

ภาคผนวก

ภาคผนวก ก

วิธีลงทะเบียน Android Map API

วิธีลงทะเบียน Android Map API

สิ่งที่จะต้องทำเพื่อเรียกใช้งาน Google Maps จะมีดังนี้

- นำรหัส SHA1 จาก Keystore ไปขอ API Key สำหรับ Google Maps ที่ Google Developer Console เพื่อกำหนดโปรเจกของแอปพลิเคชัน
- เพิ่ม Dependencies ของ Google Play Services ลงในโปรเจกของแอปพลิเคชัน
- กำหนด Layout XML สำหรับแสดง Google Maps
- เรียกใช้งานด้วยคำสั่งสำหรับ Google Maps เพื่อใช้งานตามต้องการ

การขอ API Key สำหรับ Google Maps

การขอ API Key นั้นอาจจะมีขั้นตอนยุ่งยากเล็กน้อย แต่ก็ต้องทำ เพราะว่า API Key เป็นสิ่งที่ทาง Google ใช้ระบุว่าแอปของนักพัฒนาคนไหนมีการเรียกใช้งาน Google Maps บ้าง ถึงแม้ว่า Google Maps จะให้ใช้งานฟรีสำหรับพีเจอร์เบื้องต้น (พีเจอร์พิเศษถึงจะต้องเสียเงิน)

โดย API Key นั้นจะสัมพันธ์กับชุดรหัส SHA1 ใน Keystore ที่ผู้ที่หลงเข้ามาอ่านใช้ Build แอปพลิเคชันตัวนั้นๆ และสัมพันธ์กับ Package ของแอปพลิเคชันที่เรียกใช้งาน ดังนั้น API Key นี้จะผูกกับแอปพลิเคชันตัวเดียวเท่านั้น ไม่สามารถนำไปใช้กับแอปตัวอื่นได้

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 ในขณะที่กำลังพัฒนาแอปพลิเคชันอยู่ ยังไม่ได้ Public ขึ้น Google Play Store หรือ 3rd Party App Store
- ใช้ SHA1 จาก Signed Keystore เมื่อพัฒนาแอปพลิเคชันเสร็จแล้ว และพร้อมจะ Public ขึ้น Google Play Store หรือ 3rd Party App Store

วิธีหา Debug Keystore

OS X หรือ Linux

```
~/.android/debug.keystore
```

Windows

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

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 Google ก็ได้กำหนดตายตัว

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

OS X และต้องการ SHA1 ของ Debug Keystore ก็จะต้องใช้คำสั่งดังนี้

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

ผลที่ได้จะเป็นดังนี้

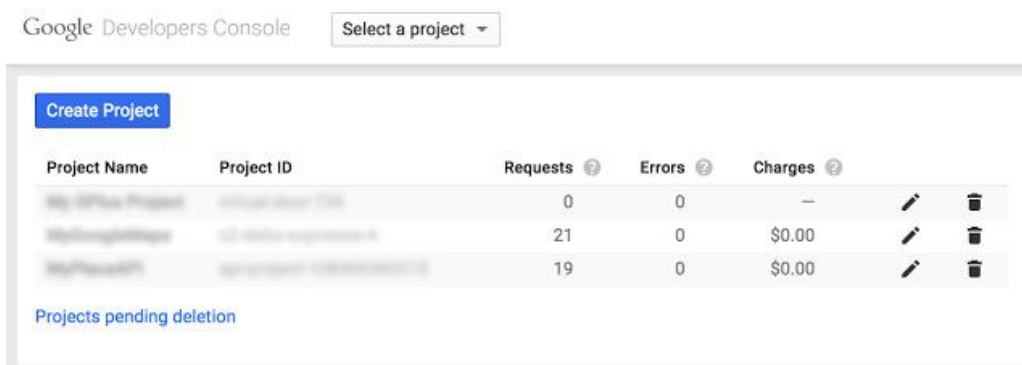
```
Akexorcist-MBP:~$ keytool -list -v -keystore ~/.android/debug.keystore
-alias androiddebugkey -storepass android -keypass android
Alias name: androiddebugkey
Creation date: Feb 13, 2015
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=Android Debug, O=Android, C=US
Issuer: CN=Android Debug, O=Android, C=US
Serial number: 54dd78b3
Valid from: Fri Feb 13 11:08:19 ICT 2015 until: Sun Feb 05 11:08:19 ICT 2045
Certificate fingerprints:
  MD5: 06:C1:A4:8F:3B:1B:08:59:D2:FB:C9:AC:EF:52:B6:A0
  SHA1: 7D:3C:4F:55:41:E1:E3:20:62:60:F3:00:F6:22:D6:E5:E9:96:30:0D
  SHA256: E6:E1:CE:63:EE:12:29:A2:49:32:ED:C6:F7:73:0F:19:D9:3A:7C:9A:E8:EA:48:41:
13:4A:06:4E:C9:73:C5:6D
  Signature algorithm name: SHA1withRSA
  Version: 3
Akexorcist-MBP:~$
```

