

## Get Started:

# Guide Documentation v1.0 for “All In One Android Earning App with Admin Panel”

**Created:** 20/05/2022

**Last Modified:** 20/05/2022

**Author:**

- AADevelopers

**Help/Support:** [aadevelopers963@gmail.com](mailto:aadevelopers963@gmail.com)

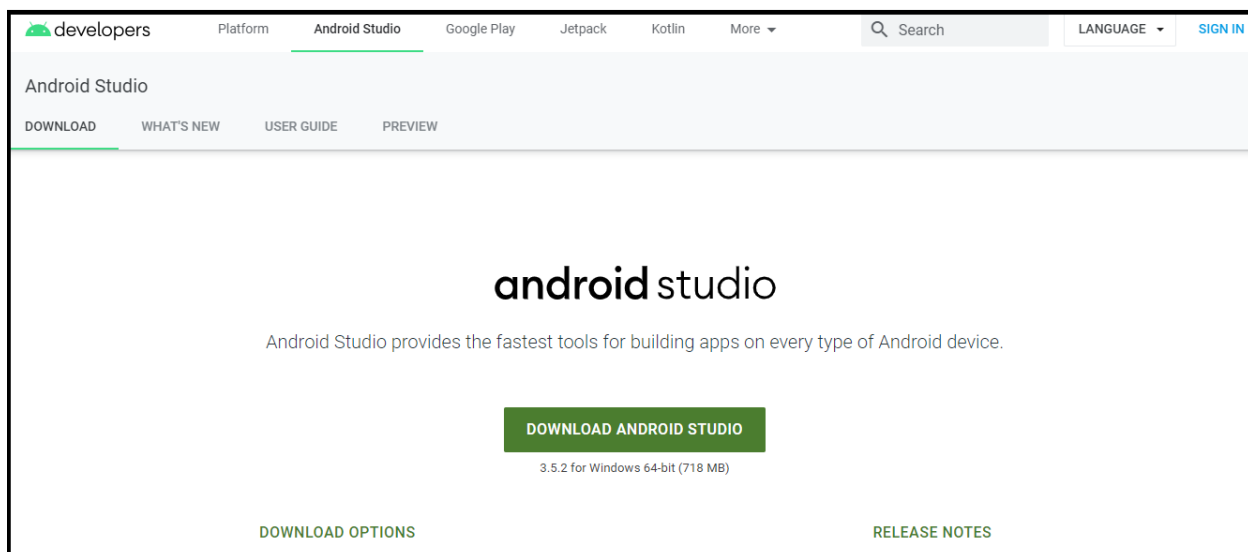
*Thank you very much for purchasing our app. If you have any issue or question that this document does not cover, feel free to contact us at [aadevelopers963@gmail.com](mailto:aadevelopers963@gmail.com)*

## 1. Android Studio

### Download Android studio

To use this app project you need to download the android studio on your computer first please click on download android studio to download on your computer. After Download the Android Studio you need to install it on your computer/Mac.

[DOWNLOAD ANDROID STUDIO](#)

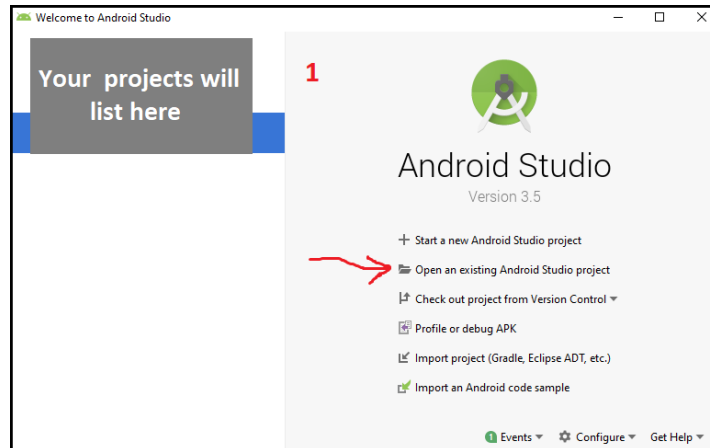


## 2. Open Project

### Open Project in Android studio

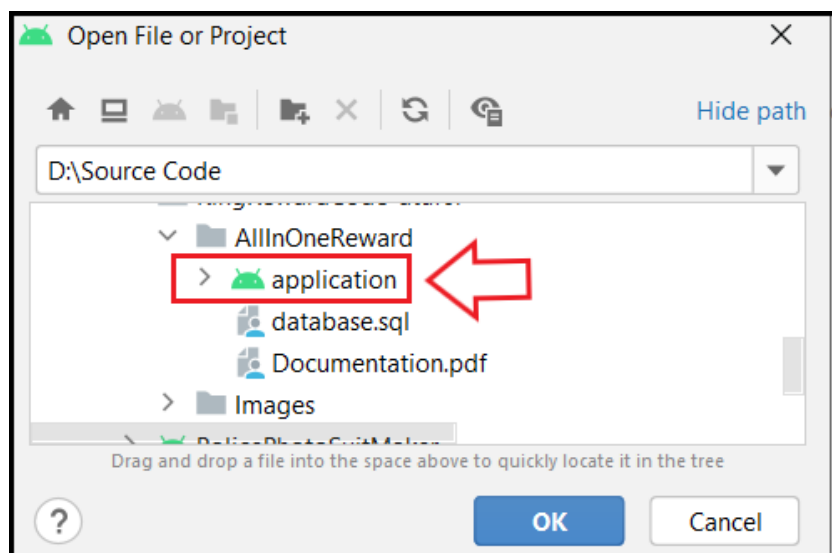
Step: 1

Open android studio and click on Open an existing Android Studio Project.



