

My Map Ready to Go Documentation 2021

Get Help By Shareing Your Location with Friends

*Track Your
Friends/Family's
Current Location*

- World-Wide*
- Search*
- Navigation*
- NearBy Places*
- Tracking*



Instruction:1

Import the project into your Android studio.

To import the code into your Android studio do the following steps:

i)Unzip the Project File

ii)Open Android Studio exe

iii)Click on Open an Existing Project on Android Studio

iv)Select our Android Project

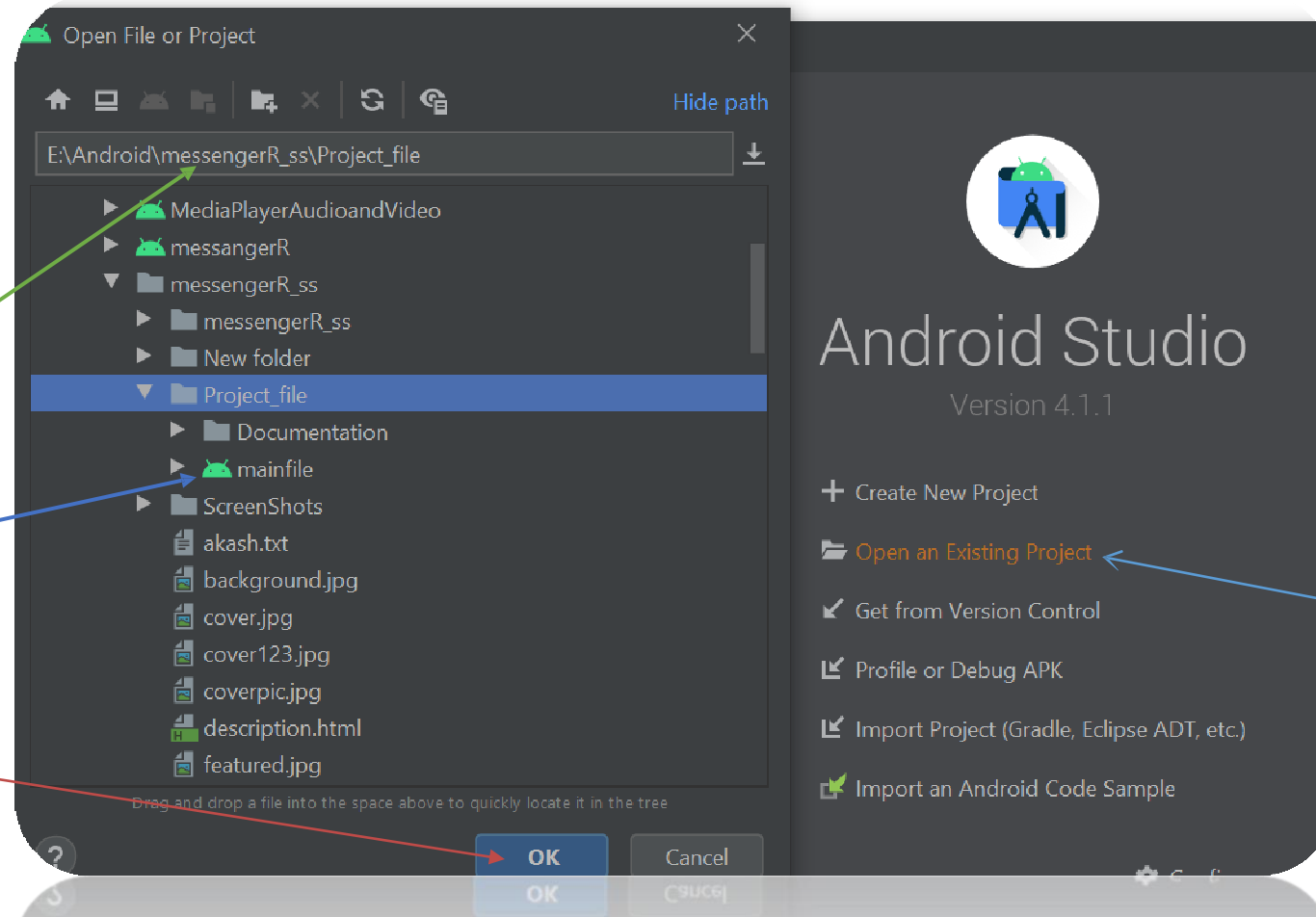
[In Android Studio Opening dialog,when you choose the option "Open an Existing Project" ,You have to put the path of project root path. Then Select our project file ... See Examples]

Examples

step1

Unzip the Project file

Step 3
put the Root location of
the unzip file
and Select our main
project
Then
click ok. Files will be
imported automatically
on Android Studio within
less than one minute



Step 2

After Open Android Studio
Click on Open an Existing
Project

Instruction 2:

Changing the basics of project like app name, package name, icon, color, ui designs as you preferences...

=>You can use ours also but you have to change package name must for creating your own network....

⇒You can change the app_name and icon from manifest file...if you want to change colors and ui designs, just go to XML files and change these...

⇒import your designs at drawable folder and set these designs from xml File...Our Designs are open source...You can also use these.

Samplaes

Import your icon in drawable folder and set your app icon here

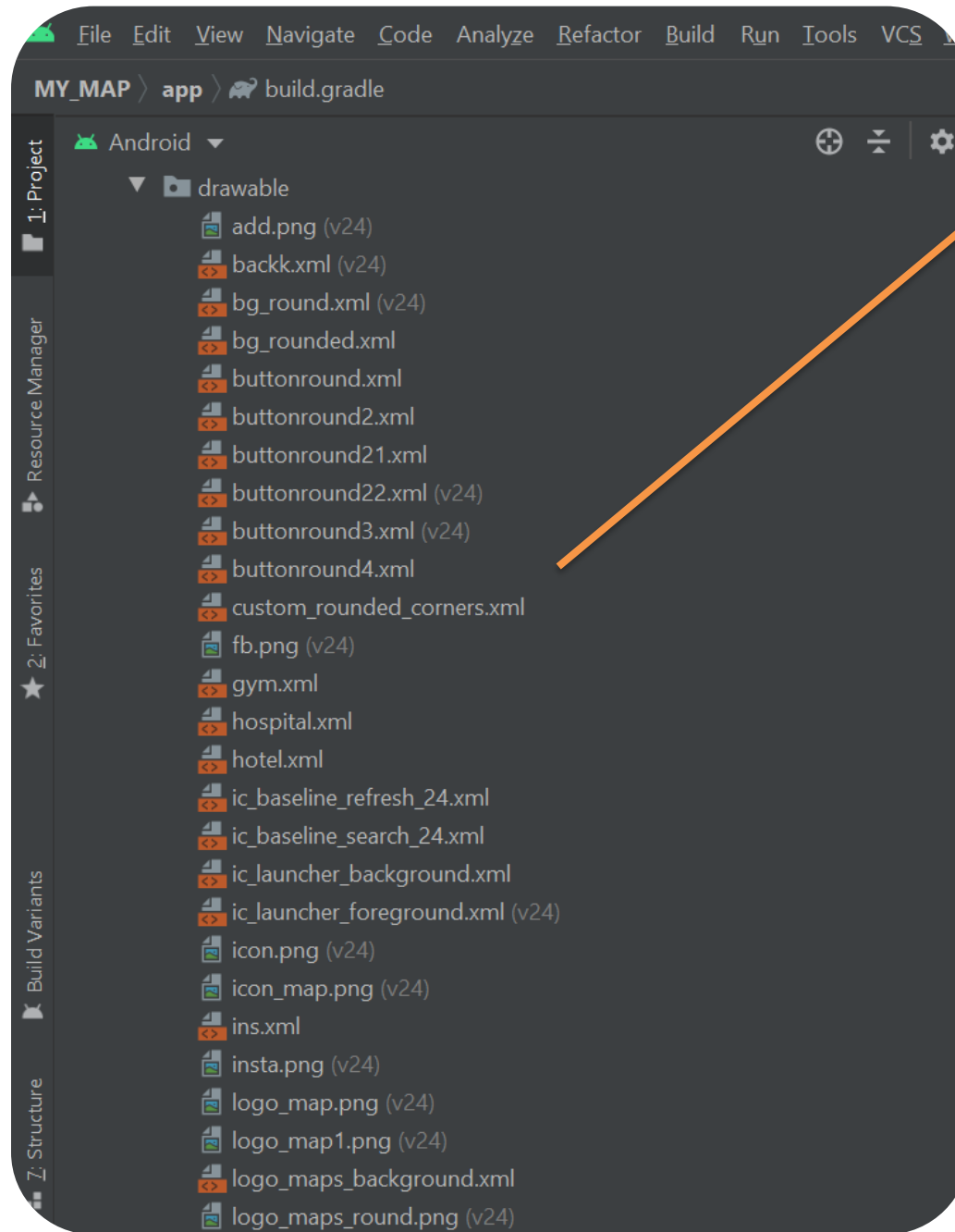
Step 1
Go to manifests file

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.softexploitmaps.mapping.ceo">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NOTIFICATION_POLICY" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/logo_map1"
        android:label="My_Map"
        android:roundIcon="@drawable/logo_map1"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".LoginActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

You can set your App name here



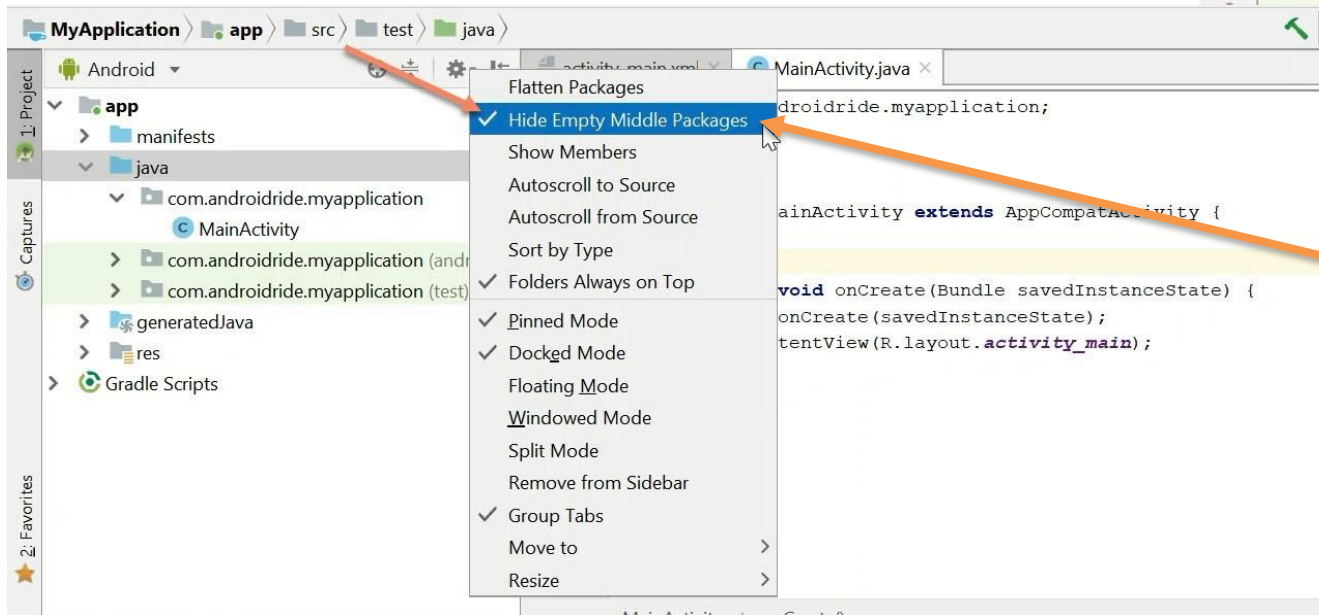
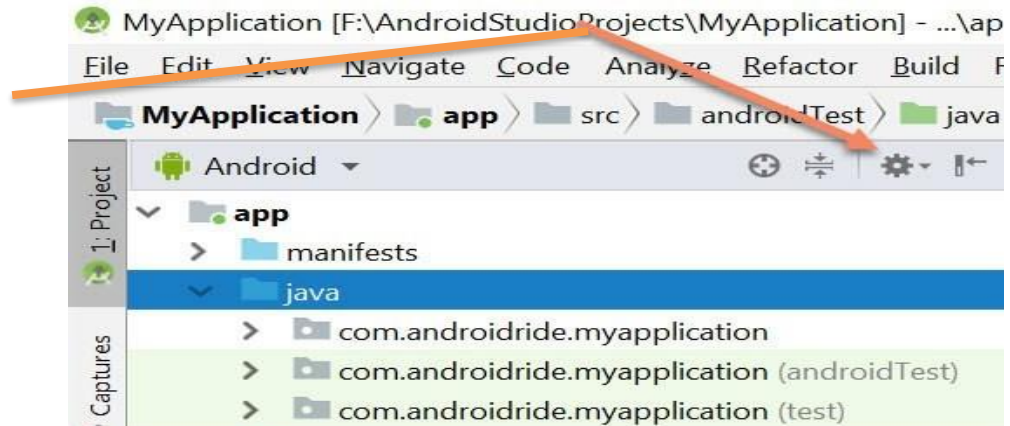
This is our Drawable Folder. You can copy your image and just paste it here in Drawable Folder. The image will be imported Automatically. Then Go to xml files and you can set your Designs there. Icons are also imported here in Drawable File

You can Replace our photo with your selected photo with the same name. EX: Our landing page's image name is bg.jpg... You can copy your photo And paste it into drawable folder then delete our bg.jpg picture and rename your photo as bg.jpg ...Your app's landing image will be changed and set with your picture...take a look on our drawable folder...

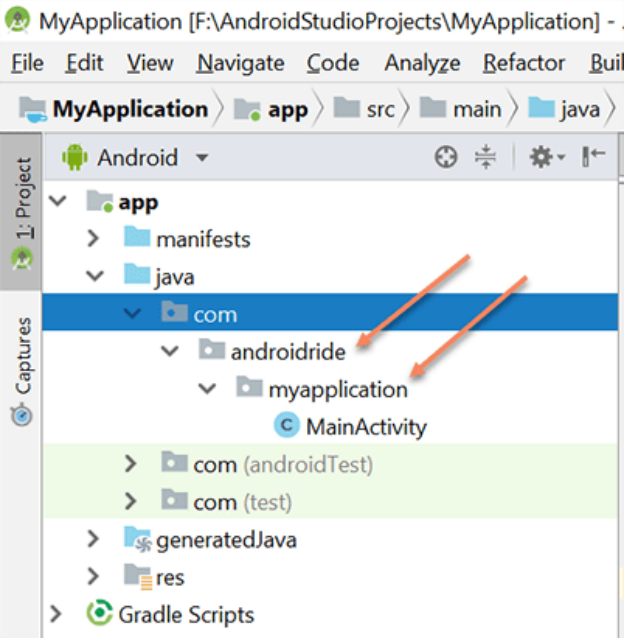
Methods To Change Or Rename Android App Package Name in Android Studio

In this example, we will change package name **com.androidride.myapplication** to **com.xyz.yourapplication**.

