

## บทที่ 13 การเขียนโปรแกรมเรียกระบุตำแหน่ง (location) และแผนที่ (map) จากกูเกิล (google)

### สาระบทเรียน

1. ความรู้พื้นฐานเกี่ยวกับ GPS Map
2. การขอ API Key สำหรับ Google Maps
3. แสดงแผนที่บนหน้าจอ



รูปที่ 13-1 บริการแผนที่จากกูเกิล

จากรูปที่ 13-1 แสดงหนึ่งในบริการของกูเกิลที่ได้รับความนิยมเป็นอย่างสูงในปัจจุบันนั่นคือ ระบบแผนที่ที่มีความสามารถที่เกี่ยวข้องกับเส้นทางหรือการเดินทางซึ่งทำงานร่วมกับระบบแผนที่ ในกรณีนี้เราต้องการพัฒนา Android Apps ร่วมกับระบบแผนที่ของกูเกิลเรียกว่า Google Maps Android V2

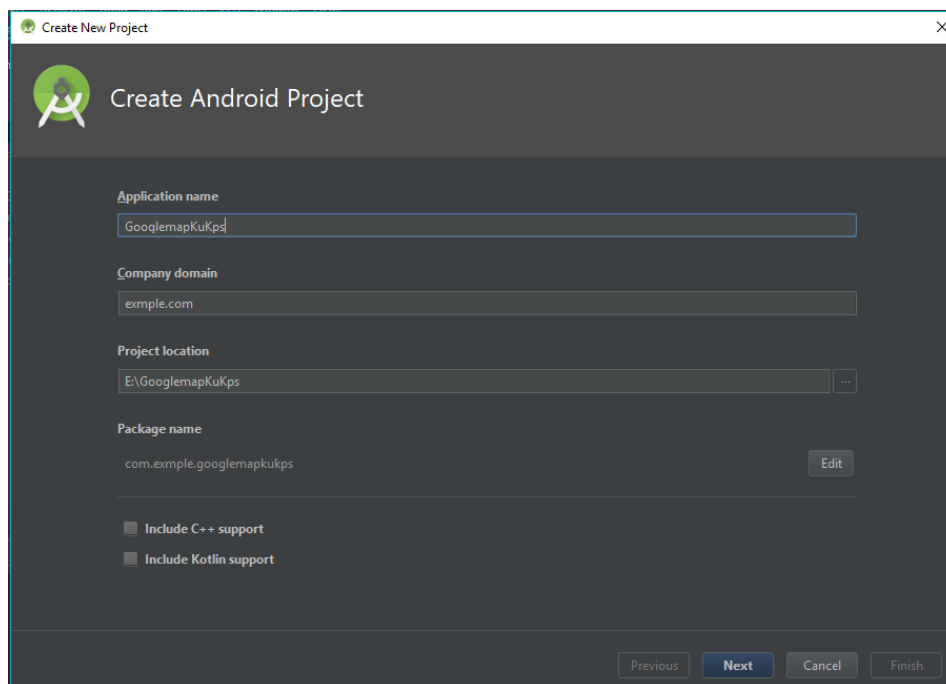
Google Maps คือ บริการแผนที่ของ กูเกิล ซึ่งให้บริการ Services ที่เกี่ยวข้องกับแผนที่ทั้งหมด โดยในปัจจุบัน แผนที่ของกูเกิลนั้นมีอยู่หลากหลายประเภทมาก อาทิเช่นที่เราใช้บริการแผนที่บนเว็บไซต์ หรือ App บนสมาร์ตโฟน (Smartphone) และทางช่องทางอื่นๆ อีกมากมายนับไม่ถ้วน โดย Services เหล่านี้เราสามารถเรียกใช้งานได้ฟรี ในกรณีที่ผ่านแอปฯต่างๆ ไป แต่ถ้าในกรณีที่เราจะมีการเรียกใช้งานในเว็บไซต์หรือแอปฯที่พัฒนาขึ้นเอง Google Maps ก็จะมี API ให้ใช้งานได้เช่นเดียวกัน แต่เซอร์วิส (Services) ต่างๆของกูเกิลนั้นมีข้อจำกัดในการใช้งาน แต่ถ้าหากต้องการใช้ในปริมาณที่สูงขึ้น ก็จะต้องซื้อแพ็คเกจ (Package) ที่ทาง Google Maps มีมาให้ ซึ่งโดยปกติจะมีการจำกัดจำนวนครั้งที่เรียกใช้งาน

## ความรู้พื้นฐานเกี่ยวกับ GPS Map

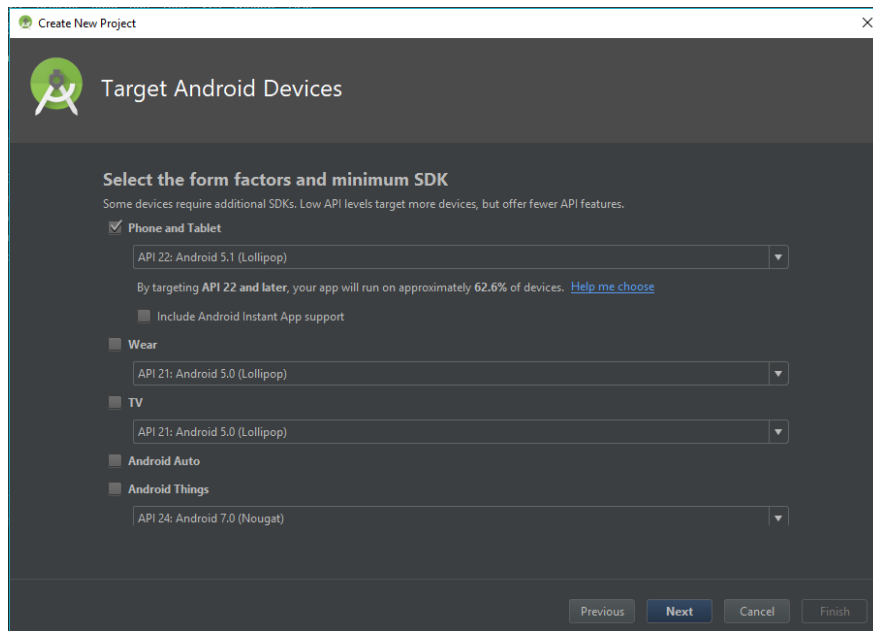
ระบบกำหนดตำแหน่งบนโลกหรือ GPS (Global Positioning System) เป็นระบบระบุตำแหน่งบนพื้นโลกโดยคำนวณจากค่าสัญญาณนาฬิกาที่ส่งมาจากดาวเทียมที่โคจรรอบโลก ทำให้ทราบตำแหน่งที่แน่นอนจึงสามารถระบุตำแหน่งต่างๆ ของอุปกรณ์รับสัญญาณต่างๆ ได้อย่างถูกต้องดาวเทียม GPS เป็นดาวเทียมที่ใช้บนวงโคจรระดับกลาง หรือ MEO (Medium Earth Orbit) ซึ่งโดยทั่วไปนั้นจะมีวงโคจรที่ระดับความสูงประมาณ 20,000 กิโลเมตรซึ่งการยืนยันตำแหน่งต่างๆ จะอาศัยค่าพิกัดจากดาวเทียมอย่างน้อย 4 ดวง โดยปกติแล้วดาวเทียมแต่ละดวงจะใช้เวลาโคจรรอบโลกเป็นเวลา 12 ชั่วโมงต่อหนึ่งรอบ โดยมีความเร็วประมาณ 4 กิโลเมตร/วินาที(km/s)และการโคจรในแต่ละรอบนั้นจะแบ่งออกเป็น 6 ระนาบ ซึ่งแต่ละระนาบจะมีดาวเทียมจำนวน 4 ดวงโดยทำมุมที่ 55 องศา (Degree) ดังนั้น โดยรวมแล้วทั้งระบบจะต้องมีดาวเทียมอย่างน้อยจำนวน 24 ดวงทั้งนี้โดยทั่วไปแล้วอุปกรณ์รับสัญญาณในปัจจุบันจะผนวกเข้ากับระบบแผนที่เพื่อใช้ระบุรายละเอียดของสถานที่ต่างๆ รวมไปถึงสามารถนำมาประยุกต์ใช้ในการนำทางได้อีกด้วย

