

Documentation

Rocket LMS Mobile App

Table of contents

Introduction

Included Features

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 files

Publish on Google Play Store

Introduction

Rocket LMS mobile app is an Android mobile application for Rocket LMS. Rocket LMS is an online course marketplace that helps you to run your online education business easily.

Rocket LMS mobile app uses the Rocket LMS as the engine. It means it uses the Rocket LMS web version data. By using this mobile app you will make your online education business more efficient by allowing users to use your platform in their pockets. Your platform users will be able to learn, conduct live classes, take quizzes, manage their courses everywhere without any time and location limitations.

Included Features

We tried to include a pile of useful features to the Rocket LMS mobile app. You can see list of the included feature.

- Native Kotlin code for the best performance.
- Clean design
- Supports live classes, video courses, and text courses.
- Offline course view.
- Text to speech feature for better usability.
- Online quizzes and results.
- Meeting booking system
- Easy to customize
- Multilanguage app and content
- Micro-interactions and animations
- Financial and sales reports
- Notifications and notices
- Advanced learning page
- Blog posts
- Gamification Badges
- Categories and Trend categories
- Google and Facebook login
- Youtube and Vimeo support
- Reviews and comments
- Support and chat system
- Subscriptions
- Discounts and coupons
- Special offer countdown
- Featured courses
- Course pricing plans
- Special native video player

- Report system
- Certificate management
- Organizational education
- Prerequisites system
- Grid and list layout
- Quiz review system
- Quiz statistics
- No network page
- Course wish list
- Comments and reviews
- Separate dashboards for different user roles
- Learning progress
- Referral system
- Offline payments
- Payout process
- Instructor profile
- Add courses to reminder
- Share courses
- Advanced cart
- Course and instructors search
- Course filters
- RTL & LTR Support
- Financial approval
- Account Settings
- Animated onboarding screens
- Error handling

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 \(Web version\)](#)

2- [Rocket LMS plugins bundle](#)

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

Note: You can get the compatible version by contacting us in CRM.

Required Softwares and libraries:

Before going ahead, you need to prepare your local PC to run the app. You need to download and install these cases

1-[Android Studio](#)

2-[JDK and JRE](#) (Both of them will be installed together by installing JDK)

3-[Android SDK \(V26.1.1\)](#)

How to get support and license?

Before everything, you should get your item license from our [CRM](#). Please create a ticket in CRM from the side menu and send your item purchase code to send the license file to you.

If you need support, you can purchase a support package according to your business requirements. You will be able to send support tickets in CRM.

You can also get video tutorials by contacting us in CRM.

How to connect the mobile app to the web version?

Rocket LMS mobile app uses the web version as the engine it means all of the users, courses, and any other data will be stored in the web version and all data will be accessible through the mobile app and the web version at the same time.

```
BuildVars.kt
1 package com.online.course.manager
2
3 import com.online.course.R
4
5
6 object BuildVars {
7     @JvmField
8     var LOGS_ENABLED = true
9     const val API_KEY = "1234"
10    const val BASE_URL = "https://sample-site.com/api/development/"
11    const val CERT_DOWNLOAD_URL = "${BASE_URL}panel/quizzes/results/{}/download"
12    const val PAYMENT_GATEWAY_URL = "${BASE_URL}panel/payments/request"
13
14    class DefaultLang(val value: String) {
15        companion object {
16            const val CODE = "en"
17            const val NAME = "English"
18        }
19    }
20 }
```

To connect your mobile app to the Rocket LMS web version, edit the "app\src\main\java\com\online\course\manager\BuildVars.kt" file using an IDE like Android Studio and edit the "API_KEY" and "BASE_URL" parameters.

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

The API_KEY value should be the same as API_KEY in the env file for the web version. You can define every value for this parameter but it should be the same in the env file and the BuildVars.kt file.

You should define your website URL as the "BASE_URL" for example, if you want to configure the mobile app for "https://sample-site.com", you should define "<https://sample-site.com/api/development/>" as the base URL. Make sure it ends with "/".

