Android Authentication

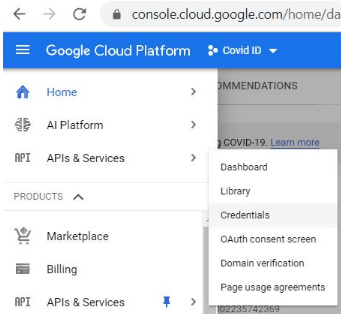
For Google service (maps, firestore, authentication-google sign in) that your application uses you will generation an API key (this is an identifier that will be sent to the API server saying who you are). You can generate multiple service keys.   For each such API key, you will want to restrict access to the corresponding service(s) based on a signature certificate key used to sign the application (to make it secure). You want to restrict access to Google Services so that you are not attacked by others flooding the services (and costing you money) with requests. **Restrictions happen by listing what you want to restrict (android versus web versus ios) for an API key AND giving all of the permitted SHA fingerprints associated with certificate keys used to sign incoming requests.**

>>>currently Google lets you only choose one application restriction per API key --which is why you can generate multiple API keys for your project in case you have multiple entries (Android , Web, iOS). The rest of this document shows how for a particular API key restricting it in the case of Android access (but, it works the same for any kind of access).

How this will be done is different depending on if you are running in developer mode versus production mode for your app.   AND you need to do it separately for general Google Cloud and Firebase.

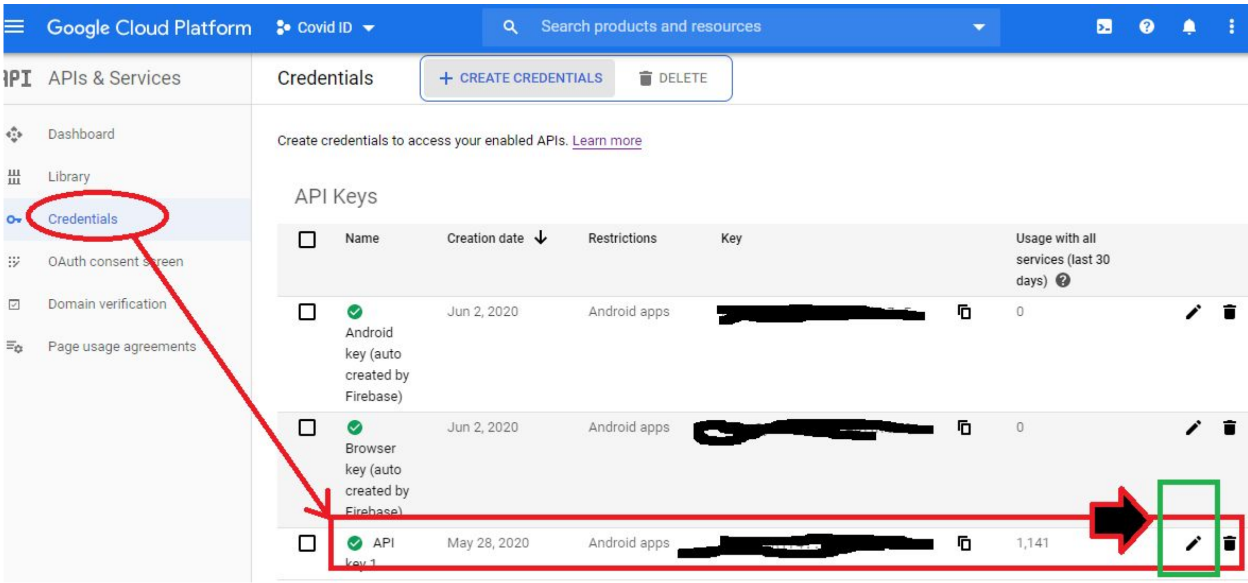
Developer mode

- Unfortunately, there is not a way to share one key for development mode and have it run. See https://stackoverflow.com/questions/54723123/how-multiple-developers-can-work-on-the-same-android-app-connected-to-a-single-f/54723198
- What you do is you need to restrict access to the APIs (services like maps, etc) that you use and register EACH developers SHA number for their local debug key.
    a. Go to http://console.cloud.google.com/
    b. Go to your project



    c. Go to APIs->Credentials

d. Find ANY and all api keys you want to restrict and select the edit button



Create credentials to access your enabled APIs. Learn more

API Keys

☐	Name	Creation date ↓	Restrictions	Key	Usage with all services (last 30 days)
☐	Android key (auto created by Firebase)	Jun 2, 2020	Android apps	[redacted]	0
☐	Browser key (auto created by Firebase)	Jun 2, 2020	Android apps	[redacted]	0
☐	API key 1	May 28, 2020	Android apps	[redacted]	1,141

