Google Maps View Documentation

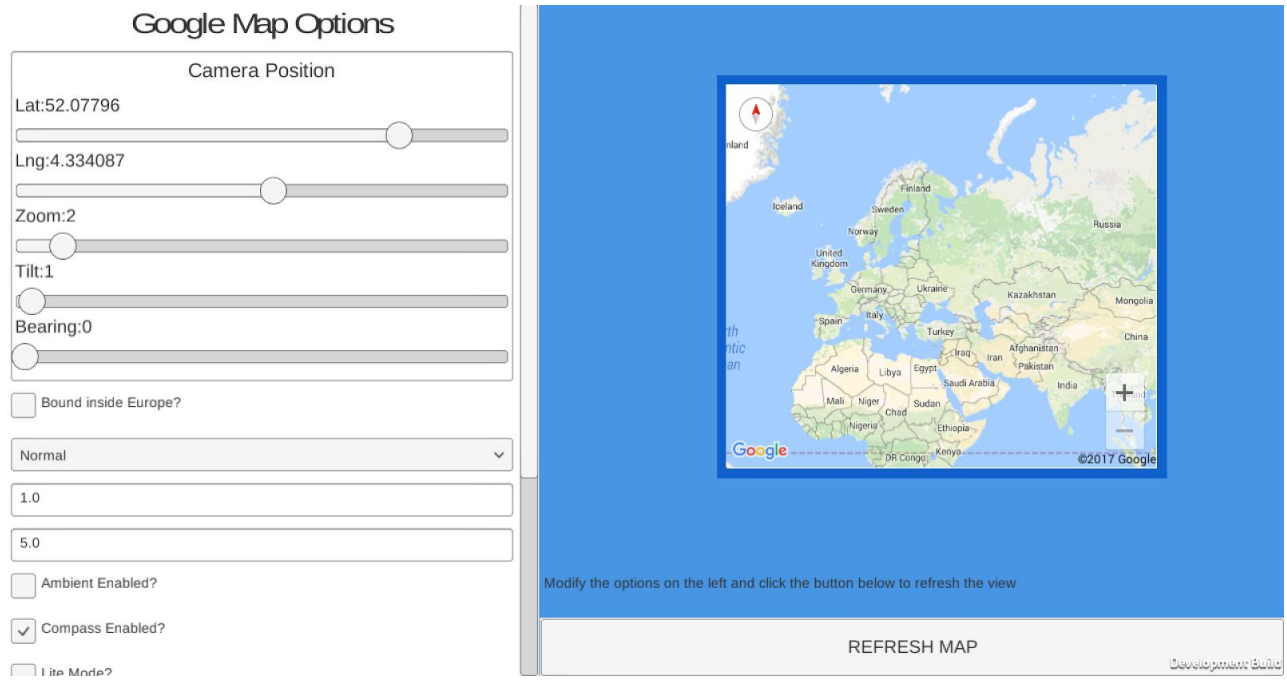
Table of Contents

For the latest up-to-date documentation always visit the Documentation Wiki. It is highly recommended to use Wiki as it is always up-to-date and easier to navigate.

For support drop me a message to leskiv.taras@gmail.com

Overview

The plugin allows you to embed Native GoogleMapsView into your Android/iOS-Unity game. Note, this is NOT a Web View and NOT a Texture, its native interactive Google Map View.



Limitations

These are the limitations to be aware of when using the plugin:

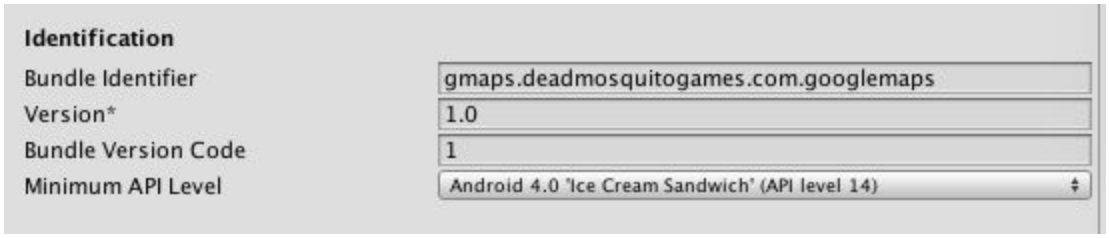
- The plugin does NOT work in Editor! It is a native Android/iOS view (not a web view) so performance is awesome but there is no way to get native Android/iOS view working in Unity Editor.
- You have to handle device orientation changes. Use Screen.orientation to determine screen orientation changes, when it changes, dismiss the view and show again passing new position rect.
- You have to handle dismissing and showing the view again in **void OnApplicationPause(bool pauseStatus)** because you might get a black screen behind map if you don't. This is very straightforward to implement (also see demo):
- You can’t move the view (e.g scroll in Unity view). The view for now is static and can’t be moved around. Repositioning might be implemented in future. Please submit an issue to this repo to request this.