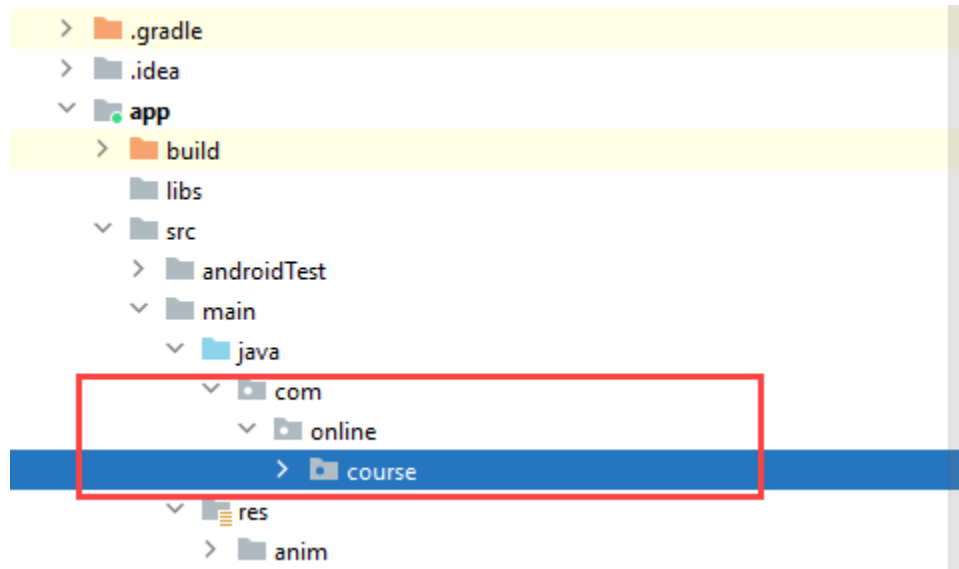
```
network_security_config.xml
1 <?xml version="1.0" encoding="utf-8"?>
2 <network-security-config>
3     <domain-config cleartextTrafficPermitted="true">
4         <domain includeSubdomains="true">yourdomain.com</domain>
5     </domain-config>
6 </network-security-config>
```

Note: If your website doesn't support HTTPS, you should also edit the "app\src\main\res\xml\network_security_config.xml" file and add your website domain address to this file according to the following template:

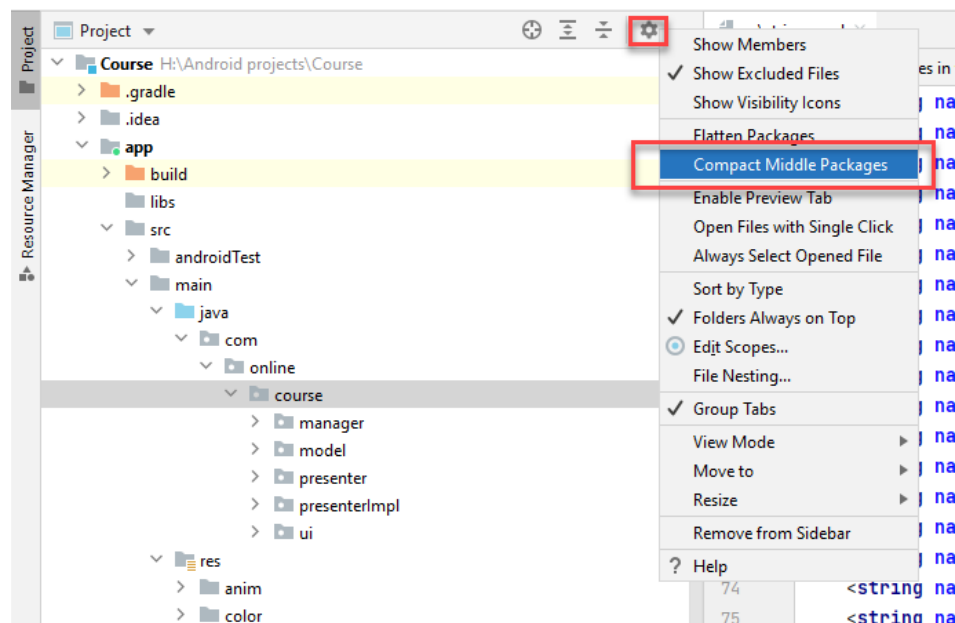
<domain includeSubdomains="true">sample-site.com</domain>

How to change the package name?

Open the project in Android studio and navigate to app/src/main/java/**com/online/course/**



Note: Make sure to click on the compact middle packages to display the package name as separate folders.