รูปที่ ก.1 แสดง SHA1

SHA1 มีชุดรหัสแสดงอยู่ 20 ชุด นำ SHA1 ที่ได้จากตรงนี้ไปขอ API Key จาก Google Developer Console

ขอ API Key จาก Google Developer Console

เข้าหน้าเว็บ Google Developer Console แล้ว Login ด้วย Google Account จากนั้นก็จะเข้าสู่หน้าแรกของเว็บ กดปุ่ม Create Project



รูปที่ ก.2 API Key จาก Google Developer Console

จากนั้นให้พิมพ์ชื่อโปรเจกของเรา และกดปุ่ม Create

New Project

Project name ?

Awesome App

Your project ID will be awesome-app-977 ? Edit

Show advanced options...

Create Cancel

รูปที่ ก.3 API Key จาก Google Developer Console (ต่อ)

เมื่อสร้างเสร็จแล้วก็จะเข้าสู่หน้า Overview ของโปรเจกทันที ให้กดเลือกที่ APIs & auth > APIs ที่อยู่ฝั่งซ้ายมือของหน้าเว็บ

Google Developers Console Awesome App Sign up for a free trial.

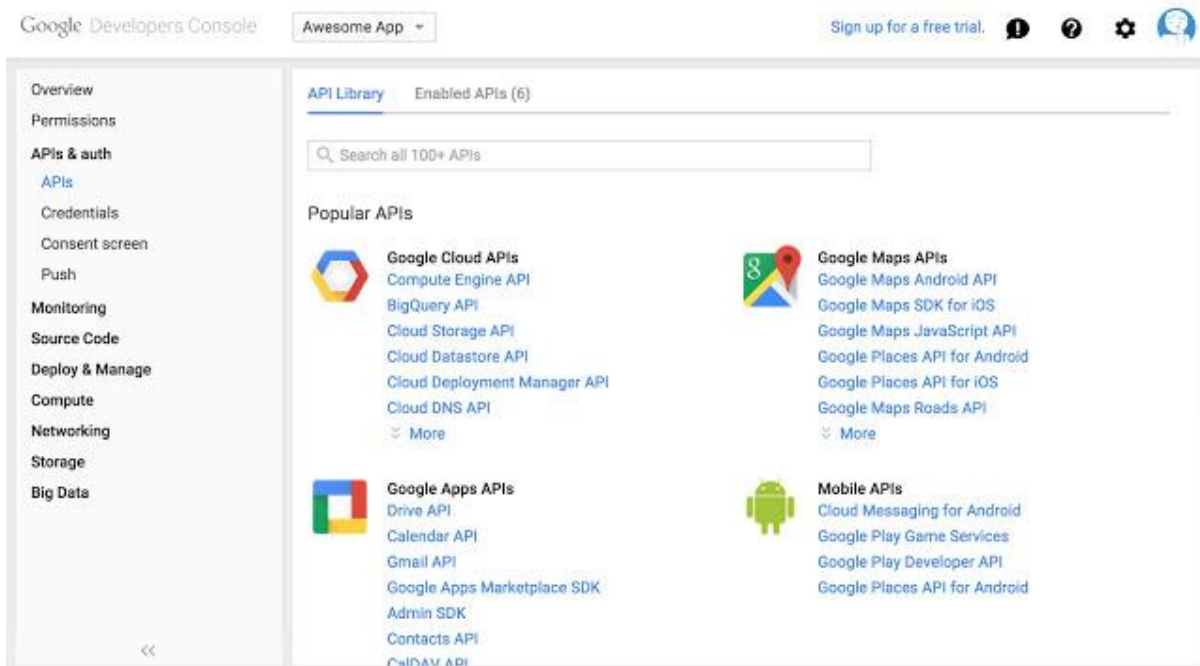
Project ID: awesome-app-977 Project Number: 411230074956 Estimated charges this month: \$0.00

