

# Android Studio

## Table of Contents

Getting started.....2

Open Project.....2

Firebase .....3

Application ID.....7

Change Logo..... 9

Change Color..... 11

Change Text..... 12

Change Images ..... 13

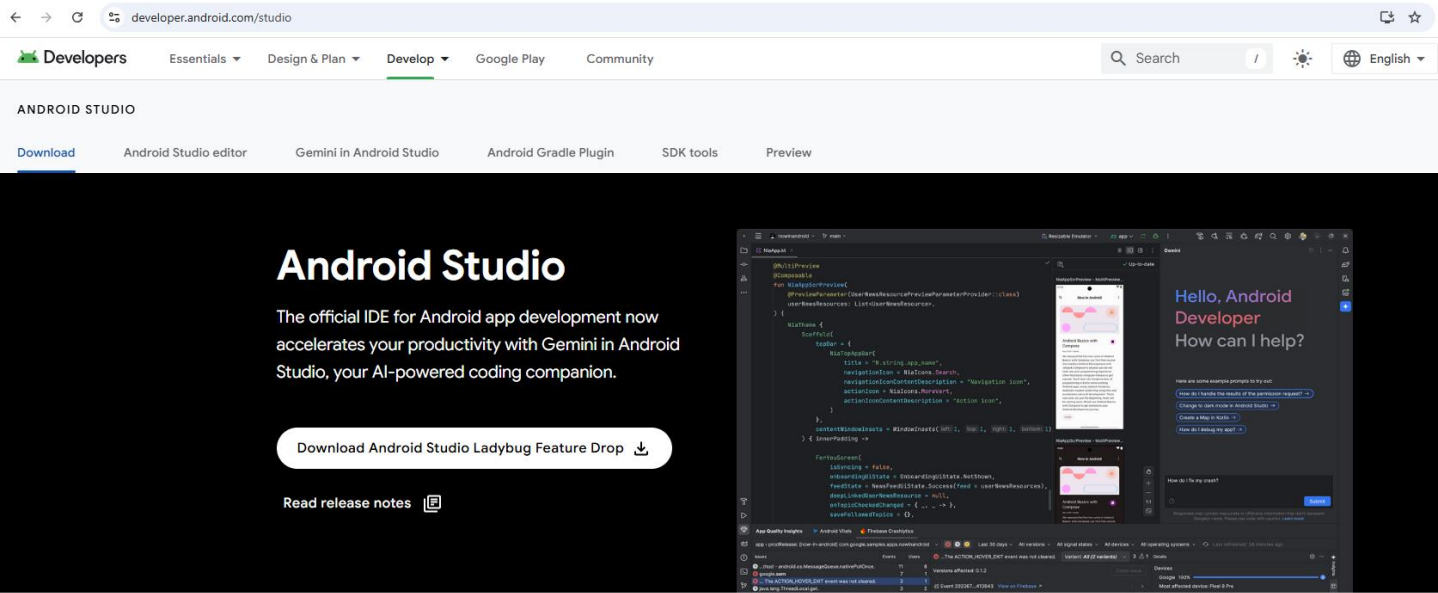
Change Font.....14

# Getting started

## Download and Install Android Studio

You can download latest Android Studio from this url :

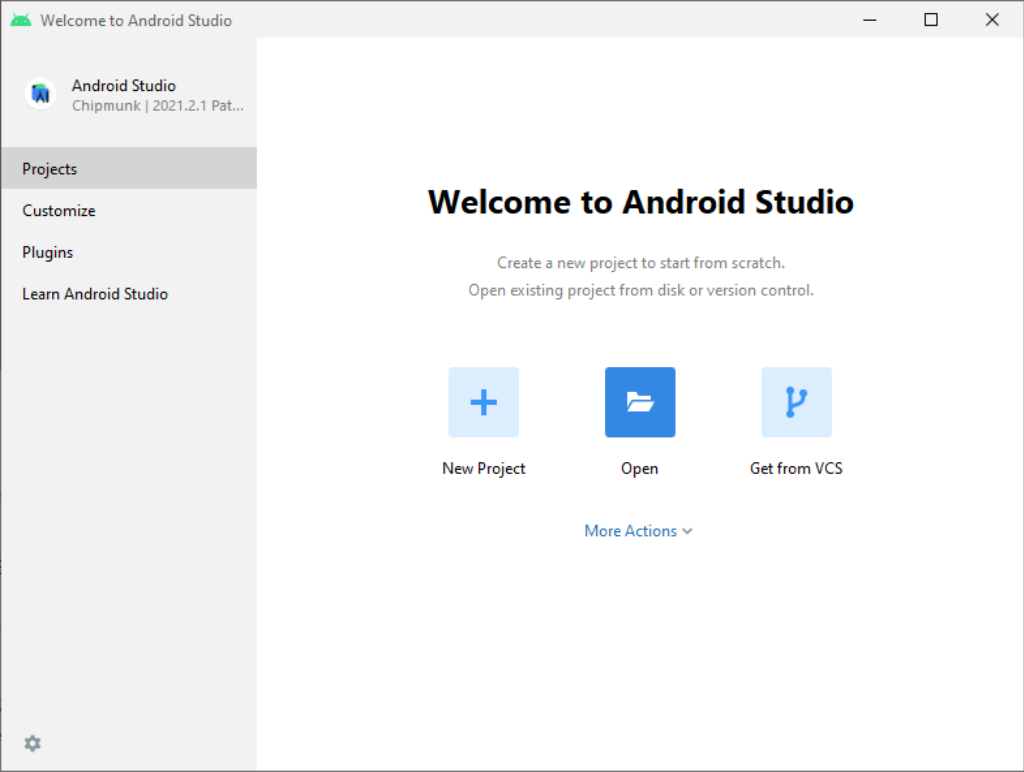
<https://developer.android.com/studio/>



# Open Project

Extract **.zip** you download from Codecanyon, you will see some file and folder. We named project folder with **"TheSingleStream"**, the project location inside **android\_studio** folder.

