

Smart POS

Offline Point of Sale System for Android

Documentation

Start

Created: 31/03/2020

Latest update: 18/11/2024

Version: SmartPos_Version_7.7

By: OnlineSoftSell

E-mail: onlinesoftsell@gmail.com

Website: www.onlinesoftsell.com

Thank you for purchasing my app. If you have any questions that are beyond the scope of this help file, please feel free to email. Thank you so much!

We are following CodeCanyon rules and regulation, so under this license you can only publish single product on play store. And you can't sell this source code anywhere. Please follow Standard Licenses: <https://codecanyon.net/licenses/standard/>

So, if you want to publish multiple app please get multiple licenses, otherwise your app will be blocked after publishing.

Introduction

Smart POS is a very useful offline point of sale application for android mobile device. By using Smart POS, you can easily maintain your all kinds of products and store all sales records. You can store customer and supplier's information. You can easily see your daily, monthly and yearly reports with bar chart. Smart POS has multiple language (English, Spanish, Bangla) facility. Also, there is backup facility. You can back up your database and save it to device storage and you can import it. You can manage your sells smartly.

Requirements for Smart Pos:

- Android Studio
- SDK 33 (Recommended to use)

Android Studio Setup Process

1. Download Android Studio - <https://developer.android.com/studio/>
2. Install Android Studio - <https://developer.android.com/studio/install>
3. After download setup Android Studio and Download SDK 33
4. Learn more about Android Studio - <https://developer.android.com/studio/intro/>

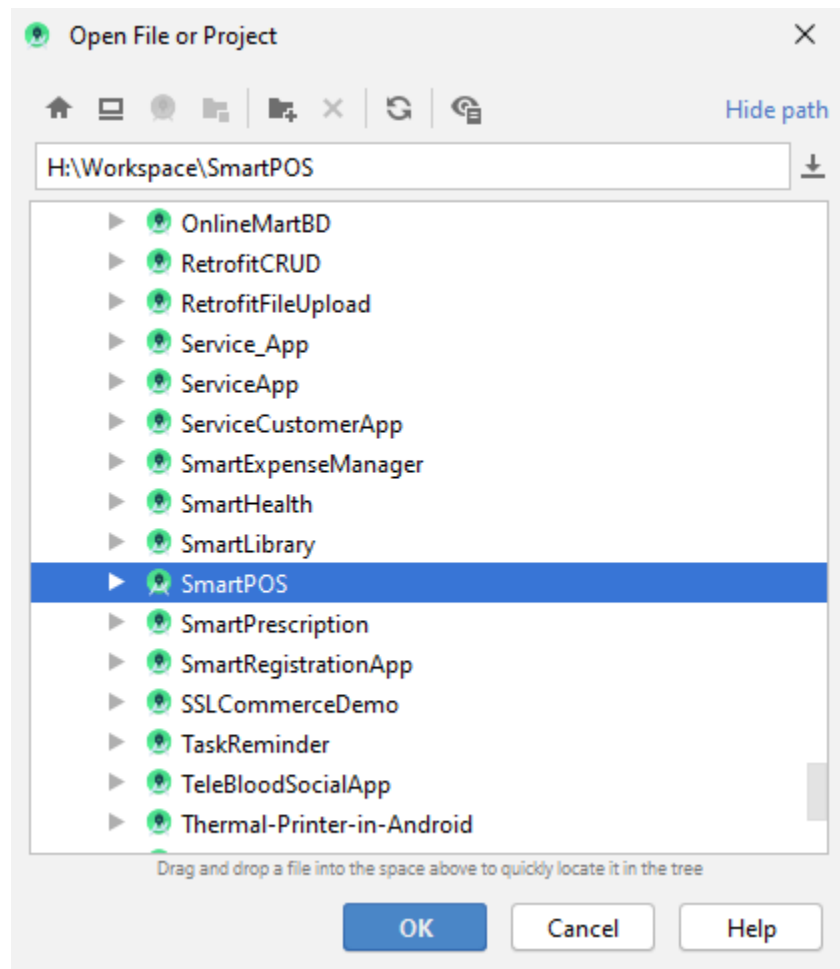
Setup

Android Studio Setup Process:

To develop an app, you will need minimum Android studio latest version. This project is built with SDK 33. So, after installing the Android studio run it and open SDK manager to download build version and SDK tools. Most of the time Android Studio will automatically download the files if you fetch any problem then check this below setup.