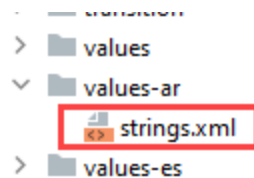


After that, you will see 3 folders. These 3 folders are package name. Press Shift+f6 and rename each folder according to the favorite package name and click on the "Refactor" button. For

example, if you want to change the package name to net.gold.academy, you should rename the first folder to "new", the second one to "gold" and the third folder to "academy".

How to translate Rocket LMS mobile app into my native language?

To translate the Rocket LMS mobile app to your native language, navigate to "app\src\main\res" and create a folder using your language ISO code ([Find your language ISO 639-1 code](#)) according to the following template:



```
<string name="dashboard">Tablero</string>
<string name="classes">Cursos</string>
<string name="meetings">Reuniones</string>
<string name="quizzes">Cuestionarios</string>
<string name="certificates">Certificados</string>
<string name="favorites">Favoritas</string>
<string name="comments">Comentarios</string>
<string name="financial">Financiera</string>
<string name="support">Apoyo</string>
<string name="subscription">Suscripción</string>
<string name="featured_classes">Cursos destacados</string>
<string name="newest_classes">Cursos más nuevos</string>
<string name="view_all">Ver todo</string>
<string name="bestRated">mejor calificado</string>
<string name="best_selling">Más vendido</string>
```

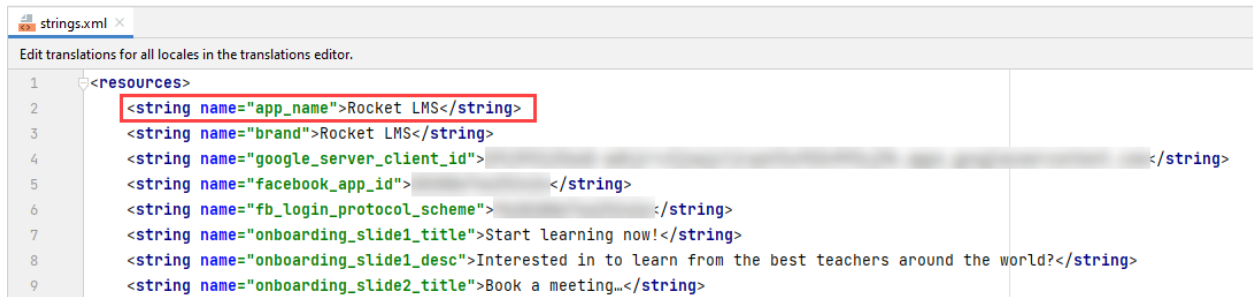
Translate this part

"values-yourlanguageisocode" for example, if you want to translate your language to Arabic, you should create "values-ar" folder.

After that go to the "app\src\main\res\values" folder and copy the "strings.xml" file to your language folder.

Translate the specified part in the "strings.xml" according to the following example using Android Studio. Your language file is ready.

How to change the app name?



To change the app name to your favorite name, edit the "app\src\main\res\values\strings.xml" file using Android Studio and change the "app_name" parameter in the first line.

How to change the app colors?



To change the app colors, edit the "app\src\main\res\values\colors.xml" file using Android Studio and change the color codes (RGB and RGBA) to your favorite colors.

Useful reference: [Convert RGBA to HEX](#)

How to change the splash screen logo?

To change the splash screen icon, create a 64x64 transparent “.png” icon and replace it with the "app\src\main\res\drawable\splash_logo.png" file.

How to change the app icon?

First, generate your app icon by using the [Android app icon generator tool](#). This tool helps you to generate the app icon for different resolutions with a single click. Make sure to put the app icon file name "app_icon.png".

After downloading your app icons, replace them with the following paths:

app\src\main\res\mipmap-hdpi

app\src\main\res\mipmap-mdpi

app\src\main\res\mipmap-xhdpi

app\src\main\res\mipmap-xxhdpi

app\src\main\res\mipmap-xxxhdpi

How to build APK?

Click on the "Build/Rebuild Project" menu from the top navigation bar and wait until APK files are generated. You can share these APK files by uploading them to your server.

The APK file will be available in \app\build\intermediates\apk\debug path.

If you want to generate APK files for the app stores like Google Play Store, you should click on the "Build/Generate Signed APK".

The APK file will be available in \app\release path.

How to publish the Rocket LMS mobile app on Play Store?

You need to change your app package name according to the previous steps and build a signed APK. After that, you can publish your mobile app on Google Play Store according to the following video tutorial.

[How to Publish an Android App to Google Play](#)