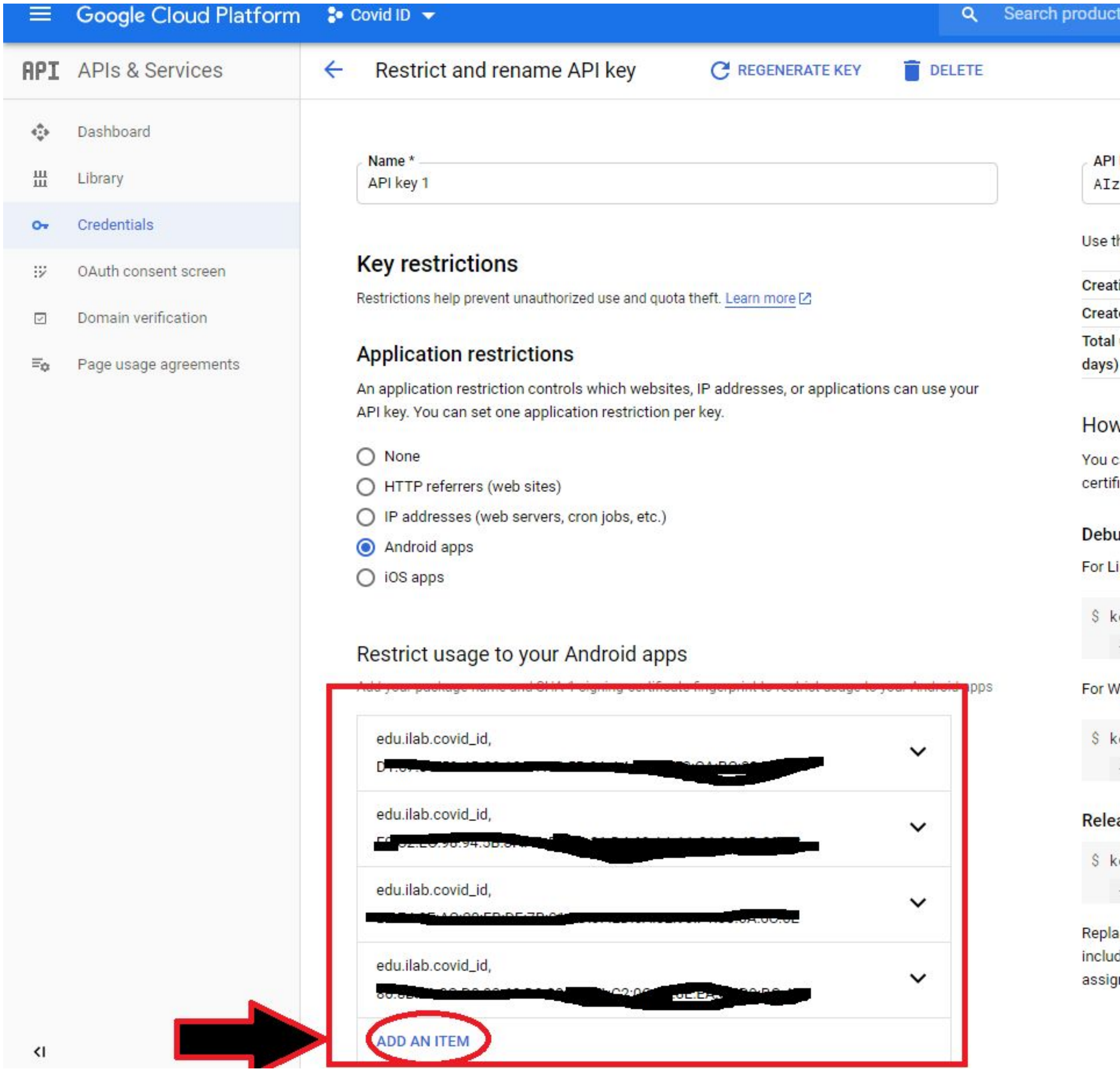
e. NOW add in for each of your developers one at a time the package name of app with THEIR SHA certificate fignerprints/numbers that are associated with their debug key (stored in .android directory on machine by default ---can find using keytool application that comes with java...search on web how to do see.)

MAC/Unix:

```
keytool -list -v -keystore ~/.android/debug.keystore -alias androiddebugkey
-storepass android -keypass android
```

WINDOWS:

```
keytool -list -v -keystore "%USERPROFILE%\.android\debug.keystore" -alias
androiddebugkey -storepass android -keypass android
(assumes you have environmental variable USERPROFILE defined otherwise
specify entire path)
```



Restrict and rename API key

Name *
API key 1

Key restrictions

Restrictions help prevent unauthorized use and quota theft. Learn more

Application restrictions

An application restriction controls which websites, IP addresses, or applications can use your API key. You can set one application restriction per key.

