

ARPA STUDIO

Percentage Calculator

Documentation

Aneesh Anchan

12/14/2021

© Copyrights 2021 All rights reserved. No part of this publication may be reproduced in any material (including photocopy or storing in any medium by electronic means and whether or not transiently or incidentally to some other use of this publication) without the written permission of the copyright holder.

Table Of Content

	PageNo.
1. Introduction	3
2. Installation Plans	6
3. Opening the Project	7
4. Changes in the App (Package Name)	8
5. Firebase setup	5
6. Adding More Apps	6
7. Adding Contact Information	7
8. Managing your new updates	8
9. Admob Integration	9
10. Notification	10
11. Notification	11

Introduction

The very basic idea of this app came to help the students out with the calculation of their Marks percentage, Cgpa and Sgpa. Thus there born a calculator that solves various kinds of calculation related to percentage.

This is the Percentage Calculator app that is designed in a unique way that it can solve all your calculation related to your Percentage, CGPA, Discount, Sales tax and Marks in a very accurate and understandable way.

We have the following calculations offering are as follows:

- * A Percentage Calculator %
- * Marks % Percentage Calculator
- * CGPA Calculation
- * SGPA Calculator
- * GPA to % percentage calculator
- * Discount % percentage calculator
- * Sales Tax calculator
- * Tip calculator
- * Inflation Calculator
- * Doubling Time
- * Margin & Markup Calculator

% Percentage Calculator

In this Percentage Calculator app we offer various operation like general Percentage calculator % , Ratio to % percentage calculator Change in % percentage Calculator which will be helping you in various chores of life.

Marks % Percentage Calculator

We have designed a unique way for the calculations of Marks which allows you to add your obtained marks and total marks of subject. This will calculate your total % percentage obtained.

CGPA Calculation

The CGPA calculation will allow you to calculate your CGPA by entering your SGPA value for each semester. And giving you flexibility in your calculation.

SGPA Calculation

By adding each subject having the credit value the obtained marks in the subject and the total marks in the subject we can find the value of SGPA value with its percentage value for that semester.

GPA to % Percentage Calculator

By entering the value of the % percentage we can find the value of GPA and vice versa

Discount % percentage Calculator

We can enter each item with its actual value, total discount and number of item we can find the value total discount % percentage, total discount amount, total price of all items and the price at which the item is sold or bought.

Sales tax calculator

A sales tax calculator will allow you calculate sales tax percentage and tax amount by entering the respective values.

Tip Calculator

This Tip calculator will allow to calculate tip per person , total bill per person and tip amount based on the minimum required value entered.

Inflation Calculator

An inflation calculator shows what will be the worth of a quantity of money after a certain period of time. It also shows what will be the worth of the same amount of money if invested.

Doubling Time

Doubling Time Calculator will calculate for how much time it will take for something for grow by 100% when it grows by some percentage % each time period.

For example, if one is investing 100\$ with a yearly returns of 10% one will have 200\$ after 7 years and 3 months.

Margin and Markup Calculator

Margin and Markup Calculator will be helpful for your business that will measures how much of every dollar in sales you keep after paying expenses. In the margin calculation example above, you keep \$0.25 for every dollar you make. The greater the margin, the greater the percentage of revenue you keep when you make a sale. For Markup Calculator is a business tool most often used to calculate your sale price. Just enter the cost and markup, and the price you should charge will be computed instantly. It can also be used to calculate the cost - in this case, provide your revenue and markup.

Along with that we can have a feature that will allow to marks some favourites calculations, which will be helpful for you for easy access.

% percentage Calculators strong feature is that just by entering the value that you know the remaining one will be found automatically. That means any value can be the source or the result for calculation.

Installation Plans

The image displays three pricing cards for app installation services. Each card is a rounded rectangle with a header, a list of features, and a price at the bottom.