When your Android Studio ready you can open it and you will see display like this :

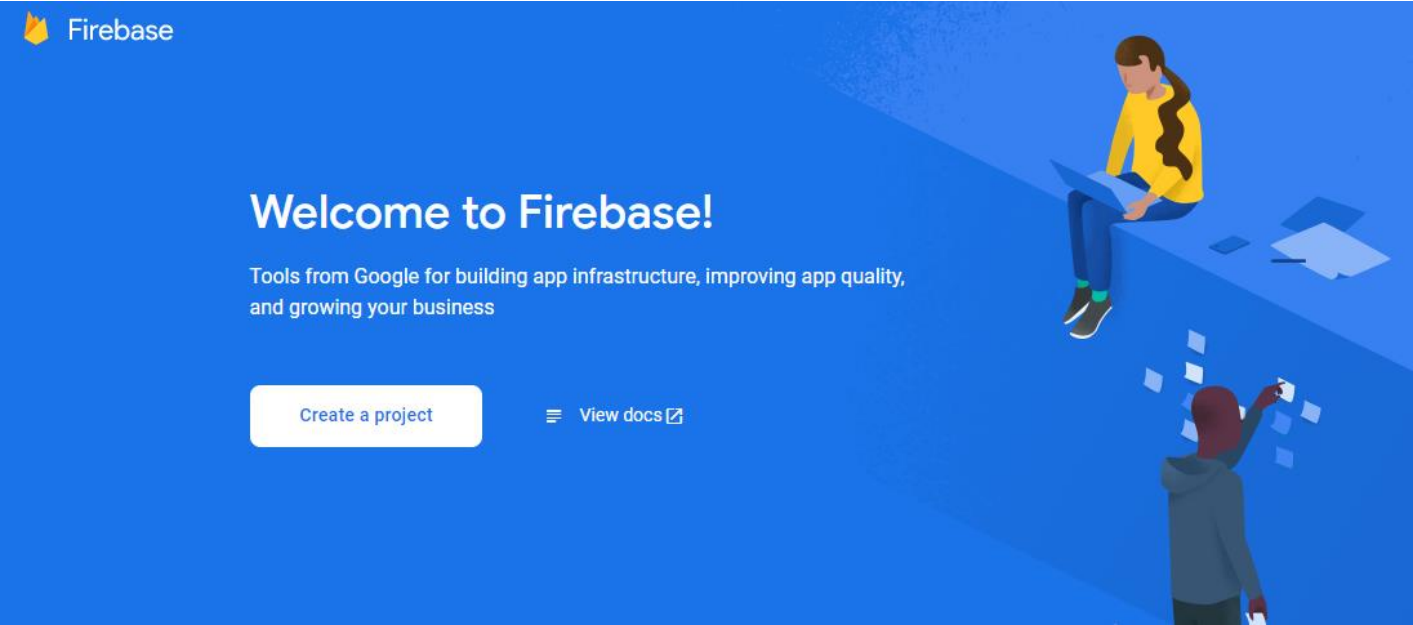


1. select **"Open"**.
2. Browse location Project and press **"OK"** button.
3. Wait for few minutes until all process import has finished.

# Firestore

Before you do step change the package name, you must do this step for firestore project configuration

Please note : this step is mandatory to generate **google-services.json** file



Visit [Firestore Console](#)

Please register or log in first and then select create new project. In the dialog window enter the name of the project that will be created and the contents of the state and then click the create project to continue.

×

Create a project (Step 1 of 3)

Let's start with a name for your project<sup>®</sup>

Project name

My Android Project

✎

my-android-project-fc98a

☒

I accept the [Firestore terms](#)

☒



I confirm that I will use Firestore exclusively for purposes relating to my trade, business, craft, or profession.

Continue

## Google Analytics for your Firebase project

Google Analytics is a free and unlimited analytics solution that enables targeting, reporting, and more in Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing, and Cloud Functions.

Google Analytics enables:

-  A/B testing ⓘ
-  Crash-free users ⓘ
-  User segmentation & targeting across  
Firebase products ⓘ
-  Event-based Cloud Functions triggers ⓘ
-  Free unlimited reporting ⓘ

☒ Enable Google Analytics for this project  
Recommended

[Previous](#)

Continue

## Configure Google Analytics

Analytics location ⓘ

United States ▾

Data sharing settings and Google Analytics terms

- ☒ Use the default settings for sharing Google Analytics data. [Learn more](#) ⓘ
- ✕ Share your Analytics data with Google to improve Google Products and Services

✓ Share your Analytics data with Google to enable Benchmarking

✓ Share your Analytics data with Google to enable Technical Support

✓ Share your Analytics data with Google Account Specialists

☒ I accept the [Google Analytics terms](#) ⓘ

Upon project creation, a new Google Analytics property will be created and linked to your Firebase project. This link will enable data flow between the products. Data exported from your Google Analytics property into Firebase is subject to the Firebase terms of service, while Firebase data imported into Google Analytics is subject to the Google Analytics terms of service. [Learn more](#) ⓘ.

[Previous](#)

