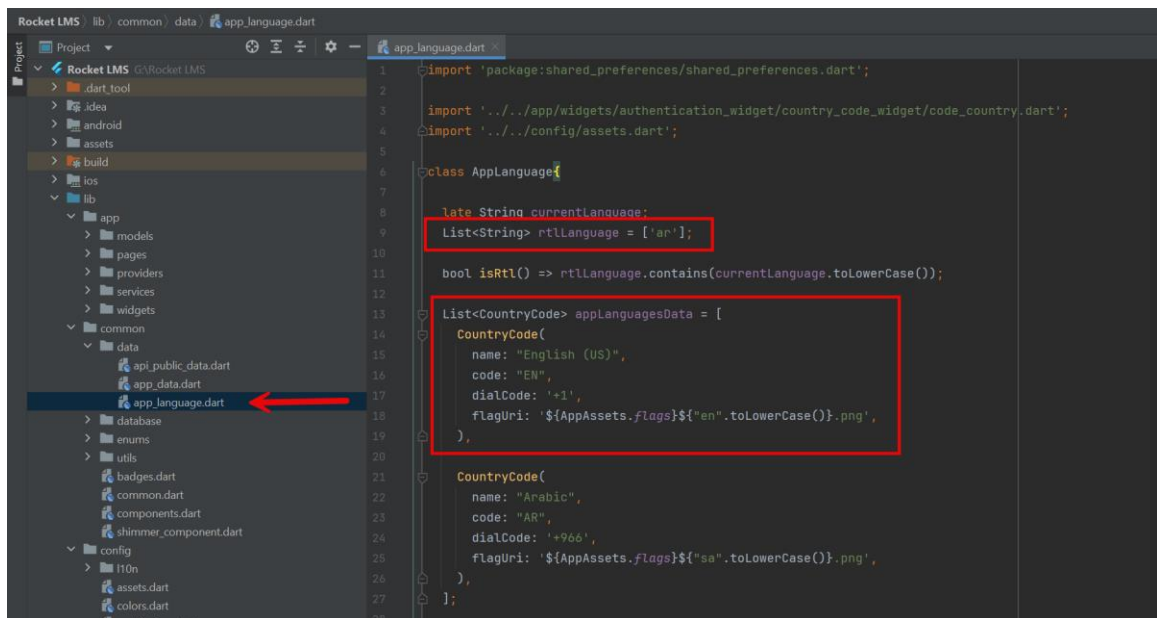
Change Package Name in Xcode Project Settings:

1. Open Xcode and your project.

2. Go to the Runner target.

3. In the Identity section, change the Bundle Identifier.

How to add a new language?

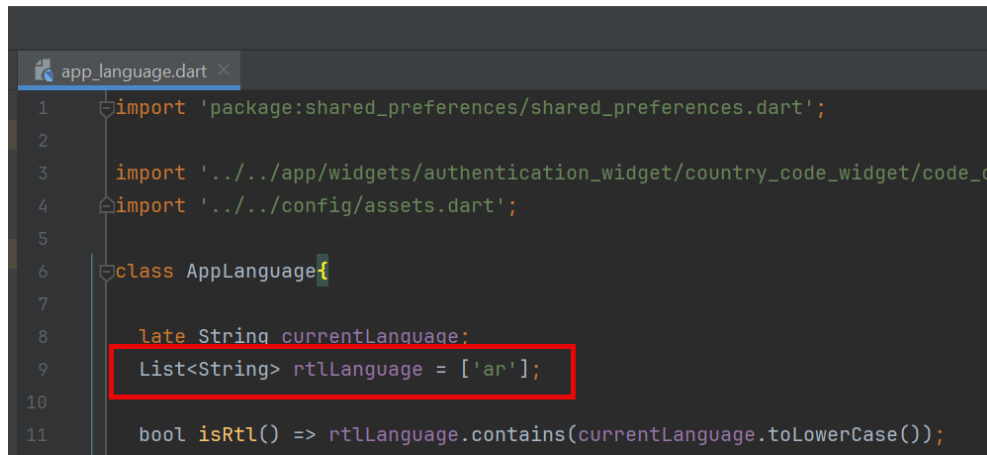1. Open the `\lib\common\data\app_language.dart` file using an IDE like Android Studio.