Step: 2

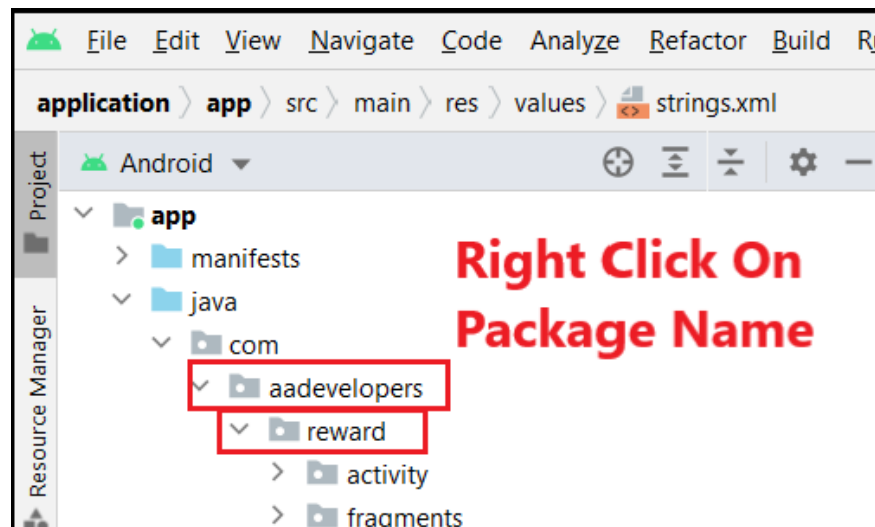
Select **application** from the **AllInOneReward** Project and click on the Ok.



### 3. Change Package Name

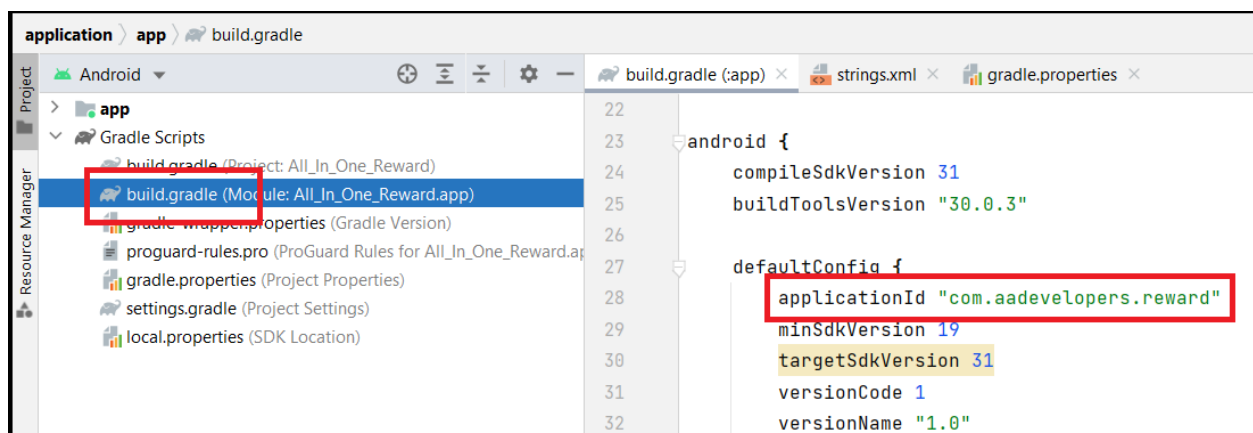
Step :1

Right click on **com.aadevelopers.reward** inside Java Folder and Select **Rename** in Refactor Option then click on **Rename Package** and enter your Package name what you want to give. Then Click on **Do Reflector** and Android Studio will automatically do default setting.



Step: 2

Go to **build.gradle** file and change **Application ID** with your new Package name and click on sync now & make sure you have to enter the same name here as Your Package Name.



#### 4. Change App Name, Applovin SDK Key, Privacy Policy Link etc

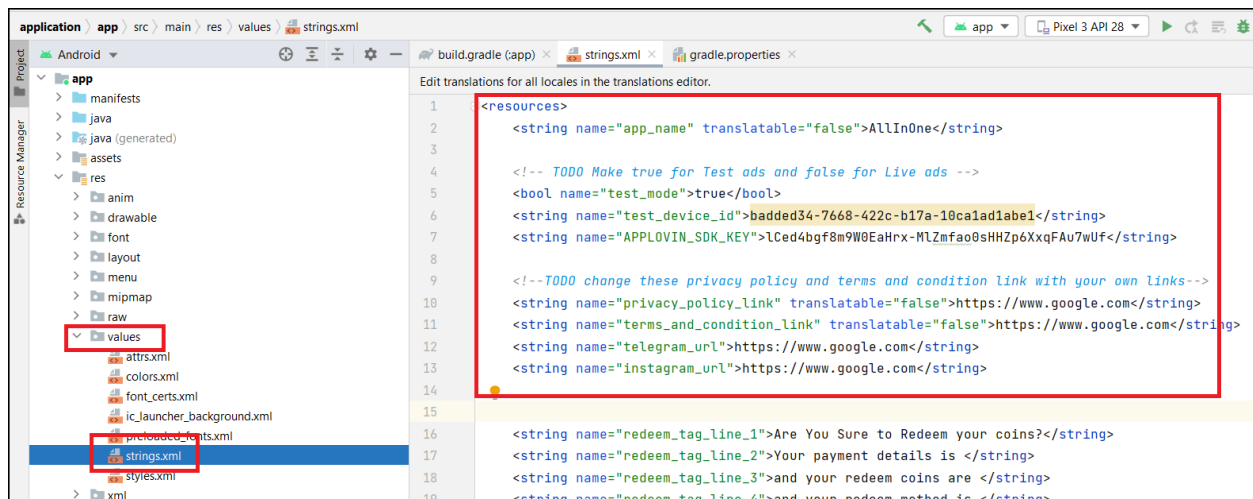
---

Step: 1

To change App Name, Applovin SDK Key, Privacy Policy Link, Social Profile Links and other details.