Step 1: Open Android Studio project, Select package name or Java, Click the gear icon.

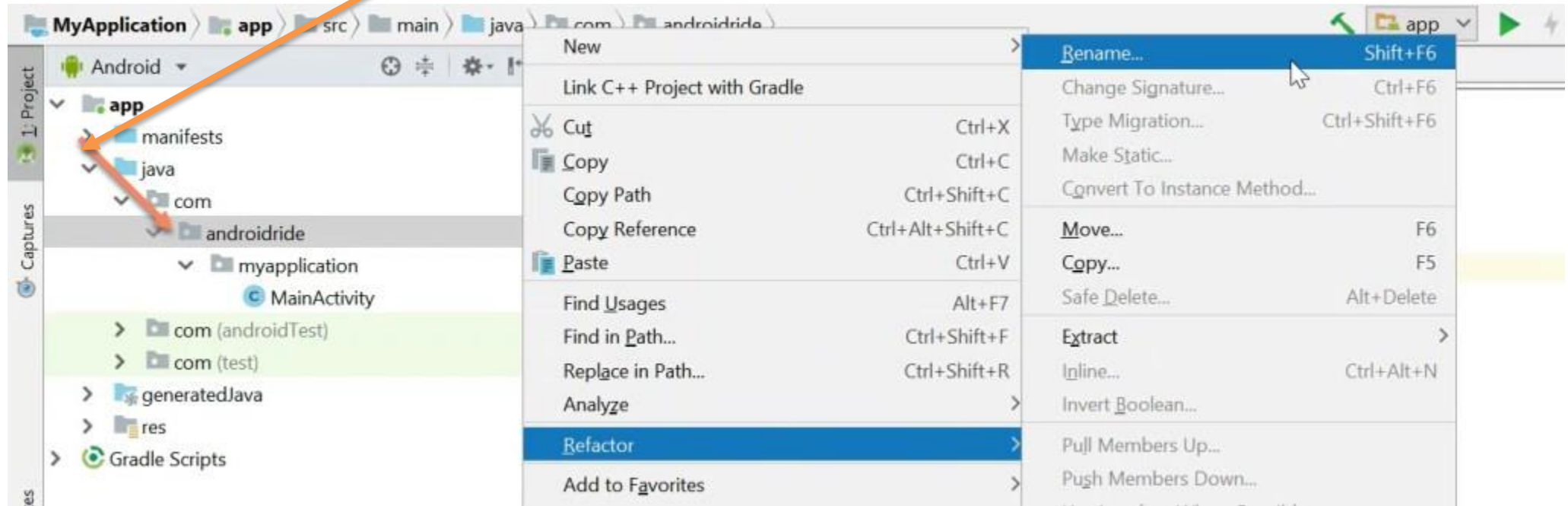


Step 2: Deselect Compact Middle Packages. This breaks your folder structure into parts.

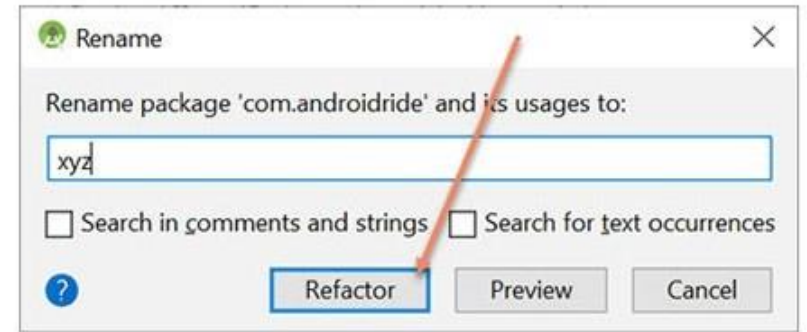
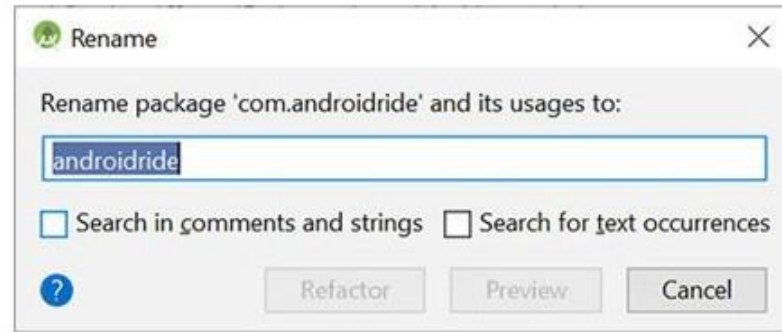
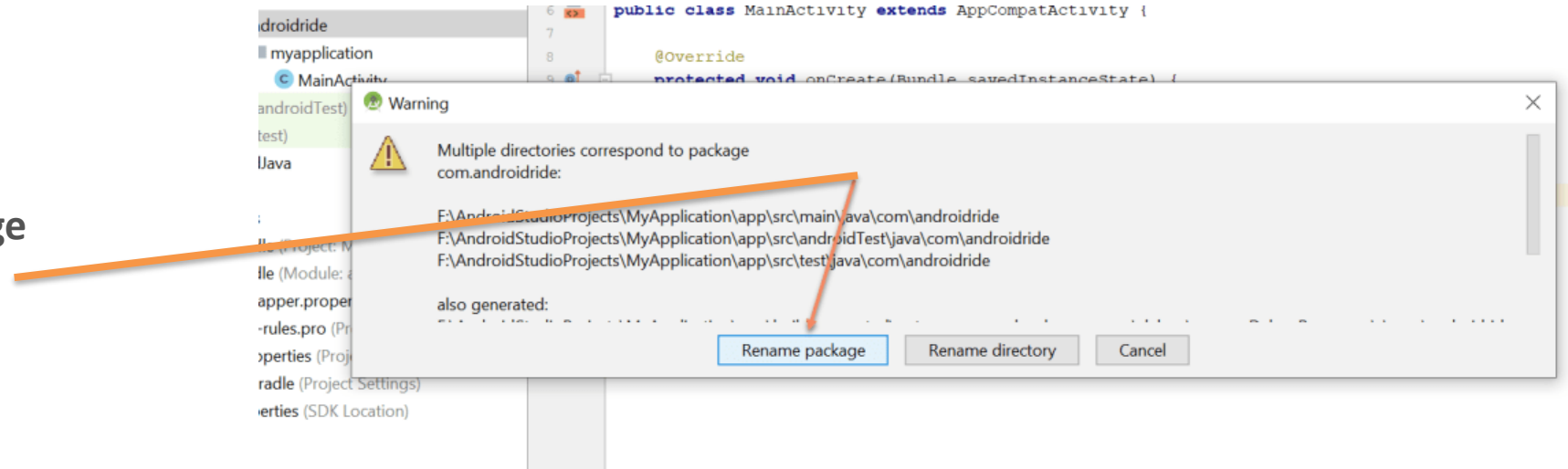


The package is broken down.

Step 3 : Right Click on AndroidRide, **Refactor -> Rename** . You can also use **Shift + F6**.



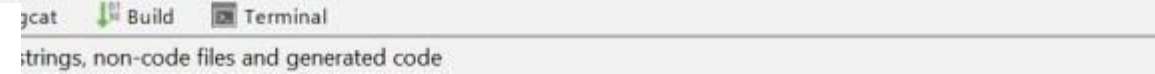
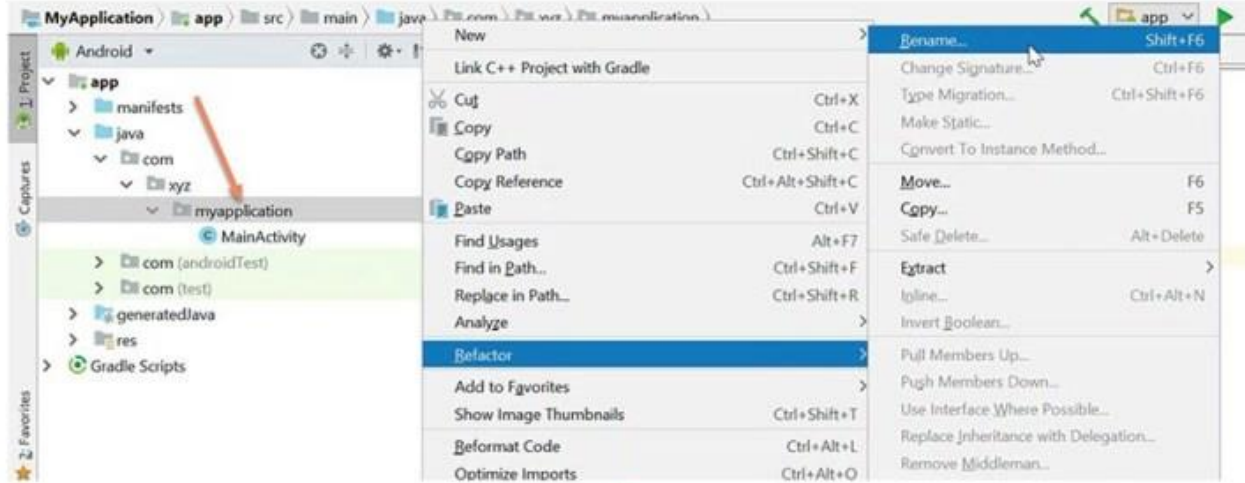
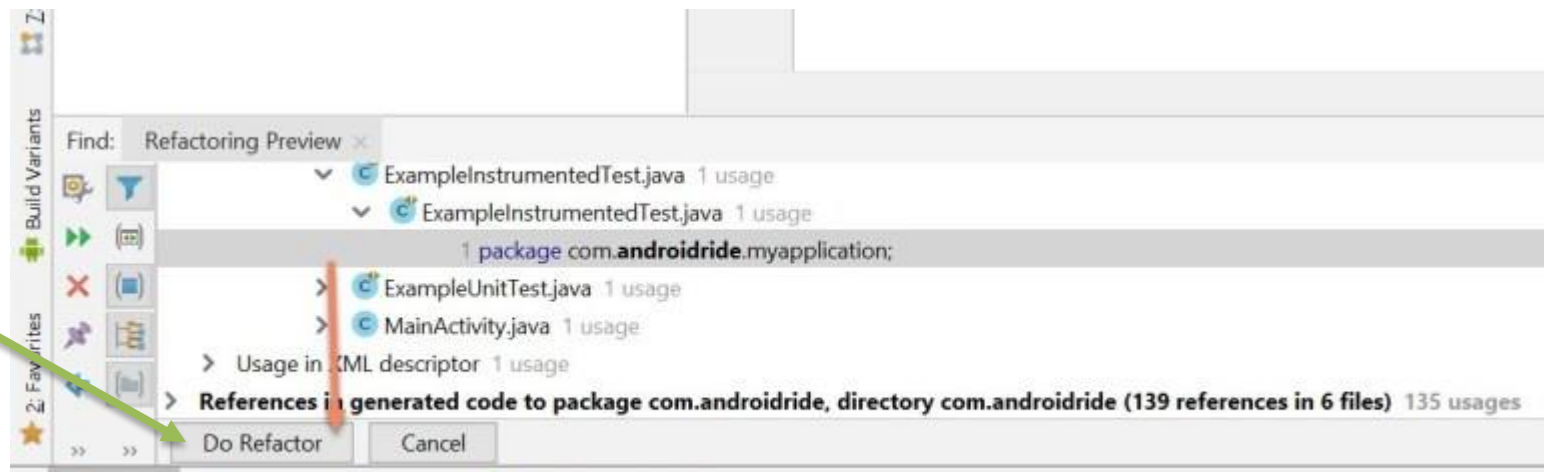
Step 4 : Click on **Rename package**



Step 5 : Clear AndroidRide and put new Name in it.
Here **"xyz"**

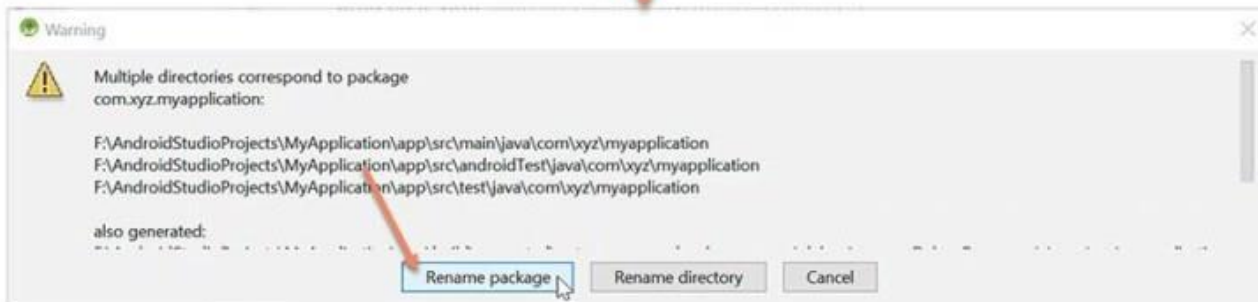
Step 6:

If This dialog will come, click on do Refactor otherwise not need

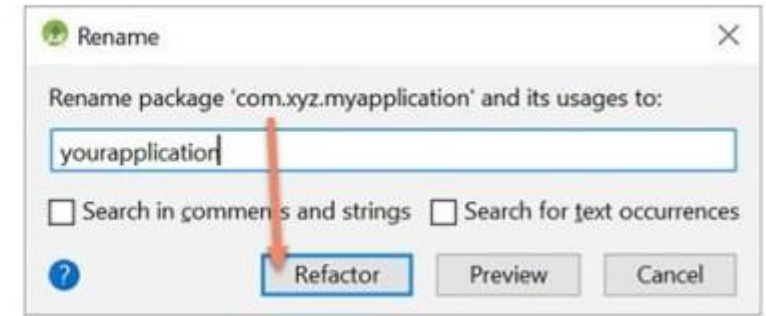
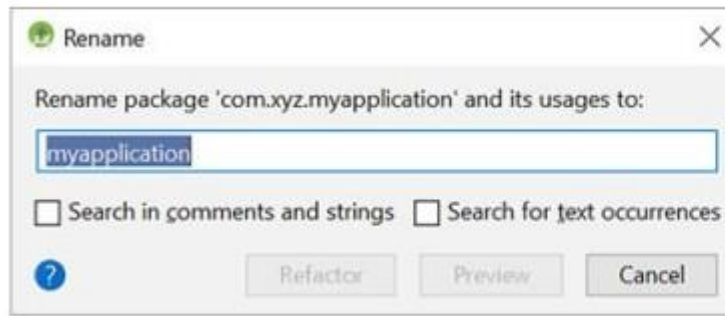


Step:7

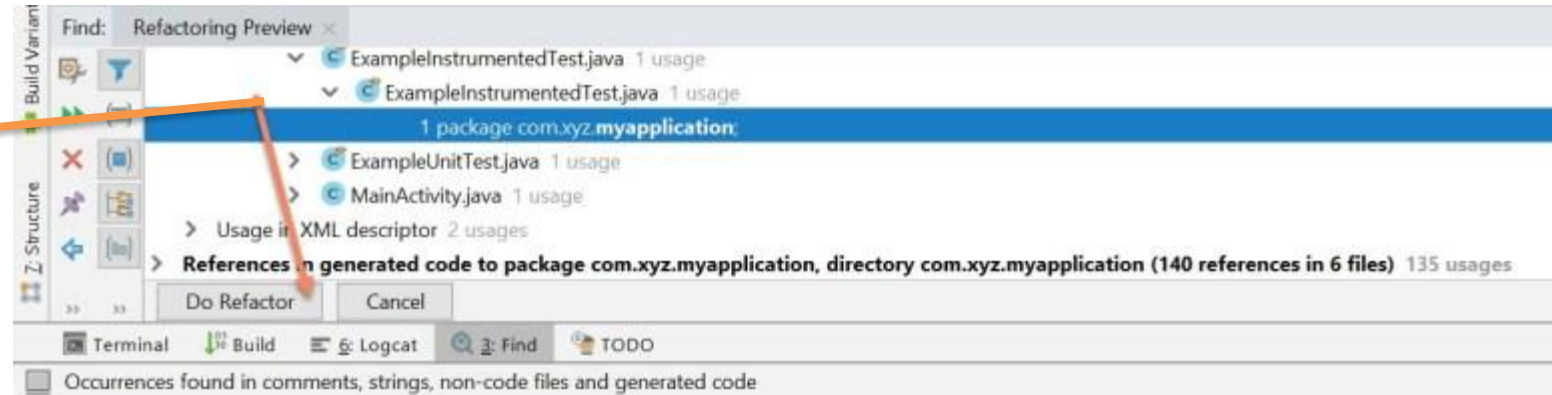
- Repeat the same process.
- Right click on **myapplication**,Refactor -> **Rename** and **Rename package**.



