



Eber Driver Documentation

● Install Android Studio

You can check steps for download and install android studio for mac / windows / linux from [here](#)

★ [download the latest version of Android Studio](#).

★ Here are the steps for installing studio in ubuntu

To install Android Studio on ubuntu, proceed as follows:

1. Unpack the .zip file you downloaded to an appropriate location for your applications, such as within /usr/local/ for your user profile, or /opt/ for shared users.
To launch Android Studio, open a terminal, navigate to the android-studio/bin/ directory, and open terminal there execute studio.sh by entering `./studio.sh`
2. Select whether you want to import previous Android Studio settings or not, then click **OK**.
3. The Android Studio Setup Wizard guides you through the rest of the setup, which includes downloading Android SDK components that are required for development.
4. The following video shows each step of the recommended setup procedure. <https://developer.android.com/studio/videos/studio-install-linux.mp4>

● Changes In Projects(android)

1.Open Project in Android Studio

File->open

2.Change package name

-Find <PROJECT_ROOT>\app\build.gradle and Change package Name
(replace your package name here)

```
applicationId "com.elluminatiinc.taxi.driver" //replace your package name here
```

3.Change BASE_URL in this build.gradle file

-You can set BASE_URL according to flavours.
-Default flavour is “developer” so for running your project in your sever
you need to change your developer BASE_URL

```
productFlavors {  
    production {  
        flavorDimensions "default"  
        buildConfigField "String", "BASE_URL", '"https://eber.appemporio.net/'"  
    }  
  
    staging {  
        flavorDimensions "default"  
        buildConfigField "String", "BASE_URL", '"https://staging.appemporio.net/'"  
    }  
  
    developer {  
        flavorDimensions "default"  
        buildConfigField "String", "BASE_URL",  
            '"https://eberdeveloper.appemporio.net/'"  
    }  
  
    local {  
        flavorDimensions "default"  
        buildConfigField "String", "BASE_URL", '"http://192.168.0.141:5000/'"  
    }  
}
```

4.Change in build.gradle file

-Change version code and version number to 1

```
versionCode 1  
versionName "1.0.0"
```

5. Change package name in provider_paths.xml file(both app)

-For FileProvider to request content URIs for the images/ subdirectory of your private file area.

-We are using this for getting content uri of image captured through camera

-Goto eber->src->res->xml->provider_paths.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<paths>  
  <external-path  
    name="my_images"  
    path="Android/data/com.elluminatiinc.taxi.driver/files/Pictures" />  
    //replace your package name here  
</paths>
```

★ Change your Theme color

File name : eber -> src -> main -> res ->values->colors.xml

Driver app

```
<color name="color_app_status_bar_green">#fad014</color>  
<color name="color_app_theme_dark">#fad014</color>  
<color name="color_app_theme_light">#fad014</color>  
<color name="color_app_button">#fad014</color>  
<color name="color_app_button_tint">#fad014</color>  
<color name="color_end_trip_progress">#fad014</color>  
<color name="color_app_red_path">#fad014</color>
```

★ Change drawable xml icon color

File name : eber -> src -> main -> res ->drawable

★ Change drawable png icon color

Ex. splash screen, image logo, notification icon etc...

File name : eber -> src -> main -> res

drawable-hdpi

drawable-xhdpi

Drawable-xxhdpi

★ Change include subdomain in

src->main->res->xml->network_security_config.xml

```
<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
  <domain-config cleartextTrafficPermitted="true">