Android Setup

This section describes how to get Google Maps View running in your app or demo that is included with plugin.

1. Change the default Bundle id to your Bundle Id (package name)

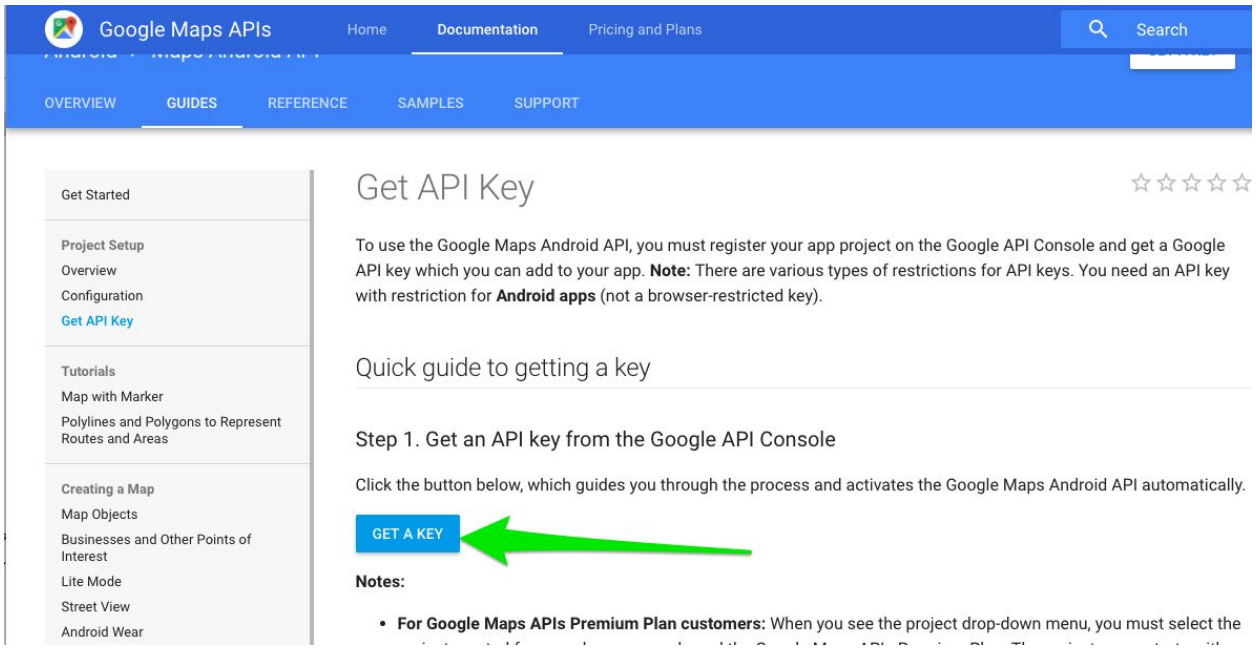
Go to Unity Android Player Settings and set the Bundle Id as your package name, e.g. `gmaps.deadmosquitogames.com.googlemaps` and save it. **This is important as Google API Key in the next step is bound to the package you set.** In this document I will refer to your package as ${YOUR_PACKAGE_NAME}