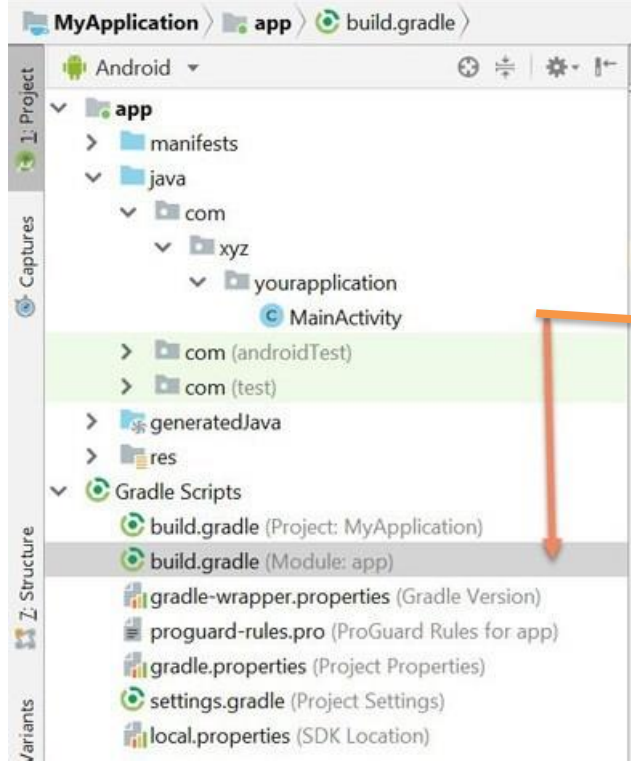
Step 8:
Change **myapplication** to
yourapplication



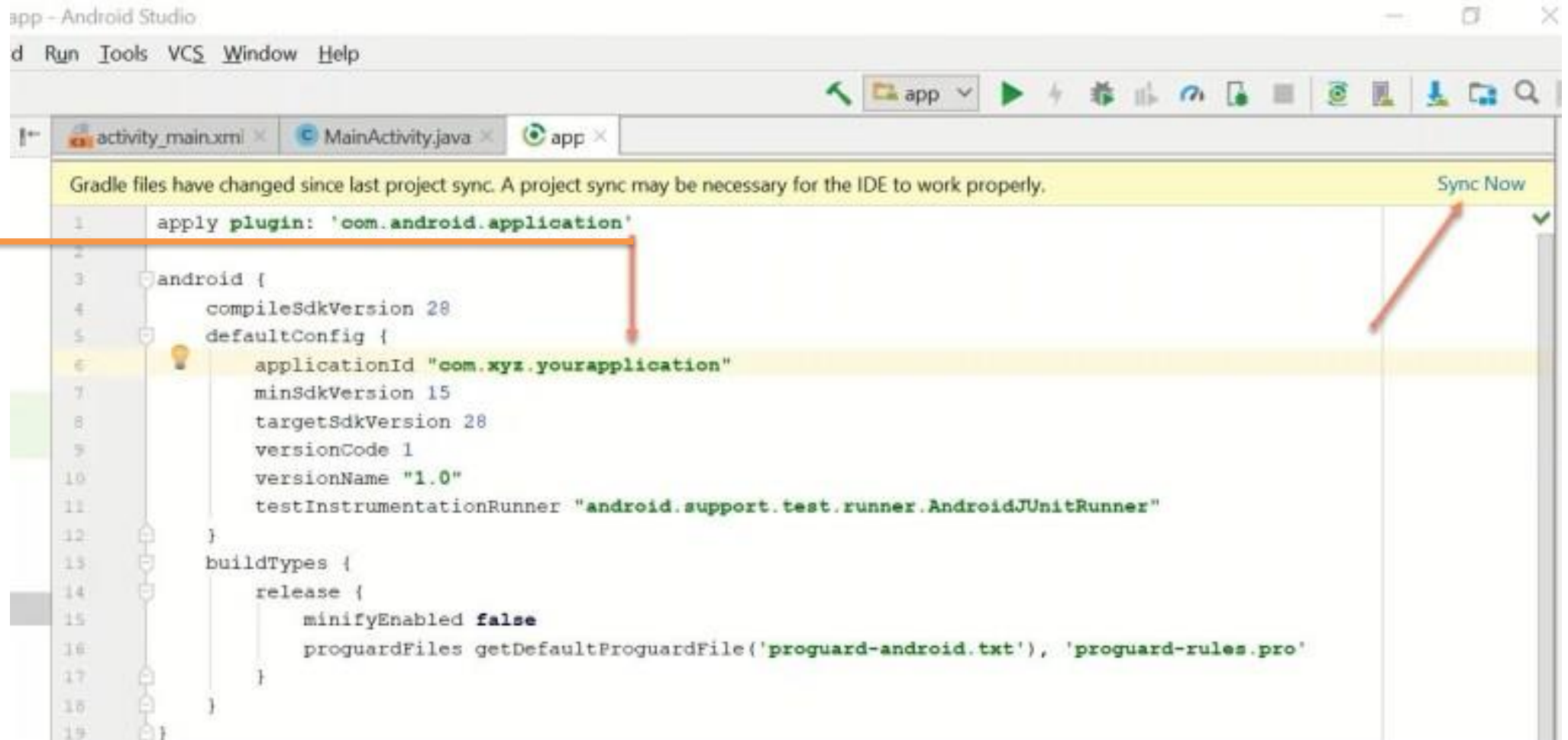
•Step 9: Click on **Do Refactor**.



MyApplication [F:\AndroidStudioProjects\MyApplication] - app -
File Edit View Navigate Code Analyze Refactor Build R



Step 10: After changing the package name,
open **build.gradle(module)** and change **Application Id**



The screenshot shows the Android Studio interface with the following elements:

- Menu bar: Run, Tools, VCS, Window, Help
- Toolbar: Back, Forward, Run, Stop, Refresh, Sync, etc.
- Tab bar: activity_main.xml, MainActivity.java, app
- Notification bar: Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly. Sync Now
- Code editor (build.gradle):

```
1  apply plugin: 'com.android.application'
2
3  android {
4      compileSdkVersion 28
5      defaultConfig {
6          applicationId "com.xyz.yourapplication"
7          minSdkVersion 15
8          targetSdkVersion 28
9          versionCode 1
10         versionName "1.0"
11         testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
12     }
13     buildTypes {
14         release {
15             minifyEnabled false
16             proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
17         }
18     }
19 }
```

An orange box highlights the `applicationId` property on line 6. A red arrow points from this box to the `Sync Now` button. Another red arrow points from the `Sync Now` button to the right side of the IDE.

change

applicationid **“com.androidride.myapplication”** to **“com.xyz.yourapplication”**.

-After changing applicationid, click on **sync now**.

Congrats. Package name successfully changed.

Instruction 3:

This project is completely built for Use. If you want to create Your own Network, Just you have create an account on <https://console.firebase.google.com/> Click Here and then create a project in It , Then, setup your android app inside that project ...For setup:

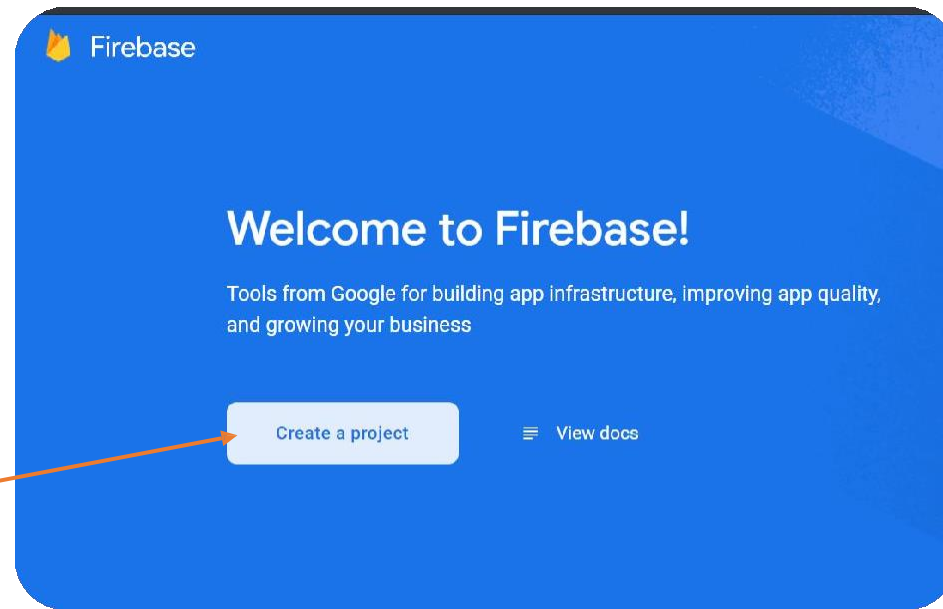
⇒ After Opening the project and Click on add app Android icon option then give your package name and download the config file from firebase . Following firebase instructions delete our existing config file from the project and in that position paste your downloaded config file. That's it. You can find the location of our config file from the firebase's instruction . After Clicking Continue to console your network is Created Successfully....

⇒ For Push Notifications System go to Firebase's Project overview->Project Settings
->Cloud Messaging->copy the server key token and then go to our project file->java
->package name file->Notifications>APIService.java file
->"Authorization : key = 'paste the token here'" ...Your push Notification Service is Build Completely...That's it. Network is Setup with Push Notifications

- Samples

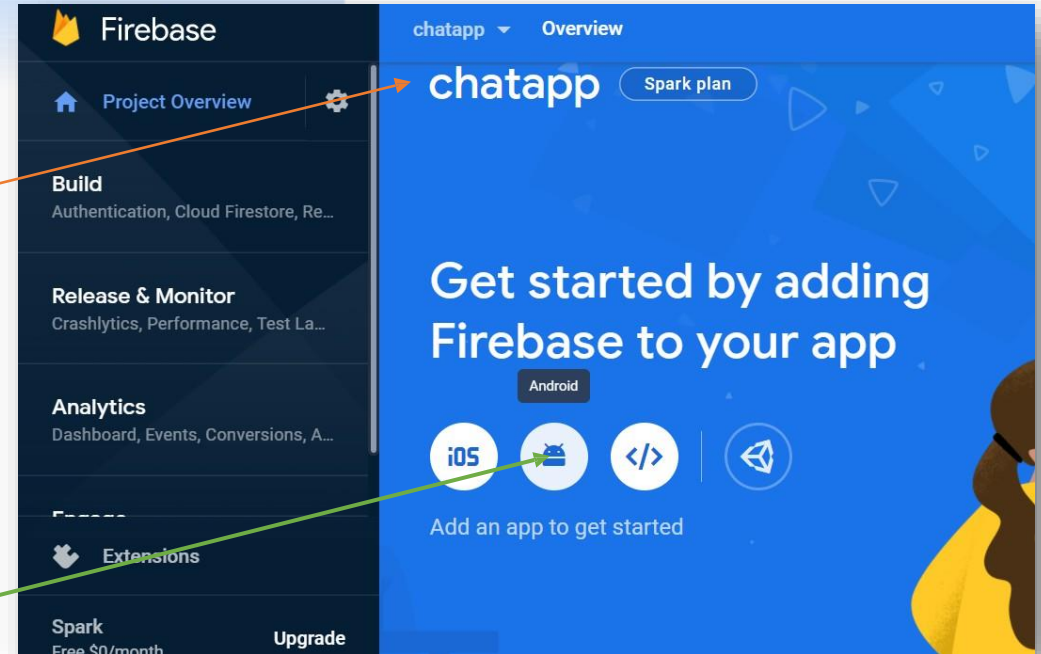
You can login to
firebase account
with your email id
Then,
Create a Project here