## ตัวอย่างการสร้างโปรเจกต์เกี่ยวกับการระบุตำแหน่ง

1. สร้าง Project โดยมี Application name = GooglemapKuKps

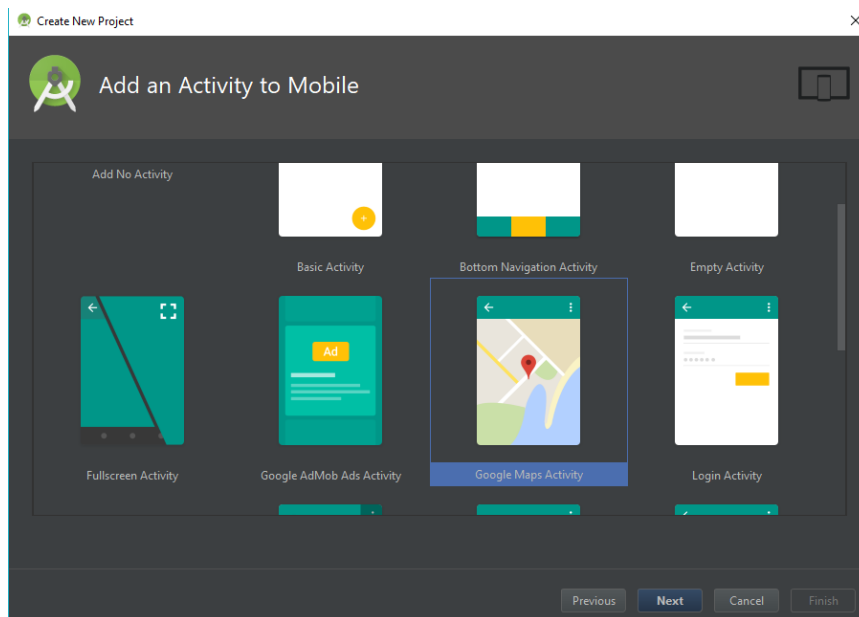


ให้ตั้งชื่อเสร็จแล้ว กด Next

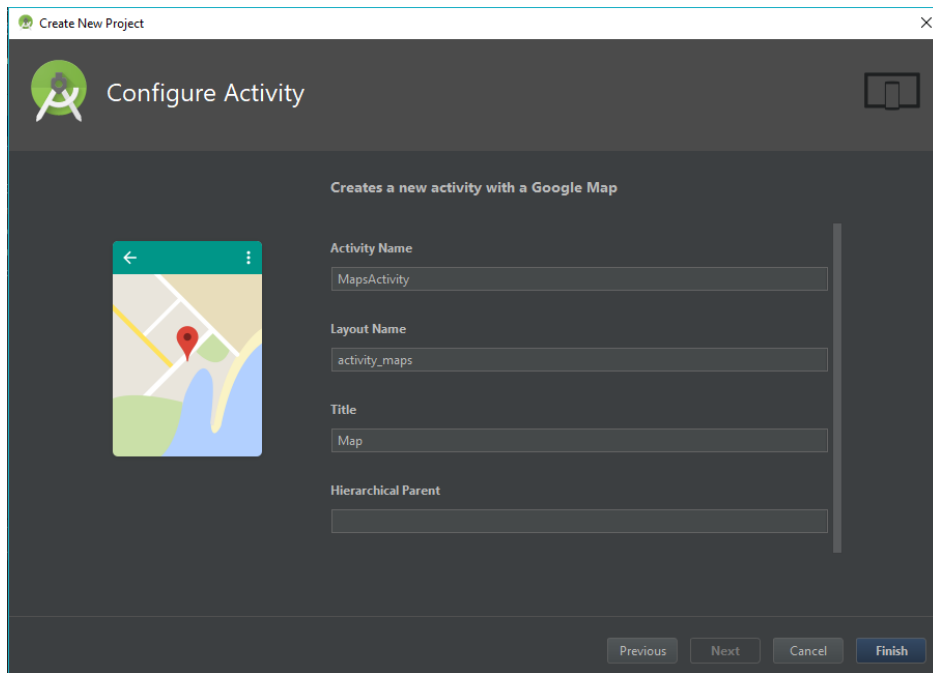


กด Next

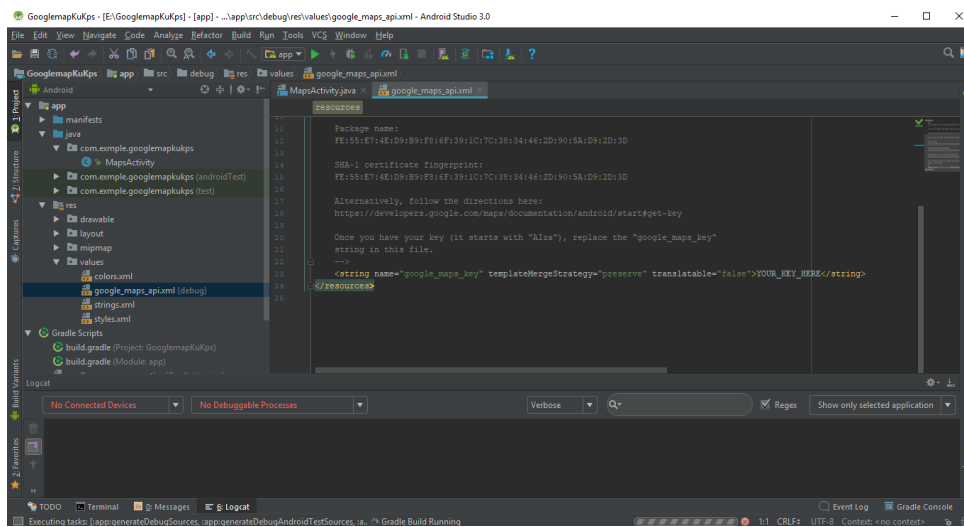
- สร้างแอกติวิตี้ที่ควรเลือกสร้างเป็น Google Maps Activity เพราะเราจะใช้ Google Maps



เลือกแล้วกด Next หน้าสุดท้ายก็จะเป็นการตั้งชื่อแอกติวิตี้เสร็จแล้วกด Finish ต่อมาจะเป็นหน้าแรกของโปรเจกต์



เมื่อเสร็จแล้วก็แสดงตามภาพดังนี้



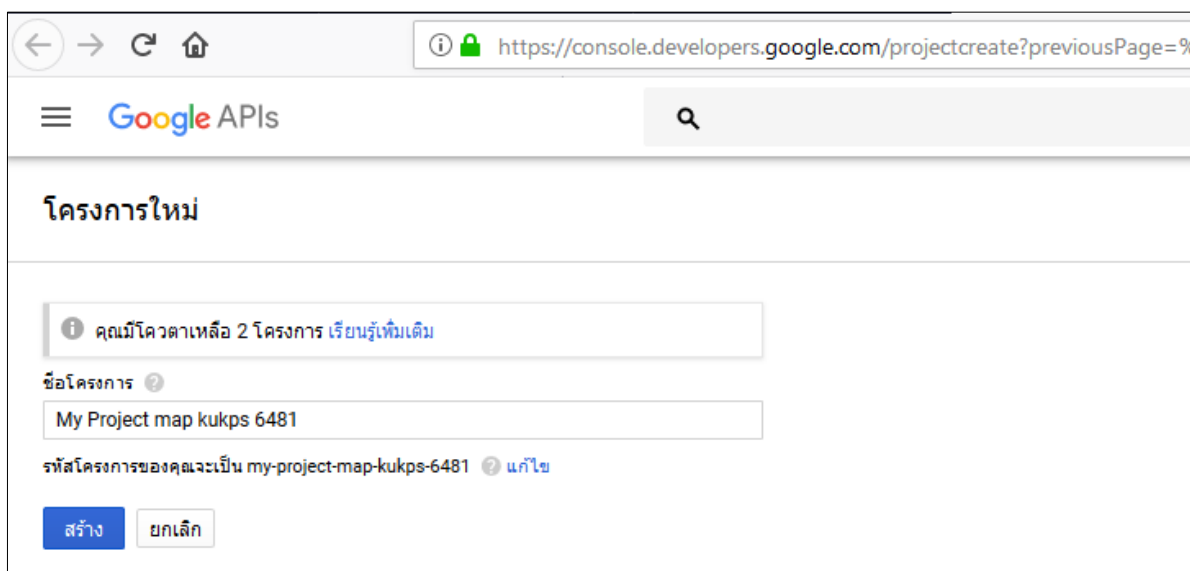
ในภาพจะมีค่าของ YOUR\_KEY\_HERE ซึ่งเป็น URL ที่ต้อง copy ไปใส่ใน Browser เพื่อที่จะไปสร้าง Key—API