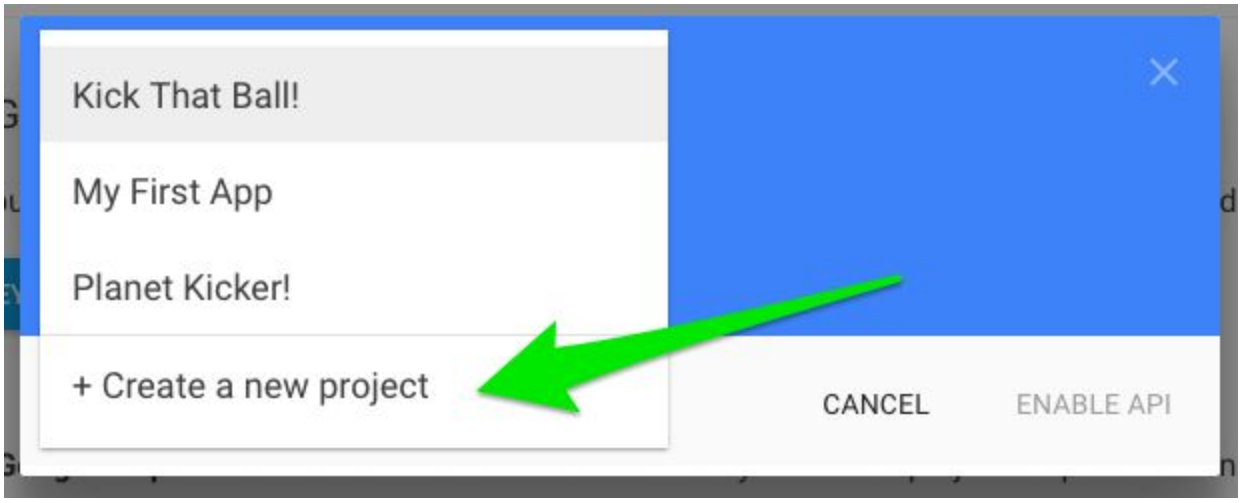
2. Obtaining Google API Key

This part is a bit tricky so please follow instructions carefully.

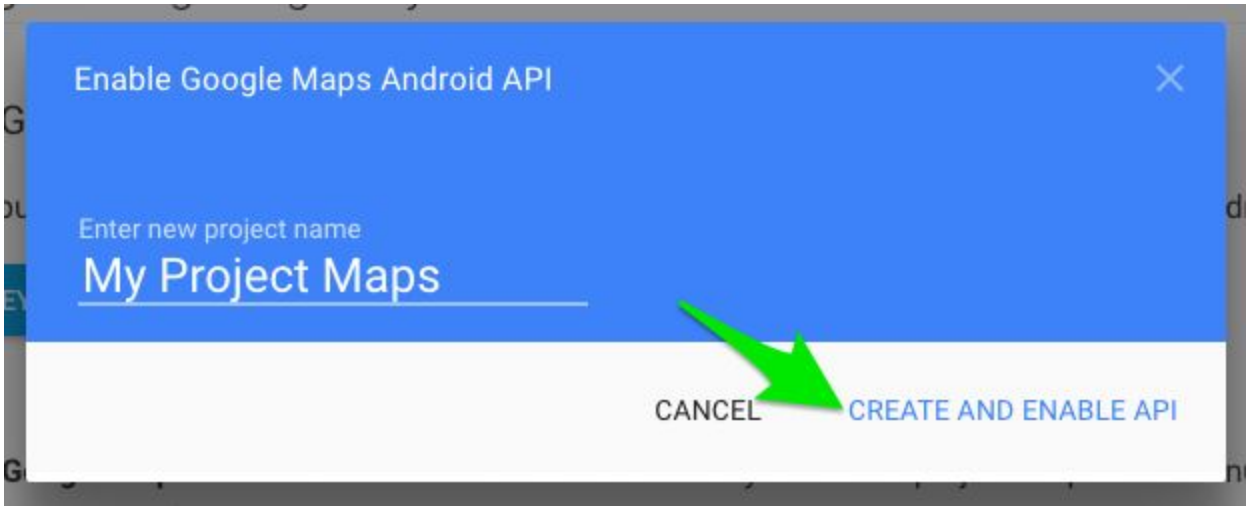
- If you don't already have a Google Console account create one and login.
- Go to https://developers.google.com/maps/documentation/android-api/signup and click **GET A KEY** button.