Project Dashboard

- Take the App Engine quickstart**
Learn "Hello World" for App Engine and deploy your first app to the cloud. Duration: ~10 minutes.
[Try App Engine](#)
- Take the Compute Engine quickstart**
Use NodeJS and MongoDB on Compute Engine to create a two-tier to-do list application. Duration: ~15 minutes.
[Try Compute Engine](#)
- Launch click-to-deploy software**
Deploy popular open stacks on Google Compute Engine just by filling out a simple form.
[Try Click to Deploy](#)
- Try BigQuery with population data**
Run queries against huge public data sets to see how BigQuery can help you analyze your own data.
[Try BigQuery](#)
- Create a storage bucket**
Store your unstructured data safely and with high availability using Google Cloud Storage.
[Create bucket](#)
- Boost your app with a Google API**
Use Google APIs with your app to harness the power of Google's services and technologies.
[Enable an API](#)

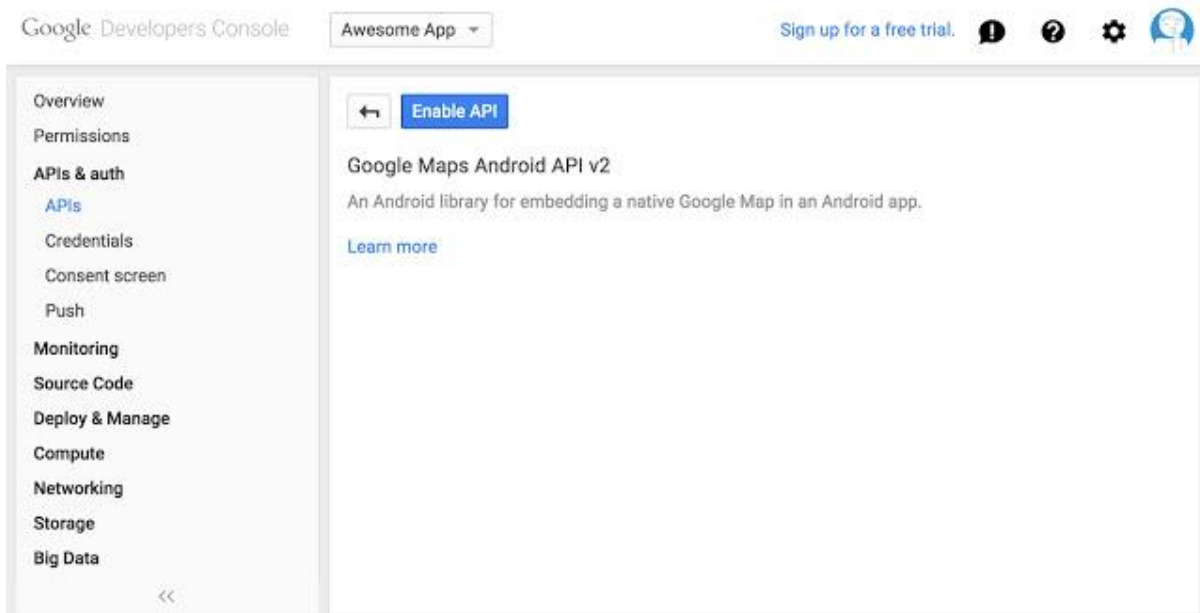
รูปที่ ก.4 API Key จาก Google Developer Console (ต่อ)