Goto **res > values > Strings > Strings.xml** file & change.

Change other details from the Admin Panel.



## 5. Change Application Logo

---

Step: 1

To change Application Logo **Rename** your Logo Name as ( ic\_launcher.png ) and make sure your logo name is the same as ic\_launcher.png otherwise you will get an Error. After you **rename** the logo right click on the **mipmap** folder inside the **res** folder and choose a new **image asset**.

Step: 2

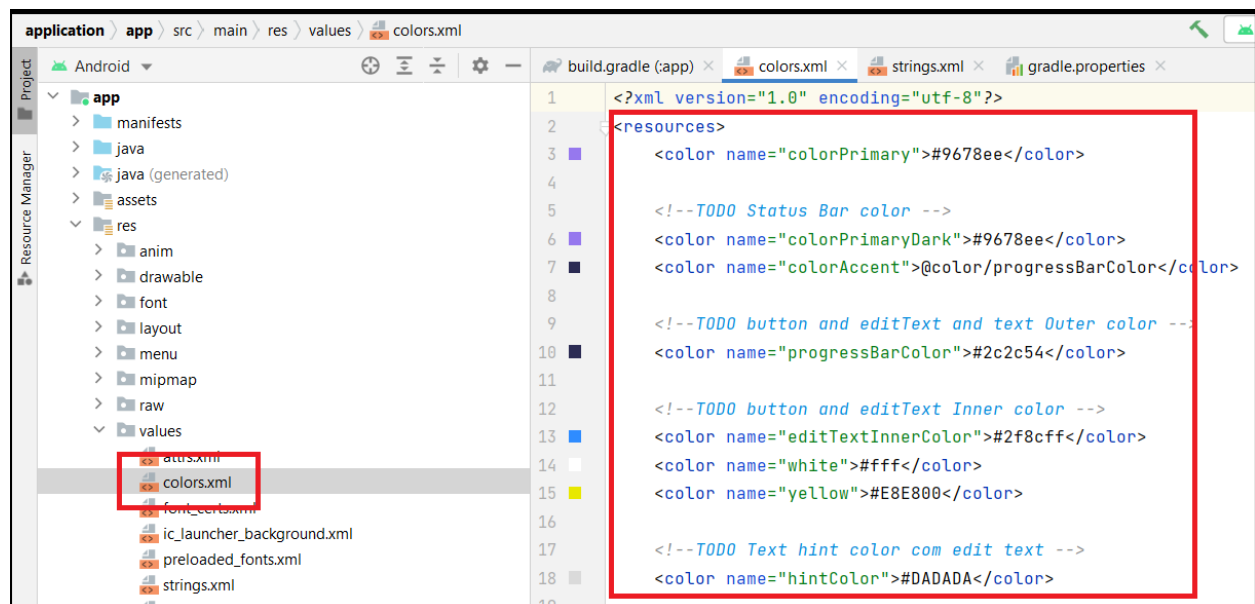
Copy your Logo from your computer and paste it in the drawable folder Inside **res** Folder make sure your logo name is ( ic\_launcher.png ) after paste click on override what be it said.

Note:- Your Logo must be 512x512 width and Height in png format.

## 6. Change application colors

---

To change the application colors Go to : values folder inside **res** folder and click on **colors.xml** file And change the application colors there.



## 7. Firebase Configuration

---

Now you need to config the firebase (firebase its a google service used to send notification)

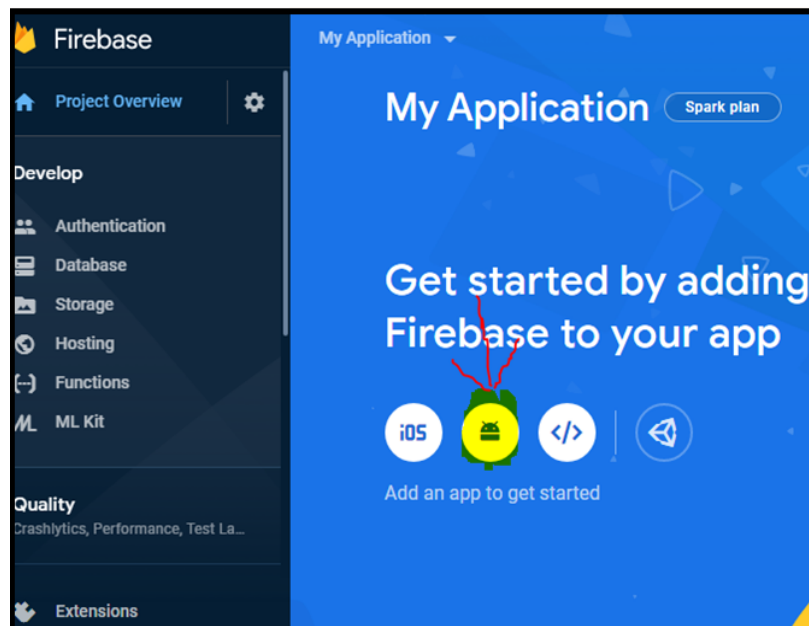
Step: 1

Go to your firebase account and **Add Project** then give a name to your project anything you want.