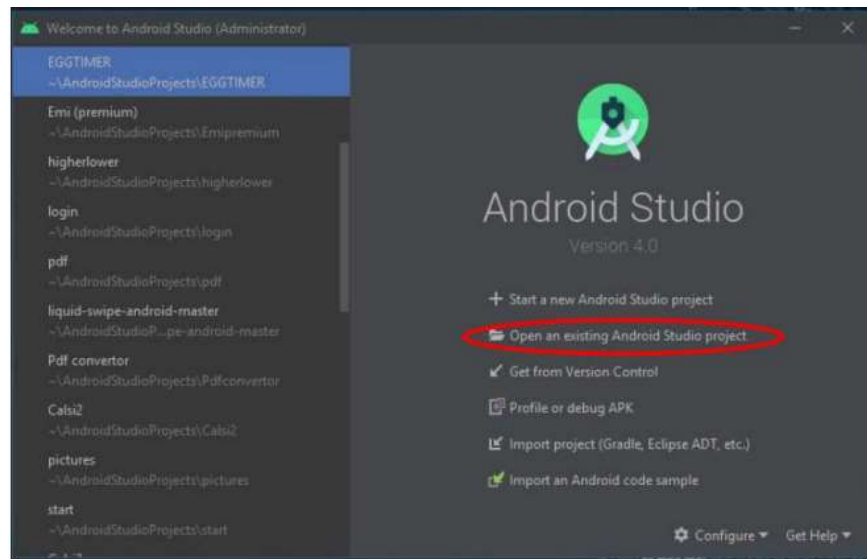
- Free Plan:** Indicated by a green header. Features include basic bug fixing, Q&A support, and no customization or installation service.
- Basic Plan:** Indicated by a red header. Features include changing the app icon, name, and splash screen; changing the package name; and changing the color. The price is \$20, shown in a red footer.
- Pro Plan:** Indicated by a red header. Features include changing the app icon, name, and splash screen; changing the package name; changing the color; Firebase and Admob integration; and adding to the Playstore. The price is \$35, shown in a red footer.

Plan	Features	Price
Free	<ul style="list-style-type: none">Basic Bug Fixing and crash fixing.Basic Q&A support.No customization & installation service	\$0
Basic Plan	<ul style="list-style-type: none">Change App icon, name and splash screen.Change Package name.Change Color.	\$ 20
Pro Plan	<ul style="list-style-type: none">Change App icon, name and splash screen.Change Package name.Change Color.Firebase and Admob integration.Add to Playstore.	\$ 35

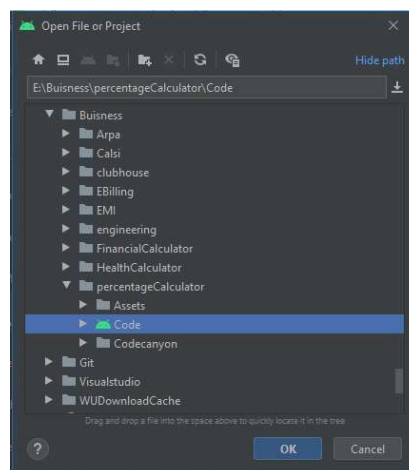
*If you need to design new icon and graphics and also you have to do major changes, the price will increase

Opening the Project

The purchased project files contain documentation, .apk file for the app and the source code for the app. For opening in android studio open your Android Studio IDE. Goto **Open an existing Android Studio project**.



Open the Code file from the project file you purchased and saved. PercentageCalculator > Code and press ok.



Let android studio takes its own time.

Changes in the App

What is the significance of the Package name?

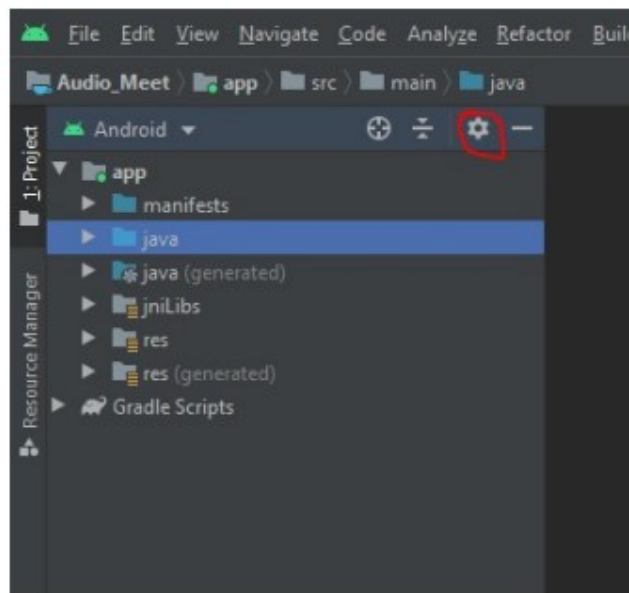
This is the unique name through which your app is identified. It is used by the Google Playstore to identify your app and also by the firebase for connecting with the app.