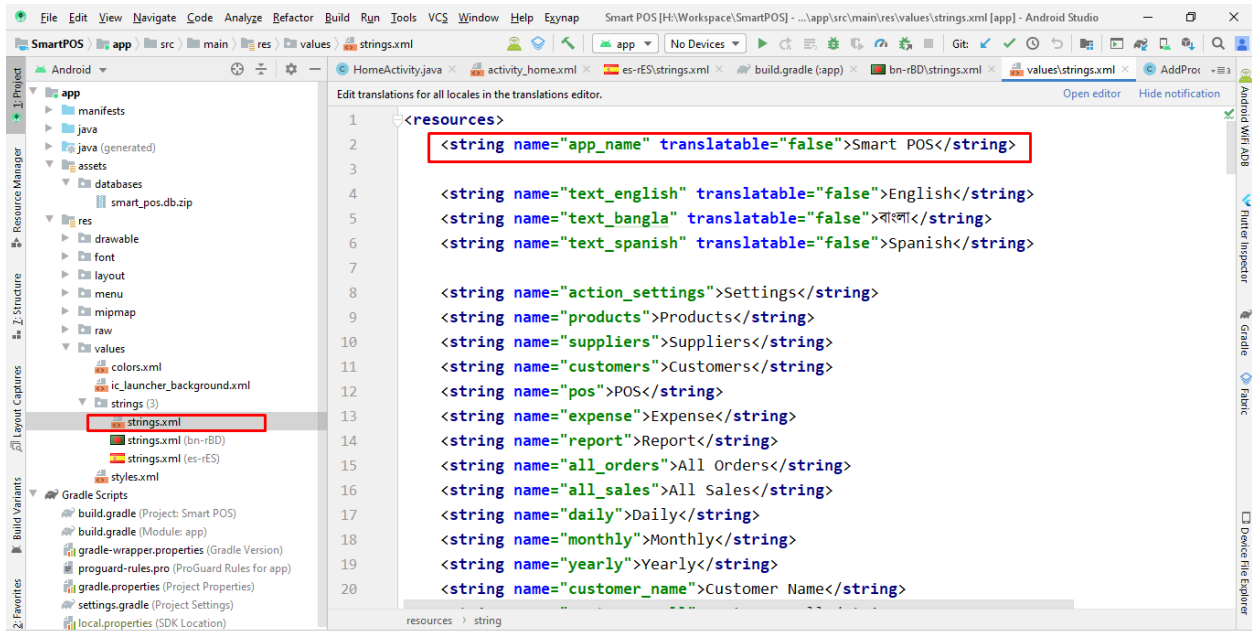
To open app source code, unzip it and open with android studio.