Step: 2

Now you need to add your application with Firebase Database just click on **Android Icon**.

Check the screenshot below.



Step: 3

Paste your package name here that you added In Android Studio. And give any of your application Nickname and click on the register app button.

× Add Firebase to your Android app

1

Register app

Android package name ⓘ

com.company.appname

App nickname (optional) ⓘ

My Android App

Debug signing certificate SHA-1 (optional) ⓘ

00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00

Required for Dynamic Links, Invites, and Google Sign-In or phone number support in Auth. Edit SHA-1s in Settings.

Step: 5

After Registering your app download this **Json** File It is very important to Connect your Application with Firebase and Click on the next button.

✓ Register app

Android package name: com.dynamicwallpapers, App nickname: My Application

2

Download config file

Download google-services.json

Switch to the Project view in Android Studio to see your project root directory.

Move the google-services.json file you just downloaded into your Android app module root directory.

google-services.json

Previous

Next

Instructions for Android Studio below | C++

Project Packages Scratches

MyApplication (~/.Desktop/MyApplication)

gradle

idea

app

build

libs

src

.gitignore

app.iml

build.gradle

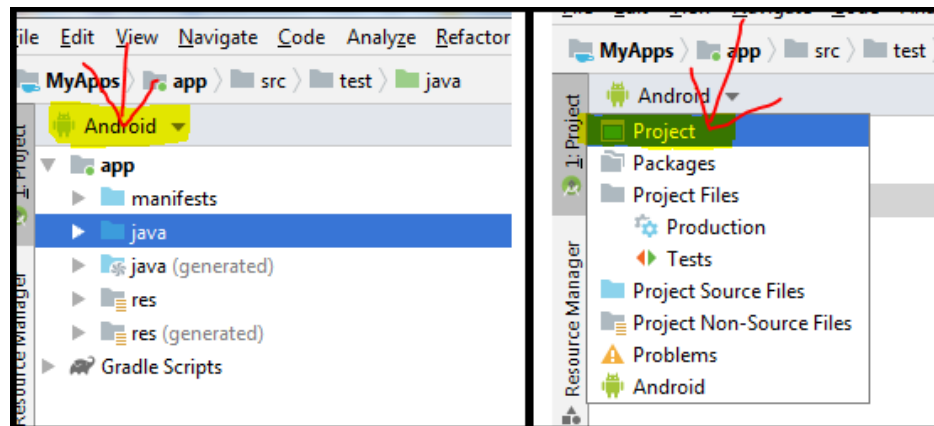
google-services.json

proguard-rules.pro

gradle

Step: 6

Go to Android Studio, click on **Android Icon** and Select Project Mode.



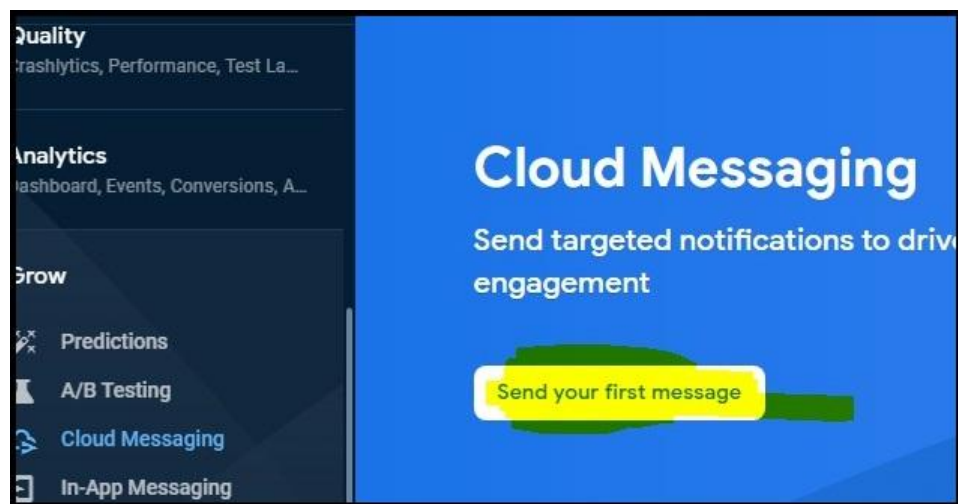
Step: 7

Delete existing json file And right click on App Folder & paste that json File from your Computer which you just downloaded from Firebase.

## 8. How To Send Notification To Users

---

Just Go to **Cloud Messaging** under **Grow** from the left side and type **Title & Message** you want to show users and in Target Option select App name and then click on review and publish it.