    <domain includeSubdomains="true">192.168.0.141</domain> //Add Your Base Url
  </domain-config>
</network-security-config>
```

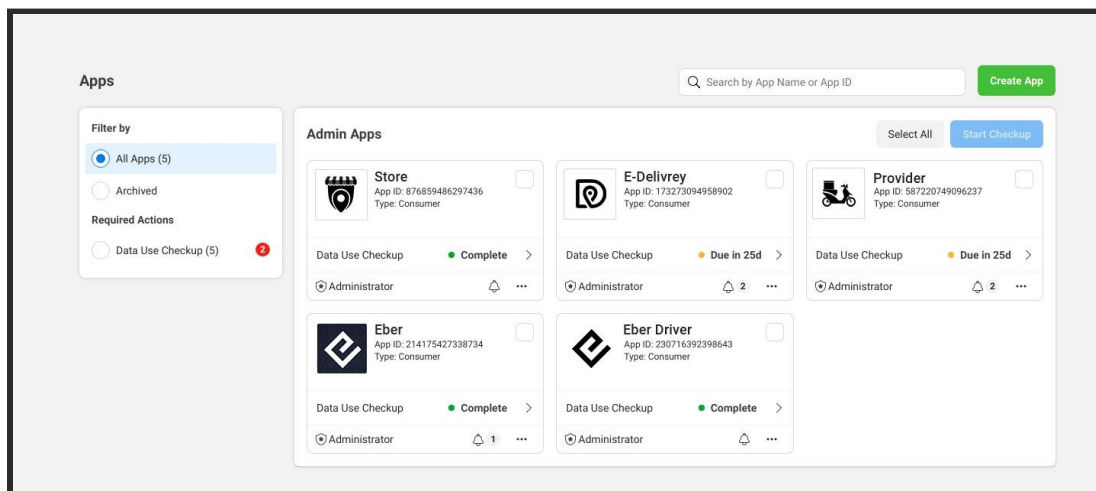
★ FACEBOOK_APP_ID and FB_LOGIN_PROTOCOL_SCHEME (facebook Login)

For enabling facebook social login

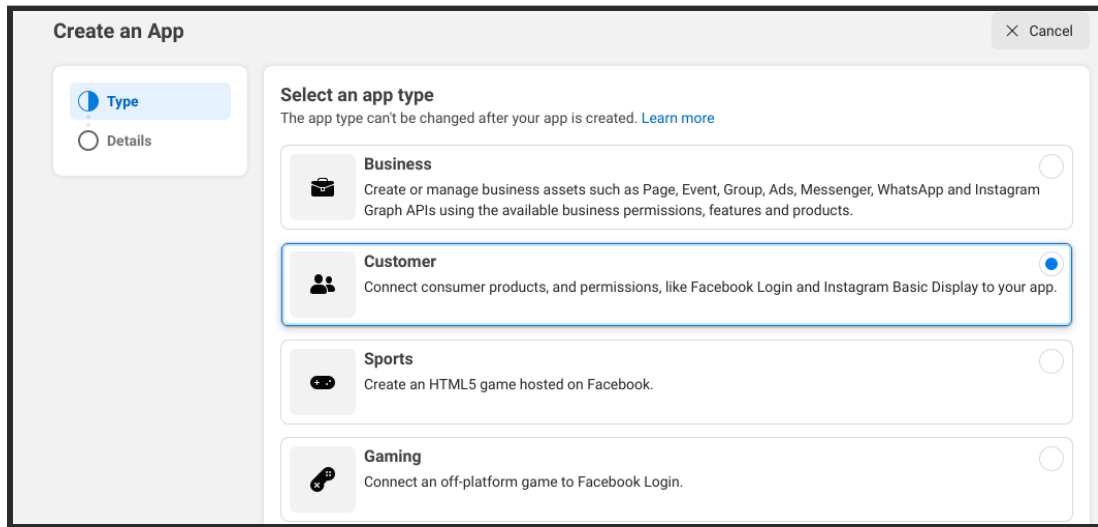
Create facebook account after open facebook developer site

<https://developers.facebook.com/apps/>

1. Create a New App



2. Click on customer after click next



The screenshot shows the 'Create an App' dialog box with the 'Type' tab selected. The 'Select an app type' section lists four options: Business, Customer, Sports, and Gaming. The 'Customer' option is selected, indicated by a blue dot and a blue border. The 'Business' option is also visible with its description: 'Create or manage business assets such as Page, Event, Group, Ads, Messenger, WhatsApp and Instagram Graph APIs using the available business permissions, features and products.'

Create an App [X] Cancel

Type Details

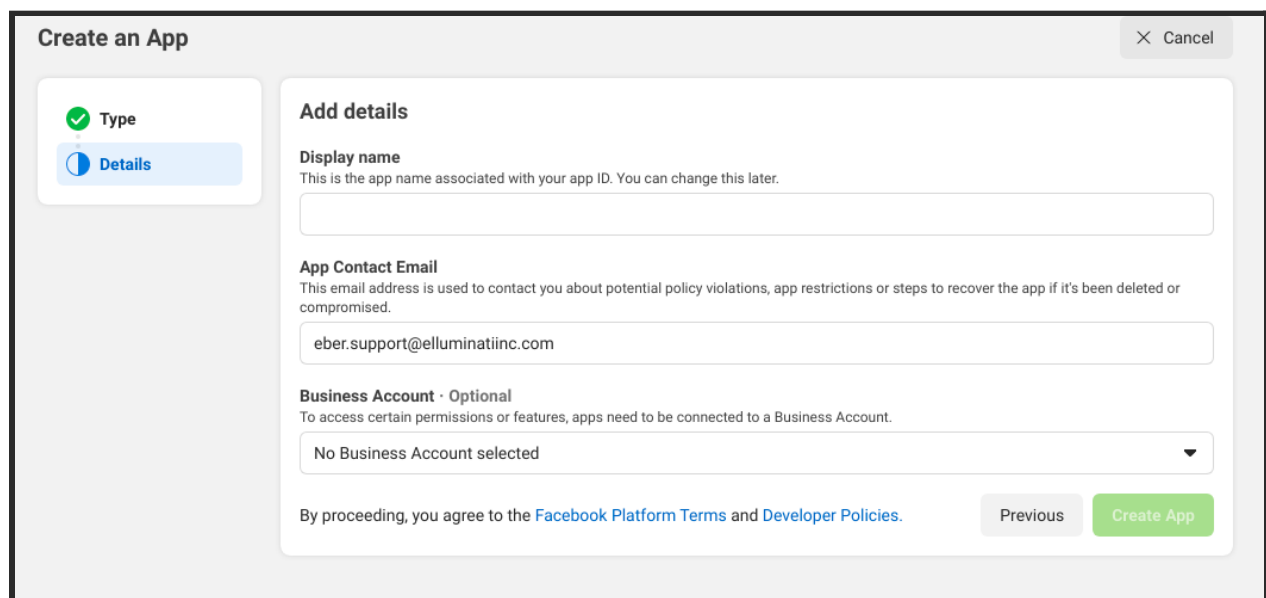
Select an app type
The app type can't be changed after your app is created. [Learn more](#)

- Business**
Create or manage business assets such as Page, Event, Group, Ads, Messenger, WhatsApp and Instagram Graph APIs using the available business permissions, features and products.
- Customer** (Selected)
Connect consumer products, and permissions, like Facebook Login and Instagram Basic Display to your app.
- Sports**
Create an HTML5 game hosted on Facebook.
- Gaming**
Connect an off-platform game to Facebook Login.

3. Add your app name

Provide the Development and Release Key Hashes for Your App

<https://developers.facebook.com/docs/facebook-login/android/#6--provide-the-development-and-release-key-hashes-for-your-app>



The screenshot shows the 'Create an App' dialog box with the 'Details' tab selected. The 'Add details' section contains fields for 'Display name', 'App Contact Email', and 'Business Account'. The 'App Contact Email' field is filled with 'eber.support@elluminatiinc.com'. The 'Business Account' dropdown is set to 'No Business Account selected'. At the bottom, there are links to 'Facebook Platform Terms' and 'Developer Policies', and buttons for 'Previous' and 'Create App'.

Create an App [X] Cancel

Type Details

Add details