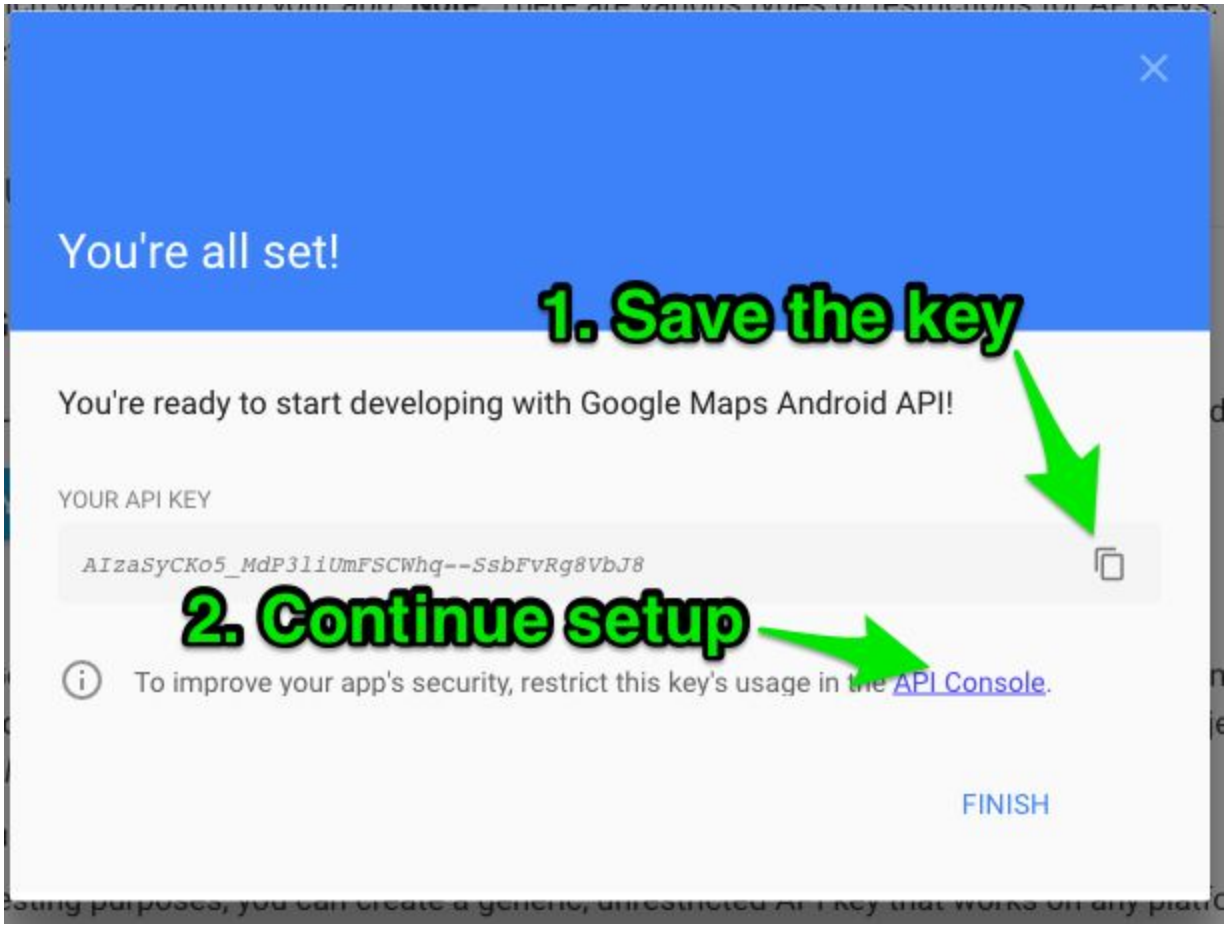
- Click on "Create a new Project" Button