2. Add your language according to the following sample:

```
CountryCode(
  name: "Your language title",
  code: "Language code",
  dialCode: 'Country code,
  flagUri: '${AppAssets.flags}${"Country Code".toLowerCase()}.png',
),
```

You should input red values according to your language.

if you need to load your language in RTL mode, you should add your language

code in the following part:

3. Create your language file in the following path:

\lib\config\l10n\

You should create the file name according to the following sample:

`app_languagecode.arb`. For example, we created `app_ar.arb` for the Arabic
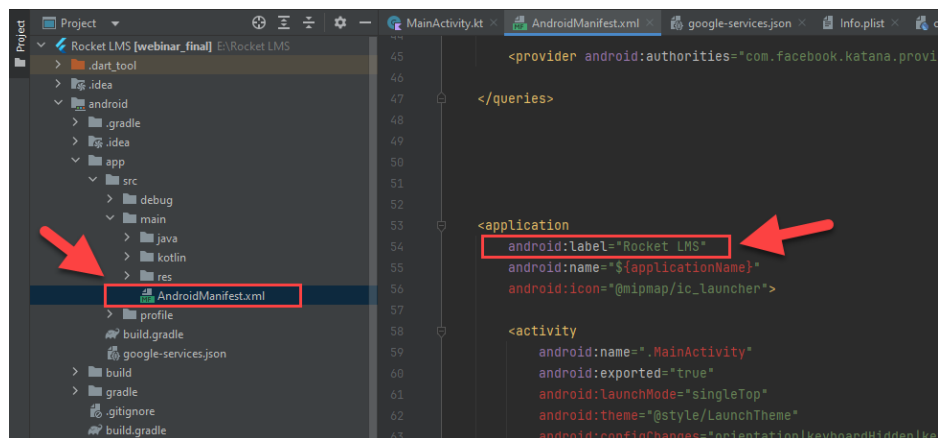
language.

Once you have created your language file, you should copy all phrases from the

`app_en.arb` file to your new language file and then change the translations

according to your language.

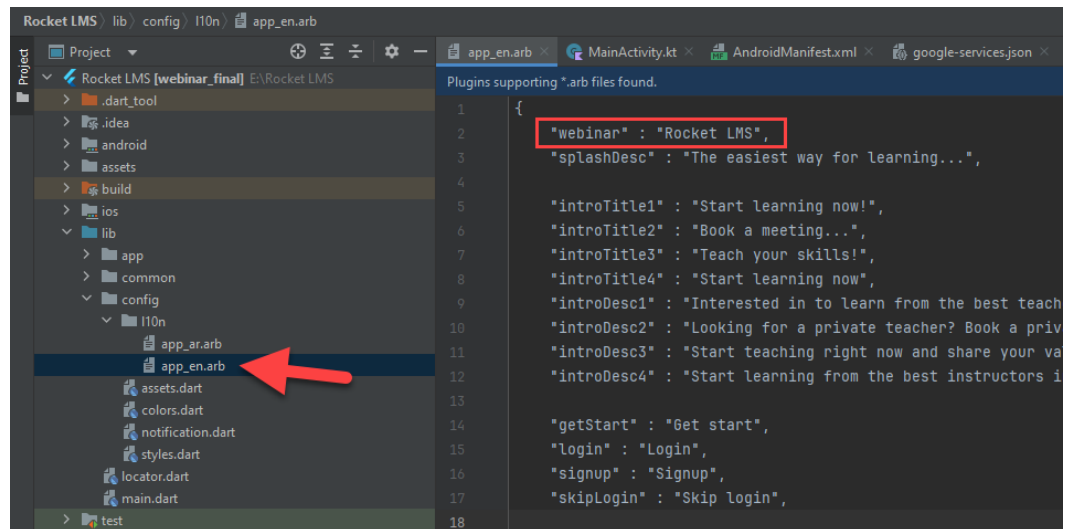You should change the second part in each line:

How to change app name?