Display name
This is the app name associated with your app ID. You can change this later.

App Contact Email
This email address is used to contact you about potential policy violations, app restrictions or steps to recover the app if it's been deleted or compromised.

eber.support@elluminatiinc.com

Business Account · Optional
To access certain permissions or features, apps need to be connected to a Business Account.

No Business Account selected

By proceeding, you agree to the [Facebook Platform Terms](#) and [Developer Policies](#).

Previous Create App

4. Click on settings -> basic -> + Add Platform -> after select android

EB4DFC6Xoq3iG2i4D5NhLsXdM9E= X

☐ Single Sign-On (SSO)
Will launch from Android notifications

☒ Log In-App Events Automatically (Recommended) ⓘ
Note: When this toggle is turned on, you should stop manually logging in-app purchase, start trial and subscribe events on Android. Otherwise you will see duplicate reporting. [Learn More](#)

☐ Google Play Credentials
Use your Google Developers credentials to reduce fraudulent in-app purchases.

Google Play

Package Names Class Name ⓘ

Install Referrer Decryption Key ⓘ

4bbee7f2f6dbab2d515e4f634098f3c4b650848e770d693bc37c715f65e1b38d

Remove App Store

+ Add Platform

Discard Save changes

5. Now you get one app id

Eber Taxi App ID: 21417542 App Type: Consumer App Mode: Development (Live)

App ID: 21417542 App Secret: Show

Display Name: Eber Taxi Namespace:

App Domains: Contact Email ⓘ: eber@gmail.com

Privacy Policy URL: https://eber.taxi/privacy.html Terms of Service URL: Terms of Service for Login dialog and App Details

User Data Deletion ⓘ: Data Deletion Instructions URL: https://eber.taxi/removefbaccount.html

App Icon (1024 x 1024):

Category: Choose a Category Find out more information about app categories here

App Purpose: This app's primary purpose is to access and use data from Facebook's Platform on behalf of:
☐ Yourself or your own business
☐ Clients: Select this option if the primary purpose of this app is to...

Discard Save changes

After past this key in FACEBOOK_APP_ID (ex.12345678901) and
FB_LOGIN_PROTOCOL_SCHEME is (ex. fb12345678901) for more
details check this video

<https://www.youtube.com/watch?v=qAN9KYhOSec>

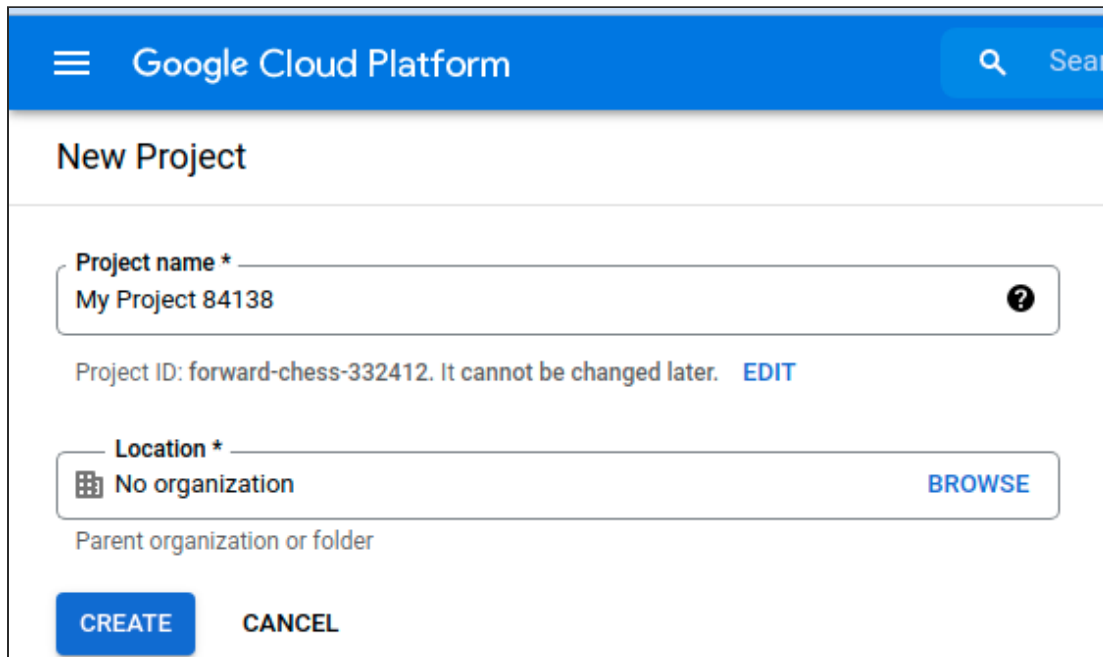
```