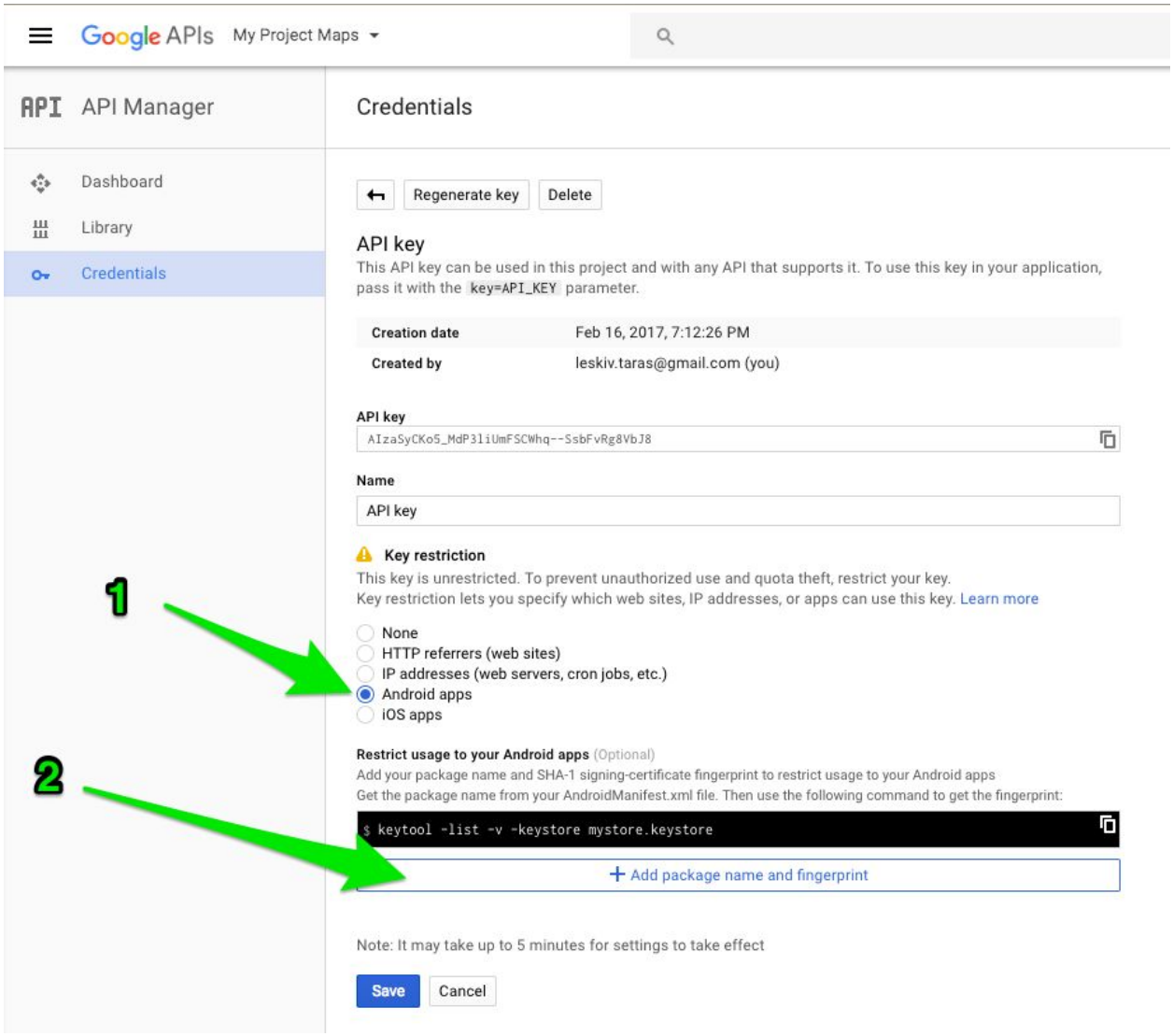
- Click on "CREATE AND ENABLE API" Button



- Now copy your key and save it, you will need it later. After this go to API Console following the link as in image.



- What we need to do now and it is very important that we restrict usage of this key to only our Android application (So other people can't use it if the obtain your key). In Key restriction section choose Android apps and click on + Add package name and fingerprint button.