Create project

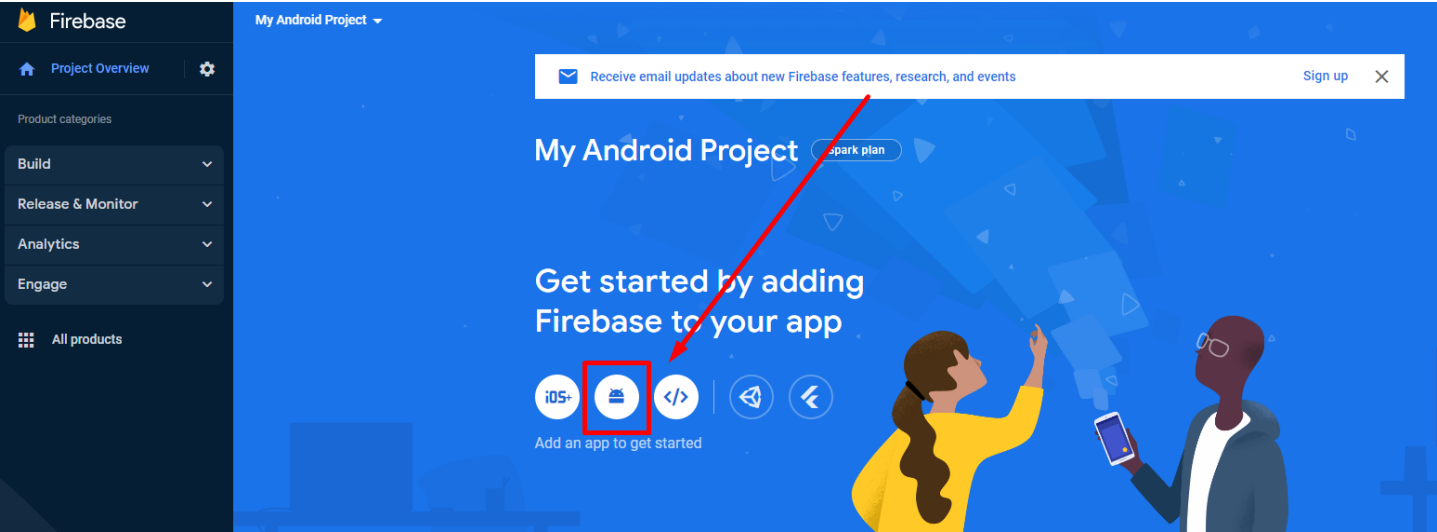


My Android Project

✓ Your new project is ready

Continue

After successfully creating a project on Firebase then the page will be redirected to the dashboard project that has been made.



Then on the dashboard Firebase select Add Firebase to your Android app. In the next dialog window enter the name of the Android application package and SHA1 of fingerprints Certificate (optional).

[illegible]

Next will download the configuration file json **google-services.json** automatically, which will be added in the directory app in the Android project.

× Add Firebase to your Android app

✓ Register app  
Android package name: com.app.webdroid, App nickname: WebDroid

2 Download and then add config file  
Instructions for Android Studio below | [Unity](#) [C++](#)

Download google-services.json

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

Move your downloaded google-services.json file into your module (app-level) root directory.

google-services.json

Next

3 Add Firebase SDK

4 Next steps

Copy the **google-services.json** file that you have downloaded and put it in the Android Studio project.

TS TheSingleStream\_1.0.0 Version control

Project

TheSingleStream [TheSingleStream\_1.0.0] C:\Solodi

- .gradle
- .idea
- app
  - build
  - libs
  - src
  - .gitignore
  - build.gradle
  - google-services.json
  - proguard-rules.pro
- build
- gradle
- .gitignore
- build.gradle
- gradle.properties

# Application ID

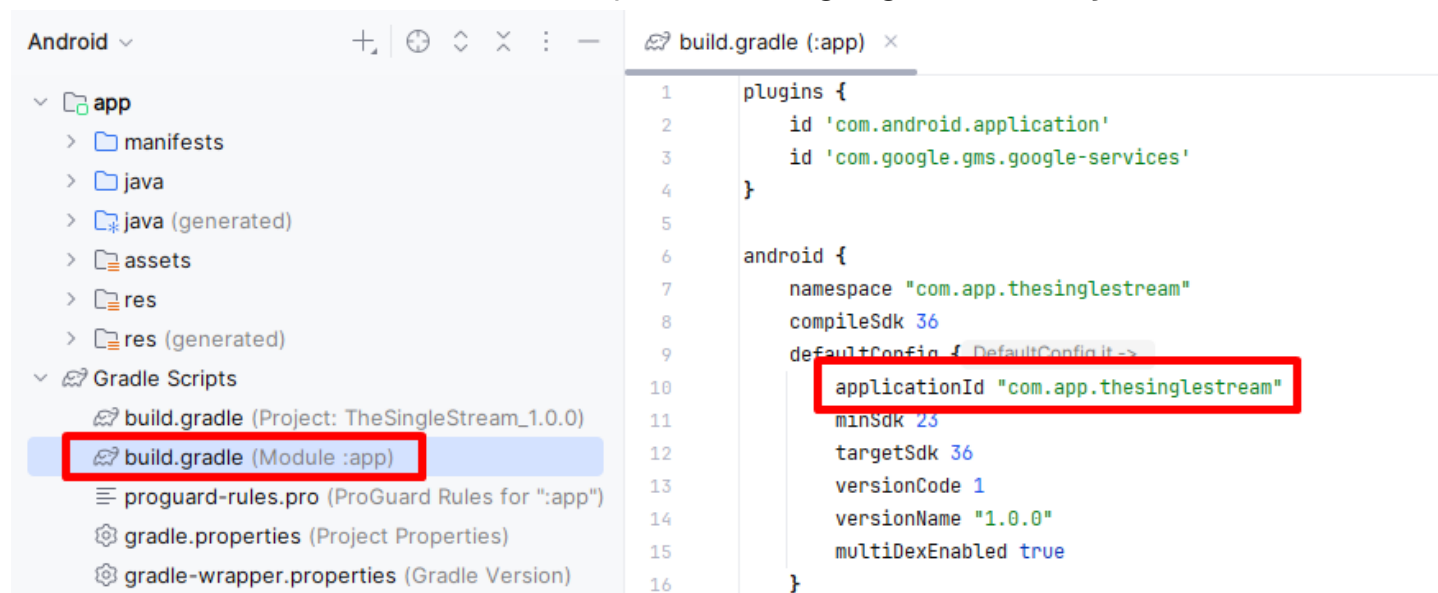
## applicationId (Required)

**applicationId** is the main domain of your application. The applicationId is not related to the package name, although some application developers name the applicationId and the package name the same.

Each application to be published must have a unique domain name, generally using 3 word arrangements (can be more than 3), example: **com.app.thesinglestream**, **com** is identified as the domain, **app** as the name of the developer and **thesinglestream** as the name of the application.

1. Open **Gradle Scripts > build.gradle (Module: app)**
2. Change the application ID with your own id name
3. Click **Sync Now**.
4. Change the id as unique as possible, because application id is very important used if you want to publish the application to the google play.