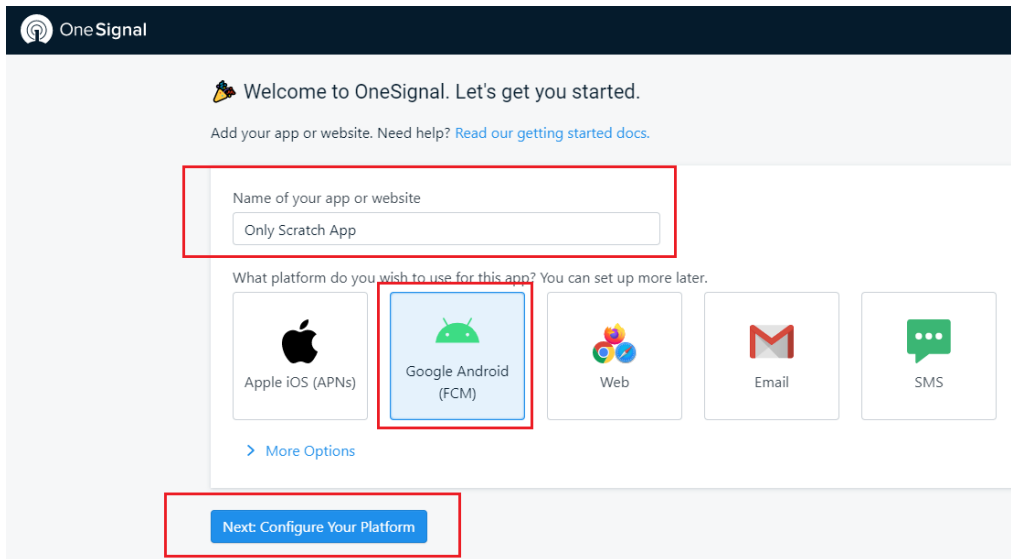




## 9. OneSignal Push Notification Setup

### Step: 1

Open your OneSignal account and Add App then select Google Android option.



The screenshot shows the OneSignal 'Add App' setup screen. At the top, there's a 'Welcome to OneSignal' message. Below it, a text input field for 'Name of your app or website' contains 'Only Scratch App'. Underneath, a question asks 'What platform do you wish to use for this app?'. Five platform options are shown: Apple iOS (APNs), Google Android (FCM), Web, Email, and SMS. The 'Google Android (FCM)' option is highlighted with a red box. At the bottom, a blue button labeled 'Next: Configure Your Platform' is also highlighted with a red box.

OneSignal

Welcome to OneSignal. Let's get you started.

Add your app or website. Need help? [Read our getting started docs.](#)

Name of your app or website

Only Scratch App

What platform do you wish to use for this app? You can set up more later.

Apple iOS (APNs)

Google Android (FCM)

Web

Email

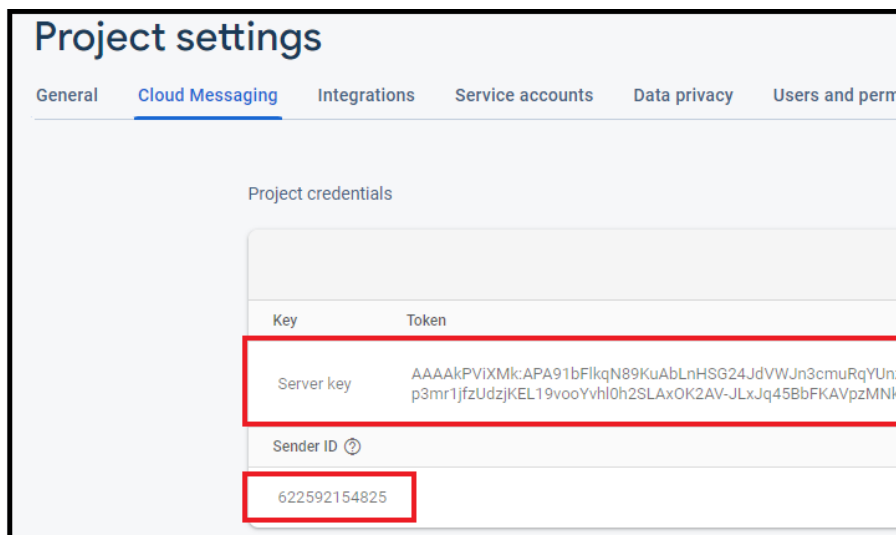
SMS

> More Options

Next: Configure Your Platform

### Step: 2

Open your firebase project and go to **Cloud Messaging** under **Project settings** then copy **Server Key** and **Sender ID** and paste it.



The screenshot shows the 'Project settings' page in Firebase, specifically the 'Cloud Messaging' tab. Under the 'Project credentials' section, there's a table with two columns: 'Key' and 'Token'. The 'Server key' is highlighted with a red box. Below the table, the 'Sender ID' is also highlighted with a red box.

Project settings

General Cloud Messaging Integrations Service accounts Data privacy Users and perm

Project credentials

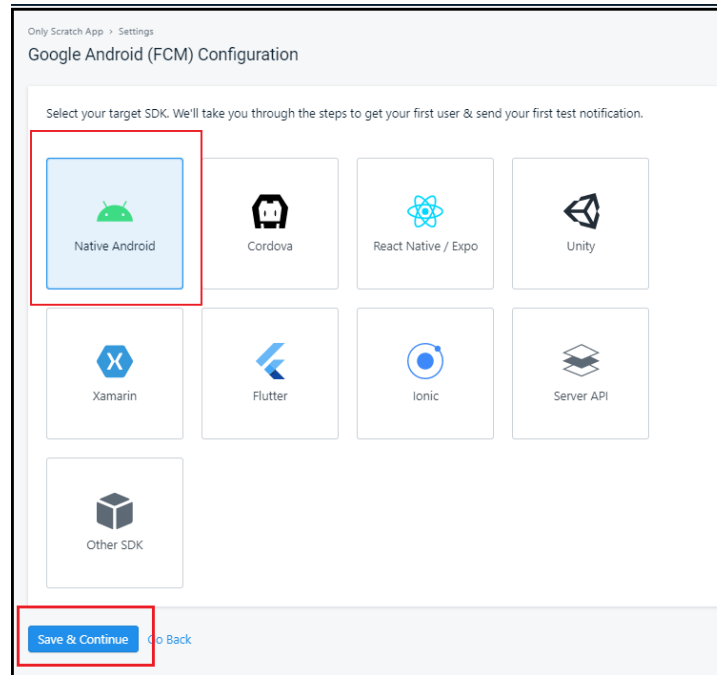
Key	Token
Server key	AAAAkPVIxMk:APA91bFlkqN89KuAbLnHSG24JdVWJn3cmuRqYUnz p3mr1jFzUdzjKEL19vooYvhl0h2SLAxOK2AV-JLxJq45BbFKAVpzMNMk