ที่หน้านี้จะเป็นการเลือก API ของ Google ว่าจะต้องการใช้ Google Services ตัวไหน ให้เลือก Google Maps Android API



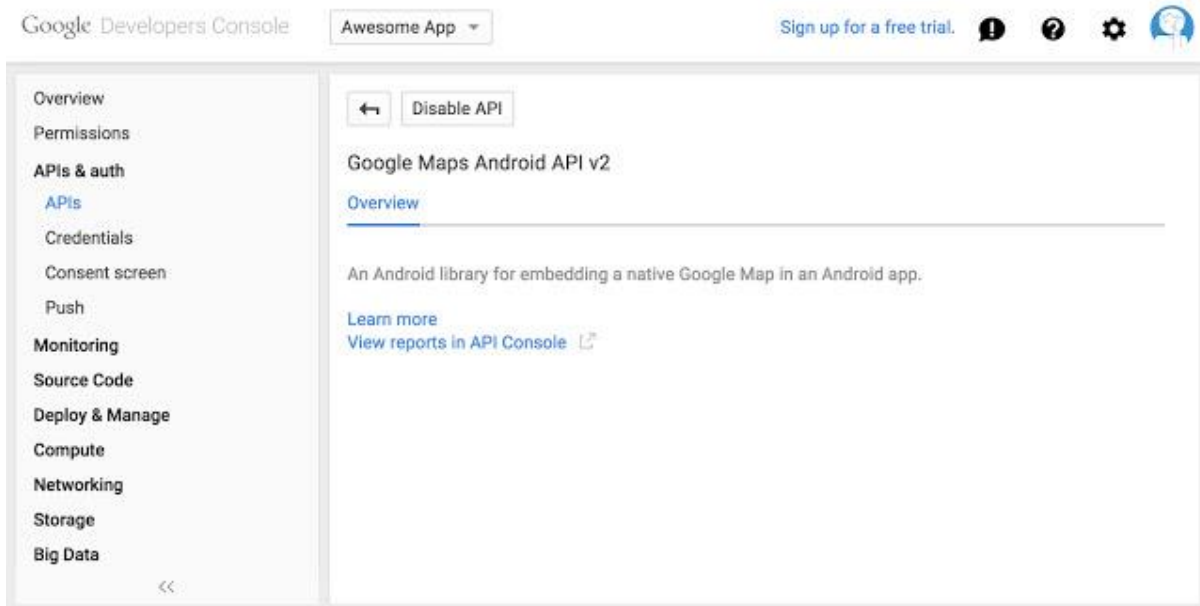
รูปที่ ก.5 API Key จาก Google Developer Console (ต่อ)