Firebase is
Completely Free and
Fast Server



We create a
project named
Chatapp

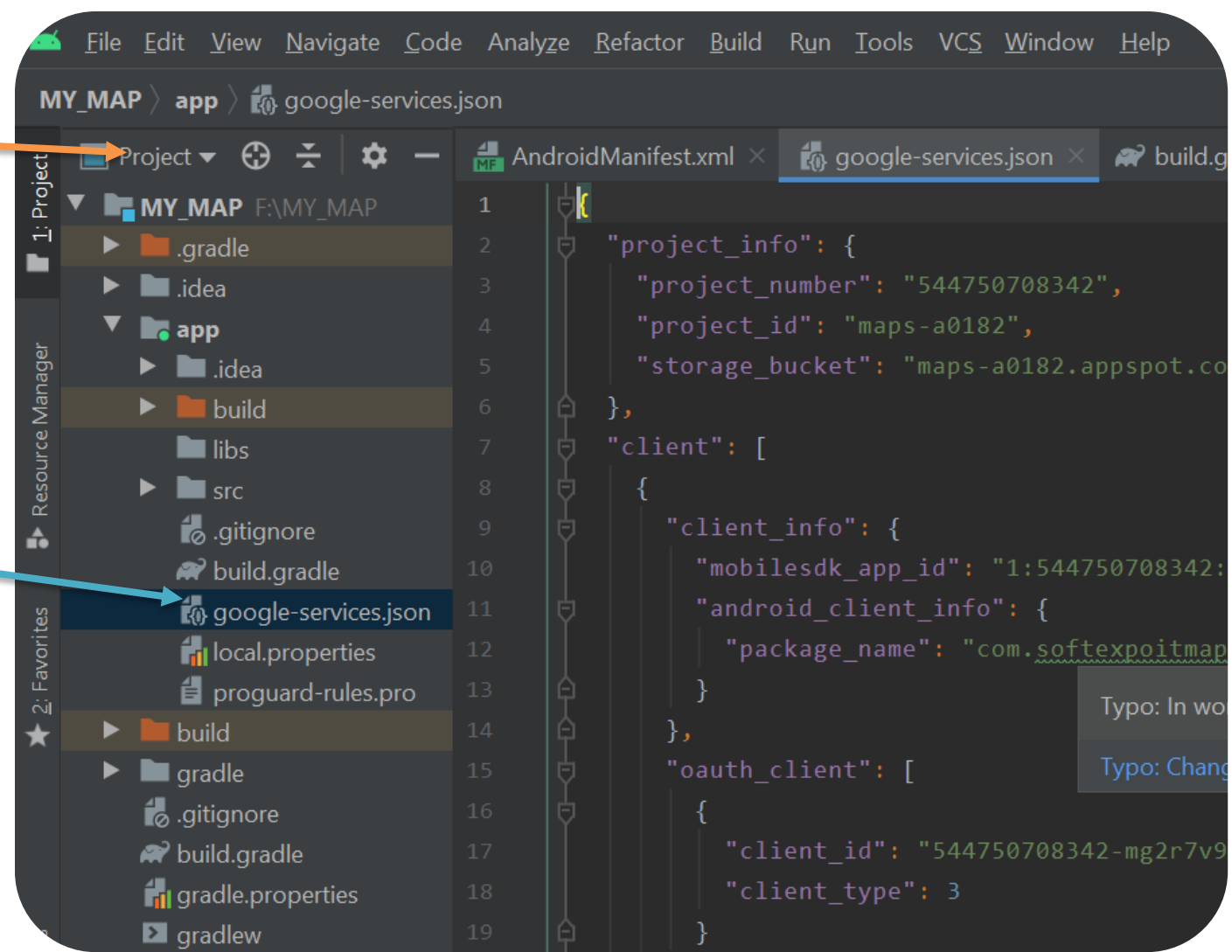
Click Here to Setup Your
Network



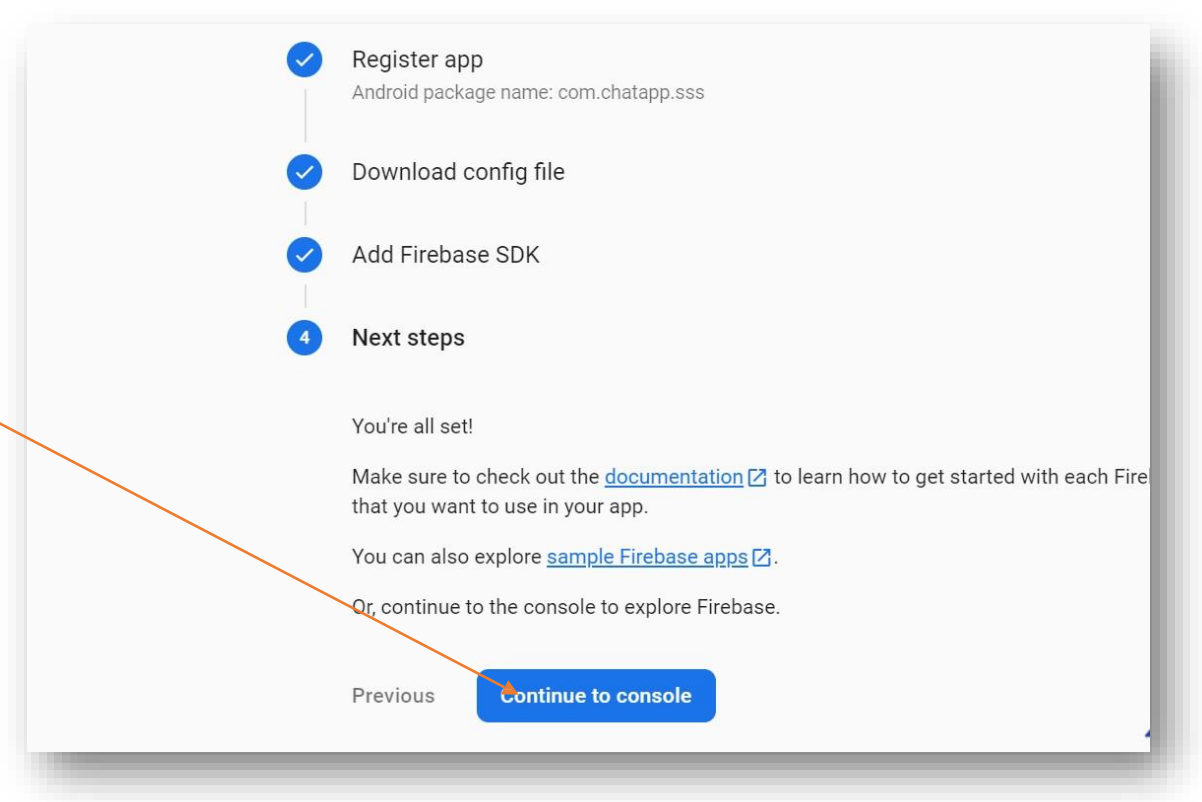
Change android to Project Here

Delete our google-services.json file from here
And paste your new Copied services.json file here
At the same position

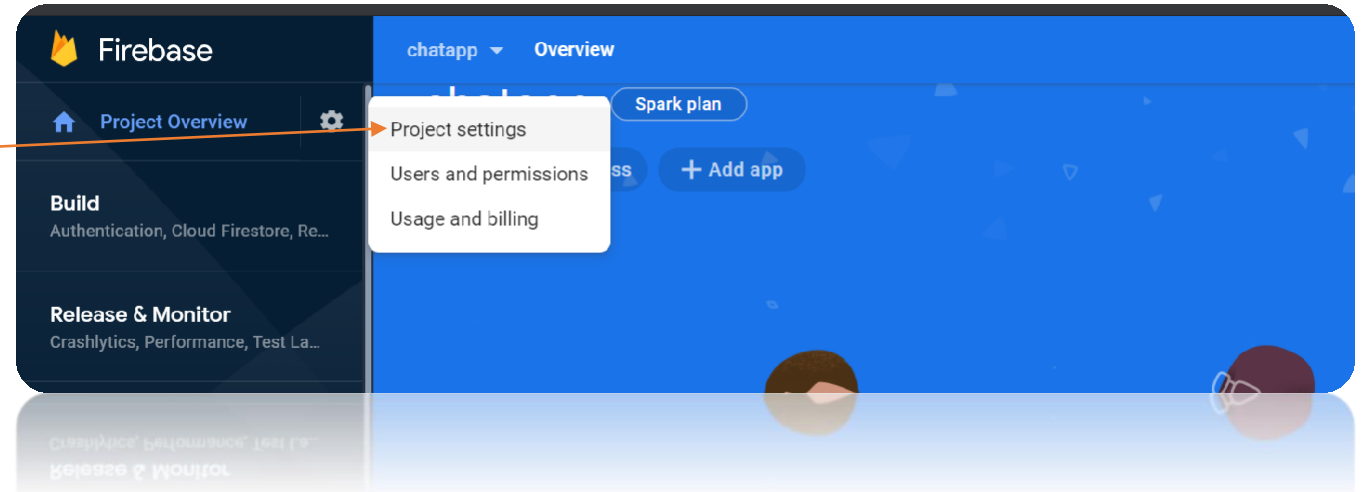
Be Careful
It is most Important part



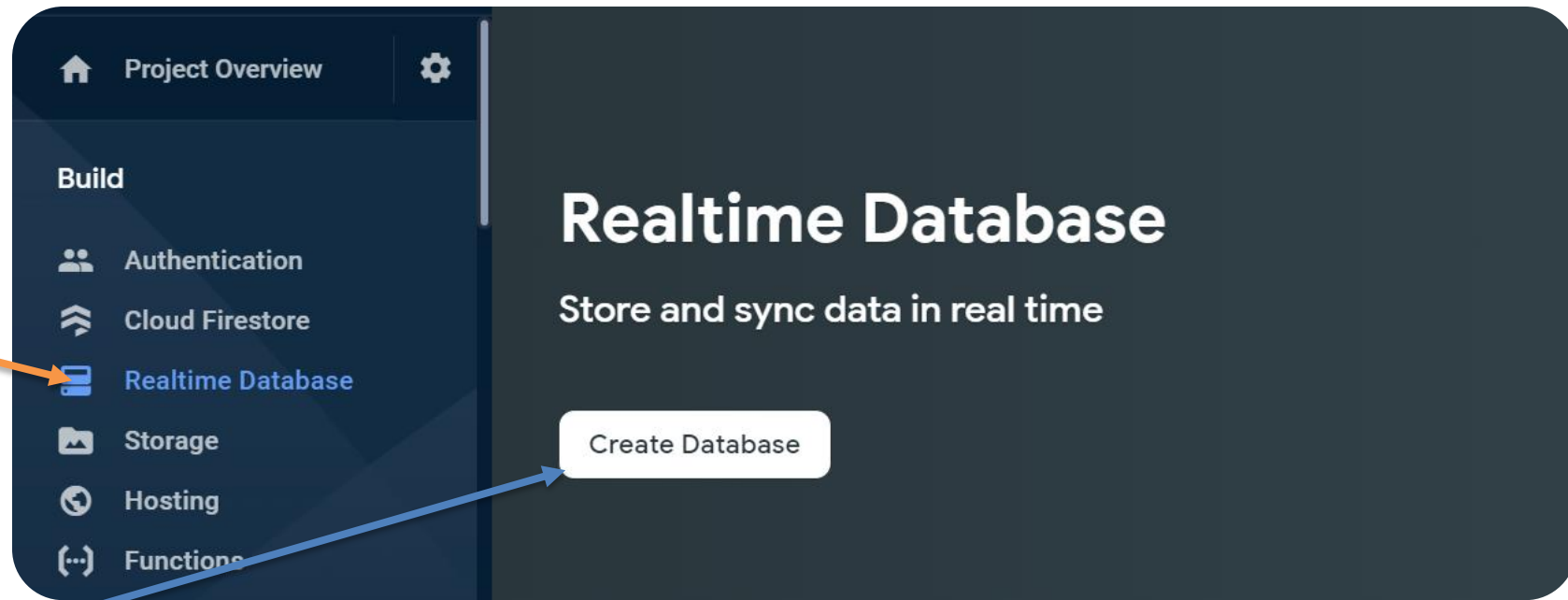
Then Click Continue the Console
Your Network Setup Finished completely



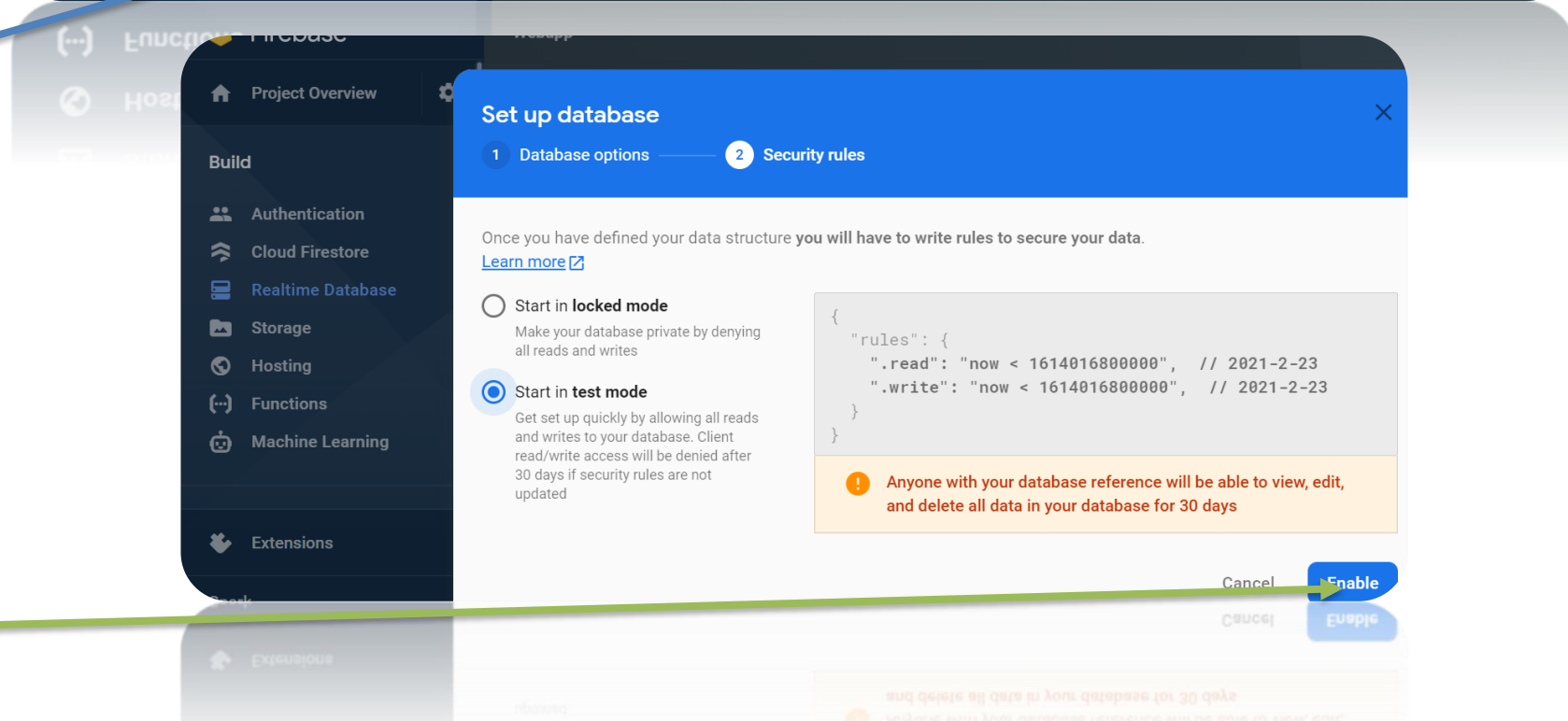
For Push notifications go to
Project Settings from Project Overview