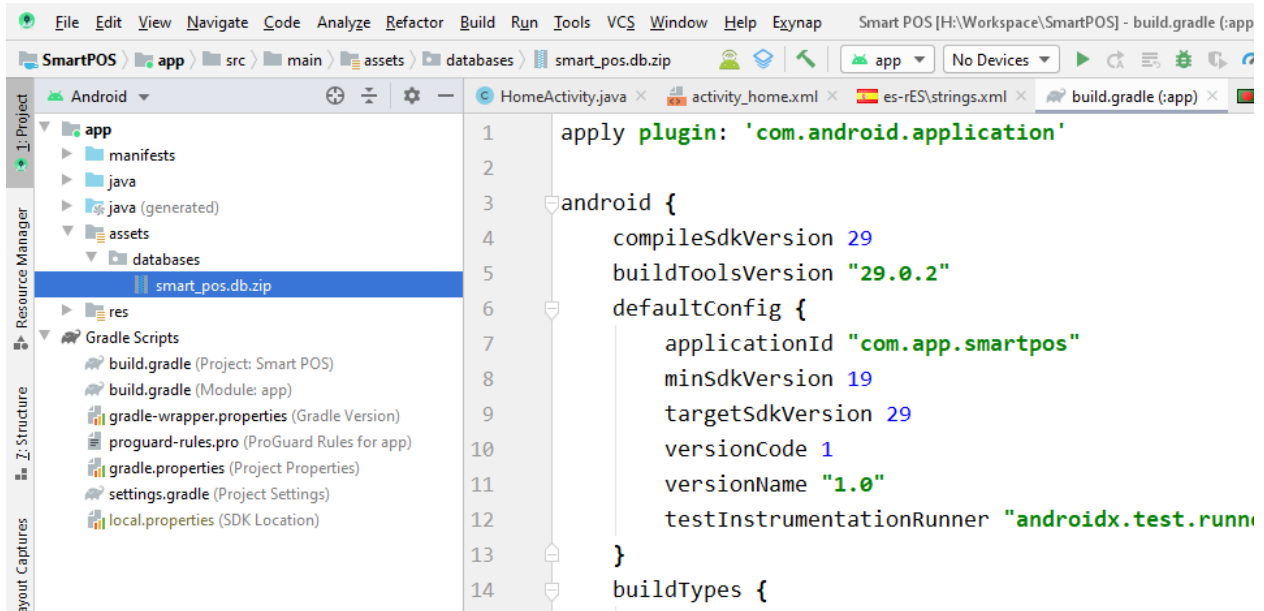
Changing App Name

To change app name, go to res>values>string.xml file and change app what you want.



Offline Database Structure and Modification

You can find database in app>assets>databases>smart_pos.db.zip in zip file.



To view database unzip and open with any SQLite browser software. Or you can download this one from the link:

<https://sqlitebrowser.org/>

You can change or modify database by using SQLite browser and zip it and replace it app>assets>databases folder with name **smart_pos.db.zip** format.

Database Table List and structure

DB Browser for SQLite - D:\DESKTOP V2\Sqlite Browser\SQLiteDatabaseBrowserPortable\Data\smart_pos.db

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Datab

Database Structure Browse Data Edit Pragma Execute SQL

Create Table Create Index Print

Name	Type	Schema
▼ Tables (11)		
> customers		CREATE TABLE "customers" ("customer_id" INTEGER PRIMARY KEY AUTOINCREMENT, "cust
> expense		CREATE TABLE "expense" ("expense_id" INTEGER PRIMARY KEY AUTOINCREMENT, "expense
> order_details		CREATE TABLE "order_details" ("order_details_id" INTEGER PRIMARY KEY AUTOINCREMENT,
> order_list		CREATE TABLE "order_list" ("order_id" INTEGER PRIMARY KEY AUTOINCREMENT, "invoice_id
> product_cart		CREATE TABLE "product_cart" ("cart_id" INTEGER PRIMARY KEY AUTOINCREMENT, "product
> product_category		CREATE TABLE "product_category" ("category_id" INTEGER PRIMARY KEY AUTOINCREMENT
> product_weight		CREATE TABLE "product_weight" ("weight_id" INTEGER PRIMARY KEY AUTOINCREMENT, "w
> products		CREATE TABLE "products" ("product_id" INTEGER PRIMARY KEY AUTOINCREMENT, "produc
> shop		CREATE TABLE "shop" ("shop_id" INTEGER PRIMARY KEY AUTOINCREMENT, "shop_name" T
> sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)
> suppliers		CREATE TABLE "suppliers" ("suppliers_id" INTEGER PRIMARY KEY AUTOINCREMENT, "suppli