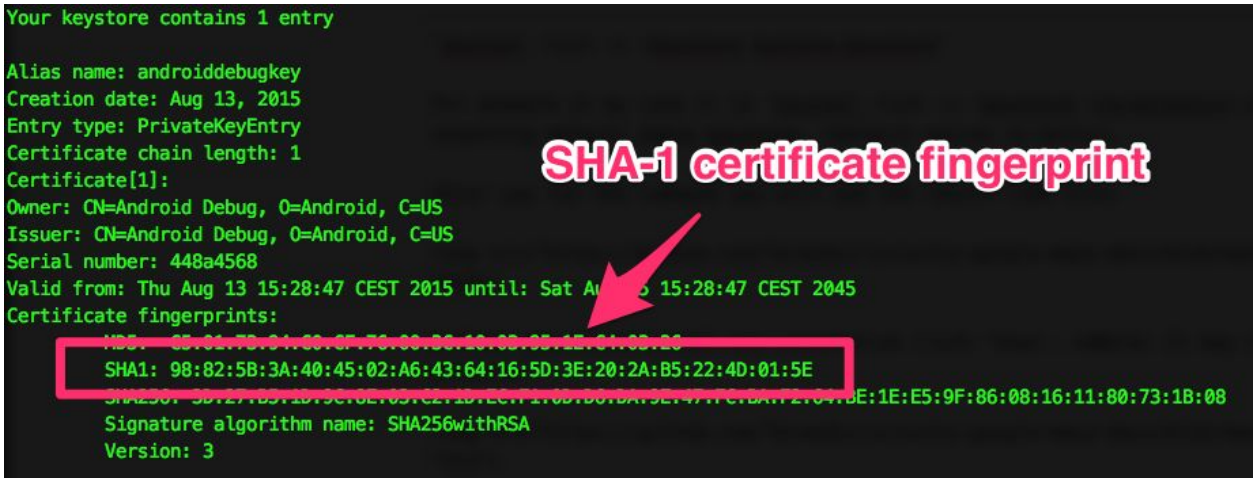


- The form will appear - put Package name from you unity project that you set up in Step 1*.
- To obtain SHA-1 certificate fingerprint run this command in your terminal pointing to your keystore that application is signed with:

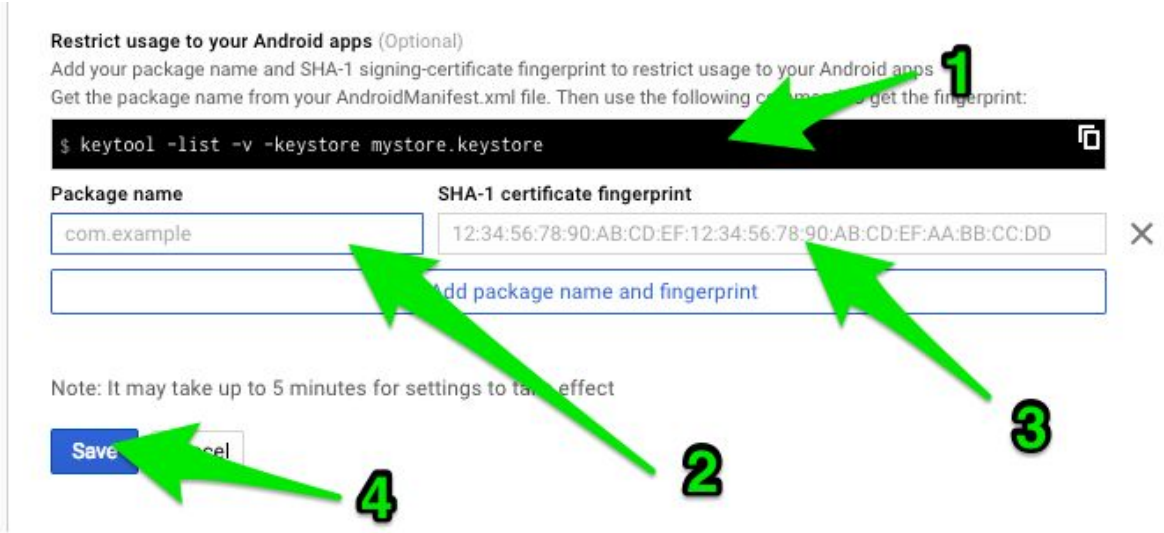
keytool -list -v -keystore mystore.keystore

For example in my case it is keytool -list -v -keystore ~tarasleskiv/.android/debug.keystore as I am using default debug keystore. (Default option in Unity).

After you run the command you will see the result like in the image below, copy SHA-1 certificate fingerprint into the form in Google Console:



- After you filled in all the information click Save. Note: It may take up to 5 minutes for settings to take effect