## การขอ API Key สำหรับ Google Maps

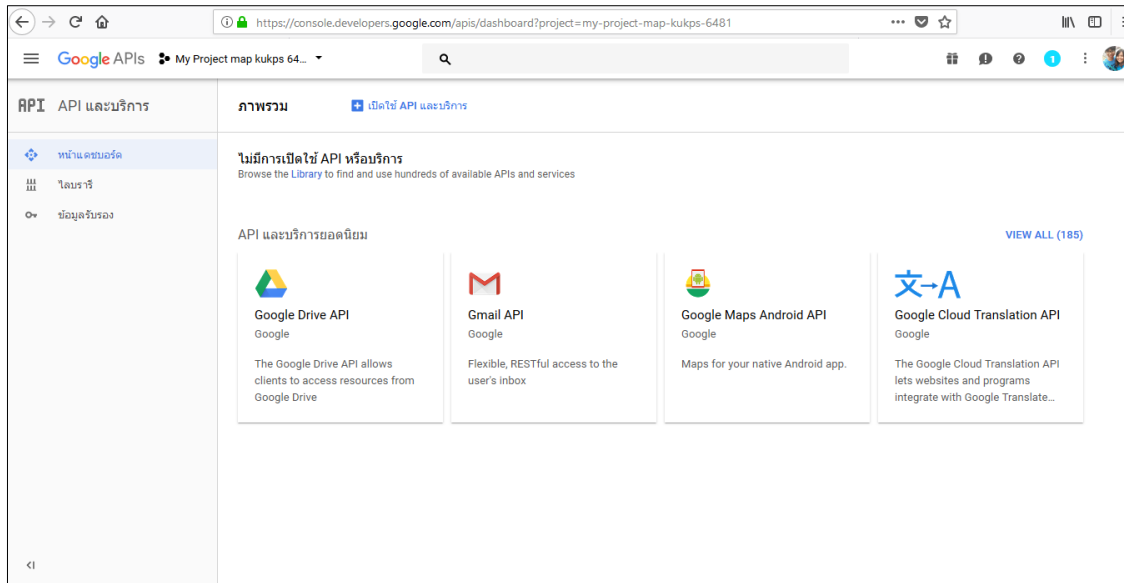
การขอ API Key นั้นอาจจะมีขั้นตอนยุ่งยากเล็กน้อย แต่ก็ต้องทำเพราะว่า API Key เป็นสิ่งที่ทางกูเกิลใช้ระบุว่าแอปพของนักพัฒนาคนไหนมีการเรียกใช้งาน Google Maps บ้าง ถึงแม้ว่า Google Maps จะให้ใช้งานฟรีสำหรับพีเจอรืเบื้องต้น (พีเจอรืพิเศษจะต้องเสียเงิน)

โดย API Key นั้นจะสัมพันธ์กับชุดการเข้ารหัสแบบ SHA1 ใน Keystore ที่ใช้ Build แอปพตัวนั้นๆ และสัมพันธ์กับแพคเกจ (Package) ของแอปพที่เรียกใช้งาน ดังนั้น API Key นี้จะผูกกับแอปพตัวเดียวเท่านั้น ไม่สามารถนำไปใช้กับแอปพตัวอื่นได้ (ทำหลายแอปพก็ต้องขอทุกตัว)

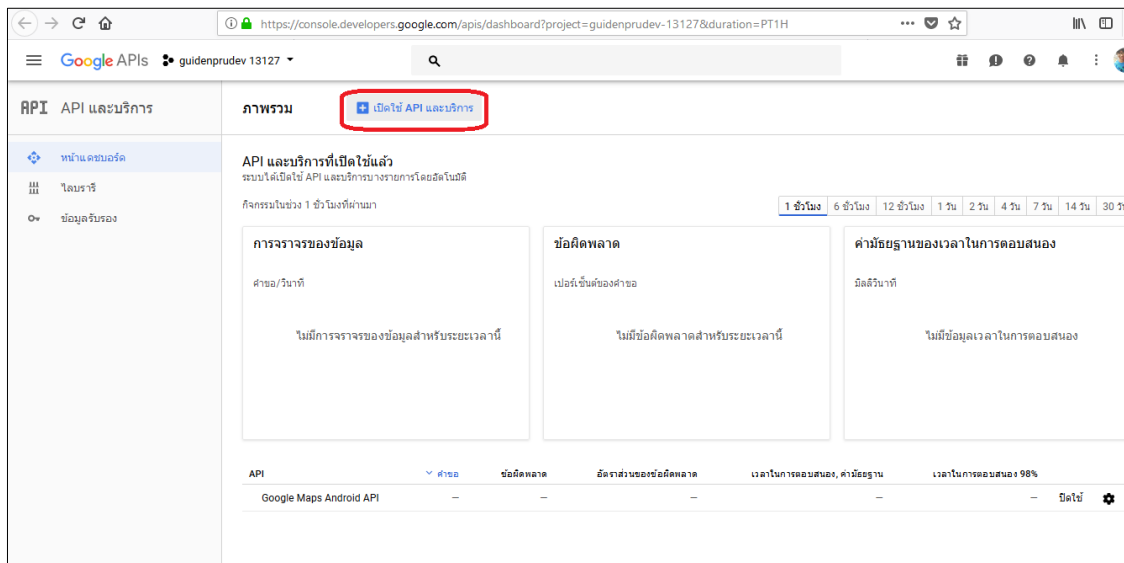
ต่อมาให้เข้าไปที่ <https://console.developers.google.com/> และสร้างโปรเจกต์(หากยังไม่มี) ถ้ายังไม่มีบัญชีของกูเกิลให้สมัครและ Login เสียก่อน จากนั้นตั้งชื่อโปรเจกต์ตามที่ต้องการ (องค์กรจะเลือกหรือไม่เลือกก็ได้)แล้วให้พิมพ์ชื่อโปรเจกต์ตามภาพ