กดที่ปุ่ม Enable API เพื่อเปิดใช้งาน Google Maps Android API



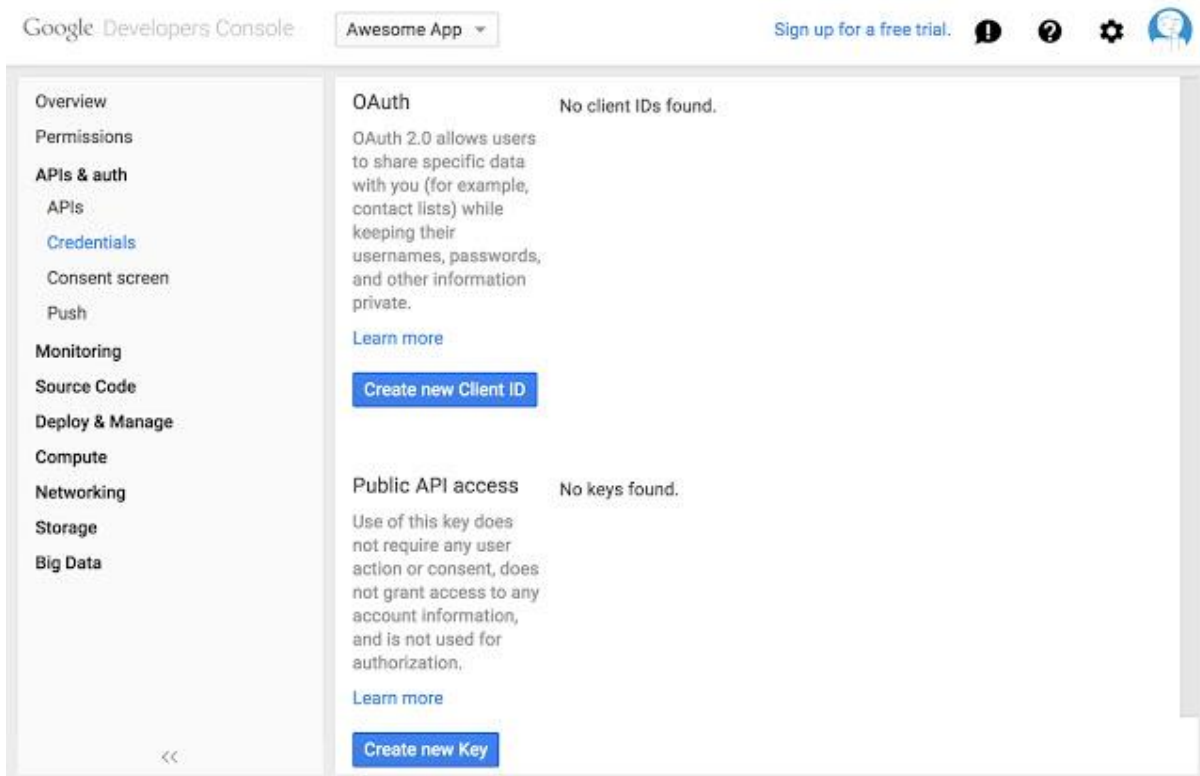
รูปที่ ก.6 API Key จาก Google Developer Console (ต่อ)

เมื่อเปิดใช้งานเรียบร้อยแล้ว ก็ต้องไปทำการขอ API Key



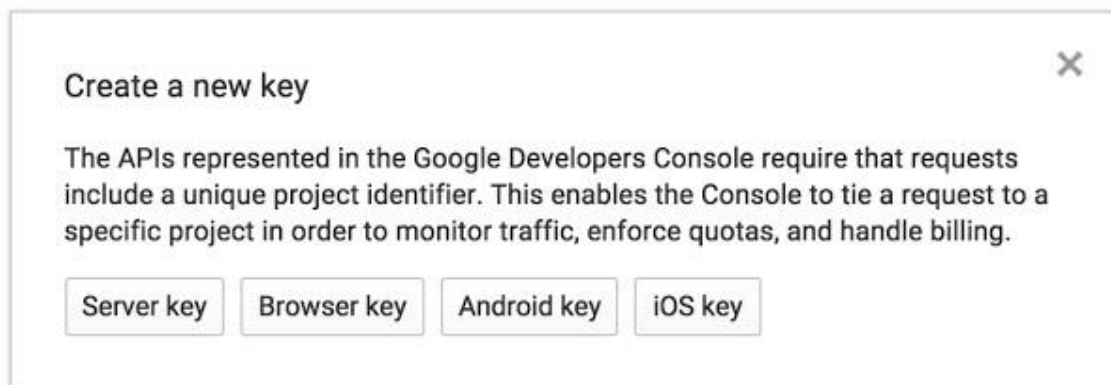
รูปที่ ก.7 API Key จาก Google Developer Console (ต่อ)