resValue "string", "FACEBOOK_APP_ID", "601526146700337"  
resValue "string", "FB_LOGIN_PROTOCOL_SCHEME", "fb601526146700337"
```


- **Google Cloud Console (Google Apis)**

- **Create Google Cloud Project**

For Using Google Apis (Google Map Api, Geocoding Api, Distance matrix Api etc) In our project we need to create project in google cloud console

1. Open the [Google Cloud Console](#).
2. Next to "Google Cloud Platform," click the Down arrow . A dialog listing current projects appears.
3. Click **New Project**. The New Project screen appears.
4. In the **Project Name** field, enter a descriptive name for your project. If you're executing a quickstart, use "Quickstart."
5. Click **Organization** and select your organization.
6. In the **Location** field, click **Browse** to display potential locations for your project.
7. Click a location and click **Select**.
8. Click **Create**. The console navigates to the Dashboard page and your project is created within a few minutes.



Google Cloud Platform

New Project

Project name *
My Project 84138 ?

Project ID: forward-chess-332412. It cannot be changed later. [EDIT](#)

Location *
No organization [BROWSE](#)

Parent organization or folder

[CREATE](#) [CANCEL](#)

For further information on GCP projects, refer to [Creating and managing projects](#).

- Activate Billing

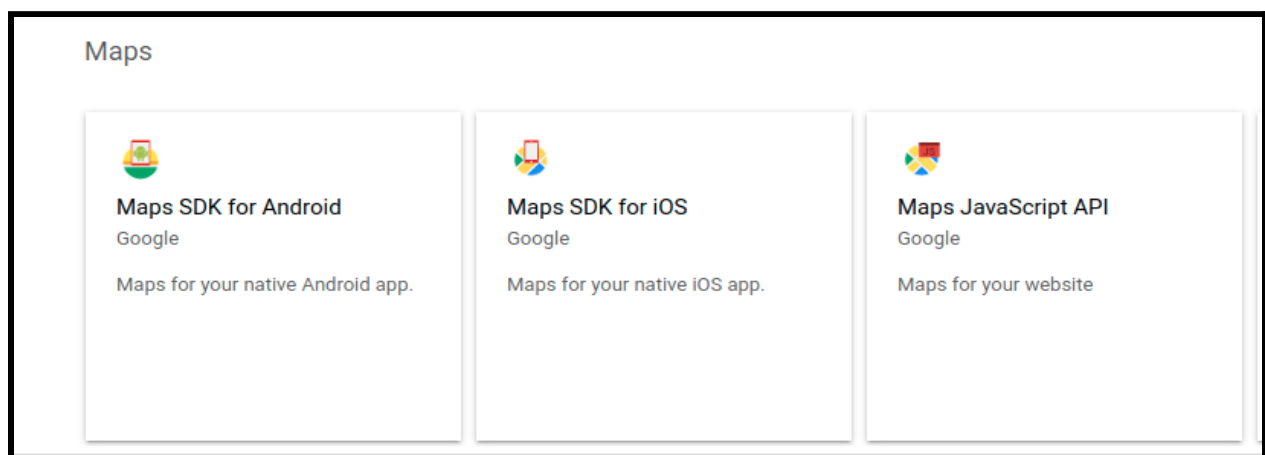
After successfully registering for a trial account you will be entitled to ~\$300 free credits that you can spend within the Google Cloud Platform (GCP). However It would recommend to set up billing by adding a valid credit / debit card.

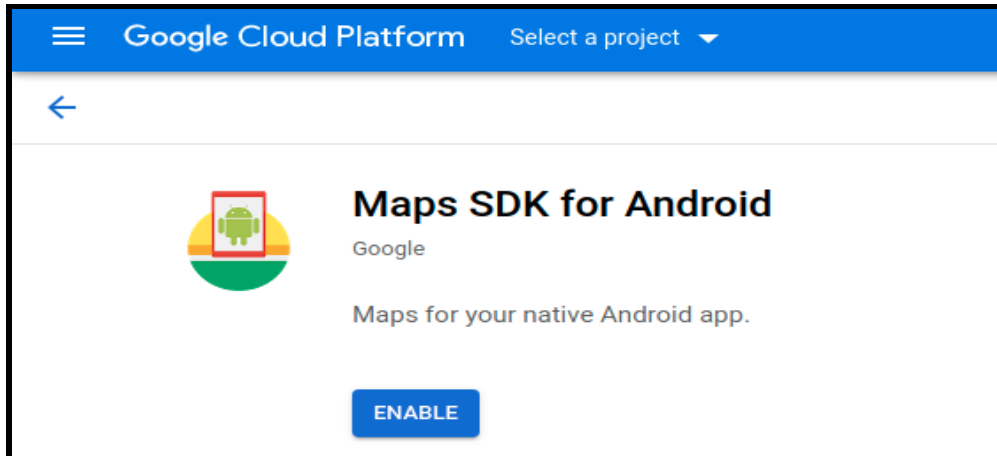
You can create a Billing Account [here](#) and its worthing remembering that one billing account can be used across multiple GCP projects.

- Enable a Google Workspace API

1. Open the [Google Cloud Console](#).
2. Next to "Google Cloud Platform," click the Down arrow and select a project.
3. In the top-left corner, click Menu - > **APIs & Services**.
4. Click **Enable APIs and Services**. The **Welcome to API Library** page appears.
5. In the search field, enter the name of the API you want to enable.
For example, type "Map API" to find the Gmail API. If you are enabling an API for a quickstart, refer to the quickstart's Prerequisites section for the API to enable.