Name	Type	Schema
▼ Tables (11)		
▼ customers		CREATE TABLE "customers" ("customer_id" INTEGER PRIMARY KEY AUTOINCREMENT, "c
customer_id	INTEGER	"customer_id" INTEGER PRIMARY KEY AUTOINCREMENT
customer_name	TEXT	"customer_name" TEXT
customer_cell	TEXT	"customer_cell" TEXT
customer_email	TEXT	"customer_email" TEXT
customer_address	TEXT	"customer_address" TEXT
▼ expense		CREATE TABLE "expense" ("expense_id" INTEGER PRIMARY KEY AUTOINCREMENT, "expe
expense_id	INTEGER	"expense_id" INTEGER PRIMARY KEY AUTOINCREMENT
expense_name	TEXT	"expense_name" TEXT
expense_note	TEXT	"expense_note" TEXT
expense_amount	TEXT	"expense_amount" TEXT
expense_date	TEXT	"expense_date" TEXT
expense_time	TEXT	"expense_time" TEXT
▼ order_details		CREATE TABLE "order_details" ("order_details_id" INTEGER PRIMARY KEY AUTOINCREME
order_details_id	INTEGER	"order_details_id" INTEGER PRIMARY KEY AUTOINCREMENT
invoice_id	TEXT	"invoice_id" TEXT
product_name	TEXT	"product_name" TEXT
product_weight	TEXT	"product_weight" TEXT
product_qty	TEXT	"product_qty" TEXT
product_price	TEXT	"product_price" TEXT
product_image	TEXT	"product_image" TEXT
product_order_date	TEXT	"product_order_date" TEXT

name	type	schema
▼ Tables (11)		
> customers		CREATE TABLE "customers" ("customer_id" INTEGER PRIMARY KEY AUTOINCREMENT, "c
> expense		CREATE TABLE "expense" ("expense_id" INTEGER PRIMARY KEY AUTOINCREMENT, "expe
> order_details		CREATE TABLE "order_details" ("order_details_id" INTEGER PRIMARY KEY AUTOINCREME
▼ order_list		CREATE TABLE "order_list" ("order_id" INTEGER PRIMARY KEY AUTOINCREMENT, "invoic
order_id	INTEGER	"order_id" INTEGER PRIMARY KEY AUTOINCREMENT
invoice_id	TEXT	"invoice_id" TEXT
order_date	TEXT	"order_date" TEXT
order_time	TEXT	"order_time" TEXT
order_type	TEXT	"order_type" TEXT
order_payment_method	TEXT	"order_payment_method" TEXT
customer_name	TEXT	"customer_name" TEXT
▼ product_cart		CREATE TABLE "product_cart" ("cart_id" INTEGER PRIMARY KEY AUTOINCREMENT, "pro
cart_id	INTEGER	"cart_id" INTEGER PRIMARY KEY AUTOINCREMENT
product_id	TEXT	"product_id" TEXT
product_weight	TEXT	"product_weight" TEXT
product_weight_unit	TEXT	"product_weight_unit" TEXT
product_price	TEXT	"product_price" TEXT
product_qty	INTEGER	"product_qty" INTEGER
▼ product_category		CREATE TABLE "product_category" ("category_id" INTEGER PRIMARY KEY AUTOINCREM
category_id	INTEGER	"category_id" INTEGER PRIMARY KEY AUTOINCREMENT
category_name	TEXT	"category_name" TEXT
▼ products		CREATE TABLE "products" ("product_id" INTEGER PRIMARY KEY AUTOINCREMENT, "pro
product_id	INTEGER	"product_id" INTEGER PRIMARY KEY AUTOINCREMENT
product_name	TEXT	"product_name" TEXT
product_code	TEXT	"product_code" TEXT
product_category	TEXT	"product_category" TEXT
product_description	TEXT	"product_description" TEXT
product_buy_price	TEXT	"product_buy_price" TEXT
product_sell_price	TEXT	"product_sell_price" TEXT
product_supplier	TEXT	"product_supplier" TEXT
product_image	TEXT	"product_image" TEXT
product_stock	TEXT	"product_stock" TEXT
product_weight_unit_id	TEXT	"product_weight_unit_id" TEXT
product_weight	TEXT	"product_weight" TEXT
▼ shop		CREATE TABLE "shop" ("shop_id" INTEGER PRIMARY KEY AUTOINCREMENT, "shop_nam
shop_id	INTEGER	"shop_id" INTEGER PRIMARY KEY AUTOINCREMENT
shop_name	TEXT	"shop_name" TEXT
shop_contact	TEXT	"shop_contact" TEXT
shop_email	TEXT	"shop_email" TEXT
shop_address	TEXT	"shop_address" TEXT
shop_currency	TEXT	"shop_currency" TEXT