Important : your **applicationId** must be the **same** as the **Android package name** you created in the firebase console when you **created google-services.json**

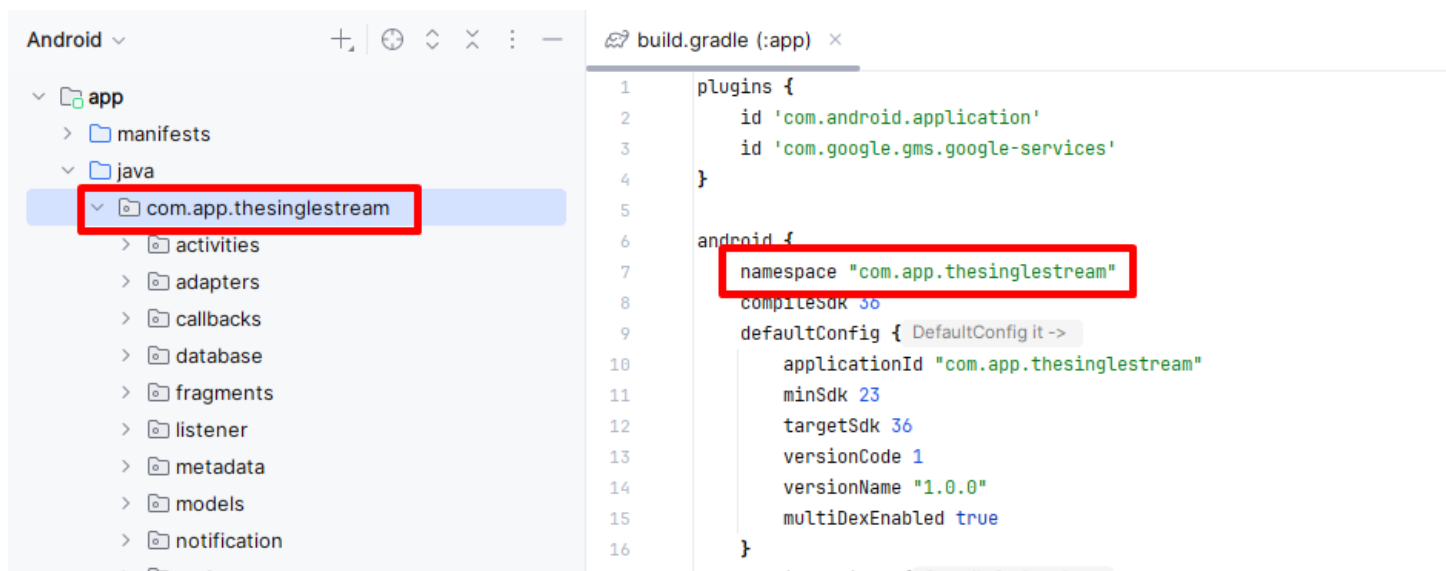
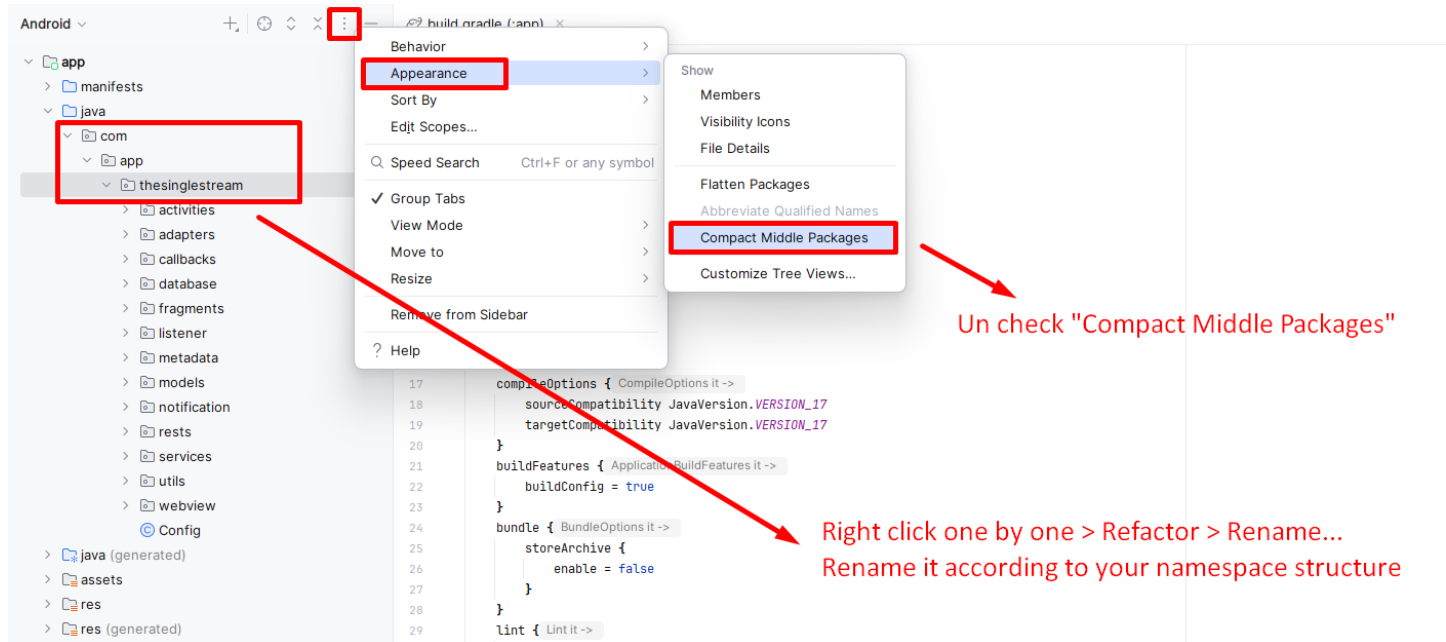


## Package name and namespace (optional)

Package name is the folder path to store class files. The package name is not required to be changed, so it uses the default source code. But we recommend changing it according to the applicationId structure, if you want to change it:

1. Click the settings icon in Android Studio
2. Select **Tree Appearance, remove the checklist** in the **Compact Middle Packages** section
3. Select one of the sub packages then **right click > Refactor > Rename...**
4. When finished, return the checklist to Compact Middle Packages so that the package name structure looks neater again

Important: the structure of the **package name** must be **the same** as the **namespace** in the **build.gradle** file, otherwise the project will be error



## Invalidate caches

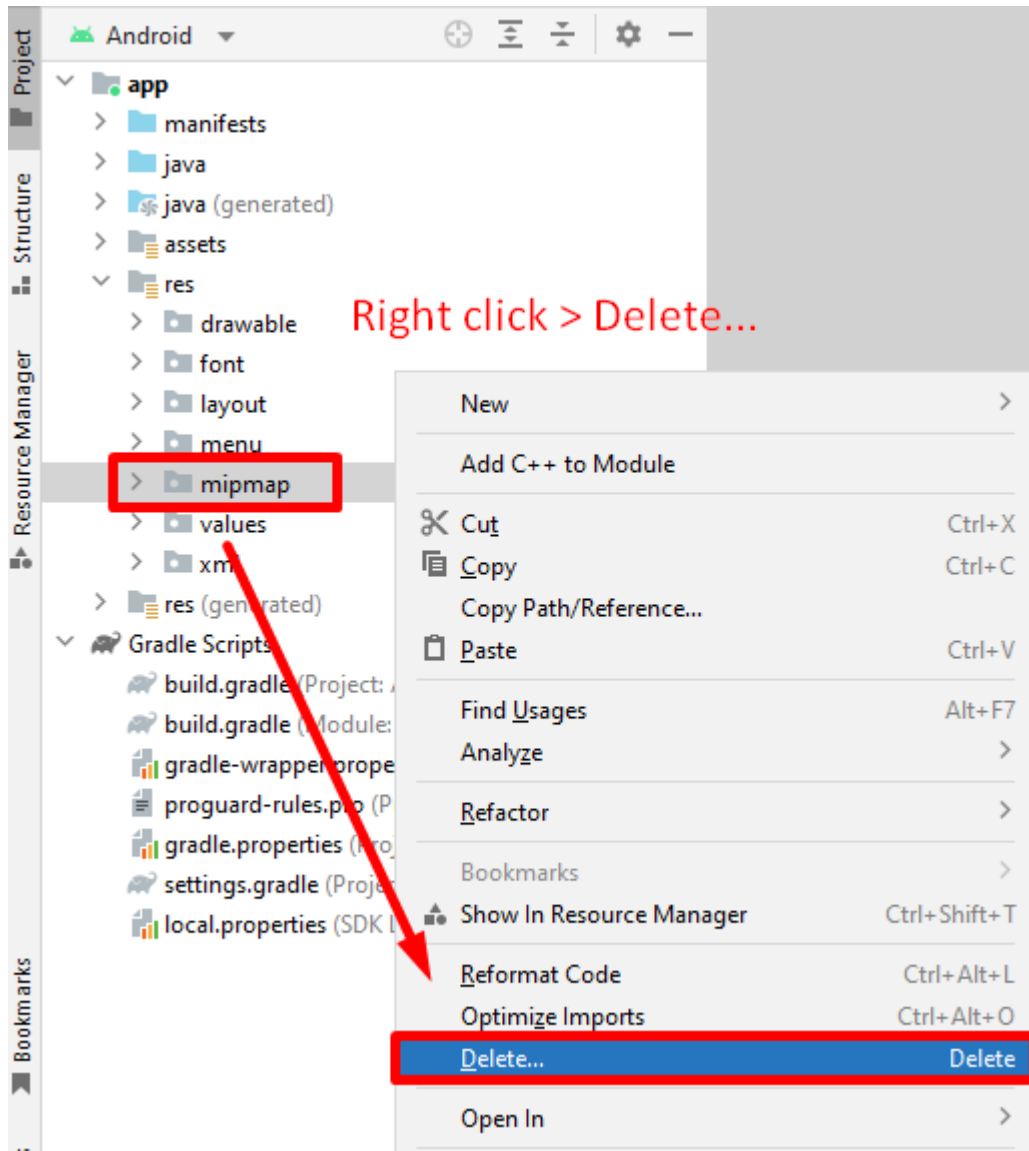
After the renaming of the package name is complete, you need to rebuild the project and invalidate the cache

1. Select **Build > Rebuild Project** and wait until the building project finished
2. Select **File > Invalidate Caches.. > Invalidate and Restart**