Sender ID ⓘ

622592154825

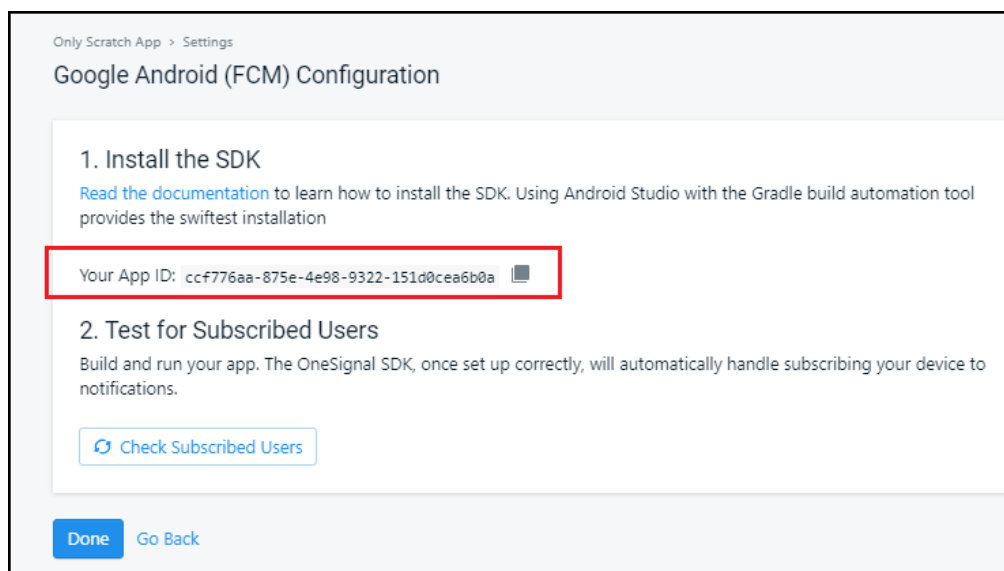
Step: 3

Select **Native Android** option and click on save & continue.



Step: 4

Now copy your OneSignal **App ID**.



Step: 5

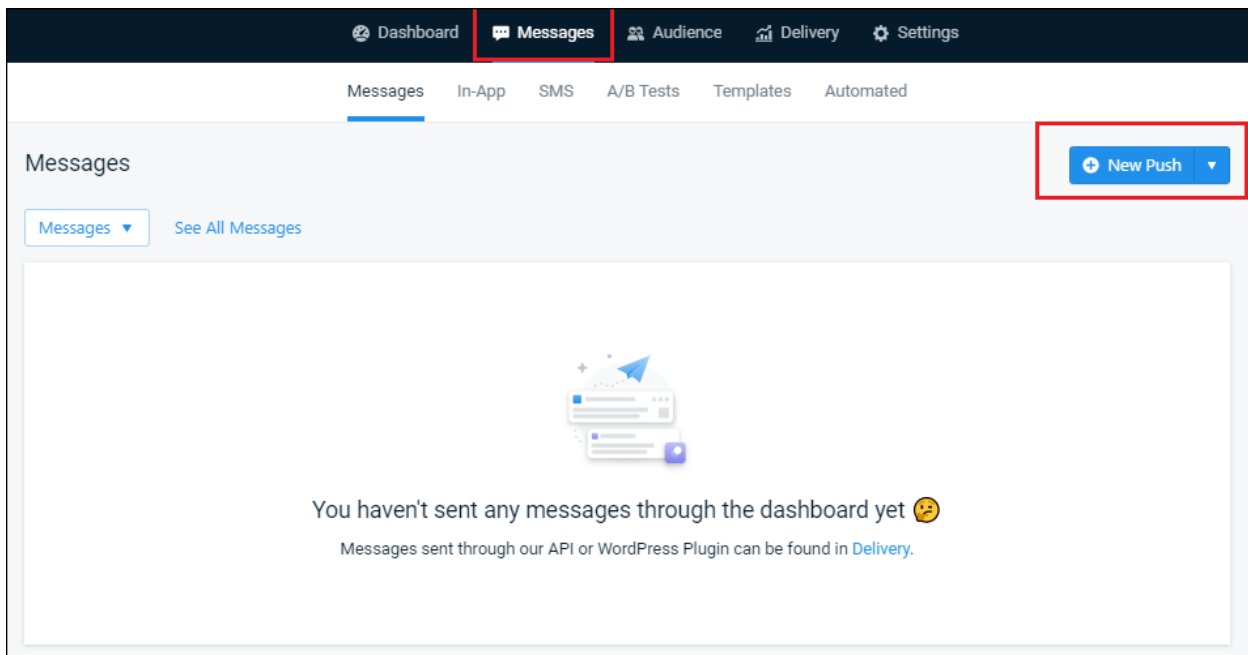
Open your app project in the Android Studio then Go to **App.java** file and paste your OneSignal **App ID** there.

```
19
20 public class App extends Application {
21     private static App mInstance;
22     public static final String TAG = App.class.getSimpleName();
23     private static final String ONESIGNAL_APP_ID = "ccf776aa-875e-4e98-9322-151d0cea6b0a";
24
25     private RequestQueue mRequestQueue;
```

Step: 7

# How to send Notification from OneSignal account.

Open your app in OneSignal account then go to the **Messages** option and click on New Push option. Enter app title, message text, external URL (optional) and send it. That's it.