Change label in AndroidManifest.xml:

- The AndroidManifest.xml file is located at android/app/src/main.

- Change the "label" value:

Change "Webinar" label translation in your language files:

- The language files are located at lib/config/l10n/.

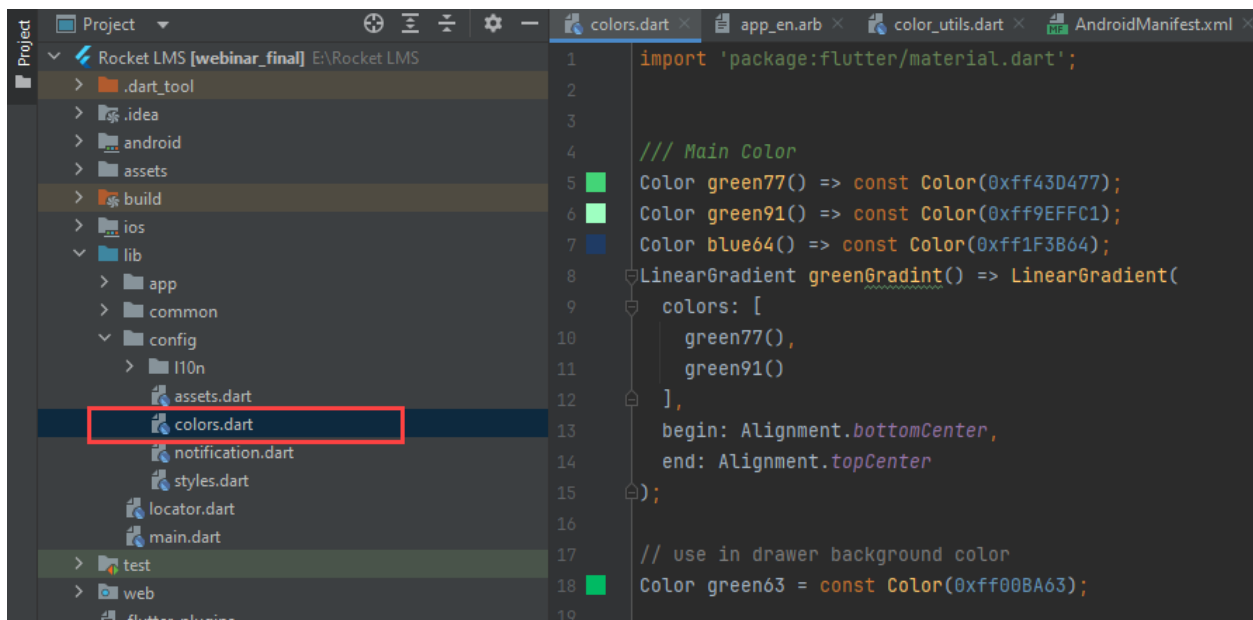- Change the "webinar" label translation:

How to change app colors?

You can change app colors by editing the following file:

/lib/config/colors.dart

We used three primary colors in the app; you can see them at the top of the mentioned file.
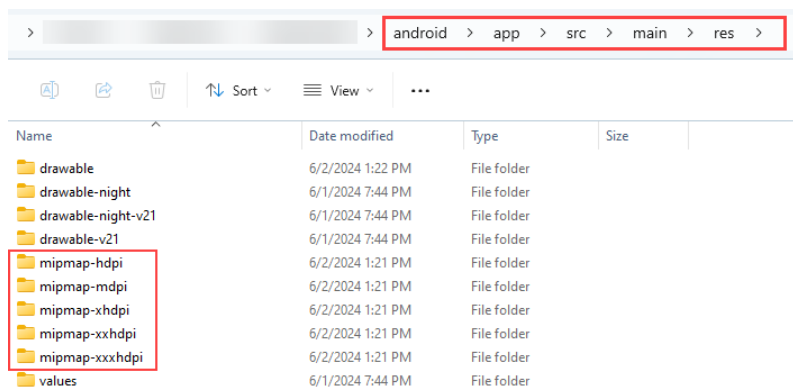

How to change splash screen logo?