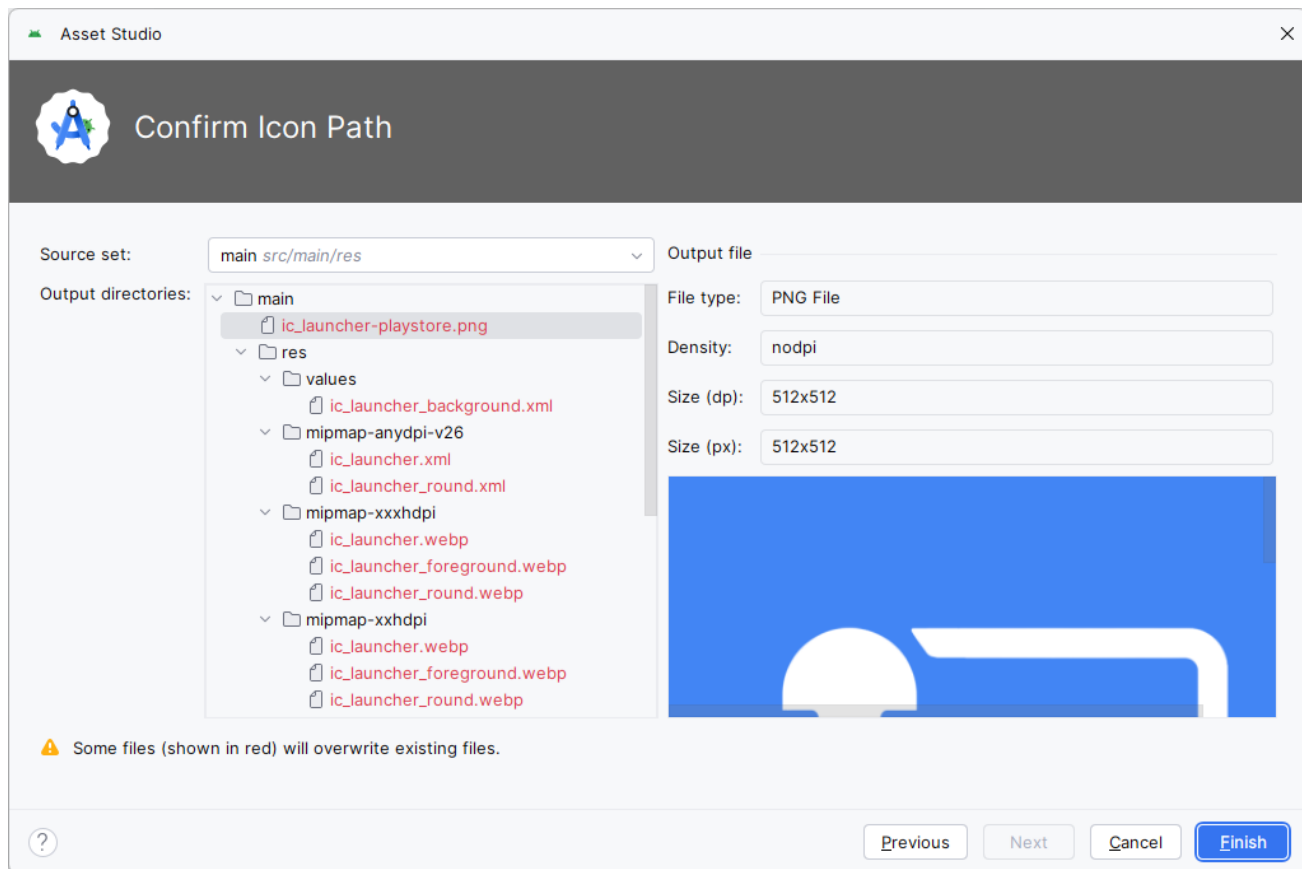
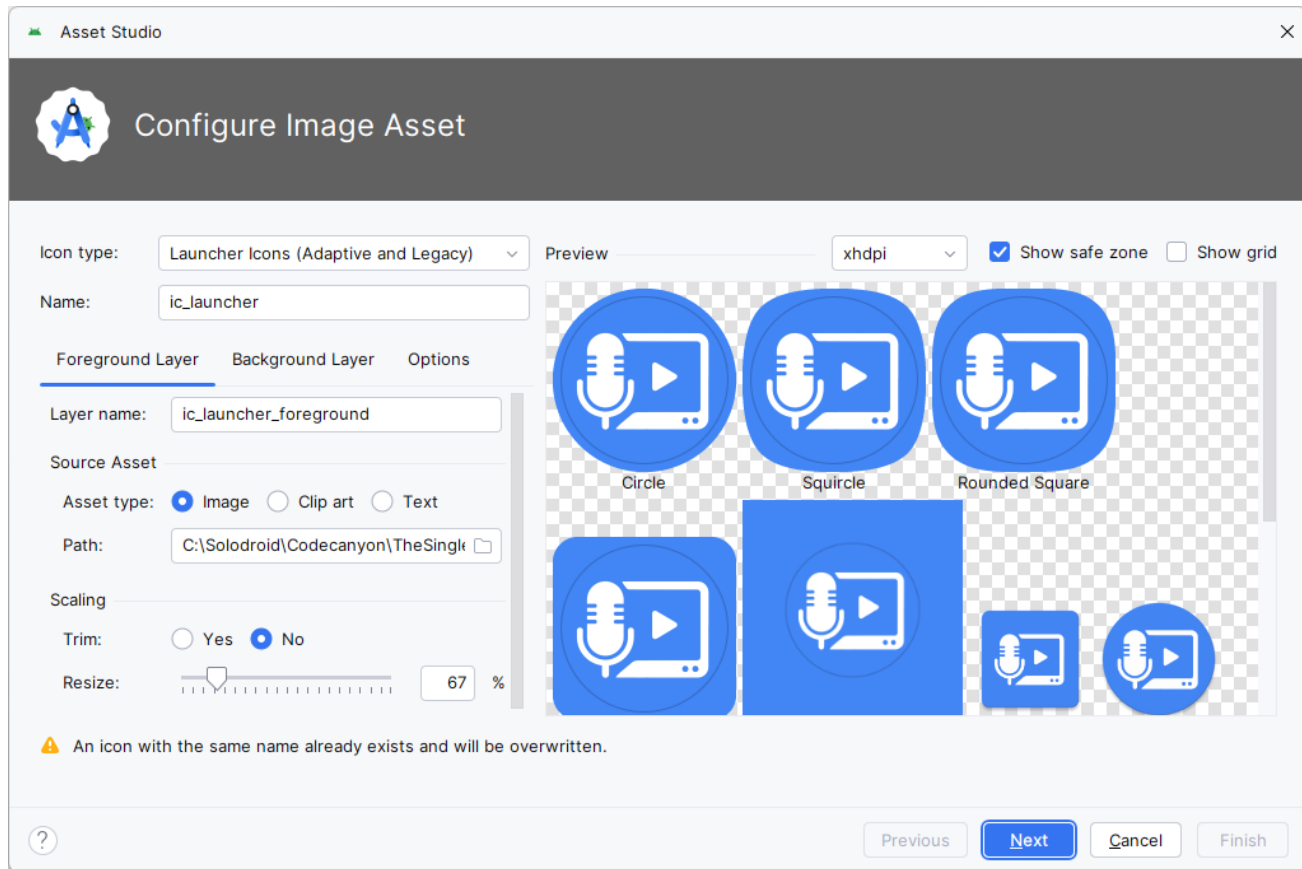
# Change Logo