## 10. Install Admin Panel on Server

---

# How to Install Admin Panel on Server:

Step: 1

Login to cPanel & create database. Database name, database user & password as well as assign all privileges to database user, you can refer below link for details description about setting database & database user.

<https://www.youtube.com/watch?v=CHwxXGPnw48>

Step: 2

Now upload the **admin\_panel.zip** file to your server's file manager and extract it. After that find the **connect.php** file change database name, database user name & database password.

Step: 3

Now go to the **php\_actions** folder and open the **fetch\_users.php** file and change database name, database user name & database password.

Step: 4

Now open **PHPMYADMIN** and go to your database which you created in the first step after that go to the **import** option and import the **database.sql** file from the Source Code folder.

Step: 5

When you completed above step's then your Admin Panel is live with location at <http://www.yourdomain.com/login.php>

Step: 6

Your **Admin Panel** default **Email & Password** for Login:

Email : admin@gmail.com

Password : 123456

## 11. How to set API URL

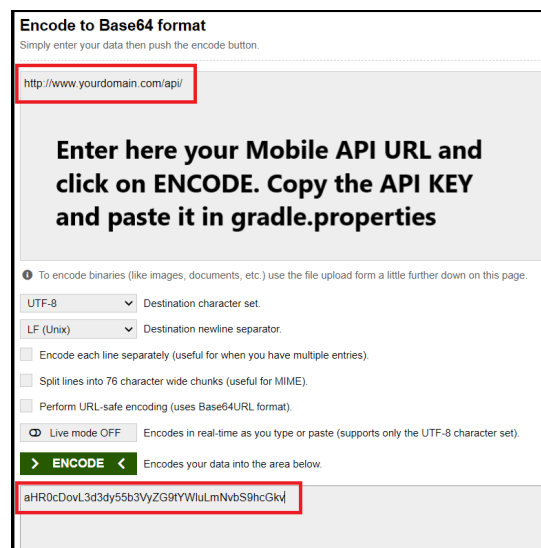
Your Mobile Api URL will be like: <http://www.yourdomain.com/api/>

Step: 1

Go to the website: <https://www.base64encode.org/>

Step: 2

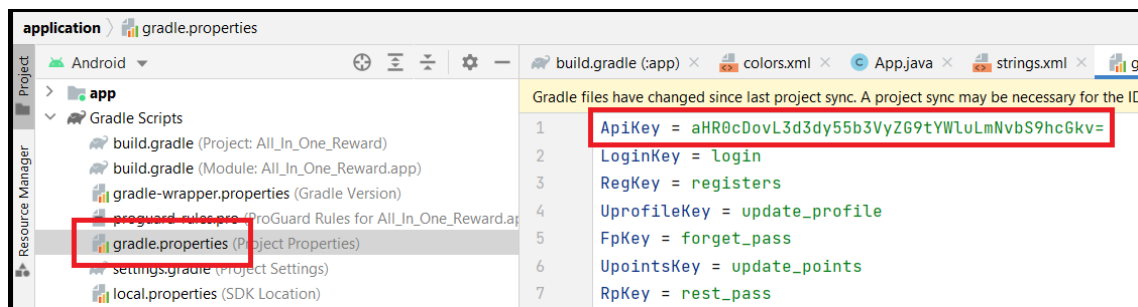
Enter your Mobile Api URL and click on the **ENCODE** button and then Copy API Key.



The screenshot shows the 'Encode to Base64 format' website. At the top, it says 'Simply enter your data then push the encode button.' Below this is a text input field containing the URL 'http://www.yourdomain.com/api/'. In the center, there is a large grey box with the text: 'Enter here your Mobile API URL and click on ENCODE. Copy the API KEY and paste it in gradle.properties'. Below this box are several options for encoding: 'Destination character set' (UTF-8), 'Destination newline separator' (LF (Unix)), 'Encode each line separately' (unchecked), 'Split lines into 76 character wide chunks' (unchecked), 'Perform URL-safe encoding' (unchecked), and 'Live mode OFF' (checked). At the bottom of these options is a green 'ENCODE' button. Below the button is a text output field containing the encoded API key: 'aHR0cDovL3d3dy55b3VyZG9tYWwLmNvbS9hcGkv'.

Step: 3

Go to **gradle.properties** and paste API Key.