Hence it must be unique and identifiable.

How to change it?

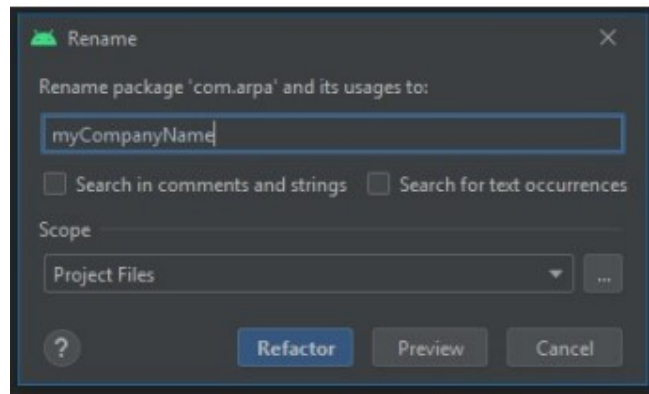
Here we go.

Click on setting in on top right corner and inside it uncheck Compact Middle Packages.



Right click on any package folder inside java>com you wish to change. Click on Refactor > Rename.

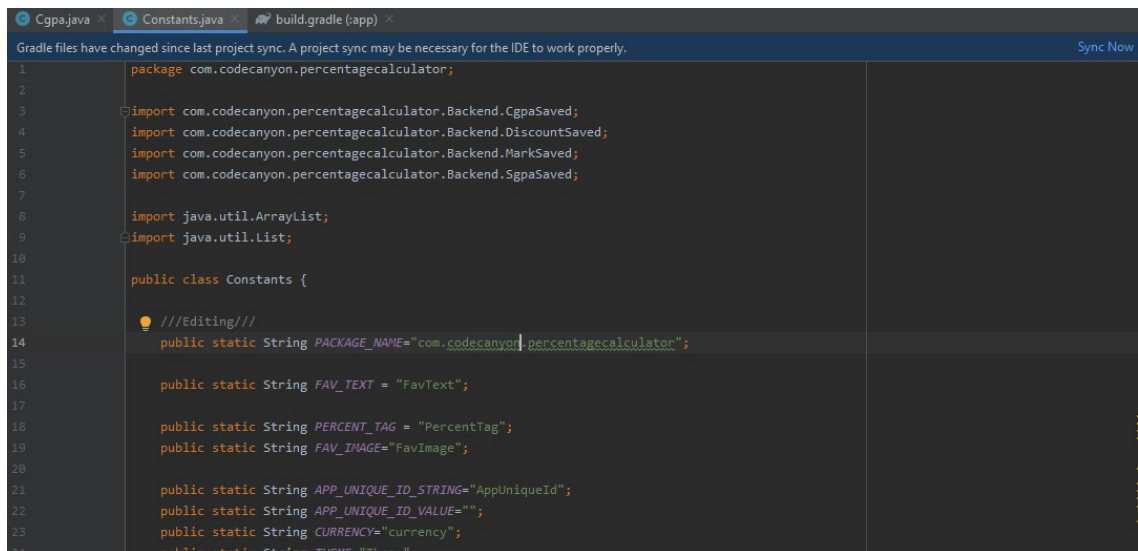
You will get a dialog like this, Enter desired name. And click on Refactor.



And after Refactor, Goto Gradle Script > build.gradle(Module:app) and make the changes as per your Package name. And then press sync button.

```
apply plugin: 'com.android.application'
apply plugin: 'com.google.gms.google-services'
android {
    compileSdkVersion 29
    defaultConfig {
        applicationId "com.arpa.clubhouse"
        minSdkVersion 18
        targetSdkVersion 29
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner 'androidx.test.runner.AndroidJUnitRunner'
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
    dependenciesInfo {
        includeInApk true
        includeInBundle true
    }
}
```

Now in Constants.java file line 15 add the changed package Name.



```
1 package com.codecanyon.percentagecalculator;
2
3 import com.codecanyon.percentagecalculator.Backend.CgpaSaved;
4 import com.codecanyon.percentagecalculator.Backend.DiscountSaved;
5 import com.codecanyon.percentagecalculator.Backend.MarkSaved;
6 import com.codecanyon.percentagecalculator.Backend.SgpaSaved;
7
8 import java.util.ArrayList;
9 import java.util.List;
10
11 public class Constants {
12
13     ///Editing///
14     public static String PACKAGE_NAME="com.codecanyon.percentagecalculator";
15
16     public static String FAV_TEXT = "FavText";
17
18     public static String PERCENT_TAG = "PercentTag";
19     public static String FAV_IMAGE="FavImage";
20
21     public static String APP_UNIQUE_ID_STRING="AppUniqueId";
22     public static String APP_UNIQUE_ID_VALUE="";
23     public static String CURRENCY="currency";
24     public static String THEME="Theme";
25 }
```