3. Put the API key inside your AndroidManifest.xml inside Unity Project.

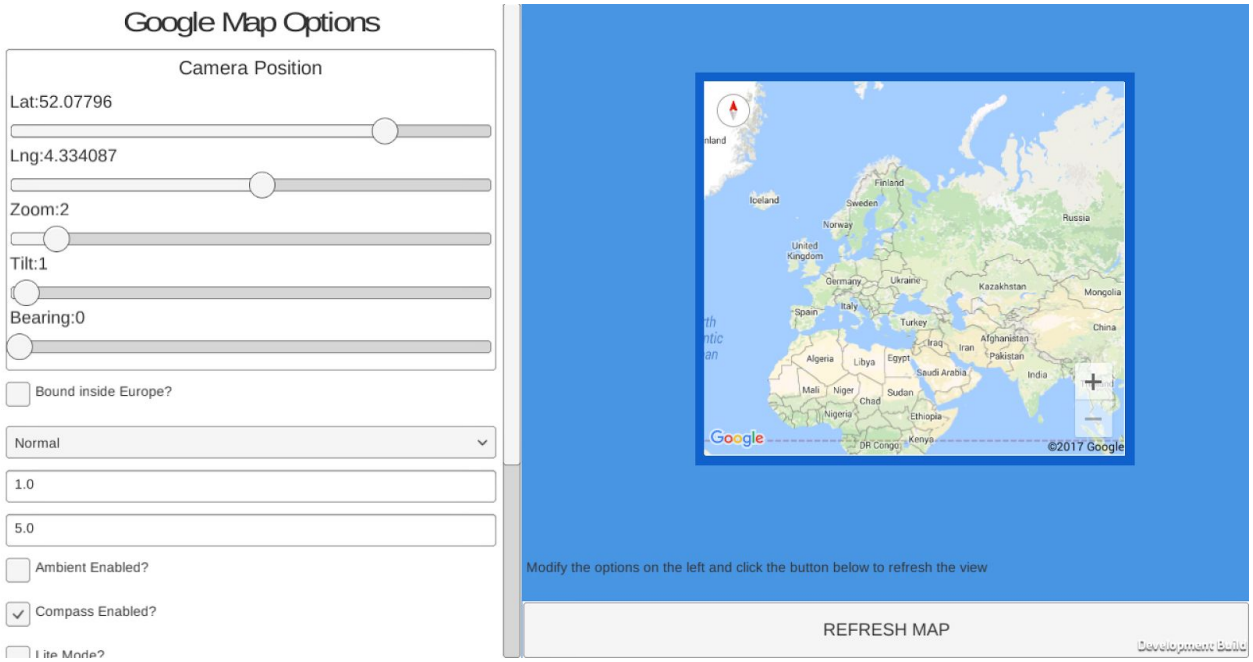
Once you have your key (it starts with "AIza"), put the following meta-data tag INSIDE application tag replacing the value with the API key that you recently retrieved.

<meta-data android:name="com.google.android.geo.API_KEY"
android:value="YOUR_GOOGLE_API_KEY_HERE" />

4. Run the Demo Scene

- Open Unity Build Settings and switch Platform to Android
- Add **GoogleMapsView/Example/ExampleScene.unity** to Scenes in Build
- Connect Android device and run the scene (Device must have Google Play Services installed)

After running the demo you will see the demo scene, now you can play around with settings. Don't forget to click "Refresh" each time you change settings.