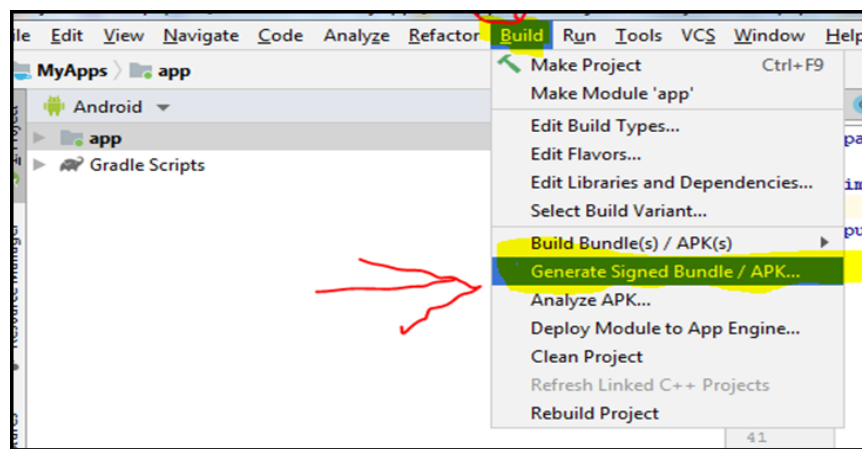
## 12. Generate Signed Apk

---

Lets See How to Generate Signed Apk Bundle For the Play Store. Apk Bundle is better than Apk because It reduces your app Size and It's lightweight and fast and also Google reminds users to upload Signed Apk Bundle to Playstore.

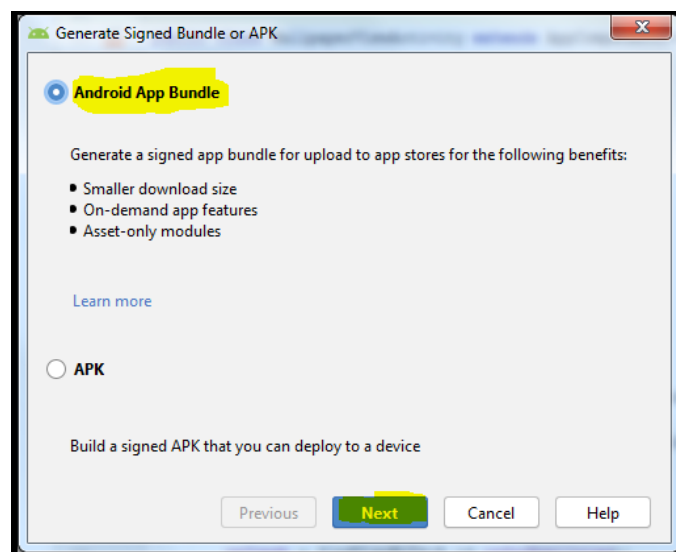
Step: 1

Click on **Build** and select Generate Signed Bundle/Apk.



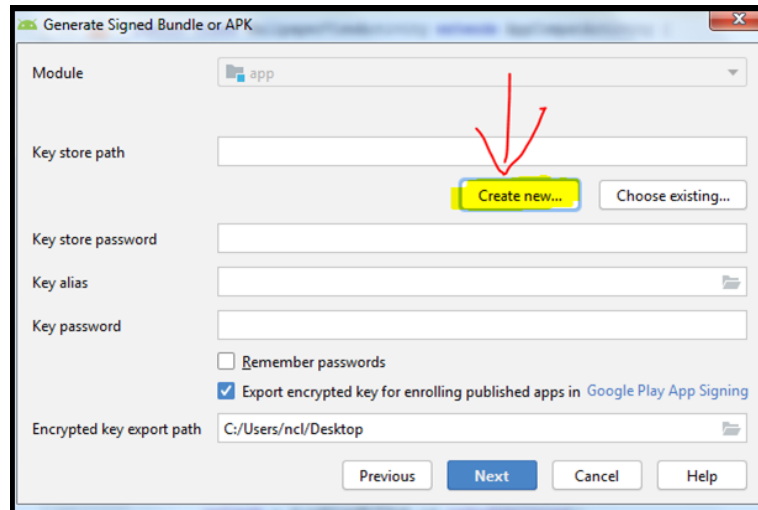
Step: 2

It's by default selected **Android App Bundle**, leave it default and click on Next button.



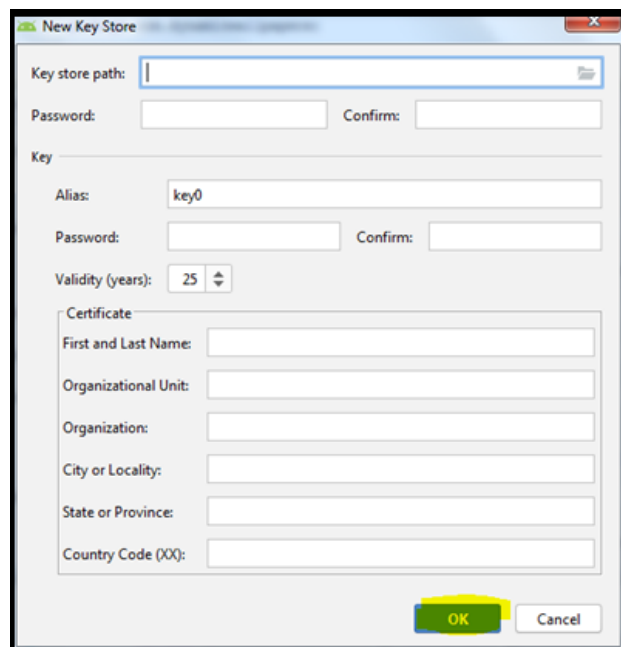
### Step: 3

Click on create New Key store path where you want to save the key file.



### Step: 4

Please check the screenshot below and fill in the details and make your key store path and click ok then click on the next button and select release mode and then click on the next button. Android Studio will automatically generate a Signed Apk Bundle for you And You can Upload It on Playstore.



**Thank You !**

If you like our app then please rate us **5 stars** on codecanyon :)

Copyright **AADevelopers** All Rights Reserved.