The first step before changing the logo, it is highly recommended to delete the mipmap folder first



Then, after you delete the mipmap folder, the next steps are as follows:

1. Make sure the selected Res Directory is "**Main**" → Finish
2. Make sure the resource name is "**ic\_launcher**" → Next
3. Browse and choose your image file icon (.png) format with square dimension and start customize your icon
4. Recommended to use **Launcher Icon (Adaptive and Legacy)**
5. You can choose 2 types of your icon type, that is **Launcher Icon (Adaptive and Legacy)** or **(Legacy Only)**
6. Open Android Studio → **File** → **New** → **Image Asset**



**Important:** To prevent errors when changing the application logo, make sure you have deleted the mipmap folder first, if not, you need to repeat this steps to change the app logo at least 2 times, because in some cases, when you change the app logo without deleting the mipmpa folder, there may be a problem and the logo is not saved in the mipmap folder, most likely this is a bugs in Android Studio, if this problem occurs, please repeat the steps to change the app logo again and the problem will be resolved.

# Change Color

1. to change the app color, you can see on the **res/values/colors.xml**
2. Enter your color code inside each of strings tag :

The screenshot displays the Android Studio interface. On the left, the 'Project' view shows the file structure of an Android app. The 'res' folder is expanded, and 'values' is further expanded, with 'colors.xml' highlighted by a red rectangle. The main editor window shows the content of 'colors.xml'. The XML code defines various color resources for the app, organized into three sections: 'app main color', 'dark theme color', and 'others'. Each color is defined using the <color> tag with a name attribute and a hex color value.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <!-- app main color -->
    <color name="color_light_primary">#4286F5</color>
    <color name="color_light_status_bar">#3A78DC</color>
    <color name="color_light_accent">#5B99FF</color>
    <color name="color_light_bottom_navigation">#FFF4F8</color>
    <color name="color_light_background">#FFFFFF</color>
    <color name="color_navigation_drawer_selected_item">#D5E5E5</color>
    <color name="color_native_ad_background">#F3F3F3</color>
    <color name="color_selected_item">#3300796B</color>
    <color name="color_light_text">#292929</color>
    <color name="color_light_icon">#292929</color>
    <color name="color_light_active_indicator">#4d4286F5</color>

    <color name="color_light_native_ad_background">#F3F3F3</color>
    <color name="color_dark_native_ad_background">#222D36</color>
    <color name="color_stroke_native_ad">#5B99FF</color>

    <color name="color_light_progress_indicator">#03A9F4</color>
    <color name="color_dark_progress_indicator">#BDC0C3</color>

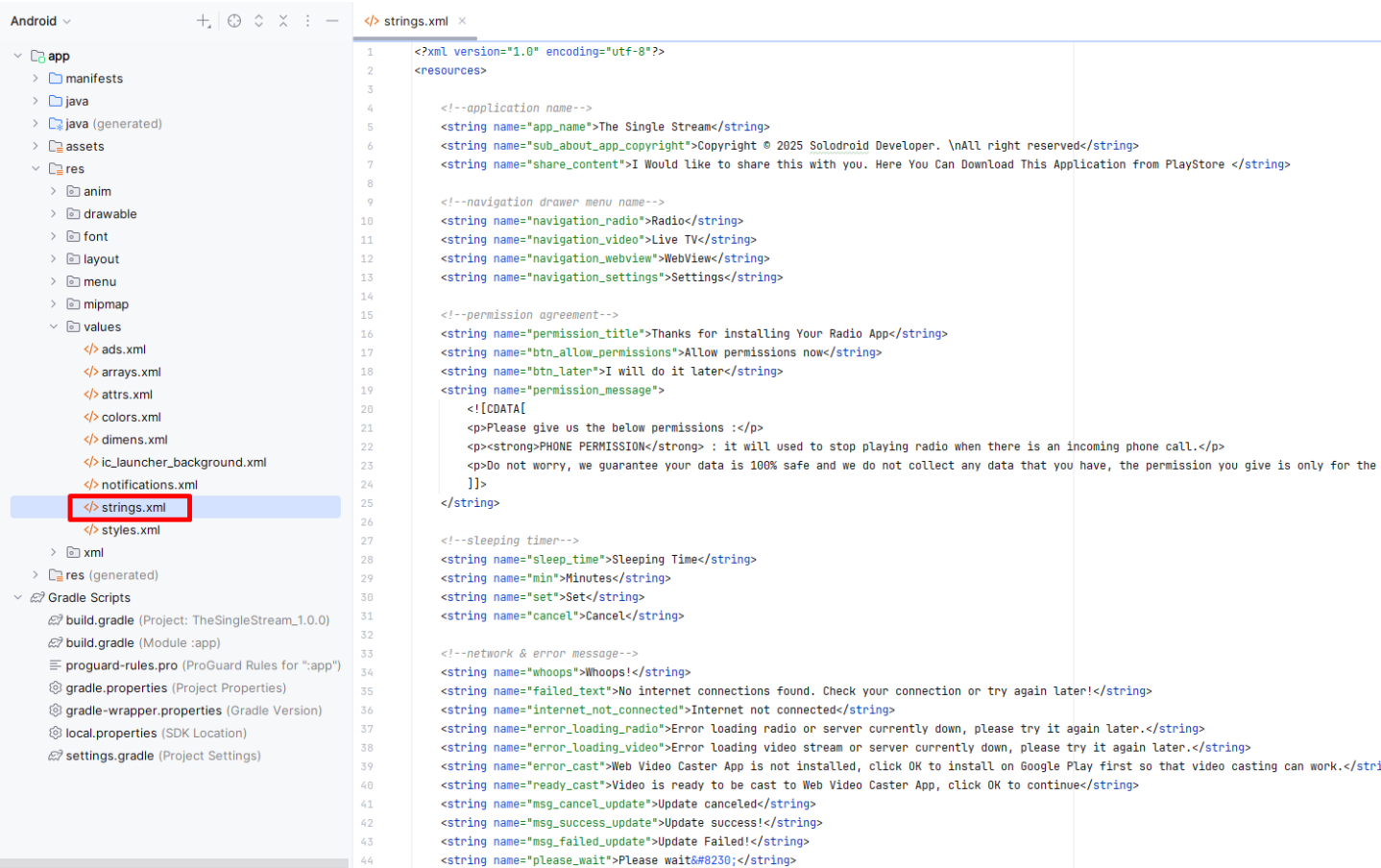
    <!-- dark theme color -->
    <color name="color_dark_status_bar">#0F1D24</color>
    <color name="color_dark_toolbar">#222D36</color>
    <color name="color_dark_accent">#5B99FF</color>
    <color name="color_dark_background">#101D24</color>
    <color name="color_dark_text">#BDC0C3</color>
    <color name="color_dark_icon">#BDC0C3</color>
    <color name="color_dark_search_bar">#222D36</color>
    <color name="color_dark_bottom_navigation">#222D36</color>
    <color name="color_dark_active_indicator">#384756</color>
    <color name="color_dark_mini_player_background">#222D36</color>
    <color name="color_dark_tab_icon_unselected">#CAC4D0</color>