iOS Setup

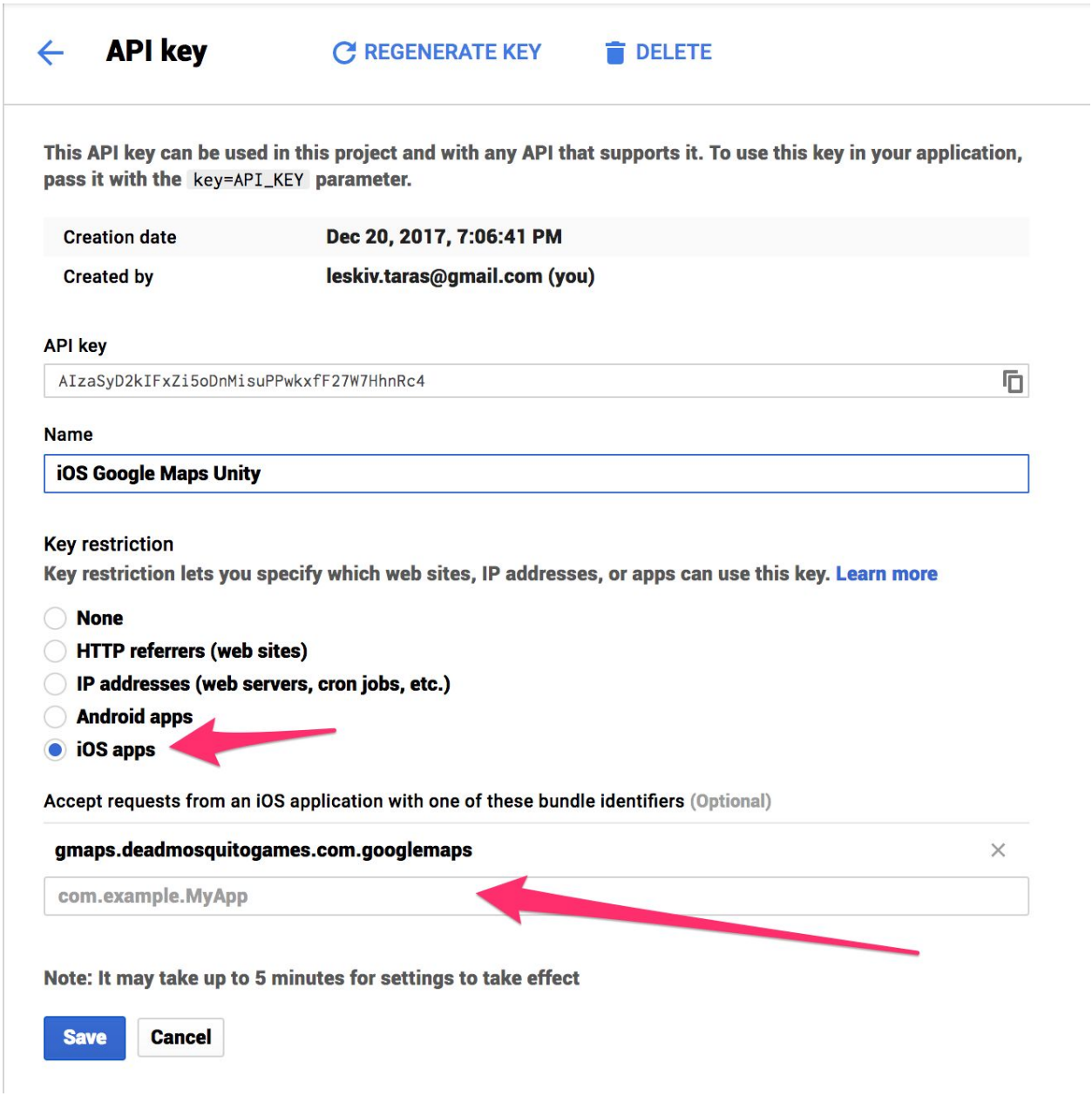
What's different on iOS?

There are a few differences and things to note about iOS implementation:

- Most of the functionality is identical but there are some methods that can be used only on Android, those methods are marked with `[GoogleMapsAndroidOnly]` attribute. This means calling this method will have no effect on ios.
- `GoogleMapsProjectPostprocessor.cs` script is used to add all the required tweaks to XCode project.
- API key is provided by calling `GoogleMapsView.SetIosApiKey(ApiKey)` unlike providing it in the manifest file on Android.

Getting Google Maps API Key from Google Developer Console

Please follow the same steps from Android setup but choose `iOS Apps` when you get to the Credentials screen. Here you can optionally restrict key usage to certain application packages.



Setting API key in the app

Unlike Android where you have to provide the key in the `AndroidManifest.xml` file in iOS you have to call the method BEFORE and calls to the API.

```
GoogleMapsView.SetIosApiKey(YourApiKey);
```

Now you are ready to show the Google Map

Usage Instructions

For reference see the Example folder inside unitypackage. It must provide exhaustive overview of plugin functionality.

Creating, Showing and Dismissing GoogleMapView

To create the google map view simply create an object and pass the created options to the constructor. After this call Show method passing rectangle that represents the position on the screen as a parameter. You can create and show multiple map view instances at the same time.

Minimum code to show the map:

```
var options = new GoogleMapsOptions();
var map = new GoogleMapsView(options);
var screenPosition = new Rect(10, 10, 300, 300);
map.Show(screenPosition);
```

Dismissing the map:

```
map.Dismiss();
```

For more detailed usage instructions visit the Usage Instructions Wiki Page.

FAQs and Known Issues

For FAQ and known issues visit the FAQ and known issues Wiki Page.