ให้เลือกไปที่ APIs &auth > Credentials แล้วกดที่ปุ่ม Create new Key ที่ Public API access เพื่อขอ API Key



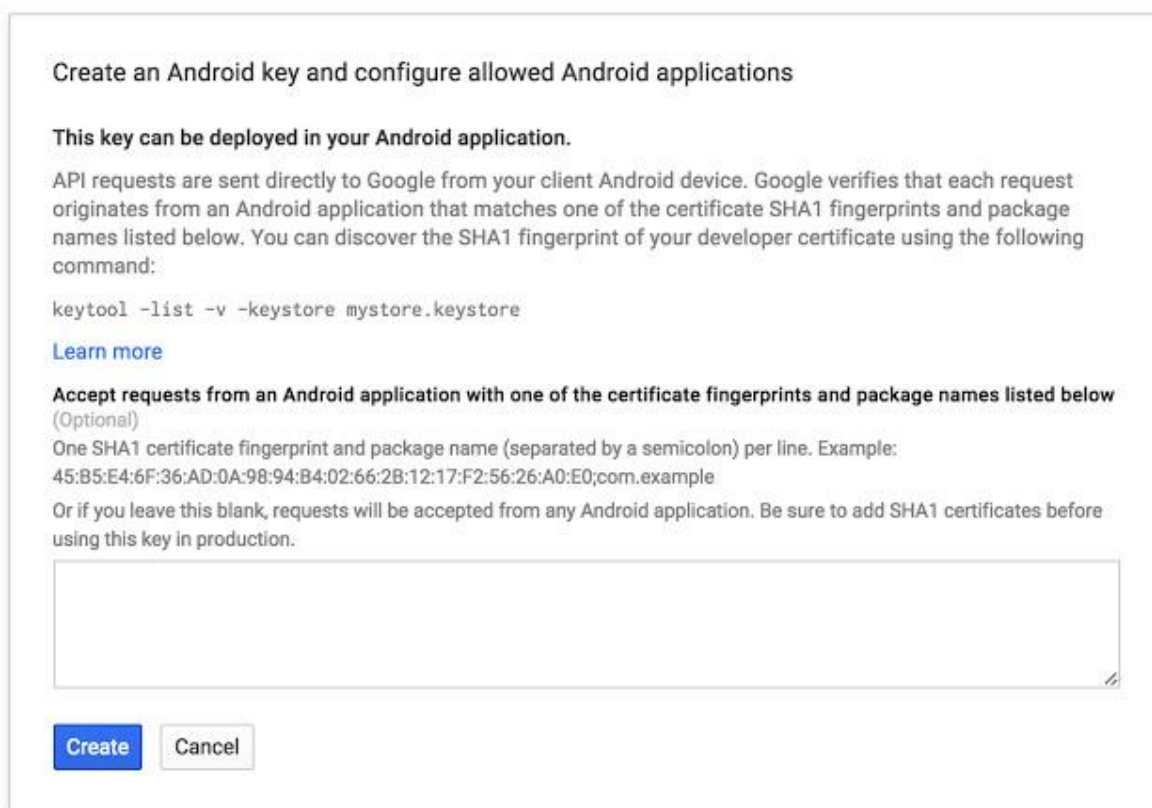
รูปที่ ก.8 API Key จาก Google Developer Console (ต่อ)

จะมีหน้าต่างเล็กๆแสดงขึ้นมาว่าจะขอ Key สำหรับอะไร ให้กดเลือกที่ปุ่ม Android Key



รูปที่ ก.9 API Key จาก Google Developer Console 21(ต่อ)

จะมีหน้าต่างให้ใส่ SHA1 กับ Package Name ของแอปพลิเคชันลงในช่องว่าง



รูปที่ ก.10 API Key จาก Google Developer Console (ต่อ)

ตัวอย่างกำหนด Package Name ไว้ว่า com.akexorcist.awesomeapp เวลากรอกลงไปในช่วงว่าง
จะต้องใส่ SHA1 ตามด้วย Package Name โดยมีเครื่องหมาย ; เป็นตัวคั่น

<sha1_key>;<package_name>

Accept requests from an Android application with one of the certificate fingerprints and package names listed below (Optional)

One SHA1 certificate fingerprint and package name (separated by a semicolon) per line. Example:

45:B5:E4:6F:36:AD:0A:98:94:B4:02:66:2B:12:17:F2:56:26:A0:E0;com.example

Or if you leave this blank, requests will be accepted from any Android application. Be sure to add SHA1 certificates before using this key in production.