Name	Type	Schema
▼ Tables (11)		
> customers		CREATE TABLE "customers" ("customer_id" INTEGER PRIMARY KEY AUTOINCREMENT, "cust
> expense		CREATE TABLE "expense" ("expense_id" INTEGER PRIMARY KEY AUTOINCREMENT, "expense
> order_details		CREATE TABLE "order_details" ("order_details_id" INTEGER PRIMARY KEY AUTOINCREMENT,
> order_list		CREATE TABLE "order_list" ("order_id" INTEGER PRIMARY KEY AUTOINCREMENT, "invoice_id
> product_cart		CREATE TABLE "product_cart" ("cart_id" INTEGER PRIMARY KEY AUTOINCREMENT, "produc
> product_category		CREATE TABLE "product_category" ("category_id" INTEGER PRIMARY KEY AUTOINCREMENT
> product_weight		CREATE TABLE "product_weight" ("weight_id" INTEGER PRIMARY KEY AUTOINCREMENT, "w
> products		CREATE TABLE "products" ("product_id" INTEGER PRIMARY KEY AUTOINCREMENT, "produc
> shop		CREATE TABLE "shop" ("shop_id" INTEGER PRIMARY KEY AUTOINCREMENT, "shop_name" T
> sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)
▼ suppliers		CREATE TABLE "suppliers" ("suppliers_id" INTEGER PRIMARY KEY AUTOINCREMENT, "suppli
suppliers_id	INTEGER	"suppliers_id" INTEGER PRIMARY KEY AUTOINCREMENT
suppliers_name	TEXT	"suppliers_name" TEXT
suppliers_contact_person	TEXT	"suppliers_contact_person" TEXT
suppliers_cell	TEXT	"suppliers_cell" TEXT
suppliers_email	TEXT	"suppliers_email" TEXT
suppliers_address	TEXT	"suppliers_address" TEXT

AdMob Ads Setup

In Smart POS app, we integrated Google AdMob Ads service. We added two types of ads.

1. Banner Ads
2. Interstitial Ads

To replace your own AdMob Ads you need to replace AdMob application ID, banner ads id and interstitial ads id.

To replace id, go to **app>res>values>string>string.xml** file and change your own AdMob ids.

```

1 <resources>
2 <string name="app_name" translatable="false">Smart POS</string>
3
4 <string name="text_english" translatable="false">English</string>
5 <string name="text_bangla" translatable="false">বাংলা</string>
6 <string name="text_spanish" translatable="false">Spanish</string>
7
8 <!--Replace your adMob application ID here -->
9 <string name="admob_application_id" translatable="false">ca-app-pub-3940256099942544~334751171</string>
10
11 <!--Replace your adMob Banner Ads ID here -->
12 <string name="admob_banner_ads_id" translatable="false">ca-app-pub-3940256099942544/6300978111</string>
13
14 <!--Replace your admob Interstitial Ads ID here -->
15 <string name="admob_interstitial_ads_id" translatable="false">ca-app-pub-3940256099942544/103</string>
16

```

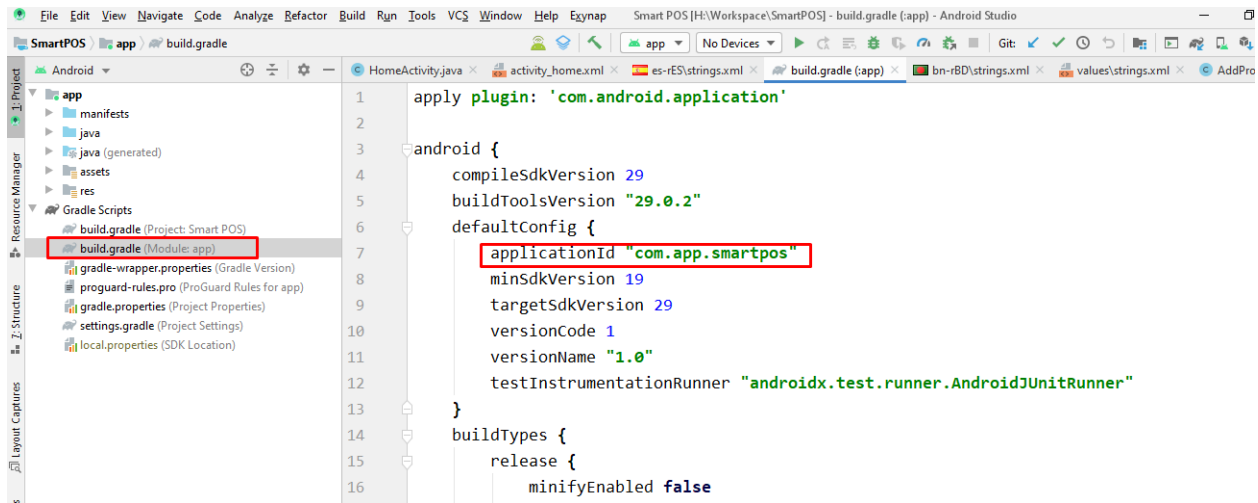
To know more about Google AdMob Ads, Visit this links:

<https://developers.google.com/admob/android/quick-start>

Publish

Package Name Changing for Play Store

Every app has a unique applicationId for identify the app. And this is also use as play store url. To change edit Gradle Scripts > build.gradle or press Shift twice and search build.gradle. “applicationId” is your app id, to publish an app on app store you must need a unique package name. After change package name must be updae firebase file. And no need to change the folder name just change the package name from gradle.

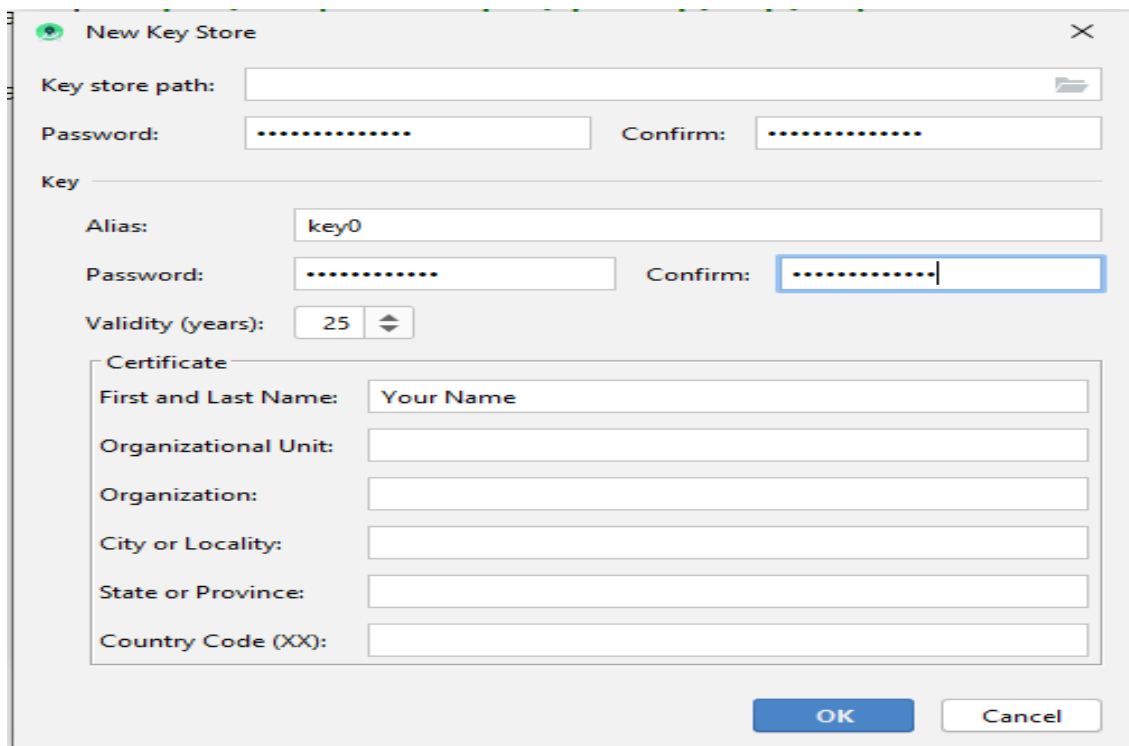
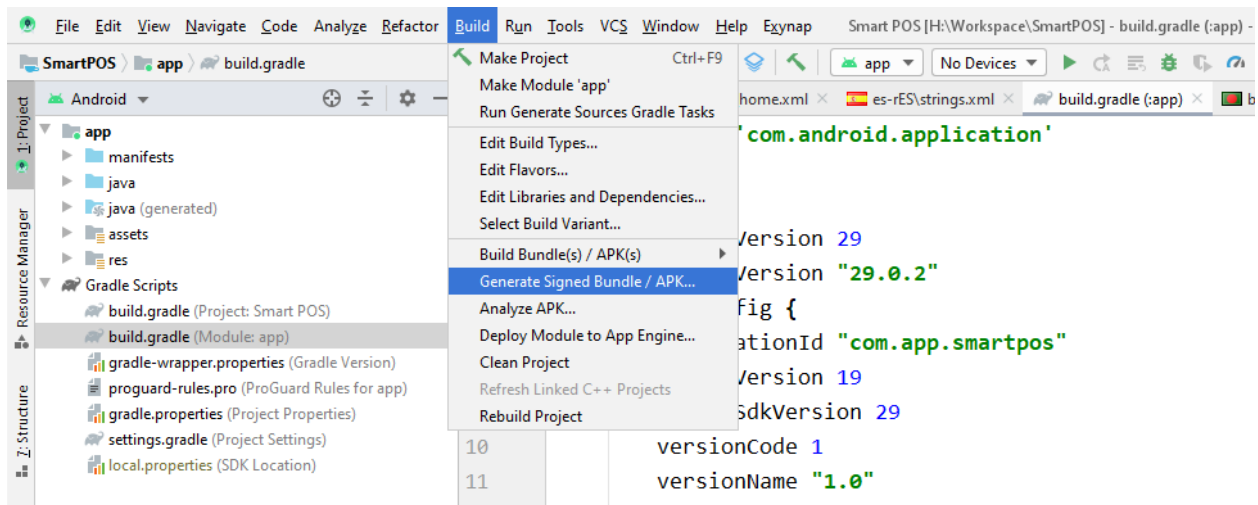