Then You have to create a database just click here



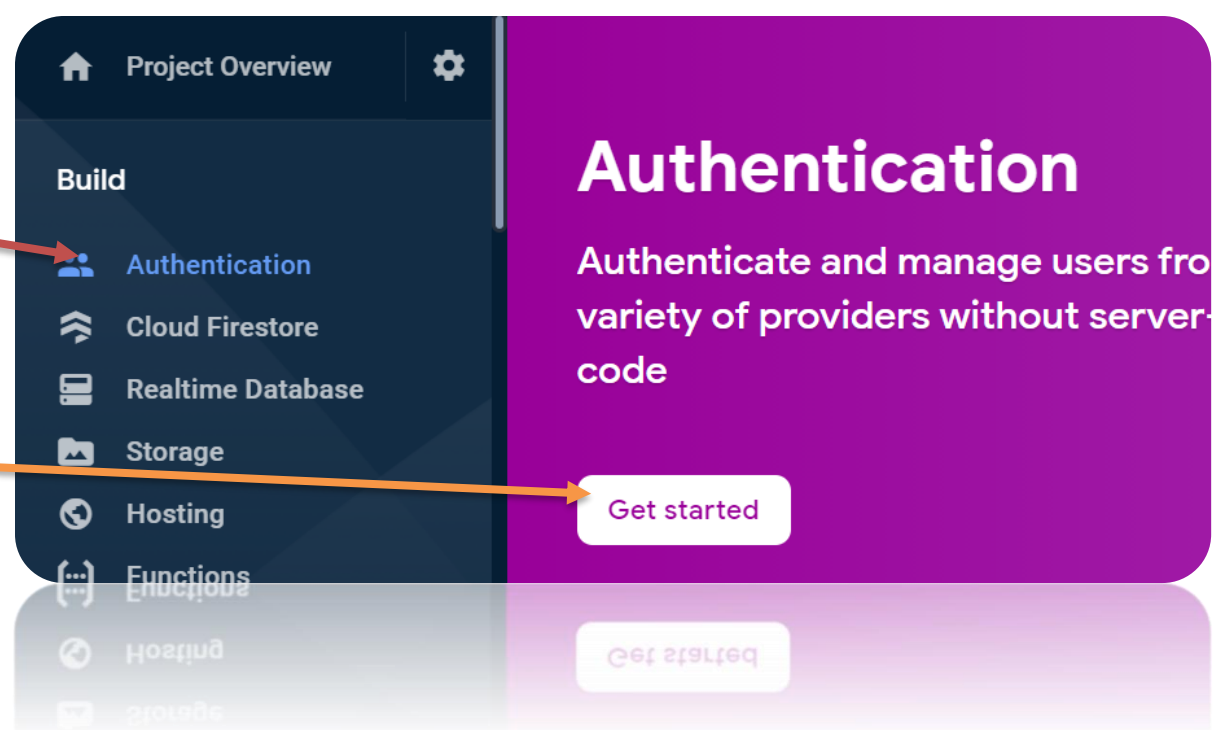
Then Click on Create Database



Then Enable it
Your Database will be
created automatically

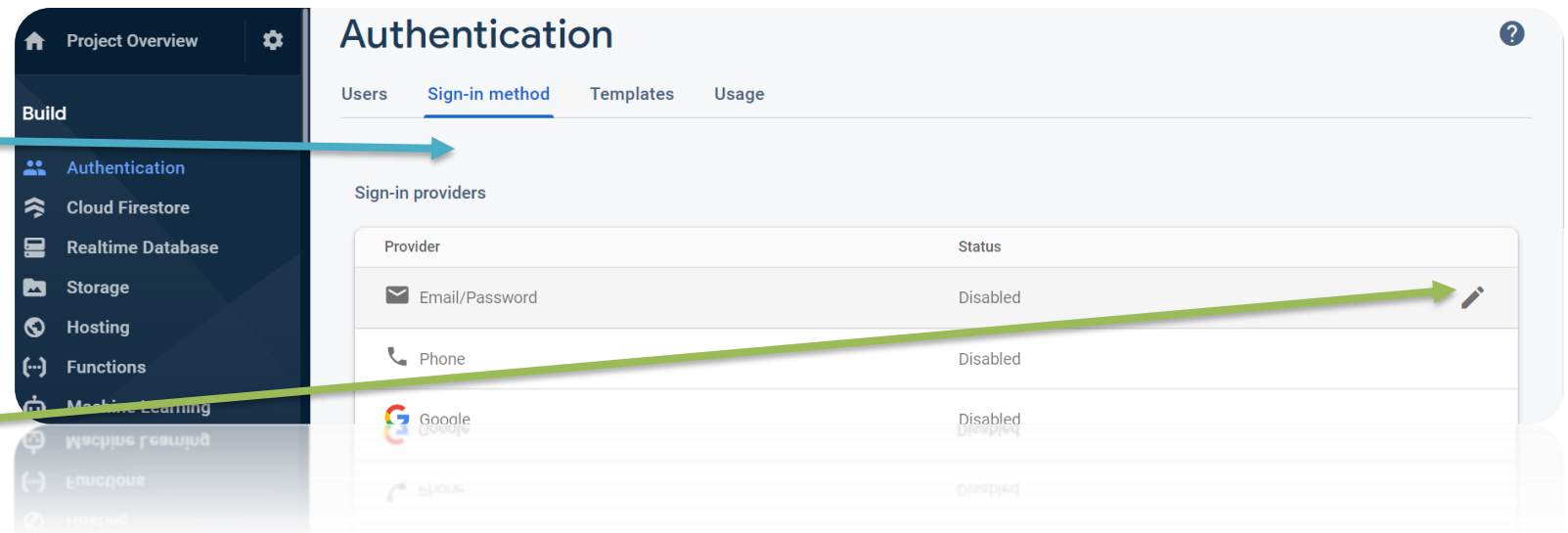
...