7D:3C:4F:55:41:E1:E3:20:62:60:F3:00:F6:22:D6:E5:E9:96:30:0D;com.akexorcist.awesomeapp

Create

Cancel

รูปที่ ก.11 API Key จาก Google Developer Console (ต่อ)

และ API Key สามารถใช้ร่วมกันกับหลายๆแอป โดยใส่หลายๆบรรทัดแบบนี้

Accept requests from an Android application with one of the certificate fingerprints and package names listed below (Optional)

One SHA1 certificate fingerprint and package name (separated by a semicolon) per line. Example:

45:B5:E4:6F:36:AD:0A:98:94:B4:02:66:2B:12:17:F2:56:26:A0:E0;com.example

Or if you leave this blank, requests will be accepted from any Android application. Be sure to add SHA1 certificates before using this key in production.

7D:3C:4F:55:41:E1:E3:20:62:60:F3:00:F6:22:D6:E5:E9:96:30:0D;com.akexorcist.awesomeapp
8A:35:1A:AA:F4:69:19:20:2D:1C:0A:F1:12:12:4D:5B:BE:FA:00:FE;com.akexorcist.betterapp

Create

Cancel

รูปที่ ก.12 API Key จาก Google Developer Console (ต่อ)

เมื่อเสร็จแล้วก็กดปุ่ม Create

ให้ดูที่ Public API Access จะมี API Key แสดงอยู่ พร้อมกับบอก Android applications ที่
กรอกลงไปเมื่อครู่ด้วย ให้เอา API Key ไปใช้งานในแอปได้เลย

นำ API Key ไปใช้ในแอปพลิเคชัน

เพิ่ม Dependencies ของ Google Play Services ลงในโปรเจก

ในกรณีที่เพิ่งสร้างโปรเจกขึ้นมาจาก Android Studio ก็ให้เปิด build.gradle ของโมดูลที่ชื่อว่า app ขึ้นมาได้เลย แต่ถ้า Import มาจาก Eclipse อีกทีก็ให้ดูว่าโมดูลตัวไหนคือโมดูลหลักของแอป แล้วจึงเปิด build.gradle ของโมดูลตัวนั้นๆขึ้นมา

จากนั้นก็เพิ่ม Dependencies เข้าไปดังนี้

```
compile 'com.google.android.gms:play-services-maps:7.5.0'
```

รูปที่ ก.13 API Key จาก Google Developer Console (ต่อ)

คำสั่งข้างบนเป็นการเพิ่ม Google Maps อย่างเดียว แต่ถ้าผู้ที่หลงเข้ามาอ่านเรียกใช้งาน Google Play Services จำนวนหลายตัวก็สามารถเพิ่มไปทั้งหมดได้ด้วยคำสั่ง

```
compile 'com.google.android.gms:play-services:7.5.0'
```

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:21.0.3'
```

```
compile 'com.google.android.gms:play-services:6.5.87'
```

รูปที่ ก.14 API Key จาก Google Developer Console (ต่อ)

กำหนดค่าใน Android Manifest

อย่างแรกสุดคือการกำหนด Meta Data สำหรับเวอร์ชันของ Google Play Services ใน Tag ของ <application> ของ AndroidManifest.xml ดังนี้

```

<meta-data
    android:name="com.google.android.gms.version"
    android:value="@integer/google_play_services_version" />

