

CHAPTER

11

ແພນທີ ແລະຕຳແໜ່ງຂອງຜູ້ໃຈ

ເນື້ອຫາໃນບົກນີ້

- ◆ การเตรียมພຣັມເພື່ອໃຊ້ງານ Google Maps Android API v2
- ◆ การແສດງແຜນທີ Google Maps ຕ້ວຍ MapFragment
- ◆ ການປັບປຸງຕຳແໜ່ງແລະນຸ້ມນອງຂອງແຜນທີ
- ◆ ການເພີມໜຸດ (Marker) ລົງບນແຜນທີ



- ◆ ການປັບແຕ່ງ Info Window ຂອງໜຸດ
- ◆ ການວາດເສັ້ນແລະຮູບທຽບຮັບແນນທີ
- ◆ ການທາດແໜ່ງຂອງຜູ້ໃຈ
- ◆ ການຕິດຕາມການເປັນແປງຕຳແໜ່ງຂອງຜູ້ໃຈ

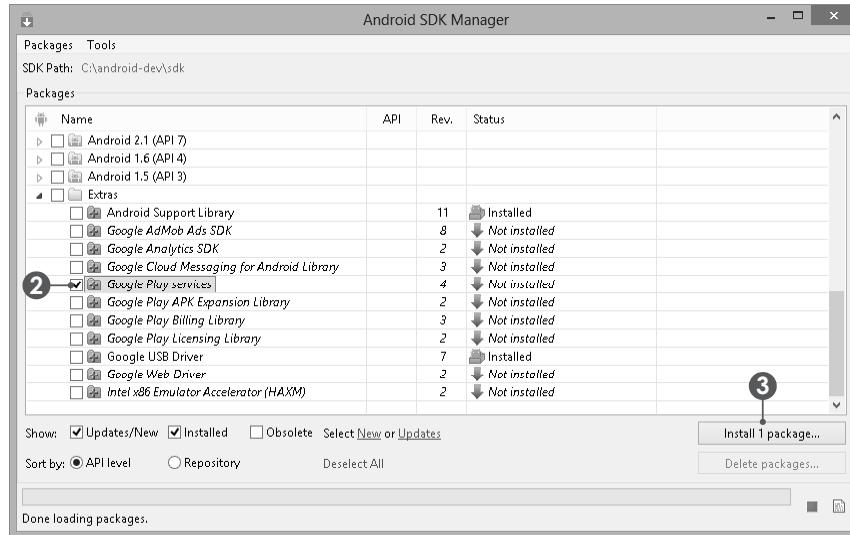
การเตรียมพร้อมเพื่อใช้งาน Google Maps Android API v2

การแสดงแผนที่ Google Maps ออกมายในแอพของเราเองจะทำได้โดยใช้控件 MapView หรือ MapFragment ที่เป็นส่วนหนึ่งของ Google Maps Android API v2 ซึ่ง API ดังกล่าวไม่ได้รวมอยู่ใน Android SDK แต่จะอยู่ใน Google Play services SDK ที่ต้องติดตั้งเพิ่มต่างหาก (เนื่องจาก API ส่วนนี้ถูกเก็บไว้ในไฟล์นามาเบอร์สโตร์โคเด)

ในหัวข้อนี้จะอธิบายวิธีการติดตั้ง Google Play services SDK รวมถึงการลงทะเบียนขอ API Key เพื่อให้เรียกใช้บริการ Google Maps ได้

การติดตั้ง Google Play services SDK

- ที่ Eclipse ให้คลิกเมนู Windows ▶ Android SDK Manager หรือคลิก  บนทูลบาร์ เพื่อเปิด Android SDK Manager