Build an android app and publish:

After renaming, build your project and check all functions. If all is ok, generate Signed APK: Click Build > Generated Signed APK. Create your Key, set your password and build it!

Preparing to publish on Play Store:

After setting up unique application ID now make a final build for play store. Go to **Menu > Build > Generate signed APK > select module > Create new key Store**, then provide your info and password for the key and select signature version for both. When you get the final build APK, just upload the file in your Play Store (<https://play.google.com/apps/publish/>) Developer Account and you are good to go. If you don't have the account then google will takes \$25 for lifetime just make payment and provide your information.



Congress your mobile application develops complete! After build successfully, you can use the APK and publish it onto any app store.

Data Export and Import from Excel (xls) file

We have added bulk product entry from excel xls file. You can easily import multiple products, customers, supplier's data from excel xls file. Also, you can export products, customers, expense, sales data into excel xls file. You can also export full database as excel xls format. To import product data, firstly you export product data into excel xls format. Then you get a product.xls file in your selected directory. Then transfer product.xls file from your mobile device to your laptop/desktop. Then open file using Microsoft Excel and add your all product data. And finally transfer this file into your mobile device and you can import all product data from this xls file.

If you want add bulk image import from xls file, you have to convert image into base64 format. Then copy base64 code and replace to *product_image* column of product.xls file.

To convert image into base64 you can use this website:

<https://www.browserling.com/tools/image-to-base64>

PDF Receipt Print

You can generate pdf receipt and print via printer. Pdf receipt support only **English** language. If you use other language, it would not be show. PDF receipt generate in 58mm POS printer page size. You can change it in code.

If you need any customization support, contact with us.

Support

What's Included with The Project Files

- Source code in Java language
- Database file
- Complete setup guideline documentation

License

- This project is not providing any icon or logo for professional use.
- App publishing or any type of custom development is not included with it.
- Icon asset credit given to www.flaticon.com
- This project is only for one app publish only and get support for that particular app.
- Under this project, we can't provide multiple product support.

Support Desk

Still, you having any questions? Just leave a comment. Or you need support from us? Just visit our Support Center.

E-mail ID: onlinesoftsell@gmail.com

Website: www.onlinesoftsell.com



THANK YOU