    <!-- others -->
    <color name="color_play_button">#4286F5</color>
    <color name="color_radio_image_border">#E1E1E1</color>
```

# Change Text

The text asset consists of the application name of another text component inside the app, to change it :

1. You can see on the **res/values/strings.xml**
2. Change value name in each strings tag according your needs.



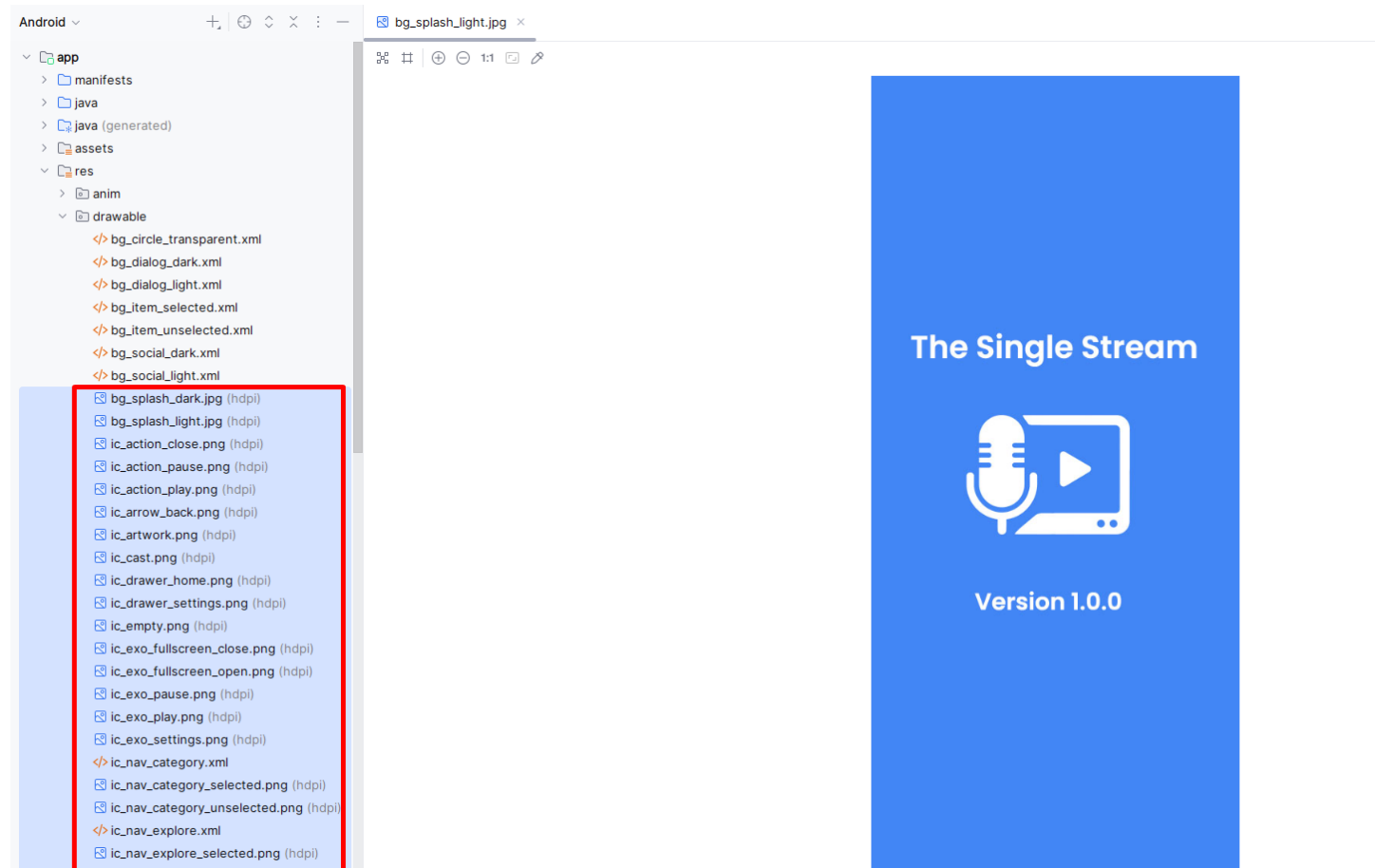
# Change Images

The image asset consists of the splash image and other image component inside the app, to change it :

1. Open **res/values/drawable** and replace with your image or icon, all image are placed on **drawable-hdpi** folder
2. You can also change every image in app like via Explorer, open Explore and go to your project directory, select:

**TheSingleStream/app/src/main/res/drawable-hdpi**

3. Replace every image which you need to custom the application and highly recommended you using same resolution for each image



# Change Font

If you want to change the app font, you can change the app font with your custom font with .ttf extension, rename your font file to **custom\_font.ttf**, then, copy and replace the existing font to :

- **res/font/custom\_font.ttf**
- **assets/font/custom\_font.ttf**