Then Click to Authentication



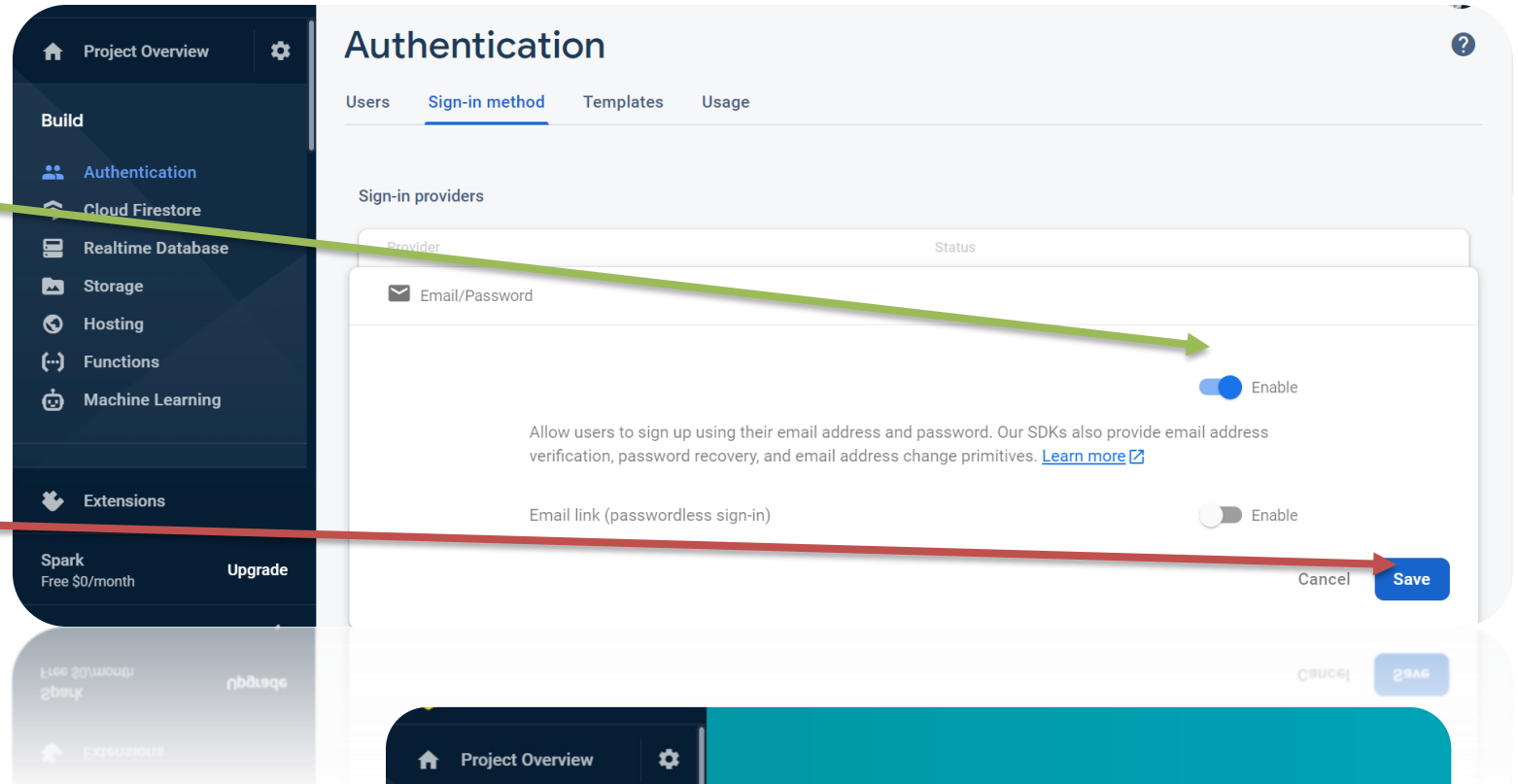
Click on get started

You can See a page like this



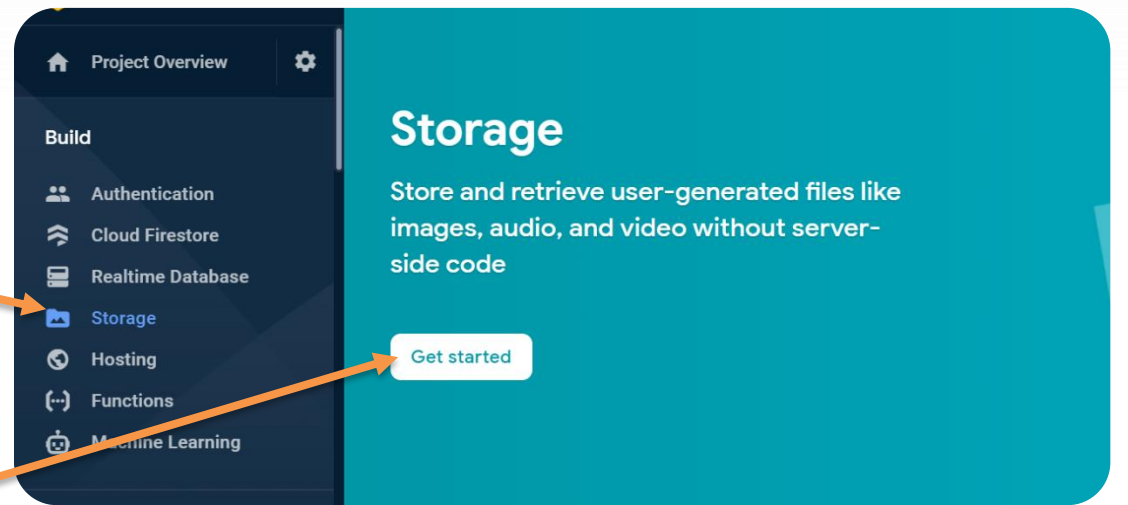
Click here

Enable Email/password



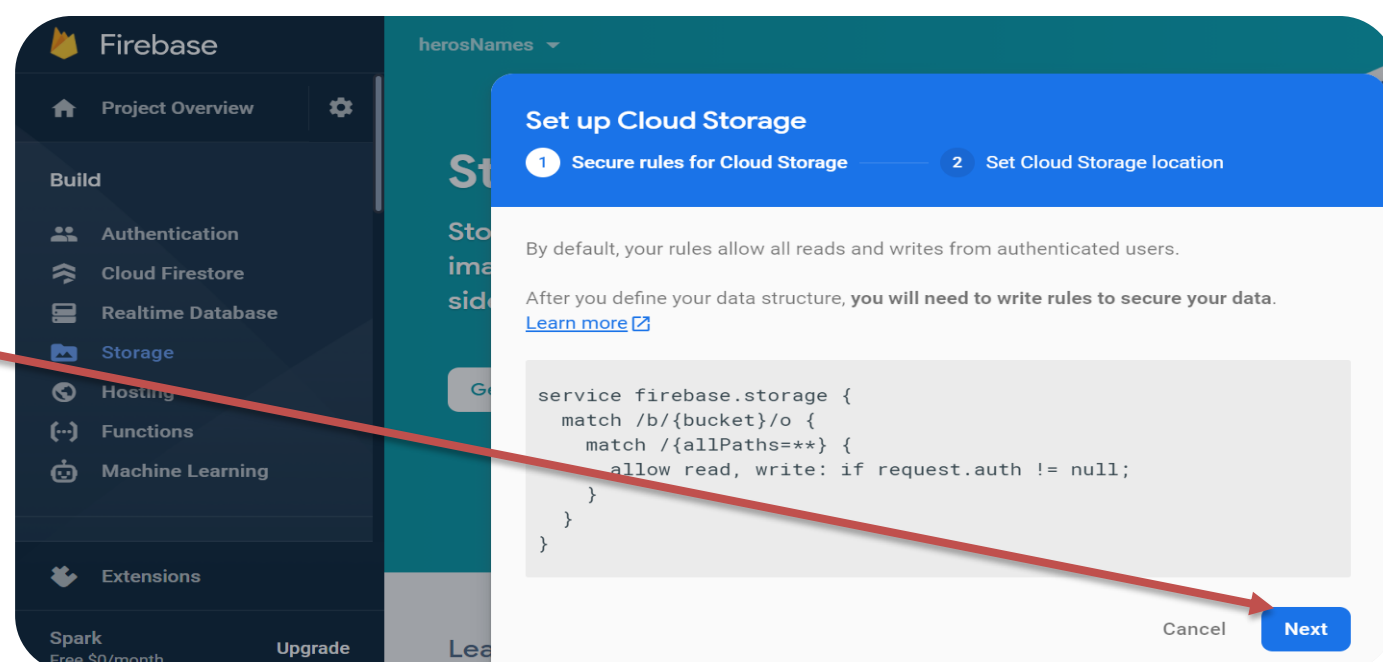
Click on save

Click On Storage

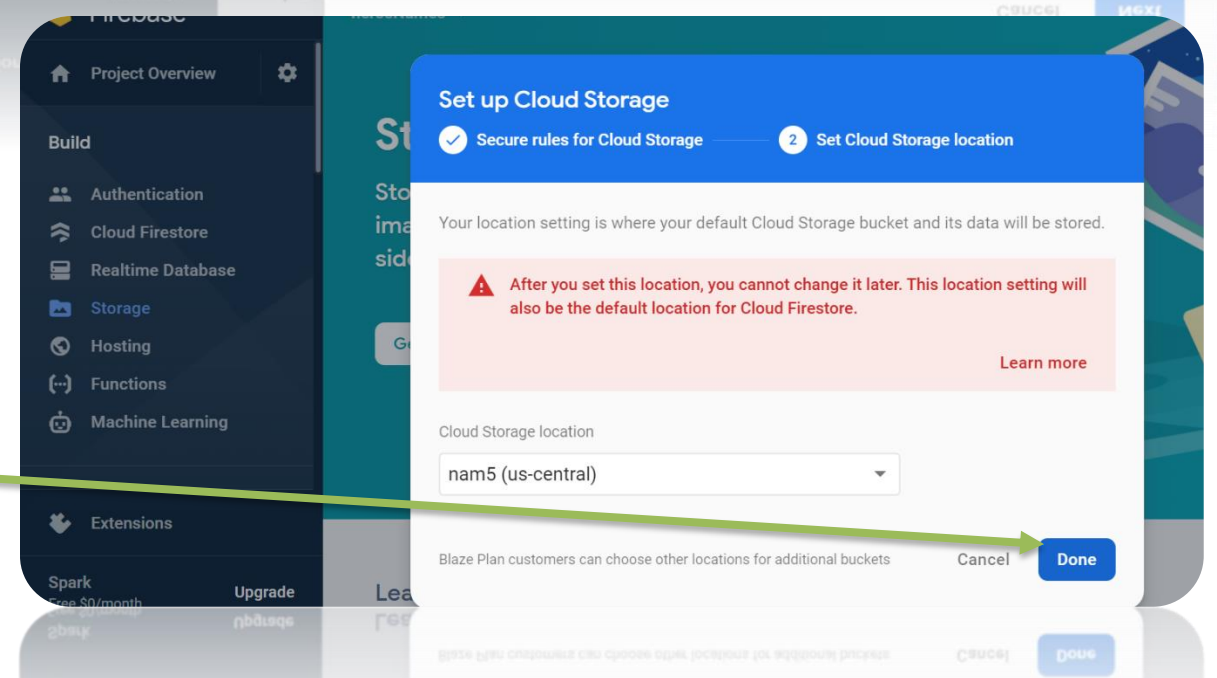


Then Click on get Started

Then click on Next

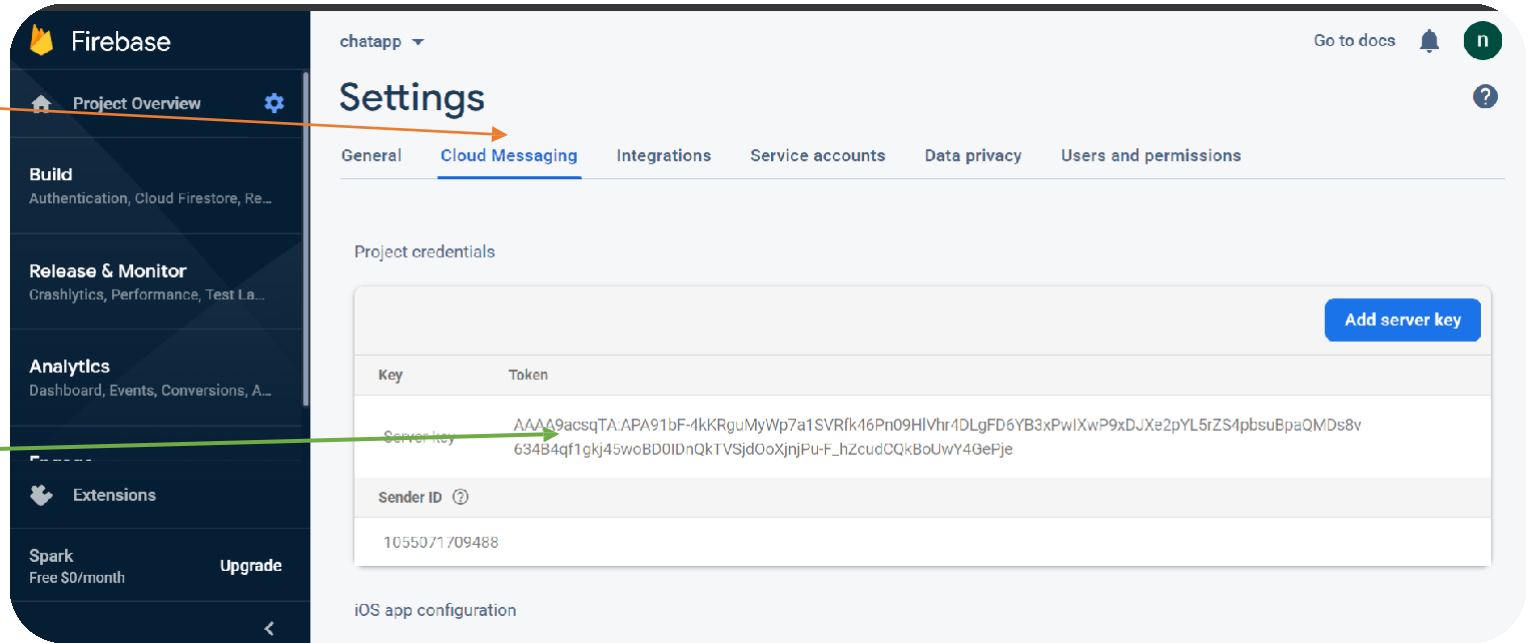


Click on Done



Your Private Network Creation is fully completed

Click on Cloud Messaging



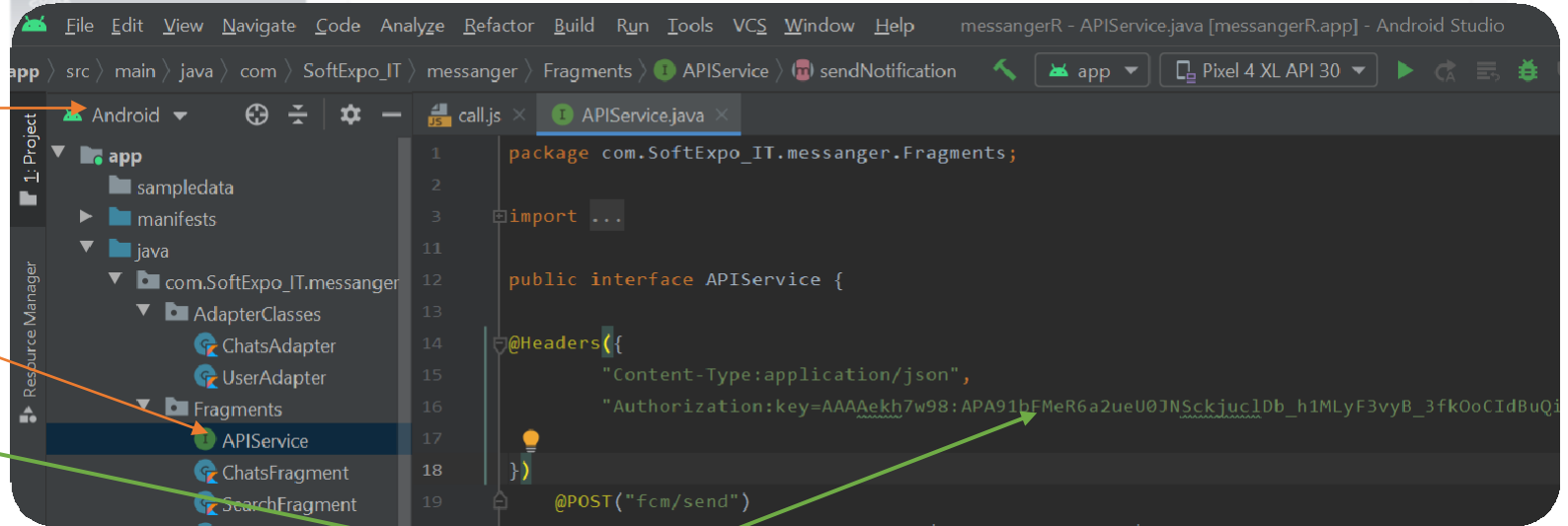
Copy Server Key Token

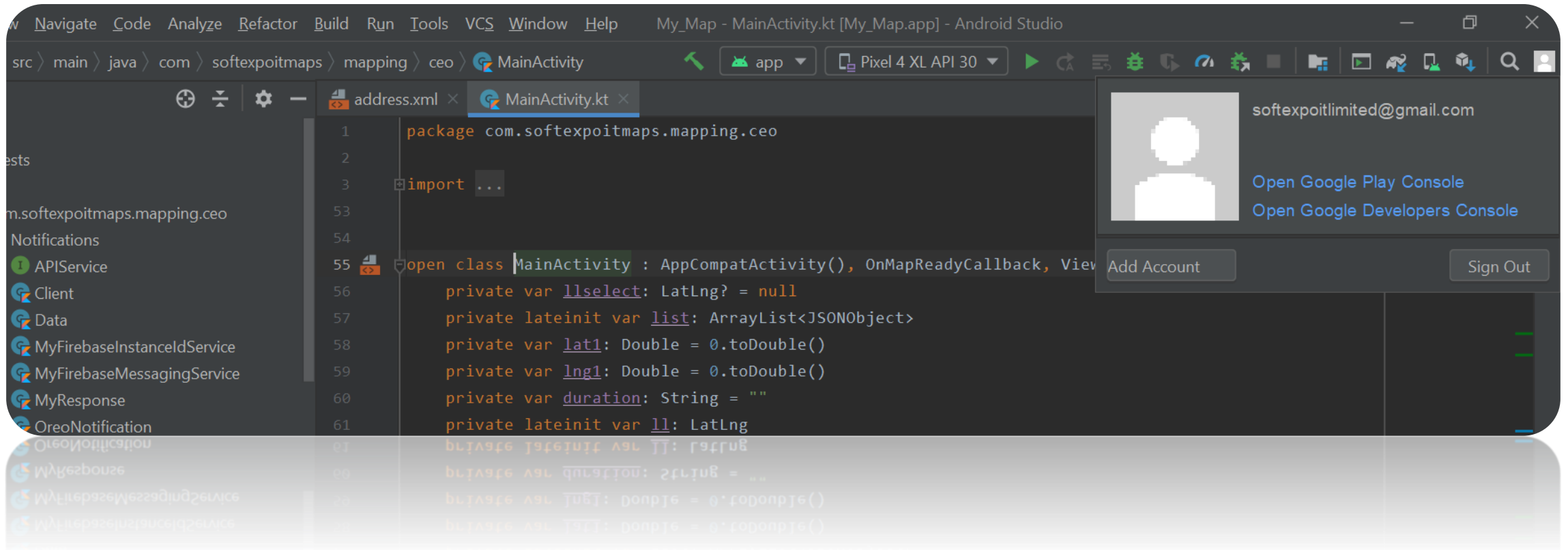
Select Android here

Go to ApiService class

Then delete the existing key value
And paste your newly copied
Token here

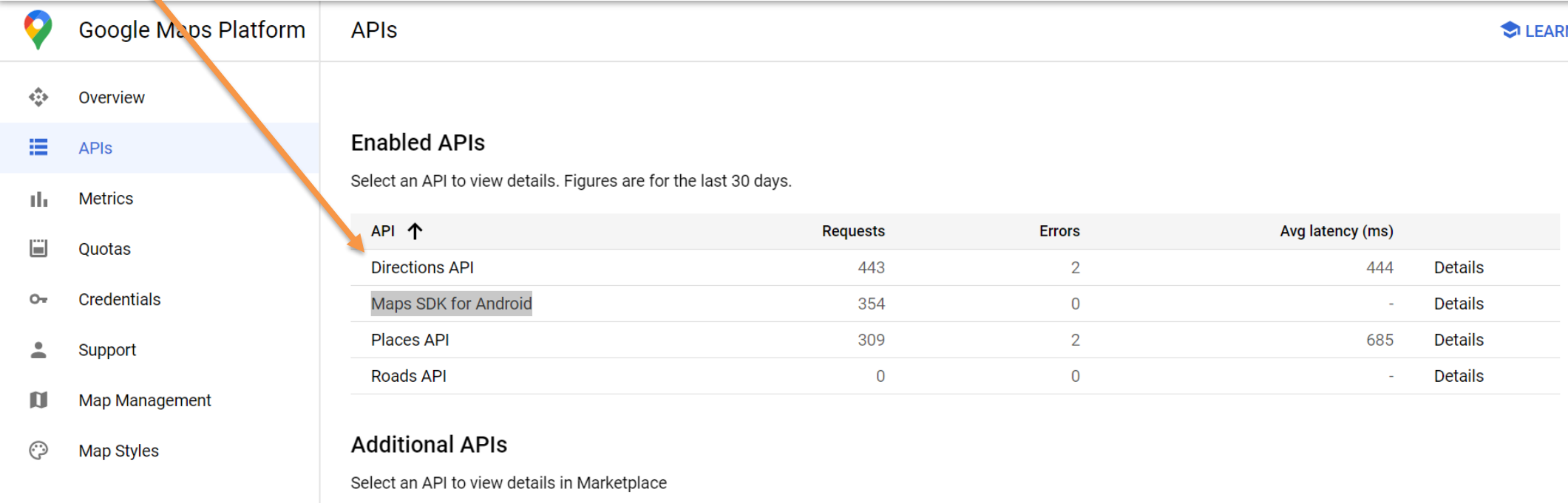
Done. Push Notification is setup 100%





Make sure you are login in Android studio with the same email id
That you used in firebase console to make the project

Here we use Google Apis For searching, Navigating, Finding Places Around The World as well as Nearby places We use Places API, Directions API, Maps SDK for Android... You have to Sign in [Google Cloud Platform](#) and Enable these Apis



Google Maps Platform APIs LEARN

- Overview
- APIs**
- Metrics
- Quotas
- Credentials
- Support
- Map Management
- Map Styles

Enabled APIs

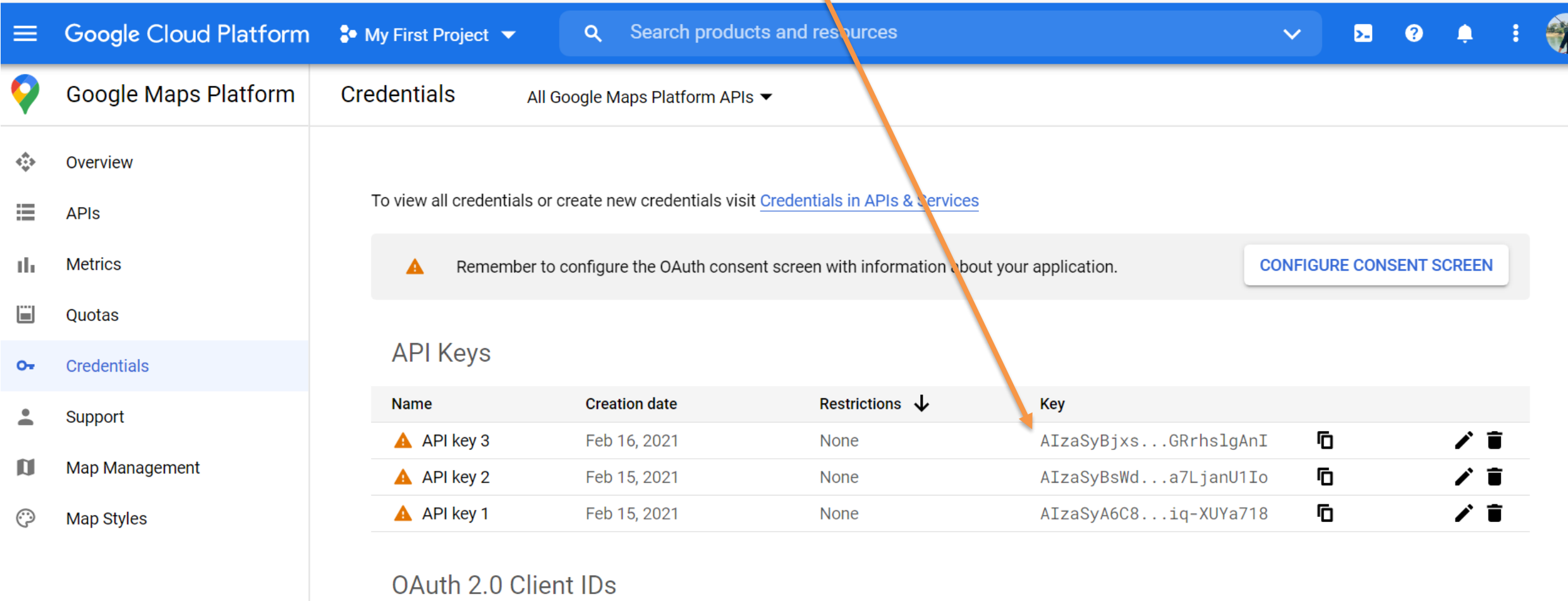
Select an API to view details. Figures are for the last 30 days.