Replace your logo file with svg format in the following path:
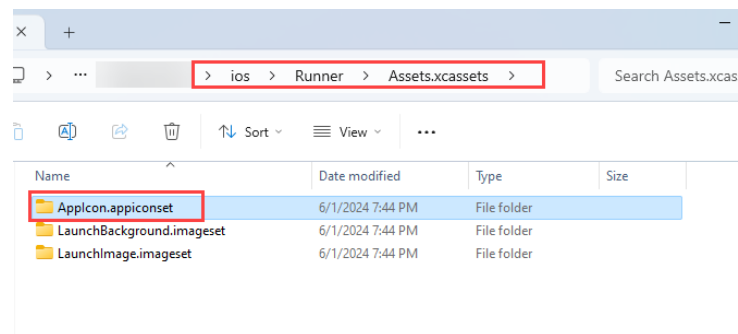
/assets/image/svg/logo-line.svg

/assets/image/svg/splash-logo.svg

How to change app icon?

For Android: Replace your logo files with the files located in the folders shown in the screenshot:

For IOS: Replace your logo files with the files located in the folder shown in the screenshot:
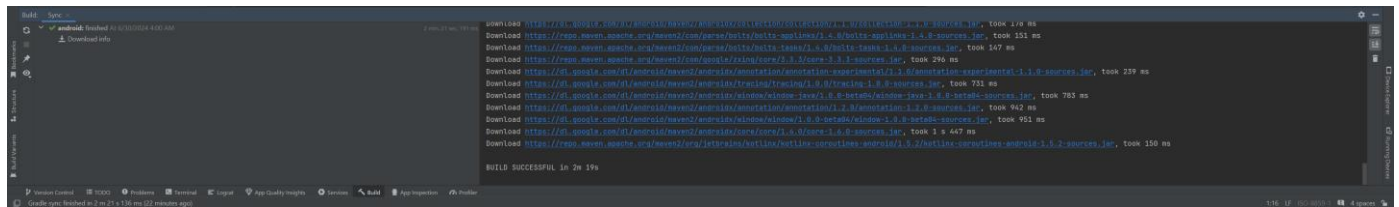

You can also use online tools for generating logo files such as:

https://www.appicon.co/

How to generate APK files?

1. Open the `android` folder in Android Studio and wait for the Gradle building process

to complete.

2. Once it finishes, open the project's root folder in Android Studio.
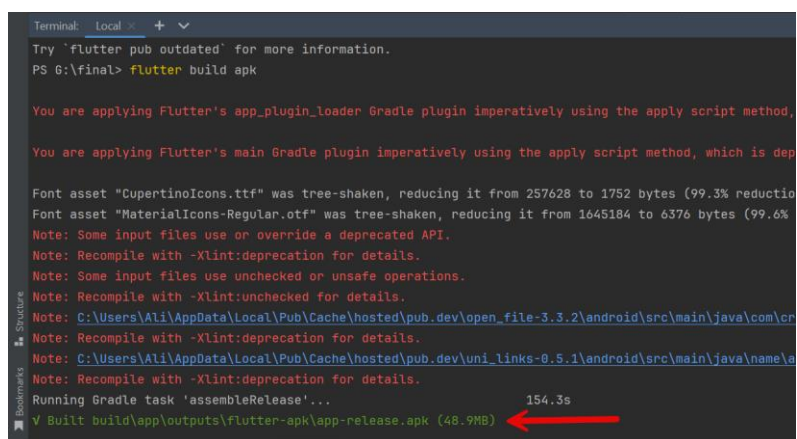
3. Run the following command in the terminal:

flutter pub get

4. Wait for the dependencies to download.

5. open terminal and run: flutter build apk

It will build an APK and show the APK located folder.

You can check the following doc for reference:

https://flutter.dev/docs/deployment/android

How to Publish on App Store and Play Store?

Google Play store:

You can host the application on Google Play store.

You will find tons of supporting videos and blogs on the internet like this:

https://www.youtube.com/watch?v=g0GNuoCOtaQ

App Store iOS:

You can host the application on Apple Store. You will find tons of supporting videos and

blogs on the internet like this:

https://www.youtube.com/watch?v=4_rcaPvDaKM