○ None
○ HTTP referrers (web sites)
○ IP addresses (web servers, cron jobs, etc.)
● Android apps
○ iOS apps

Restrict usage to your Android apps

Add your package name and SHA-1 signing certificate fingerprint to restrict usage to your Android apps

edu.ilab.covid_id,
[redacted]

edu.ilab.covid_id,
[redacted]

edu.ilab.covid_id,
[redacted]

edu.ilab.covid_id,
[redacted]

ADD AN ITEM

- Special note: when the application on the developers machine is run on an emulator OR a connected physical device it is "signed" with their debug key and it will use this certificate when it sends requests to firebase for services.   Firebase knows to ONLY allow requests with these SHA certificate numbers to make requests.

QUESTION: why are there autogenerated keys in console.cloud.google.com from Firebase???

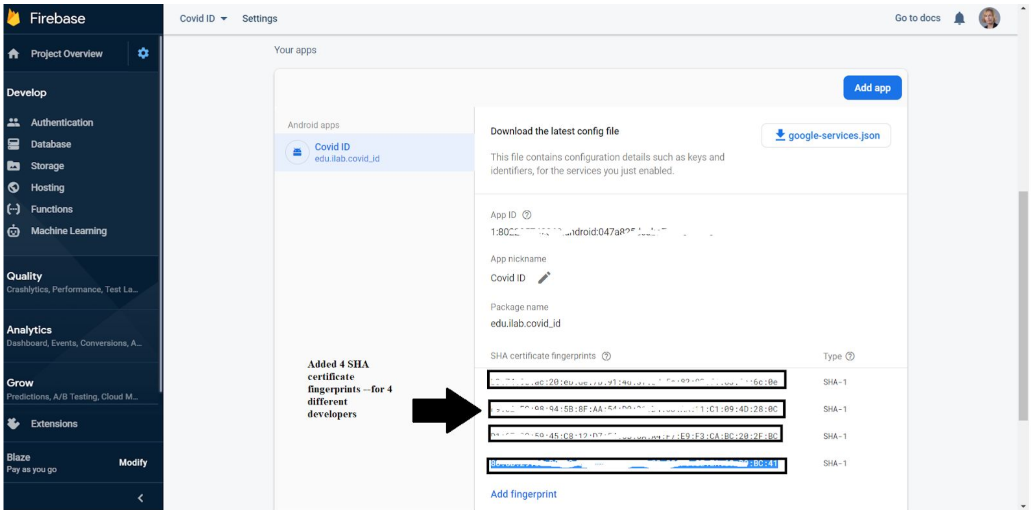
Production mode

- You will need to generate a key that will be used for signing you app. Follow the instructions on https://developer.android.com/studio/publish/app-signing#signing-manually
- I BELIEVE??? Then if you register this ONE key inside the google cloud restrictions (AND below for Firebase) it will work for ALL who download the apk (you sign the apk file with this key).
- NOTE: you can't run in developer mode as far as I know with one key that only works for signed apk file...because each developer has their own key they can't sign it with the single key...only the production engineer deploying the apk file to the playstore will do the signature with their key.

Firebase Authentication

When you choose to do Google Sign-in with Firebase it requires a different set of permissions. I believe the idea behind offering Google Sign-in through plain google cloud and differently through Firebase is Google's desire to have for basic apps Firebase be a one-stop shop (don't have to go to Google cloud) ---however the result of this is that if your app uses both Firebase services and google cloud you must know that there are different restriction settings for each and Google Cloud and Firebase have separate consoles. GO to /https://console.firebase.google.com/

1. Go to the settings for the app and the General tab
2. Restrict the apps usage as you wish and ADD all the developer SHA numbers as needed.
3. You will see a page like this Note that FOR EACH developer in development mode you must add the SHA of their machine to work. Here You can see an example where I added 4 SHA certificate fingerprints/numbers for the debug key stored on each of the 4 developer machines.



Download the latest config file

This file contains configuration details such as keys and identifiers, for the services you just enabled.

App ID
1:80[redacted]ndroid:047a8[redacted]

App nickname
Covid ID

Package name
edu.ilab.covid_id

SHA certificate fingerprints	Type
[redacted].ac:20:e[redacted]:6c:0e	SHA-1
[redacted]:98:94:5B:8F:AA:5[redacted]1:C1:09:4D:28:0C	SHA-1
[redacted]:59:45:C8:12:D7[redacted]:E9:F3:CA:BC:20:2F:BC	SHA-1
[redacted]:BC:41	SHA-1

Added 4 SHA certificate fingerprints --for 4 different developers