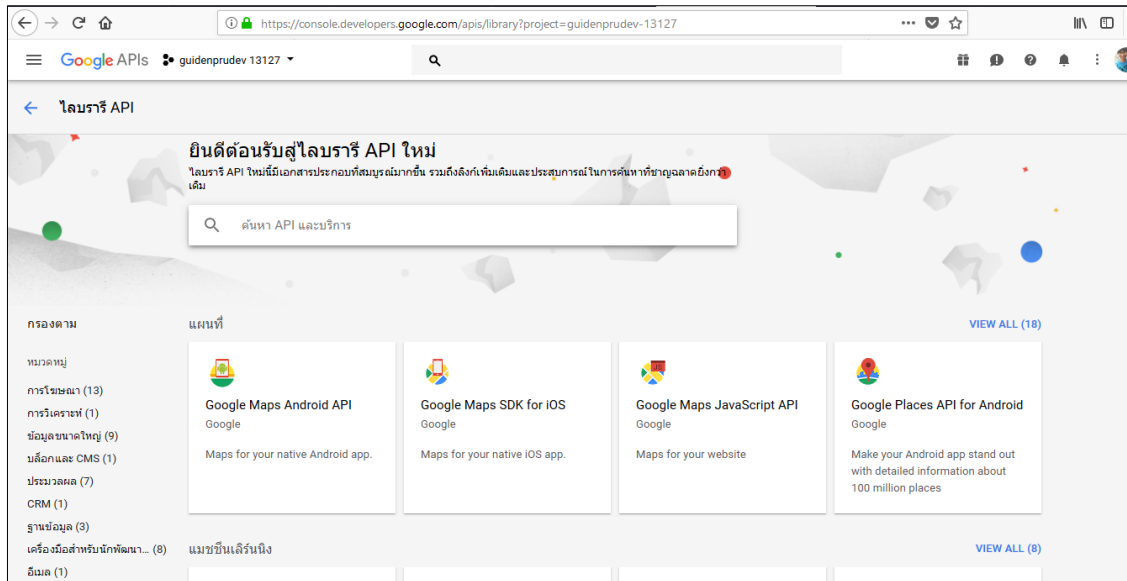
ต่อมาจะแสดงส่วนหน้าหลักของ Google APIs



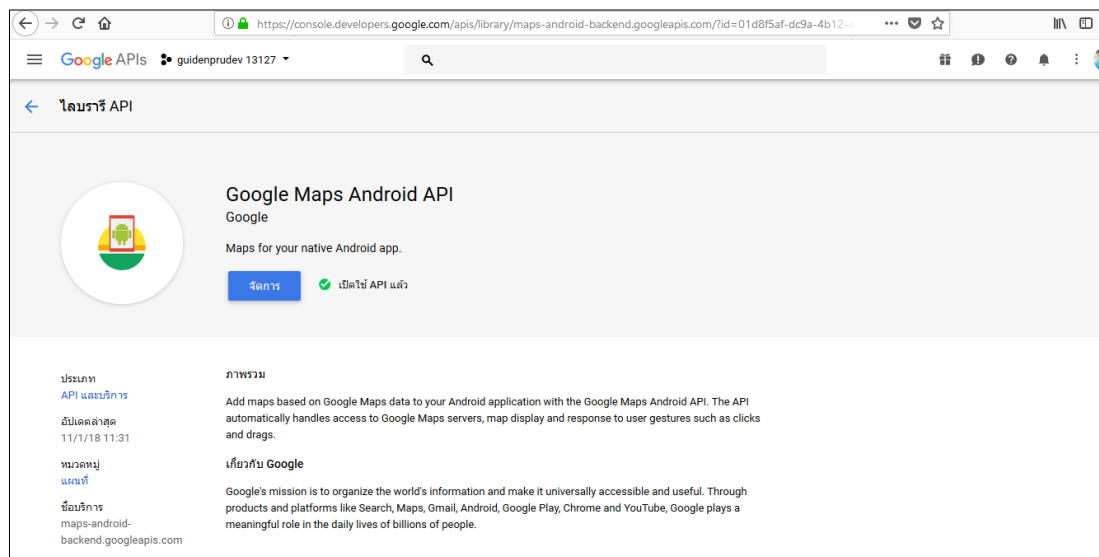
กด “เปิดใช้” หลังจากนั้นจะมีปุ่มด้านขวาให้สร้าง “ข้อมูลรับรอง” กดสร้างเพื่อให้ได้ข้อมูลคีย์ จากนั้นให้คัดลอก (copy) คีย์ API นี้ไว้แล้วกดเสร็จสิ้น



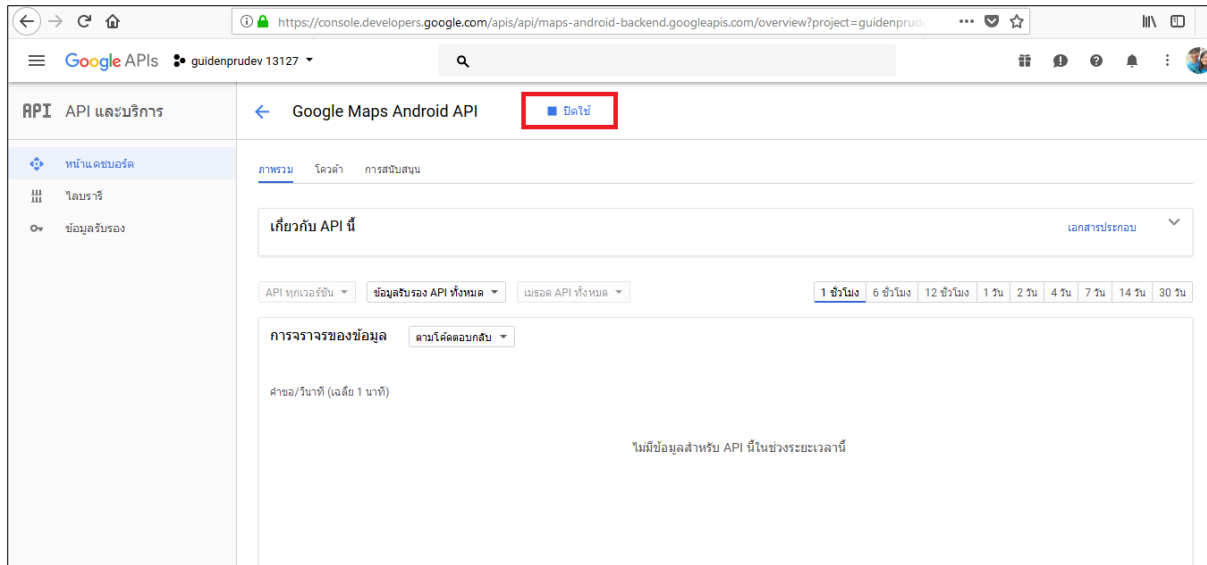
แล้วจะได้หน้าจอภาพที่ขึ้นมาว่า ยินดีต้อนรับสู่โลกบราว์ใหม่ พร้อมกับให้ดูที่แผนที่ แล้วให้สังเกตคำว่า Google Maps Android API



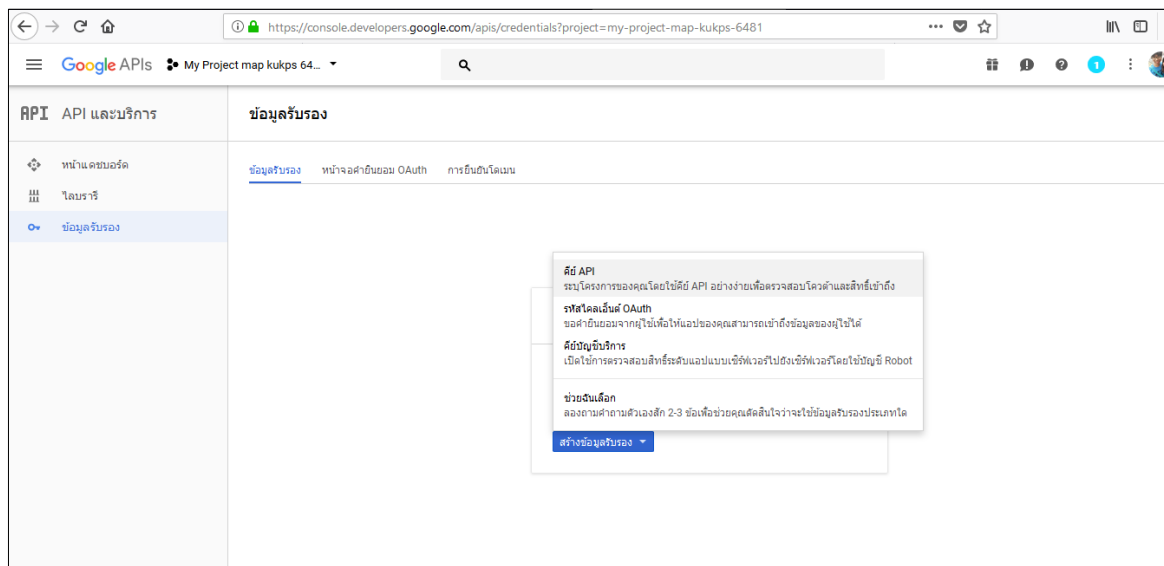
เมื่อกดแล้วก็จะสู่หน้า Google Maps Android API พร้อมกับแสดงข้อมูลรายละเอียดต่างๆ จากนั้นให้เลือกคำว่า “จัดการ”



จากหน้าจอที่ได้จะสังเกตเห็นว่าการเรียกเปิดใช้ส่วนที่เป็น googleMap Android API เรียบร้อยแล้ว



ลำดับต่อมา เลือกปุ่มคำว่า “ข้อมูลรับรอง” แล้วเลือกไปที่ สร้างข้อมูลรับรอง พร้อมเลื่อนคำว่า คีย์ API



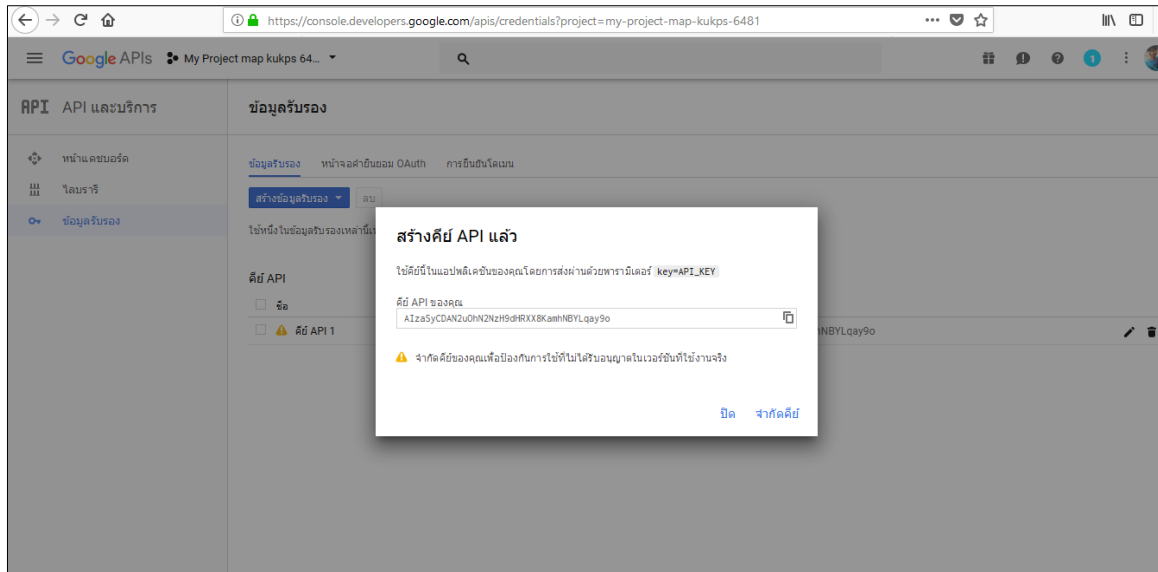
ถัดไปจะเป็นส่วนที่บอกว่าสร้างคีย์ API แล้ว ใช้คีย์นี้ในแอปพลิเคชันของคุณโดยการส่งผ่านด้วย