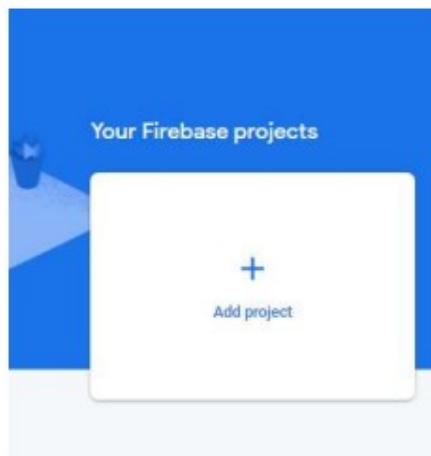
Failure in any of the steps can lead to following problems:

1. Error in each file (due to refactor not done correctly).
2. Error while opening the app (package name not added in Constants.java or build.gradle)

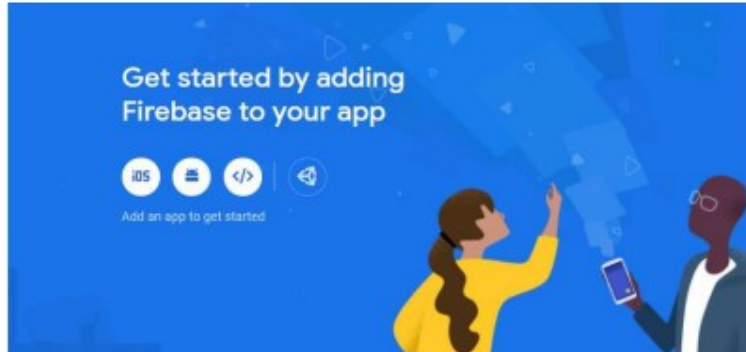
FireBase SetUp

SignUp to your Firebase account.

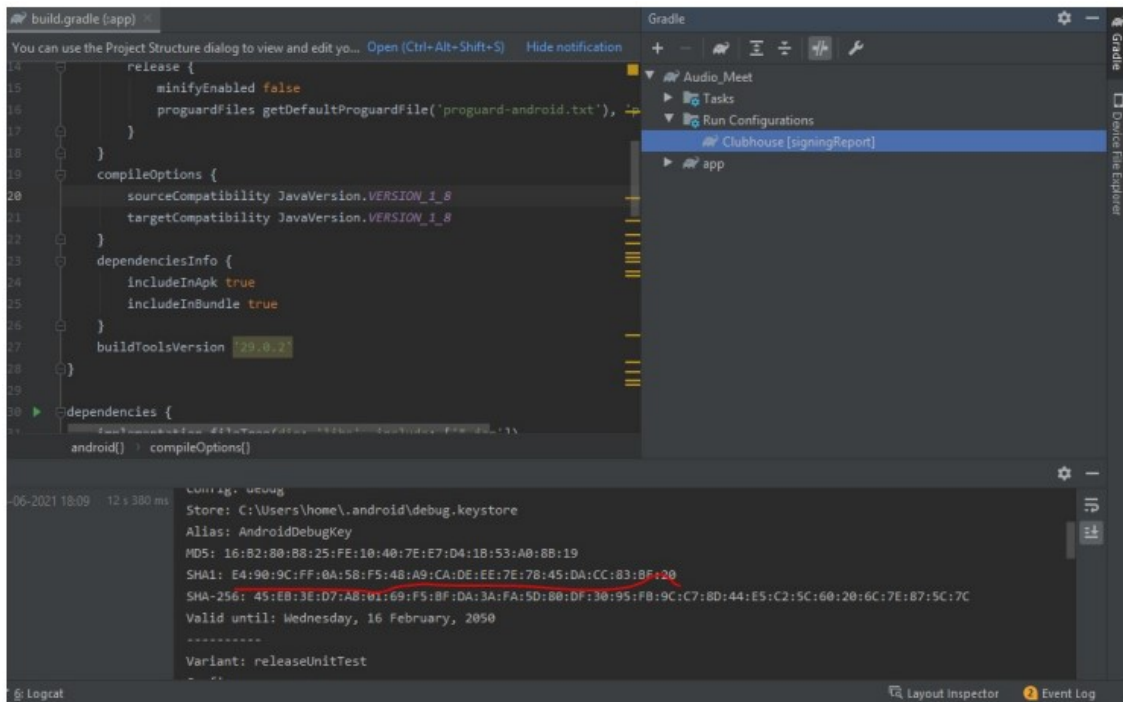
And create new project.