<application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >
    <meta-data
        android:name="com.google.android.gms.version"
        android:value="@integer/google_play_services_version" />

```

รูปที่ ก.15 API Key จาก Google Developer Console (ต่อ)

Meta Data ตัวนี้เป็นการกำหนดให้รู้ว่า Google Play Services ที่ใช้งานอยู่เป็นเวอร์ชันอะไร เพราะถ้าเวอร์ชันของแอปเราใหม่กว่าที่มีอยู่ในเครื่อง ระบบก็จะให้ผู้ใช้อัปเดต Google Apps ในเครื่องแทน

ต่อไปก็กำหนด API Key สำหรับใช้งาน Google Maps โดยจะใส่เป็น Meta Data ใน Tag ของ <application> เช่นเดิม

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

ตรง your_api_key ให้ใส่ API Key ที่ได้จาก Google Developer Console

```
<meta-data
  android:name="com.google.android.gms.version"
  android:value="@integer/google_play_services_version" />
<meta-data
  android:name="com.google.android.maps.v2.API_KEY"
  android:value="AIzaSyBLBiG7DV8xn2Azan1o7IB4I2XTj4mR9v8" />
```

รูปที่ ก.16 API Key จาก Google Developer Console(ต่อ)

ต่อไปเพิ่ม Permission เข้าไปดังนี้

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```


```
<uses-sdk
  android:minSdkVersion="8"
  android:targetSdkVersion="19" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
<!--
The ACCESS_COARSE/FINE_LOCATION permissions are not required to use
Google Maps Android API v2, but are recommended.
-->
```

รูปที่ ก.17 API Key จาก Google Developer Console(ต่อ)

- INTERNET ดาวน์โหลดแผนที่จาก Server
- ACCESS_NETWORK_STATE ตรวจสอบสถานะการเชื่อมต่อของ Cellular Data ก่อนจะดาวน์โหลดแผนที่
- ACCESS_WIFI_STATE ตรวจสอบสถานะการเชื่อมต่อของ WiFi ก่อนจะดาวน์โหลดแผนที่
- WRITE_EXTERNAL_STORAGE เพื่อทำ Cache สำหรับข้อมูลแผนที่

และจะมีเพิ่มอีกสองตัวเมื่อผู้ที่หลงเข้ามาอ่านเรียกใช้งานคำสั่งแสดงตำแหน่งปัจจุบันของผู้ใช้บน Google Maps แต่ถ้าไม่ได้ใช้งานก็ไม่จำเป็นต้องใส่ก็ได้ โดยให้เลือกอย่างใดอย่างหนึ่งก็พอ ขึ้นอยู่กับผู้ที่หลงเข้ามาอ่านว่าต้องการความแม่นยำในการระบุตำแหน่งของผู้ใช้มากน้อยแค่ไหน

```
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
```



```
<!--
The ACCESS_COARSE/FINE_LOCATION permissions are not required to use
Google Maps Android API v2, but are recommended.
-->
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-feature android:glEsVersion="0x00020000" android:required="true" />
```

รูปที่ ก.18 API Key จาก Google Developer Console(ต่อ)

- ACCESS_COARSE_LOCATION ใช้ Cellular Data หรือ WiFi เพื่อหาตำแหน่งของผู้ใช้แบบคร่าวๆ ตำแหน่งที่ได้จะไม่มีความแม่นยำมากนัก
- ACCESS_FINE_LOCATION ใช้ GPS เพื่อหาตำแหน่งของผู้ใช้ให้แม่นยำที่สุดเท่าที่ทำได้

เนื่องจาก Google Maps ใช้ OpenGL ES 2.0 ในการแสดงผลแผนที่ ดังนั้นจะต้องเพิ่ม Tag เพื่อใช้งาน OpenGL ES 2.0 ดังนี้

```
<uses-feature
  android:glEsVersion="0x00020000"
  android:required="true"/>
→
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-feature android:glEsVersion="0x00020000"
  android:required="true" />
```

รูปที่ ก.19 API Key จาก Google Developer Console(ต่อ)