พารามิเตอร์ key=API\_KEY โดยในที่นี้ key เป็น

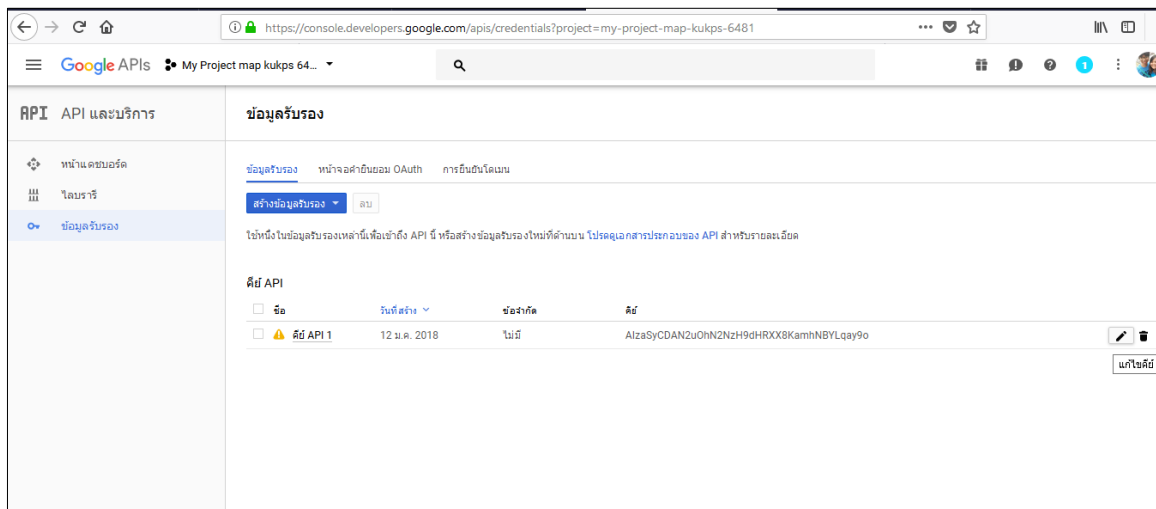
IzaSyCDAN2uOhN2NzH9dHRXX8KamhNBYLqay9o (ทุก key จะไม่ซ้ำกัน) ทำการคัดลอกแล้วกด

ปิด

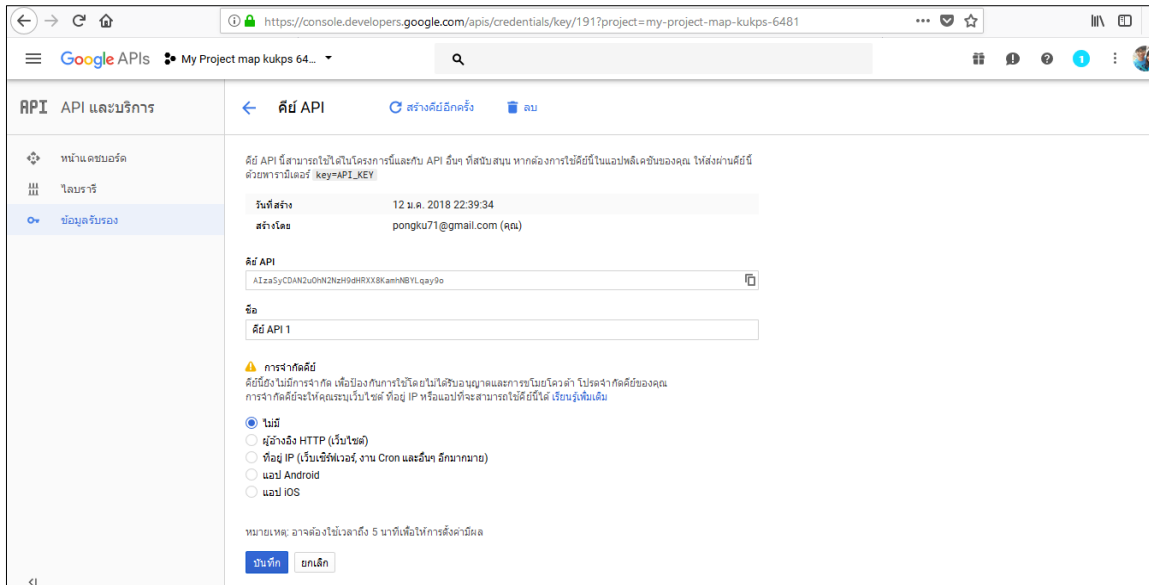




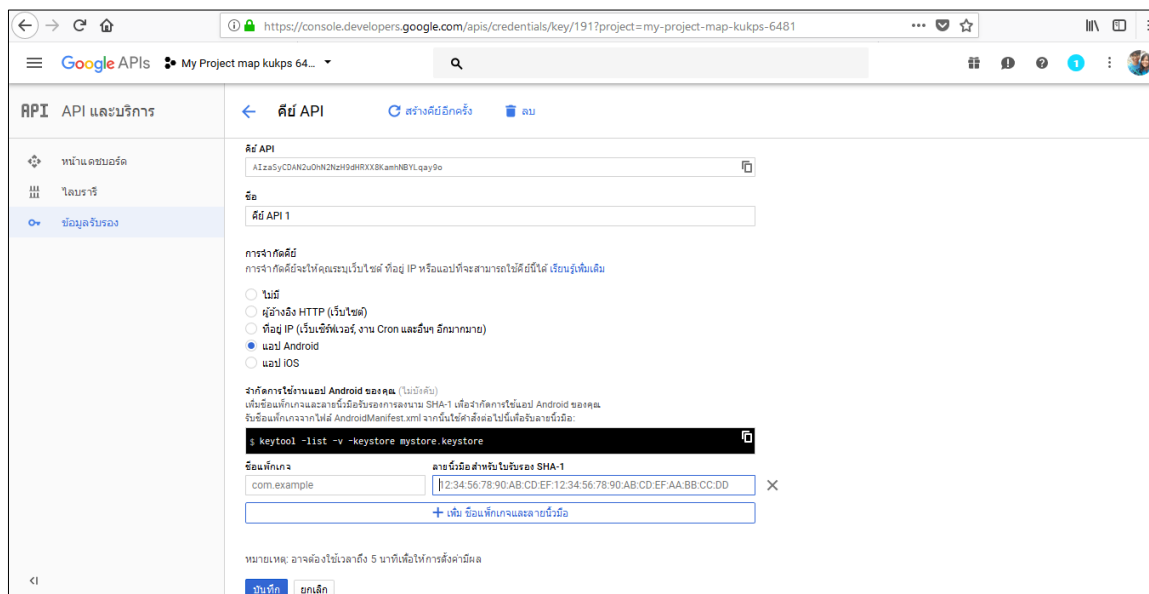
ก็จะพบได้ว่าการสร้าง API แล้ว แต่ยังไม่สามารถเรียกใช้ key ในแอนดรอยด์ได้ ซึ่งต้องเลือกที่ส่วนแก้ไขตามภาพ



ก็จะพบเข้าสู่หน้าแก้ไข โดยคีย์ API นี้สามารถใช้ได้ในโครงการนี้และกับ API อื่นๆ ที่สนับสนุน หากต้องการใช้คีย์นี้ในแอปของคุณ ให้ส่งผ่านคีย์นี้ด้วยพารามิเตอร์ key=API\_KEY



โดยส่วนการจำกัดสิทธิ์จะให้คุณระบุเว็บไซต์ ที่อยู่ IP หรือแอปที่จะสามารถใช้คีย์นี้ได้ หมายความว่า การเรียกใช้ API Key นั้นจะสัมพันธ์กับชุดรหัส SHA1 ใน Keystore ในที่นี้ให้เลือก ส่วนที่เป็นแอปแอนดรอยด์ พร้อมกับเลือกคำว่า + เพิ่ม ชื่อแพ็คเกจและลายมือ



## ชุดรหัส SHA1 จาก Keystore

ชุดรหัส SHA1 จาก Keystore ที่ใช้เพื่อขอ API Key นั้นจะมีอยู่สองแบบด้วยกันคือ SHA1 ของ Debug Keystore และ Signed Keystore โดยมีเงื่อนไขอยู่ว่า