After Creating Project, In Project overview add android app



Add the changed Package name and studio Gradles in the left corner and press on Run Configura signing Report. And copy the SHA1: par our Firebase account. In Project Overview add android app he changed Package name and your app Nick name. For SHA 1 goto Android corner and press on Run Configurations. And press on he SHA1: part. As shown below.



Note: change that bellow to app (because when you will run it won't create error).

{In Firebase}

Download google-services.json

{In Android Studio}

Now Click On Android and select Project.

Now paste the downloaded google-services.json in src.

And then press next and continue to Console.

Now GO TO Real TIME Database and press create Database.

GO TO Rules and make

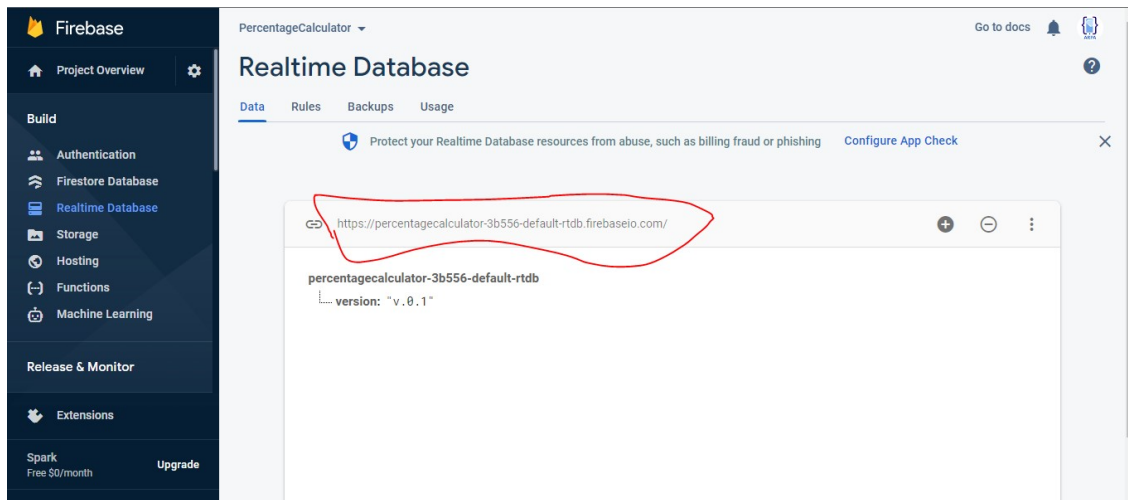
```

".read": true,

```

And then click Publish

In the Realtime database copy the bellow link



Now open Android studio

In Constants paste the Copied link in Constant.java

In line number 44

Now go to Storage and press get Started.

In the Storage Upload your app Icon

Now go to Firestore and press get Started.

Now in the rule section make **true** for allow read and then publish.