API ↑	Requests	Errors	Avg latency (ms)	
Directions API	443	2	444	Details
Maps SDK for Android	354	0	-	Details
Places API	309	2	685	Details
Roads API	0	0	-	Details

Additional APIs

Select an API to view details in Marketplace

Generate Keys for the Apis in Credentials section....Copy the keys



Google Cloud Platform My First Project Search products and resources

Google Maps Platform Credentials All Google Maps Platform APIs

Overview APIs Metrics Quotas Credentials Support Map Management Map Styles

To view all credentials or create new credentials visit [Credentials in APIs & Services](#)

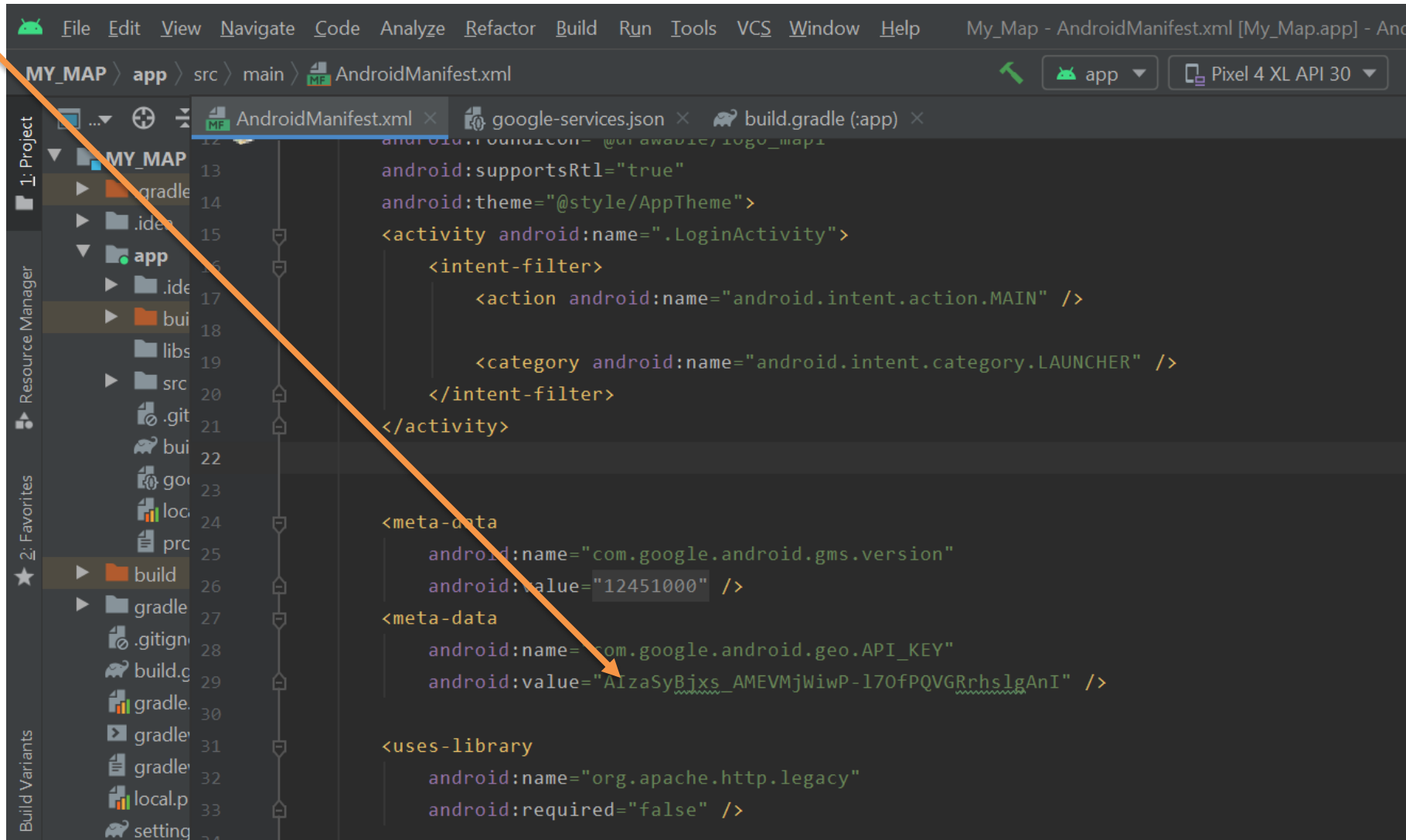
Remember to configure the OAuth consent screen with information about your application. [CONFIGURE CONSENT SCREEN](#)

API Keys

Name	Creation date	Restrictions ↓	Key		
API key 3	Feb 16, 2021	None	AIzaSyBjxs...GRrhs1gAnI	Copy	Edit Delete
API key 2	Feb 15, 2021	None	AIzaSyBsWd...a7LjanU1Io	Copy	Edit Delete
API key 1	Feb 15, 2021	None	AIzaSyA6C8...iq-XUYa718	Copy	Edit Delete

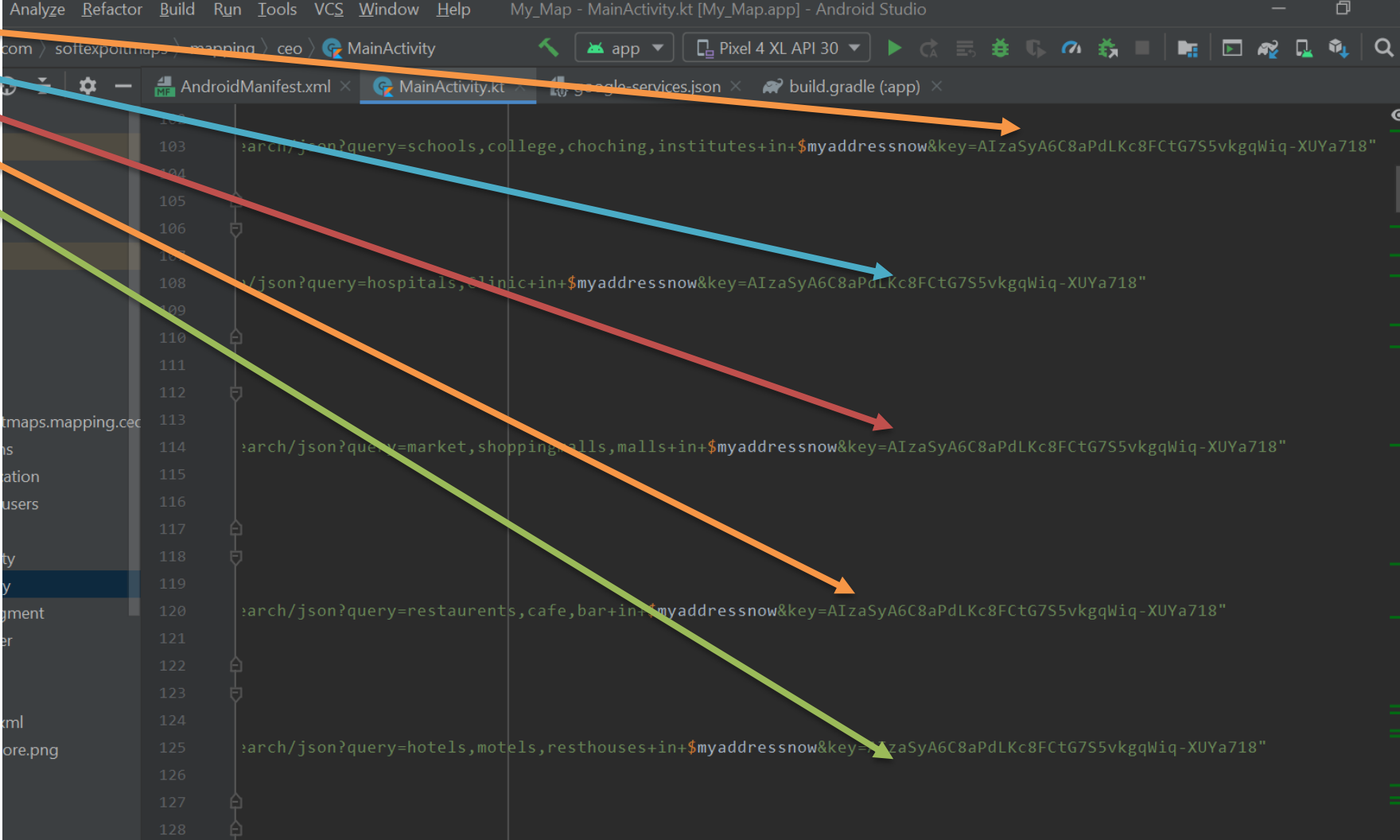
OAuth 2.0 Client IDs

Copy maps sdk api key and
paste it here



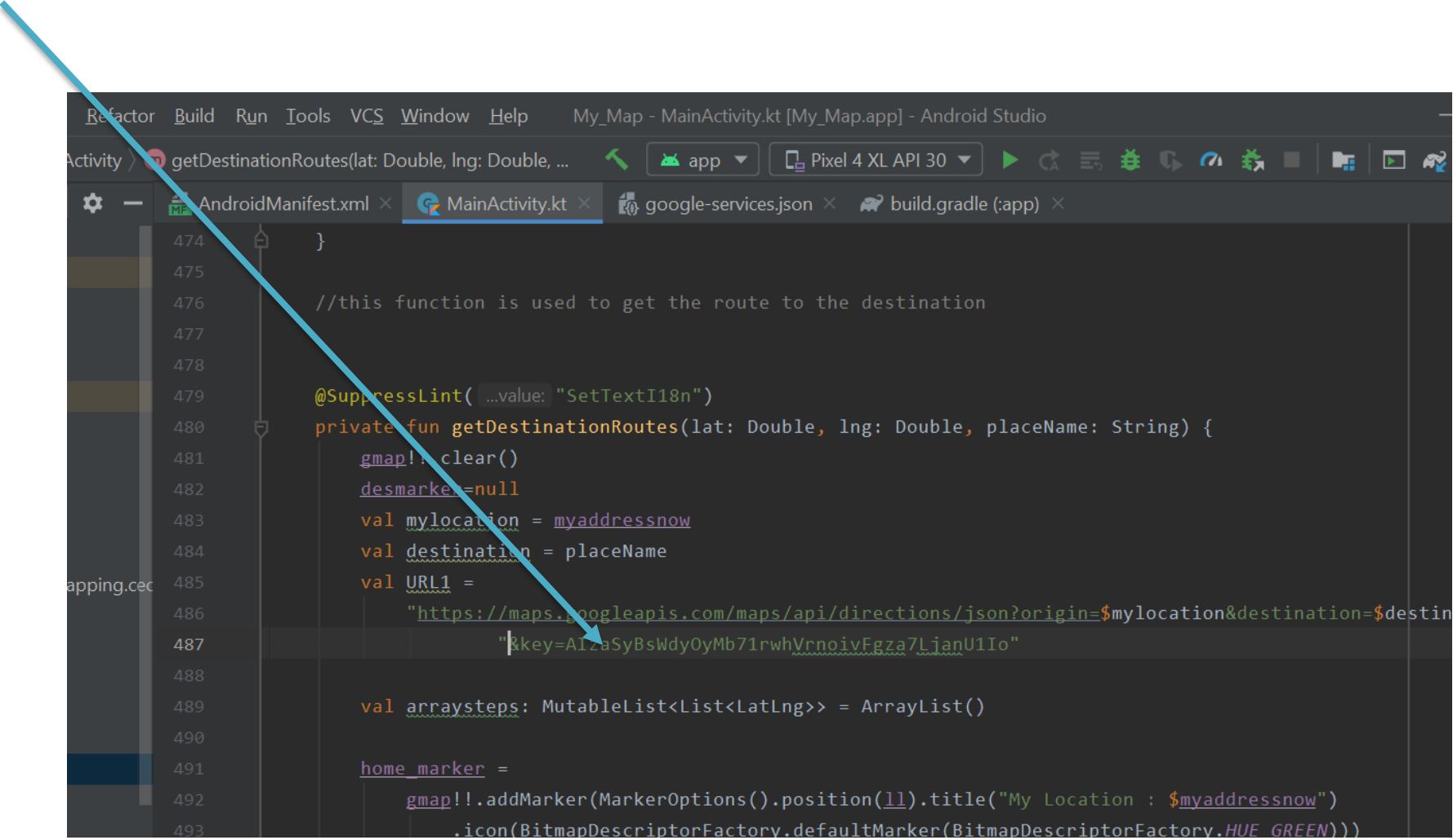
```
12 android:roundIcon="@drawable/logo_map"
13
14 android:supportsRtl="true"
15
16 android:theme="@style/AppTheme">
17   <activity android:name=".LoginActivity">
18     <intent-filter>
19       <action android:name="android.intent.action.MAIN" />
20
21       <category android:name="android.intent.category.LAUNCHER" />
22     </intent-filter>
23   </activity>
24
25   <meta-data
26     android:name="com.google.android.gms.version"
27     android:value="12451000" />
28
29   <meta-data
30     android:name="com.google.android.geo.API_KEY"
31     android:value="AIzaSyBjxs_AMEVMjWiwP-170fPQVGRrhslgAnI" />
32
33   <uses-library
34     android:name="org.apache.http.legacy"
35     android:required="false" />
```