6. Click the API to enable. The API page appears.
7. Click **Enable**. The Overview page appears.
8. To enable an additional API, repeat steps 3 - 7.

For Example:





★ Make these libraries enable

→ Maps SDK for Android

With the Maps SDK for Android, add maps to your [Android app](#) including [Wear OS](#) apps using Google Maps data, map displays, and map gesture responses. on web pages and mobile devices. Geolocation API

For more detail :-

<https://developers.google.com/maps/documentation/android-sdk/overview>

→ Geocoding API

Geocoding is the process of converting addresses (like "1600 Amphitheatre Parkway, Mountain View, CA") into geographic coordinates (like latitude 37.423021 and longitude -122.083739), which you can use to place markers on a map, or position the map.

The Geocoding API provides a direct way to access these services via an HTTP request.

For more detail :-

<https://developers.google.com/maps/documentation/geocoding/overview>

→ Distance Matrix API

The Distance Matrix API is a service that provides travel distance and time for a matrix of origins and destinations.

For more detail :-

<https://developers.google.com/maps/documentation/distance-matrix/overview>

→ Directions API

Provide directions for multiple transportation modes, featuring real-time traffic information.

For more detail :-

<https://developers.google.com/maps/documentation/directions>

→ Places API

The Places API is a service that returns information about places using HTTP requests. Places are defined within this API as establishments, geographic locations, or prominent points of interest.

For more detail :-

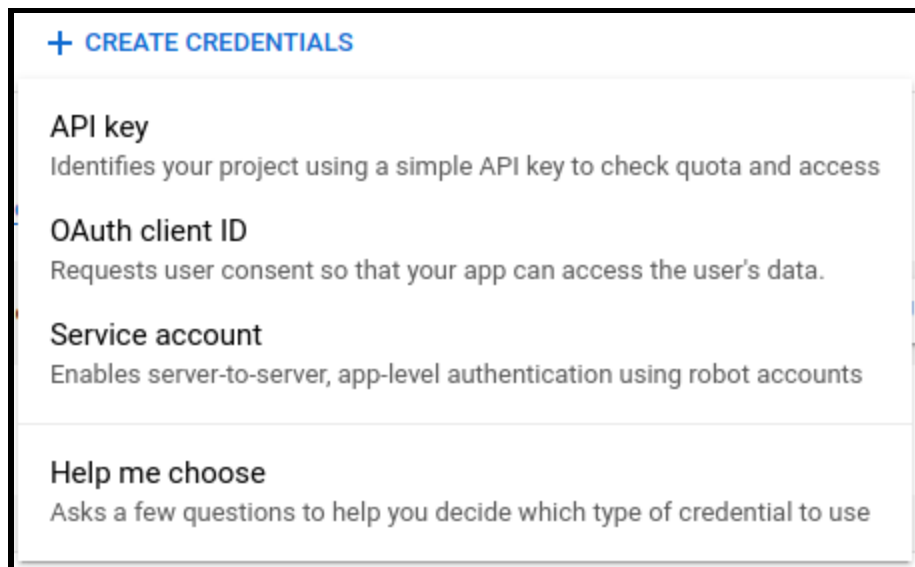
→ <https://developers.google.com/maps/documentation/places/web-service/overview>

For More information on apis you can refer :

<https://developers.google.com/maps/documentation>

- Create Api key

1. Go to the Google Maps Platform > Credentials page.
[Go to the Credentials page](#)
2. On the Credentials page, click Create credentials > API key.
The API key created dialog displays your newly created API key.
3. Click Close.
The new API key is listed on the Credentials page under API keys.
(Remember to restrict the API key before using it in production.)



- After paste this key in project build.gradle file

Goto `GOOGLE_ANDROID_API_KEY`