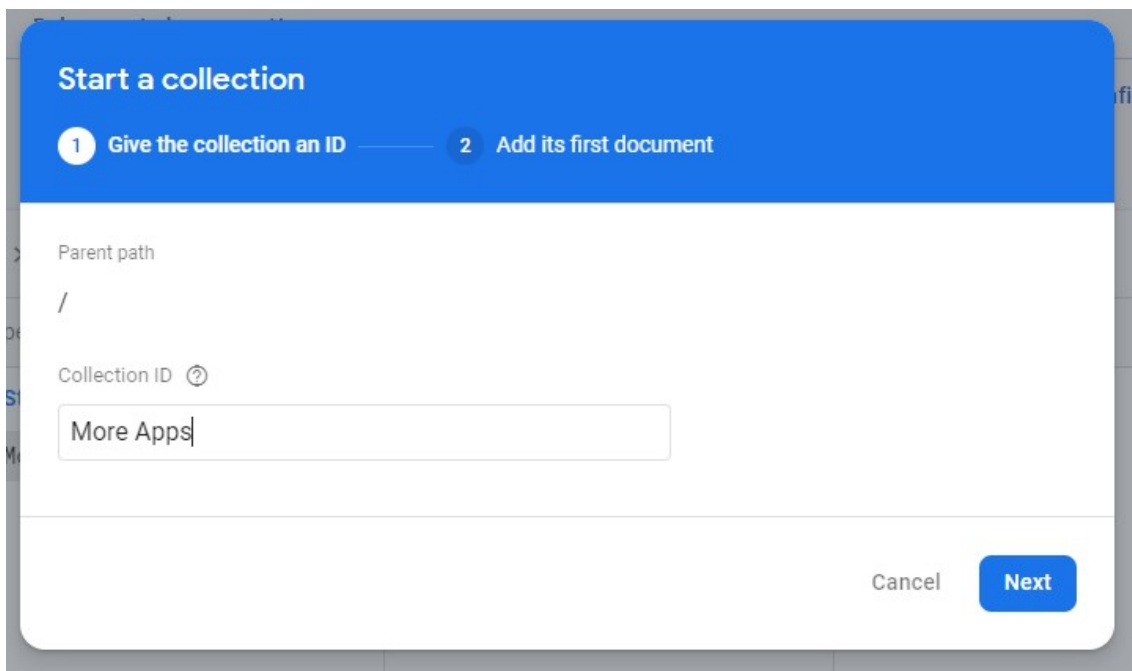
Adding More Apps

Goto FireBase

Now goto Firestore Database

Press Start Collection and write `More Apps`

Spelling errors will create crashes.



Start a collection

1 Give the collection an ID — 2 Add its first document

Parent path
/

Collection ID ?
More Apps

Cancel Next

Now press Next.

Add your first app. In the Document Id add the other app name

In the field section add icon, and paste link of the icon in Value.

Other field section add playstoreLink , and paste link of the playstore in Value.

Other field section add shortDiscription , and the description that you want to display in the app.

Document parent path ?
/More Apps

Document ID ?
Calsii

Field	Type	Value
icon	= string	<input type="text"/>
shortDiscription	= string	<input type="text"/>
playstoreLink	= string	<input type="text"/>

+ Add field

Cancel Save

And save.

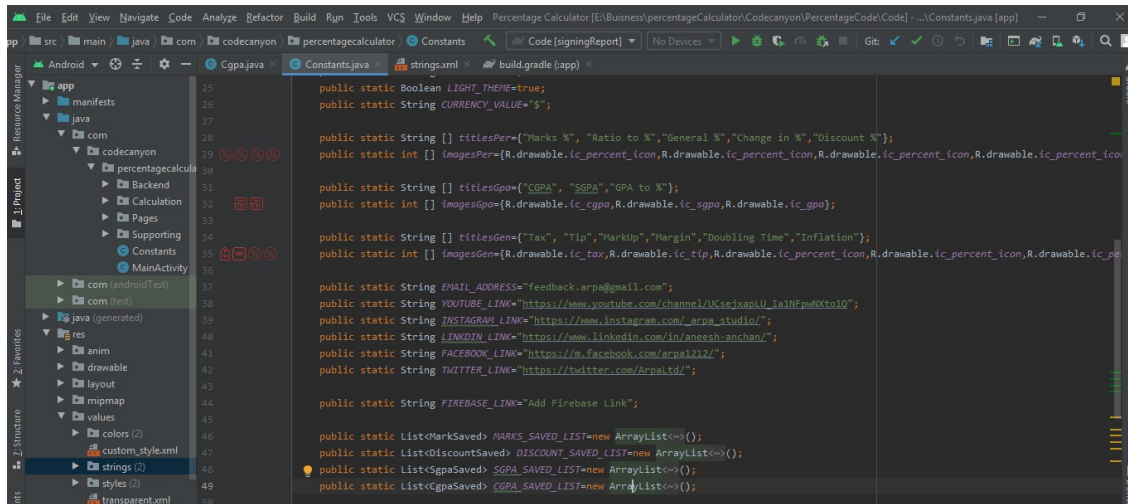
the field should not have spelling error it should be same as above shown.

Now to add other app.

Click on Add Document and repeat the same process for the app.