Copy places api key
and paste it here



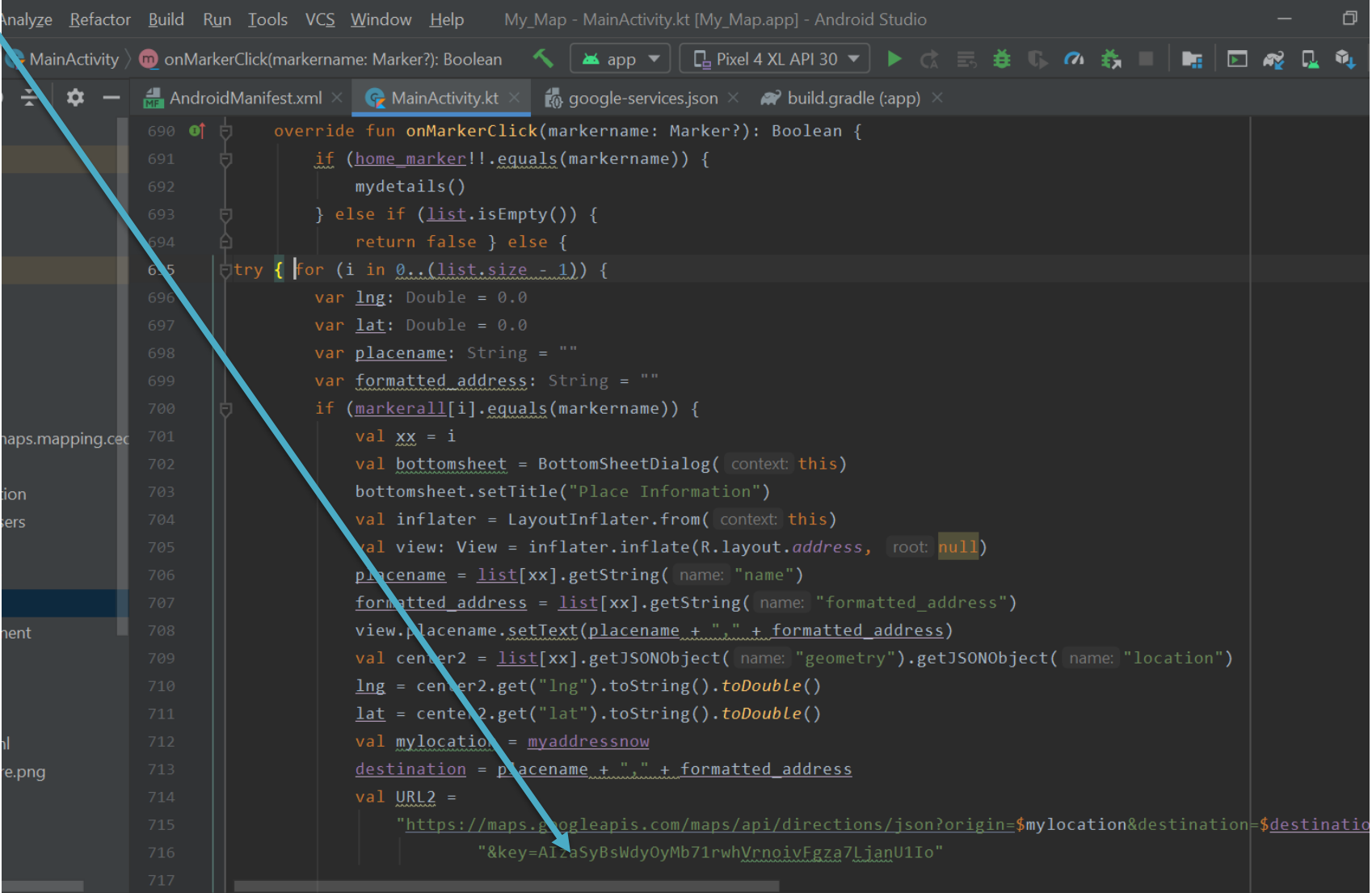
```
103     search/json?query=schools,college,choching,institutes+in+$myaddressnow&key=AIzaSyA6C8aPdLKc8FCtG7S5vkgqWiq-XUYa718"  
104  
105  
106  
107  
108     search/json?query=hospitals,clinic+in+$myaddressnow&key=AIzaSyA6C8aPdLKc8FCtG7S5vkgqWiq-XUYa718"  
109  
110  
111  
112  
113  
114     search/json?query=market,shopping,malls,malls+in+$myaddressnow&key=AIzaSyA6C8aPdLKc8FCtG7S5vkgqWiq-XUYa718"  
115  
116  
117  
118  
119  
120     search/json?query=restaurents,cafe,bar+in+$myaddressnow&key=AIzaSyA6C8aPdLKc8FCtG7S5vkgqWiq-XUYa718"  
121  
122  
123  
124  
125     search/json?query=hotels,motels,resthouses+in+$myaddressnow&key=AIzaSyA6C8aPdLKc8FCtG7S5vkgqWiq-XUYa718"  
126  
127  
128
```

Copy Directions Api key and
paste it here



```
474     }
475
476     //this function is used to get the route to the destination
477
478
479     @SuppressWarnings("SetTextI18n")
480     private fun getDestinationRoutes(lat: Double, lng: Double, placeName: String) {
481         gmap!!.clear()
482         desmarker = null
483         val mylocation = myaddressnow
484         val destination = placeName
485         val URL1 =
486             "https://maps.googleapis.com/maps/api/directions/json?origin=$mylocation&destination=$destination&key=AlzaSyBsWdyOyMb71rwhVrnoivFgza7LjanU1Io"
487
488
489         val arraysteps: MutableList<List<LatLng>> = ArrayList()
490
491         home_marker =
492             gmap!!.addMarker(MarkerOptions().position(11).title("My Location : $myaddressnow")
493                 .icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_GREEN)))
```

Copy Directions Api key and
paste it here also



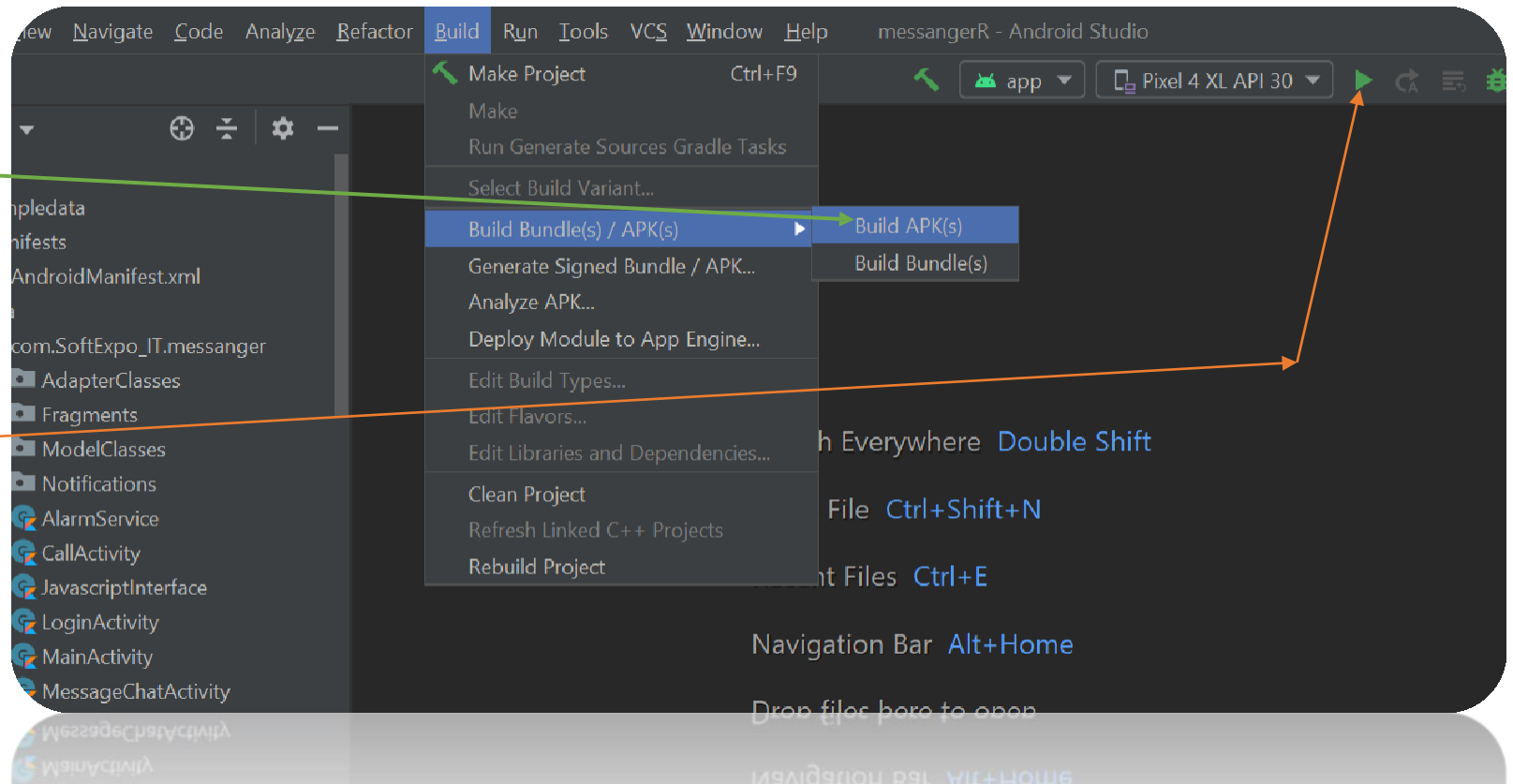
```
Analyze Refactor Build Run Tools VCS Window Help My_Map - MainActivity.kt [My_Map.app] - Android Studio
MainActivity onMarkerClick(markername: Marker?): Boolean app Pixel 4 XL API 30
AndroidManifest.xml MainActivity.kt google-services.json build.gradle (:app)
690 override fun onMarkerClick(markername: Marker?): Boolean {
691     if (home_marker!!.equals(markername)) {
692         mydetails()
693     } else if (list.isEmpty()) {
694         return false } else {
695     try { for (i in 0...(list.size - 1)) {
696         var lng: Double = 0.0
697         var lat: Double = 0.0
698         var placename: String = ""
699         var formatted_address: String = ""
700         if (markerall[i].equals(markername)) {
701             val xx = i
702             val bottomsheet = BottomSheetDialog( context: this)
703             bottomsheet.setTitle("Place Information")
704             val inflater = LayoutInflater.from( context: this)
705             val view: View = inflater.inflate(R.layout.address, root: null)
706             placename = list[xx].getString( name: "name")
707             formatted_address = list[xx].getString( name: "formatted_address")
708             view.placename.setText(placename + " " + formatted_address)
709             val center2 = list[xx].getJSONObject( name: "geometry").getJSONObject( name: "location")
710             lng = center2.get("lng").toString().toDouble()
711             lat = center2.get("lat").toString().toDouble()
712             val mylocation = myaddressnow
713             destination = placename + " " + formatted_address
714             val URL2 =
715                 "https://maps.googleapis.com/maps/api/directions/json?origin=$mylocation&destination=$destinatio
716                 "&key=A1zaSyBsWdy0yMb71rwhVrnoivFgza7LJanU1Io"
717
```

Instruction 5

- Build your Project to get your app's APK file...The process is :
Build->Build Bundle(s)/APK(s)->Build APK(s)

Build APK File

Before Build You can run
your file in emulator with
this



Instruction 6

- Deploy The APK File in the play store/Any where you want.
- You can also Distribute the APK File with any Link

That's it . These are the only steps you need to Follow .

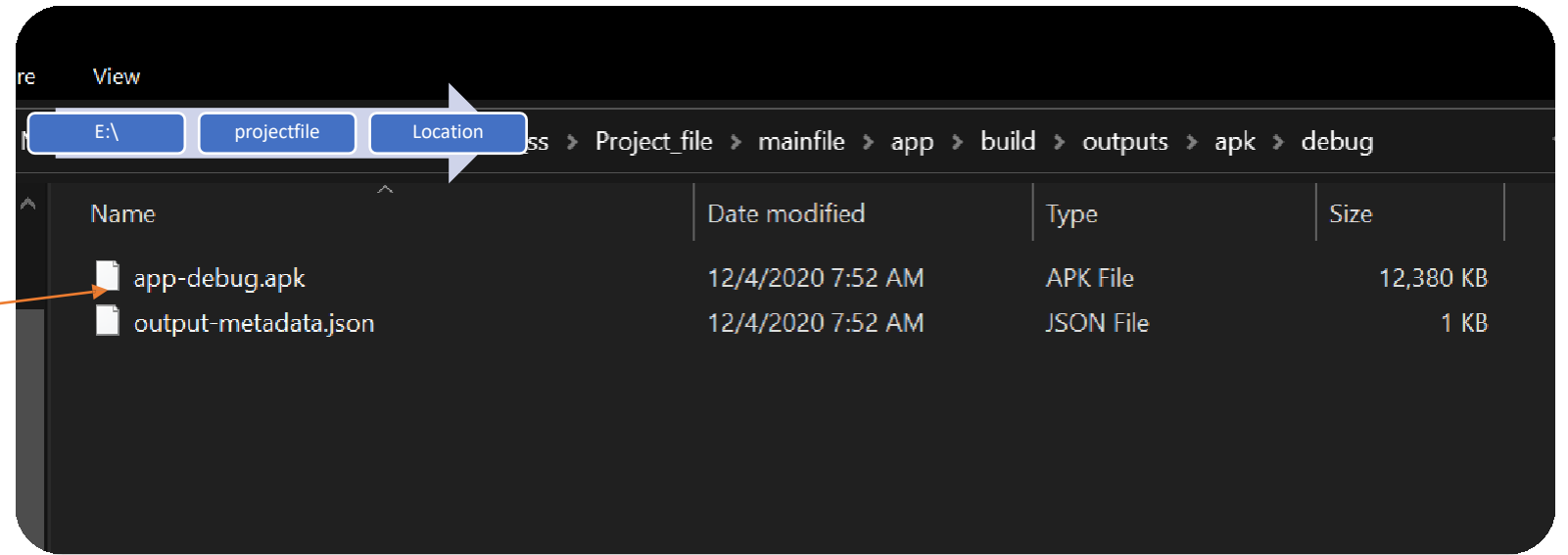
You can also get the apk file from

`Projectfile_Location\mainfile\app\build\outputs\apk\debug`

Our APK APP

So if Anyone don't want to build
APKs

They will Directly use our Released
APK File from Here



Developed By Softexploit Limited
Email : softexploitlimited@gmail.com