- ใช้ SHA1 จาก Debug Keystore ในขณะที่กำลังพัฒนาแอปอยู่และยังไม่ได้ส่งแอป ขึ้น Google Play Store หรือ 3rd Party App Store แบบสาธารณะ (Public)

- ใช้ SHA1 จาก Signed Keystore เมื่อพัฒนาแอปฯเสร็จแล้ว และพร้อมจะส่งแอปฯขึ้น Google Play Store หรือ 3rd Party App Store แบบสาธารณะ (Public)

เนื่องจาก เพราะไม่รู้ว่า API Key ต้องใช้คนละตัวกันเวลาทดสอบแอปฯกับตอนส่งแอปฯขึ้น Google Play Store หรือ 3rd Party App Store แบบสาธารณะ (Public)

### ตำแหน่งการเก็บ Debug Keystore กับ Signed Keystore

Debug Keystore จะอยู่ที่

OS X หรือ Linux

~/android/debug.keystore

Windows

C:\Users\\.android\debug.keystore

ส่วน Signed Keystore ไม่กำหนดที่อยู่ตายตัว เพราะมันคือไฟล์ Keystore ที่จะสร้างเพื่อใช้ Export แอปพลิเคชันเป็น APK ได้

### การอ่านชุดรหัส SHA1 จาก Keystore

จะต้องทำผ่าน Terminal (OS X หรือ Linux) หรือ Command Prompt (Windows) ดังนี้

```
keytool -list -v -keystore <keystore_file_path> -alias <alias_name> -storepass <alias_pass> -
keypass <keystore_pass>
```

- keystore\_file\_path คือที่อยู่ของไฟล์ Keystore ที่ต้องการอ่าน SHA1
- alias\_name Alias Name ของ Keystore นั้นๆ
- alias\_pass รหัสผ่านสำหรับ Alias Name
- keystore\_pass รหัสผ่านสำหรับ Keystore

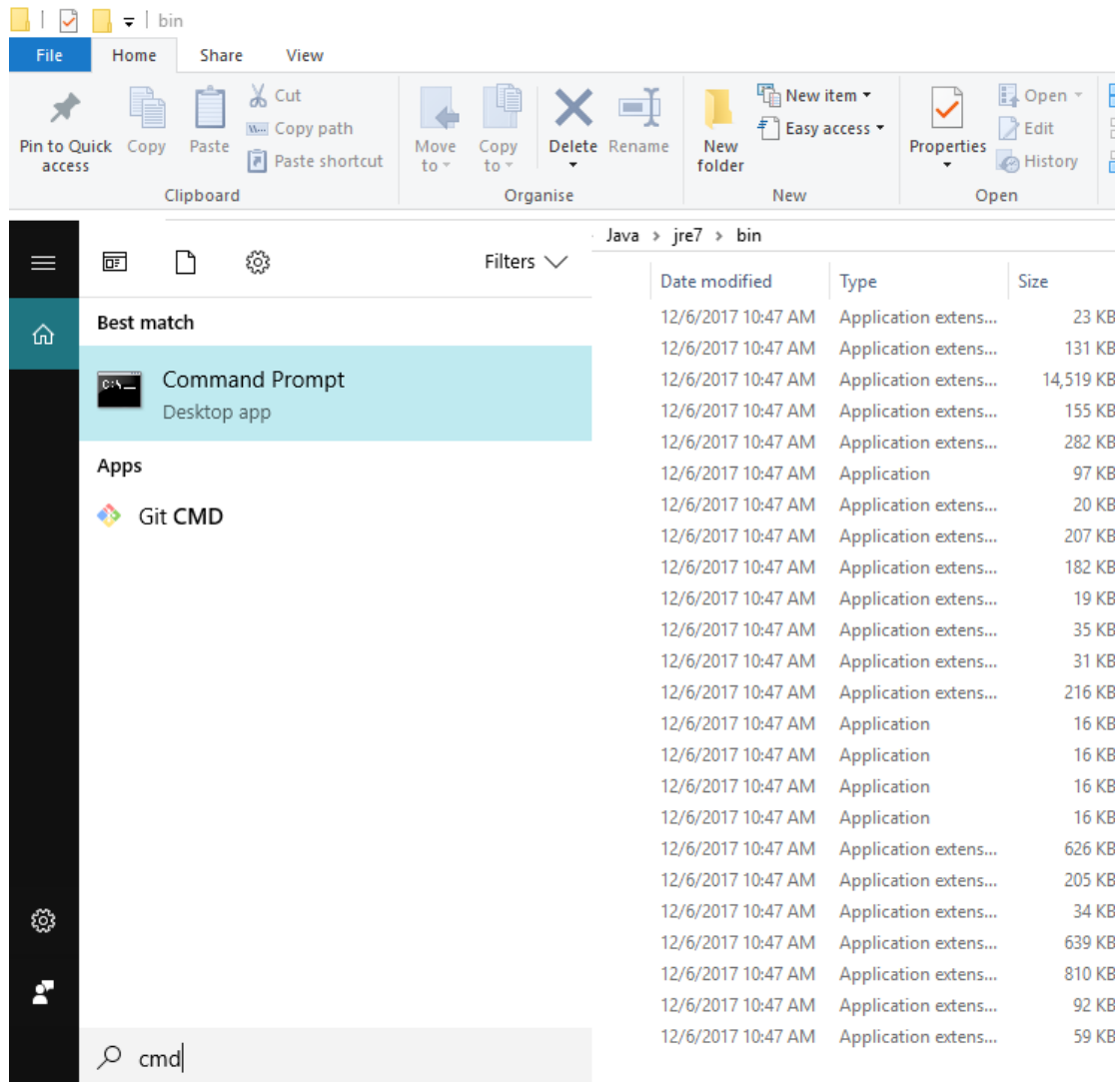
ถ้าเป็น Debug Keystore ทาง กูเกิล ก็ได้กำหนดตายตัวไว้ตั้งแต่แรกแล้ว

- Alias Name คือ androiddebugkey
- Alias Password คือ android
- Keystore Password คือ android

ยกตัวอย่างเช่น ต้องการ SHA1 ของ Debug Keystore ก็จะต้องใช้คำสั่งดังนี้

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

โดยให้เข้าไปที่ start พร้อมค้นหาคำว่า cmd



และพิมพ์คำสั่งตามนี้

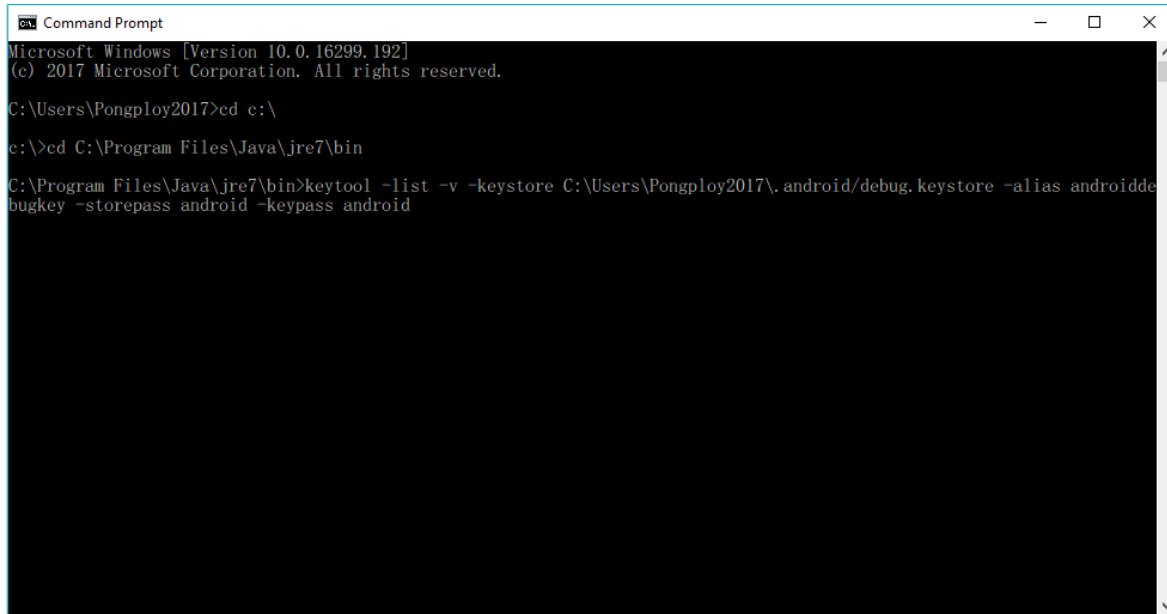
```
cd c:\ กด enter ลงมา
```

```
c:\>cd C:\Program Files\Java\jre7\bin กด enter ลงมา
```

```
แล้วพิมพ์ใส่ >keytool -list -v -keystore C:\Users\[ชื่อของเครื่อง]\.android/debug.keystore -alias
androiddebugkey -storepass android -keypass android
```