Adding Contact Information

In the Constant.java from line 37 to 42 change email, instagram Link, app all your information

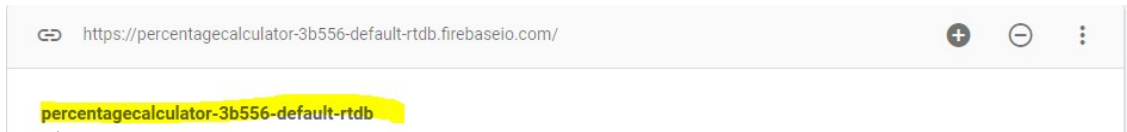


Now to go to res> values> strings.xml

Change the email with your email Address and the appLink with your appLink.

Managing the updates

Now open the firebase inside that open realtime database.

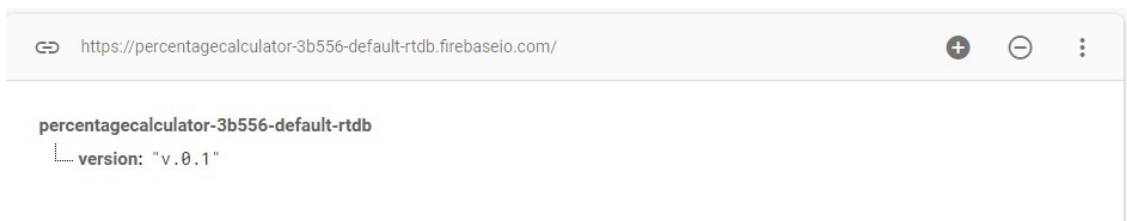


Move the cursor on to the highlighted portion and then click on plus button

In the Name write **version** and in value “v.0.1”(which is same as in string.xml line 33)

And press enter

It will look something like this



Now, when you are updating your app change the version in string.xml line 33 and in build.gradle(Module:app) the versionCode and versionName

After the update is live in the google playstore goto firebase and realtime database and change the version to the one that you have changed.

Admob Integration

Go to res > values > strings.xml

Replace the default id with banner_ads, native_ads and Interstial_ads.

*Note if your testing the app then use test ads and while publishing replace it with your placement ids.

Test ids

Banner ads: ca-app-pub-3940256099942544/6300978111

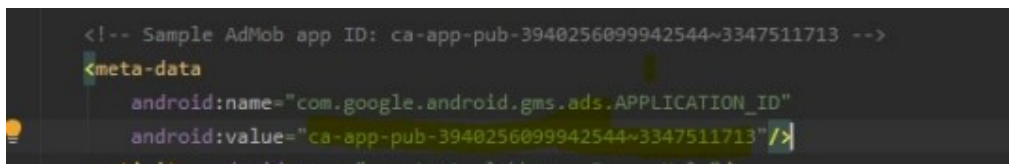
Native ads: ca-app-pub-3940256099942544/2247696110

Interstitial ads: ca-app-pub-3940256099942544/1033173712

A screenshot of an IDE showing the strings.xml file. The file contains XML code for app resources. Line 33 shows a version string. Lines 36-39 show AdMob ad IDs for interstitial, banner, rewards, and native ads respectively. The AdMob IDs are the same as those listed in the text above.

```
32
33 <string name="version">v.0.1</string>
34
35 <!-- admob ads-->
36 <string name="inter_ads">ca-app-pub-3940256099942544/1033173712</string>
37 <string name="banner_ads">ca-app-pub-3940256099942544/6300978111</string>
38 <string name="rewards_ads">ca-app-pub-3940256099942544/5224354917</string>
39 <string name="native_ads">ca-app-pub-3940256099942544/2247696110</string>
40 </resources>
```

After that go to manifests > AndroidManifesto.xml there you replace the highlighted with your app id.

A screenshot of an IDE showing the AndroidManifesto.xml file. The file contains XML code for app metadata. The android:value attribute is highlighted in yellow, indicating where to replace the app ID.

```
<!-- Sample AdMob app ID: ca-app-pub-3940256099942544~3347511713 -->
<meta-data
    android:name="com.google.android.gms.ads.APPLICATION_ID"
    android:value="ca-app-pub-3940256099942544~3347511713"/>
```

Notification

Goto Firebase and goto Messaging

Press New Campaign and then notification.

Add all the necessary data and then press next select your app package and then schedule the time and then filling necessary information and then press review and then publish.

A notification has been send successfully.