```
resValue "string", "GOOGLE_ANDROID_API_KEY", "AlzaSyDNFD-asdasdjKTSdssDASDbaksjasd"  
//paste your google key here
```

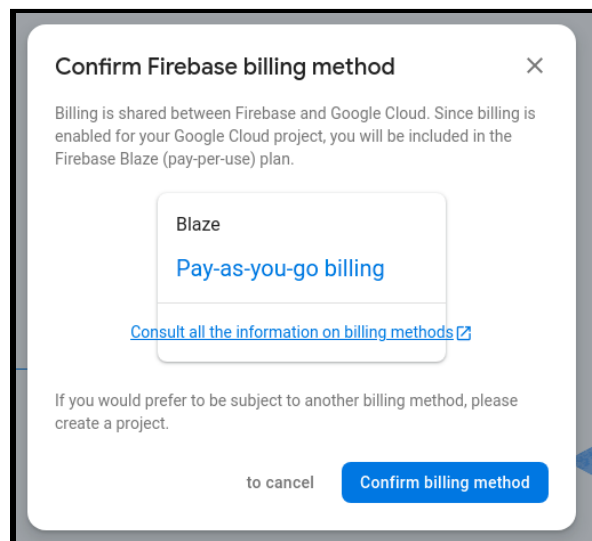
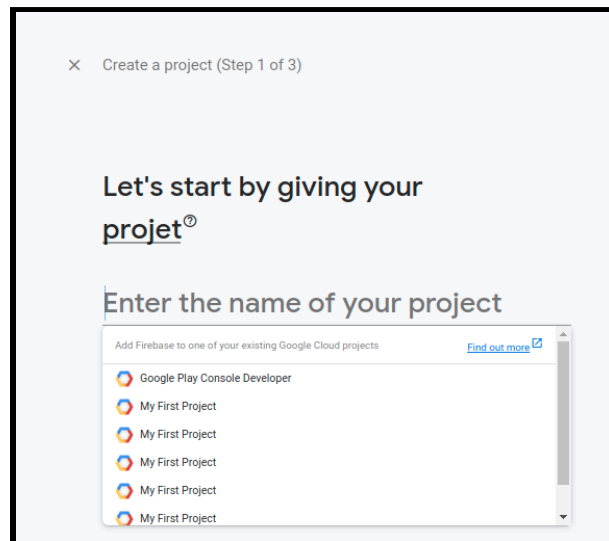
● Firebase Account

Firebase provides many utilities like cloud messaging, Crashalytics ,Analytics , RealTime Databases ,In-App Messaging , Dynamic Links etc.

You can learn more about firebase products from <https://firebase.google.com/>

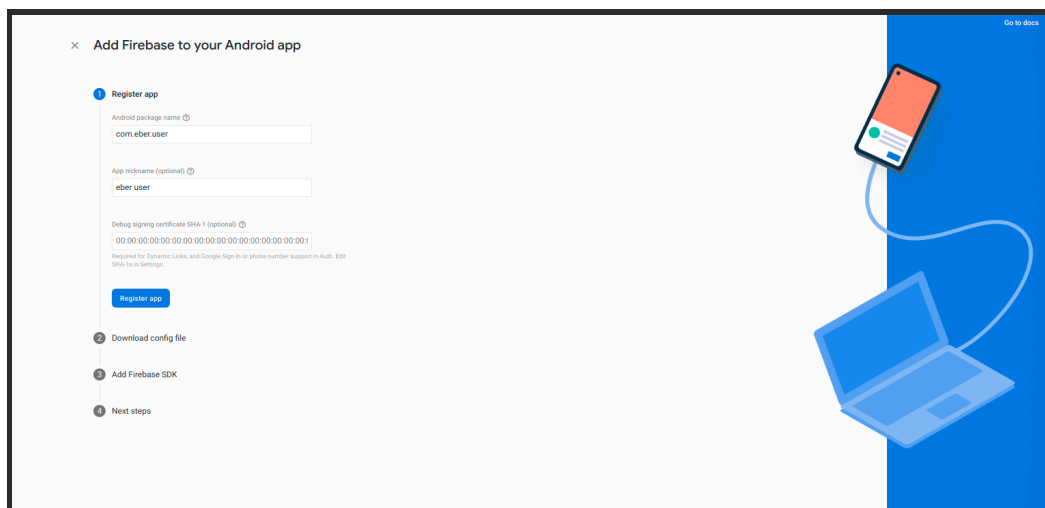
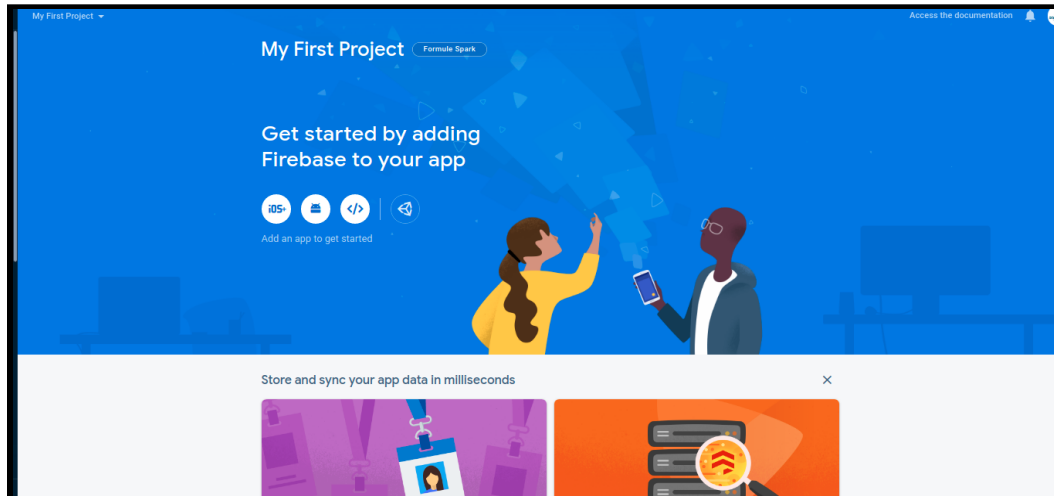
- Create Project in FirebaseConsole

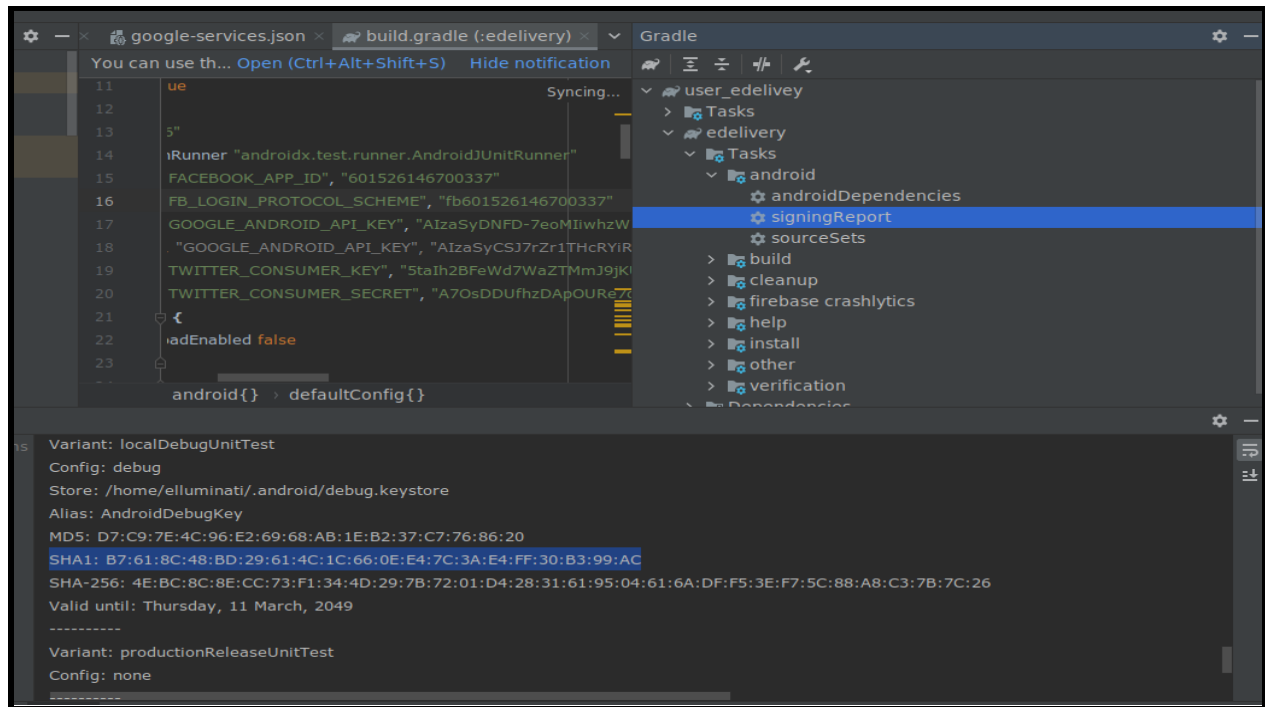
- Goto Firebase console <https://console.firebase.google.com/u/4/>
- Click Add Project
- You can see Google cloud projects you created in <https://console.cloud.google.com> here , Select Your Project and continue
- Unselect switch. You can set it up later .Continue to create project
- Confirm Your Billing Method