เช่น >keytool -list -v -keystore C:\Users\Pongploy2017\.android/debug.keystore -alias androiddebugkey -storepass android -keypass android

ตามภาพ

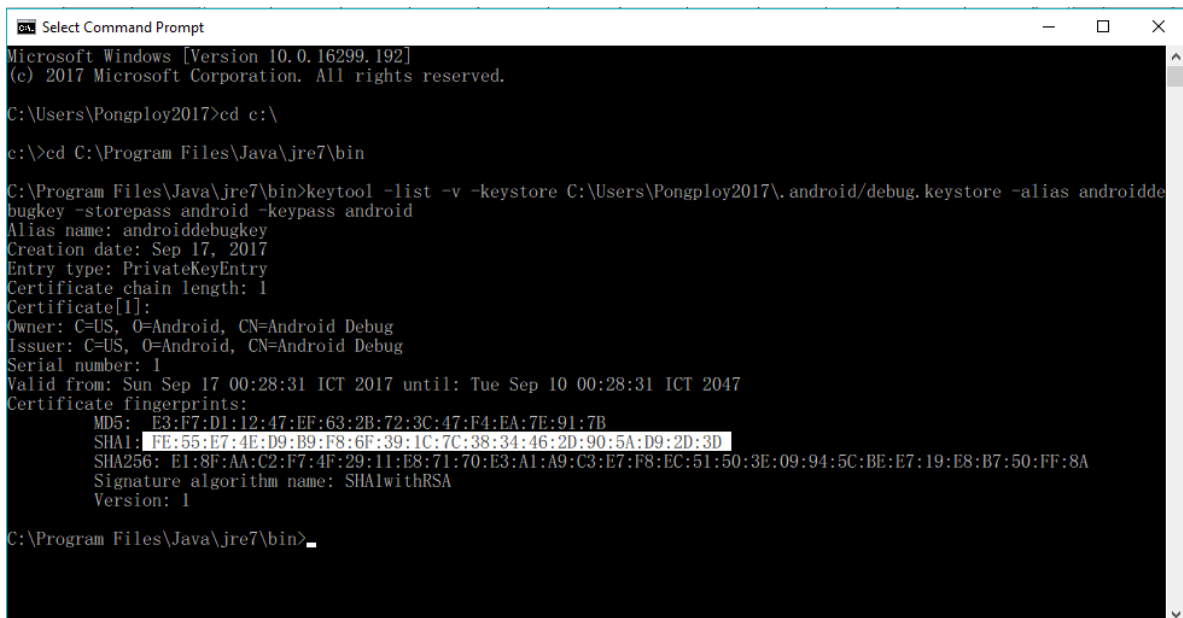


```

Command Prompt
Microsoft Windows [Version 10.0.16299.192]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Pongploy2017>cd c:\
c:\>cd C:\Program Files\Java\jre7\bin
C:\Program Files\Java\jre7\bin>keytool -list -v -keystore C:\Users\Pongploy2017\.android/debug.keystore -alias androiddebugkey -storepass android -keypass android
  
```

ก็จะได้เลข SHA1 ที่ต้องการ จะเห็นตรงแถว SHA1 มีชุดรหัสแสดงอยู่ 20 ชุดด้วยกัน ให้เอา SHA1 ที่ได้จากตรงนั้นแหละไปขอ API Key จาก Google Developer Console สามารถ copy เอาไปในเว็บ google map api ที่ได้ตั้งไว้



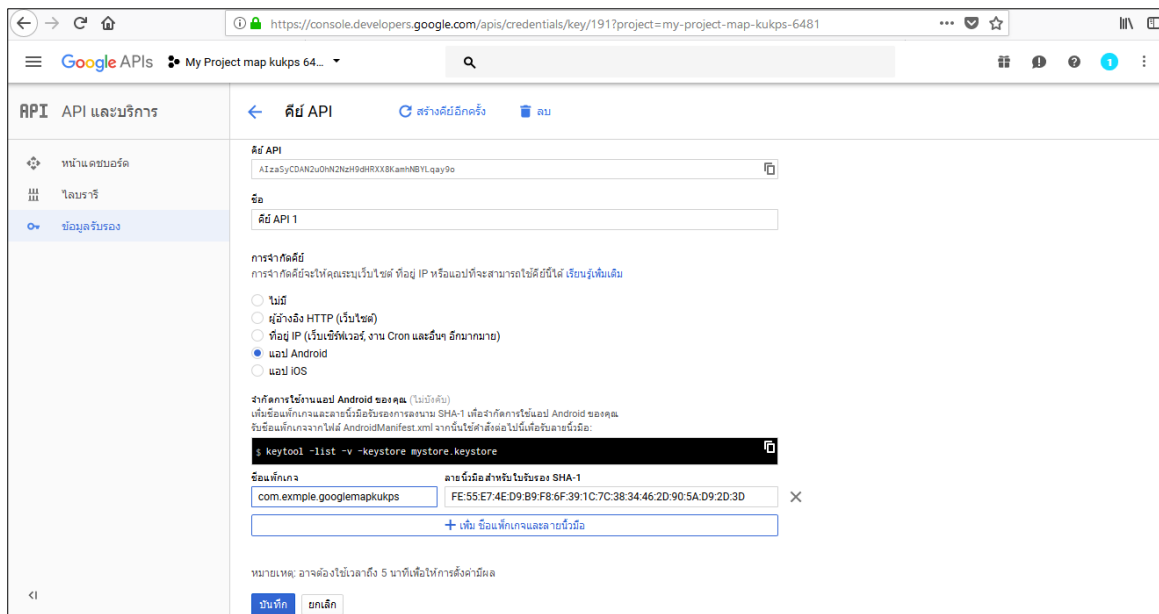
```

Select Command Prompt
Microsoft Windows [Version 10.0.16299.192]
(c) 2017 Microsoft Corporation. All rights reserved.

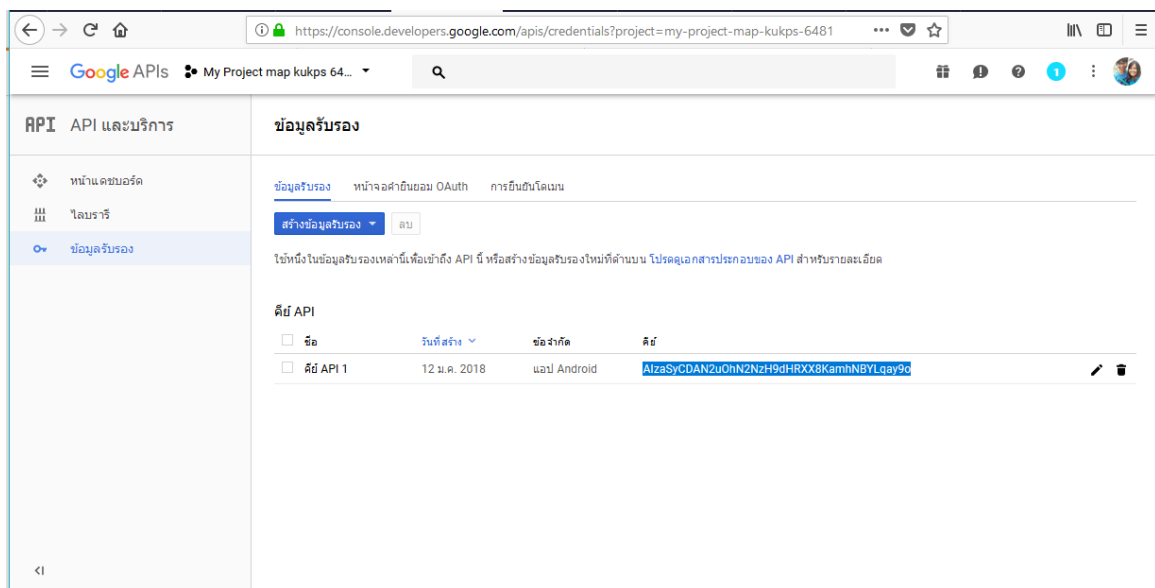
C:\Users\Pongploy2017>cd c:\
c:\>cd C:\Program Files\Java\jre7\bin
C:\Program Files\Java\jre7\bin>keytool -list -v -keystore C:\Users\Pongploy2017\.android/debug.keystore -alias androiddebugkey -storepass android -keypass android
Alias name: androiddebugkey
Creation date: Sep 17, 2017
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: C=US, O=Android, CN=Android Debug
Issuer: C=US, O=Android, CN=Android Debug
Serial number: 1
Valid from: Sun Sep 17 00:28:31 ICT 2017 until: Tue Sep 10 00:28:31 ICT 2047
Certificate fingerprints:
  MD5:   E3:F7:D1:12:47:EF:63:2B:72:3C:47:F4:EA:7E:91:7B
  SHA1:  FE:55:E7:4E:D9:B9:F8:6F:39:1C:7C:38:34:46:2D:90:5A:D9:2D:3D
  SHA256: E1:8F:AA:C2:F7:4F:29:11:E8:71:70:E3:A1:A9:C3:E7:F8:EC:51:50:3E:09:94:5C:BE:E7:19:E8:B7:50:FF:8A
Signature algorithm name: SHA1withRSA
Version: 1

C:\Program Files\Java\jre7\bin>
  