- In the next steps, you will be asked whether to set up Google Analytics.



App


- Now the Project is created. Add Android app by clicking on android icon
- Add your apps package name and App Name and SHA1
- Register your app and download **google-services.json** file






2

Download the configuration file

 Download google-services.json

Go to the **Project** page in Android Studio to view the root directory for your project.

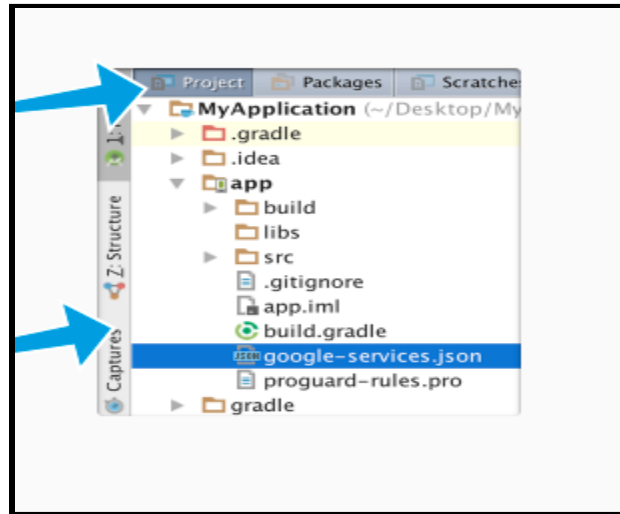
Move the google-services.json file you just downloaded to the module root directory of your Android app.



google-services.json

Following

- Add this **google-services.json** file to the module root directory of your Android app. Refer below image.



★ Create RealTimeDatabase

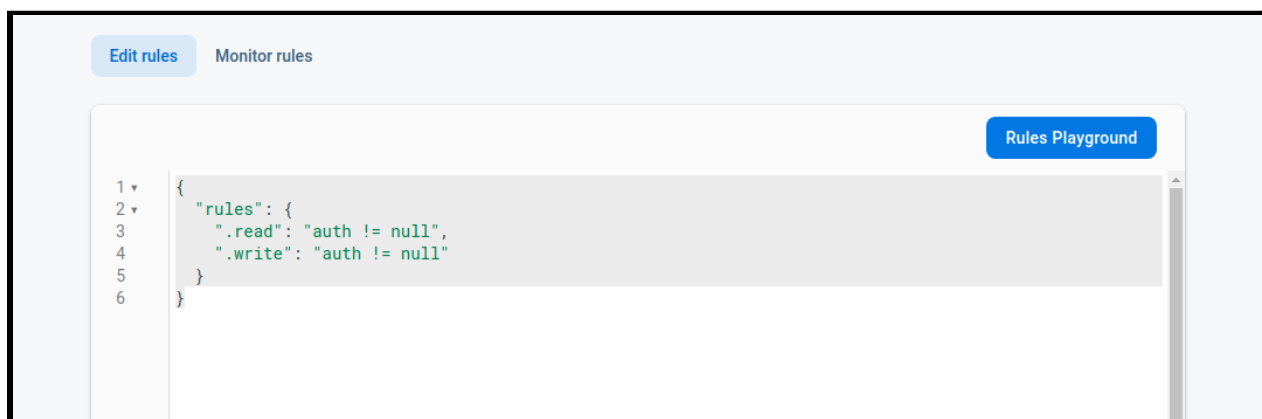
Store and sync data with our NoSQL cloud database. Data is synced across all clients in realtime, and remains available when your app goes offline.