```

และเมื่อได้เลข SHA-1 ให้วาง (paste) ค่าลายนิ้วมือสำหรับใบรับรอง SHA-1 พร้อมใส่ชื่อแพ็คเกจตามภาพ



แล้วกดบันทึก คราวนี้ก็จะสามารถใช้ Google Map API เต็มรูปแบบแล้ว สามารถดึงแผนที่ได้ตามภาพ



และให้ใส่คำว่า AlzaSyCDAN2uOhN2NzH9dHRXX8KamhNBYLqay9o ที่ได้มาแทน

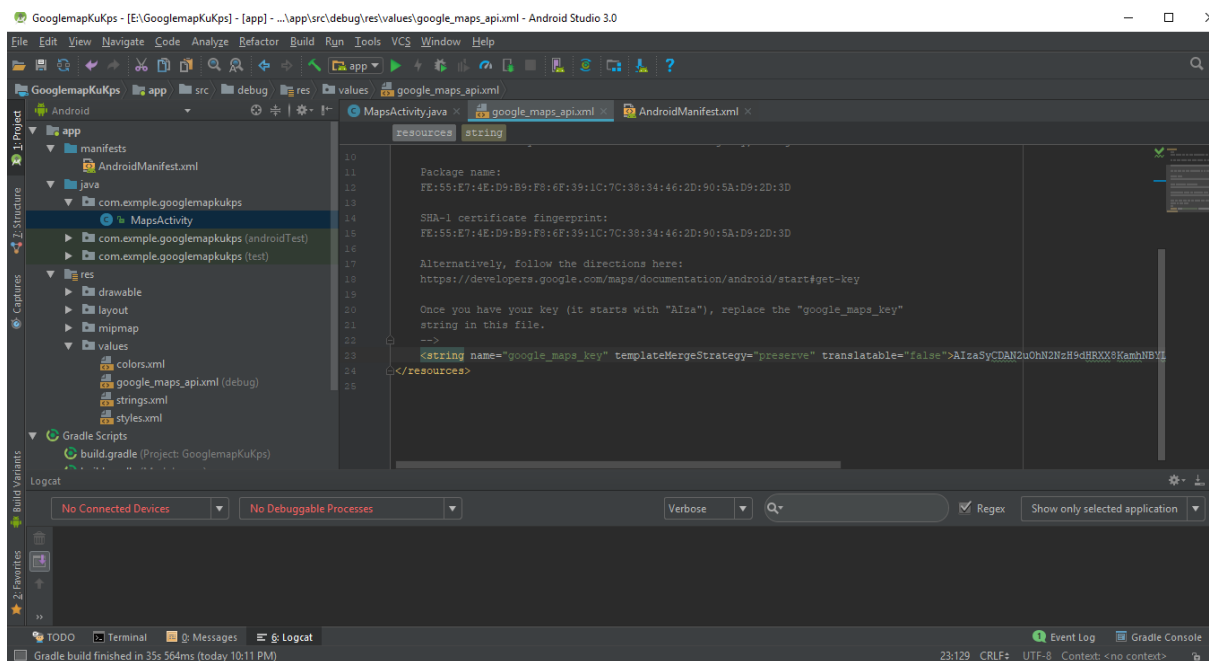
YOUR\_KEY\_HERE พร้อมรัน

เมื่อสร้างแล้วแอนดรอยด์สตูดิโอจะทำการสร้างไฟล์ชื่อว่า google\_map\_api.xml ให้เราอัตโนมัติ ให้นำคีย์ API ที่เรา copy ไว้ในขั้นตอนก่อนหน้านี้มาใส่ในไฟล์นี้

```
<string name="google_maps_key" templateMergeStrategy="preserve" translatable="false">
  YOUR_KEY_HERE
</string>
```

ถ้าหากเราเข้าไปดูที่ไฟล์ build.gradle (Module: app) จะเห็นได้ว่าแอนดรอยด์สตูดิโอจะเพิ่ม dependency ของ google play service ที่มี Google Map API ใส่ไว้ให้เราเรียบร้อยแล้วโดยไม่ต้องเพิ่มเอง

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:25.1.0'
    compile 'com.google.android.gms:play-services:10.0.1'
    testCompile 'junit:junit:4.12'
}
```



## ตัวอย่างการเขียนโปรแกรมโดยใช้งาน Google Maps API

### 1. การแสดงแผนที่บนหน้าจอ (Hello Map)

เปิดไฟล์ MapsActivity.java และ activity\_maps.xml จะเห็นว่าแอนดรอยด์สตูดิโอได้ใส่คำสั่งสำหรับแสดงแผนที่ไว้ให้เราเรียบร้อยแล้วตามภาพ

```
public class MapsActivity extends FragmentActivity implements OnMapReadyCallback {

    private GoogleMap mMap;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
        // Obtain the SupportMapFragment and get notified when the map is ready to be used.
        SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
            .findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);
    }

    /**
     * Manipulates the map once available.
     * This callback is triggered when the map is ready to be used.
     * This is where we can add markers or lines, add listeners or move the camera. In this case,
     * we just add a marker near Sydney, Australia.
     * If Google Play services is not installed on the device, the user will be prompted to install
     * it inside the SupportMapFragment. This method will only be triggered once the user has
     * installed Google Play services and returned to the app.
     */
    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;

        // Add a marker in Sydney and move the camera
        LatLng sydney = new LatLng(-34, 151);
        mMap.addMarker(new MarkerOptions().position(sydney).title("Marker in Sydney"));
        mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));
    }
}
```

```
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:map="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/map"
    android:name="com.google.android.gms.maps.SupportMapFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="th.ac.ubu.basicgooglemap.MapsActivity" />
```

อธิบายเพิ่มเติม



```
SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
    .findFragmentById(R.id.map);
mapFragment.getMapAsync(this);
```

คือการไปค้นหาแฟล็กเมนต์จาก xml เหมือนกับการ findViewById() นั่นเอง เมื่อได้ mapFragment แล้วจึงสั่ง getMapAsync() ซึ่งจะช่วยให้เมธอด onMapReady() ด้านล่างนั้นทำงาน

@Override

```
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;

    // Add a marker in Sydney and move the camera
    LatLng sydney = new LatLng(-34, 151);
    mMap.addMarker(new MarkerOptions().position(sydney).title("Marker in Sydney"));
    mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));
}
```

ซึ่งคลาส MapsActivity นั้นได้อิมพลีเมนต์ (implements) อินเตอร์เฟสชื่อว่า OnMapReadyCallback ซึ่งต้อง Override เมธอดชื่อว่า onMapReady() นั่นเอง โดยในเมธอดนี้จะทำการปักหมุด (Marker) ที่ Latitude -34 Longitude 151 และใส่ title ชื่อ "Marker in Sydney" และสุดท้ายสั่งให้แผนที่ขยับมุมมองไปที่ตำแหน่ง Latitude, Longitude เดียวกันโดยใช้คำสั่ง "mMap.moveCamera()"

ก็จะแสดงหน้าจอแผนที่ตามภาพด้านล่าง \*แนะนำให้ใช้เครื่องจริงในการรันเพราะต้องใช้พีเจอร์หลายๆอย่างที่ตัว emulator นั้นไม่มี หรือมี (Genymotion) ก็ไม่สมมุติเท่าเครื่องจริง



## แบบฝึกหัดท้ายบท

1. จงให้ความหมายของคำว่า google map
2. จงให้ความหมายคำสั่งที่ให้ส่วนนี้ พร้อมกับอธิบายความหมาย

```
keytool -list -v -keystore <keystore_file_path> -alias <alias_name> -
storepass <alias_pass> -keypass <keystore_pass>
```

3. ให้นักศึกษาค้นเอกสารคู่มือพร้อมเขียนคำสั่งพร้อมกับทำตามเอกสารเพื่อให้ได้ผลลัพธ์ที่ได้เป็นดังภาพ  
ต่อไปนี้ พร้อมกับแสดงสถานที่ของมหาวิทยาลัยของเรา