Firebase Realtime Database Security Rules determine who has read and write access to your database, how your data is structured, and what indexes exist.

For more info check this <https://firebase.google.com/docs/database>

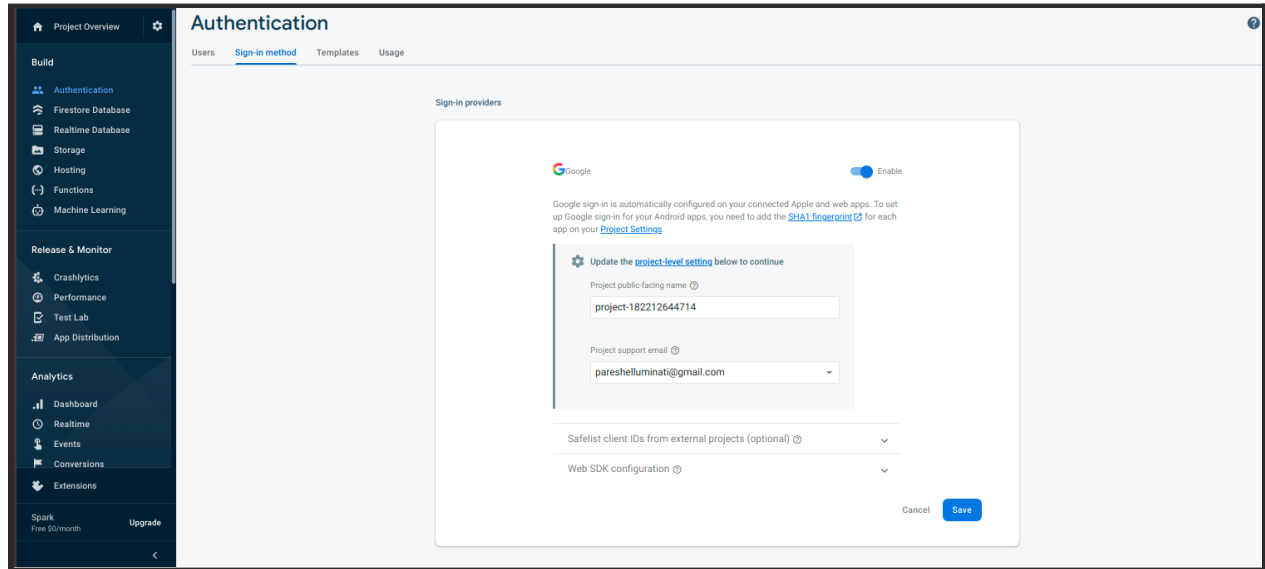
We are using Firebase realTimeDatabase for sending, retrieving ,storing chat data

- GoTo [Firebase console](#)
- Side menu -> Realtime Database -> Create Database ->select locked mode -click rules -> read true and write true ->click on publish



★ For Google Sign in :

Go to Authentication -> Get Started -> select google -> Enable -> select project support email -> save



★ For Crashlytics:

- GoTo [Firebase console](#)
- Side menu -> Crashlytics
- You can learn how to integrate crashlytics from [here](#)



★ Build project

Check build variants (check which have BASE_URL)

1. Select target device
2. Run project