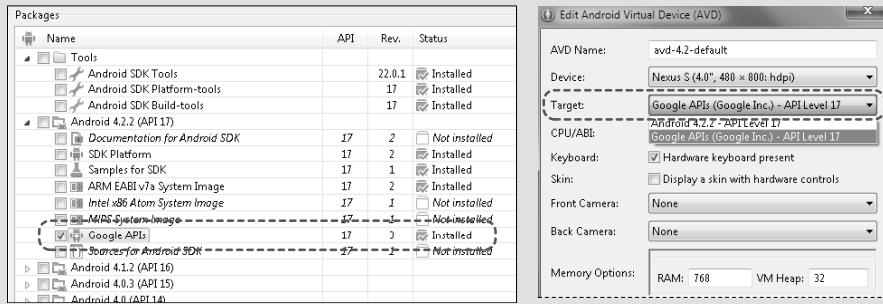


- ที่หัวข้อ Extras ให้เลือก Google Play services
 - คลิก Install จะมีหน้าจอที่แสดงเงื่อนไขการใช้งานขึ้นมา ก็ให้เลือก Accept และคลิก Install จากนั้นให้รอนกว่าจะดาวน์โหลดและติดตั้งเสร็จสมบูรณ์
- Google Play services จะถูกติดตั้งที่โฟลเดอร์ <android-sdk-folder>\extras\google\google_play_services ในเครื่องพีซีของคุณ เช่นกรณีของผู้เขียนจะอยู่ที่ C:\android-dev\sdk\extras\google\google_play_services

Google Play services จัดเตรียม API ต่างๆ (หนึ่งในนั้นคือ Maps API) มาให้ในรูปแบบของ Library Project ที่เราต้องอิมพอร์ต (import) เข้ามาใน Eclipse ดังนั้นถ้าไปเราจะอิมพอร์ต Library Project ดังกล่าว

NOTE »»

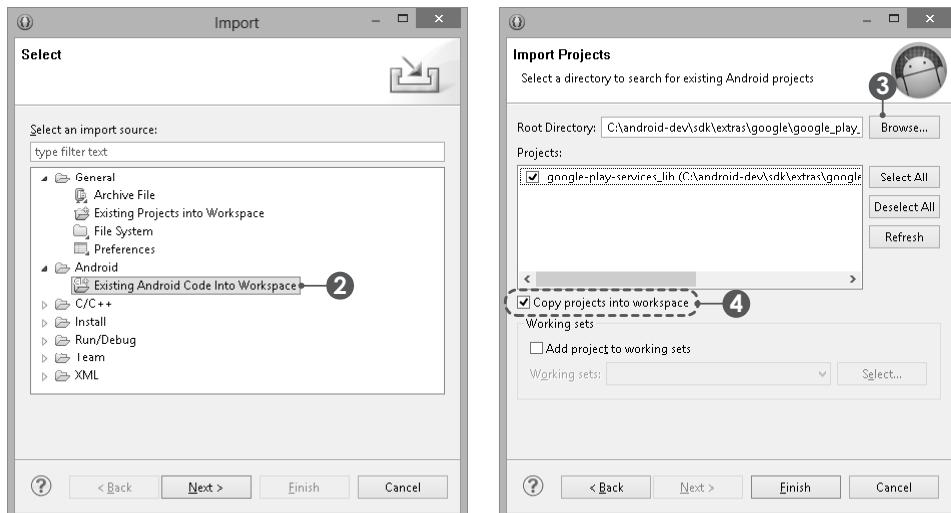
ถ้าคุณต้องการทดสอบแอพบนอีเมลเตอร์ จะต้องเลือกติดตั้ง Google APIs ที่อยู่ภายใต้หัวข้อ Android 4.2.2 (API 17) หรือสูงกว่าด้วย แล้วสร้างอีเมลเตอร์ใหม่หรือแก้ไขอีเมลเตอร์ที่สร้างไว้แล้วโดยระบุ Target เป็น Google APIs นั้น ดังรูป



กรณีที่ Google Play services จะมีอยู่ใน Google APIs ของ Android 4.2.2 หรือสูงกว่าเท่านั้นในการถึงของ อีเมลเตอร์ แต่สำหรับเครื่องจริงจะใช้ได้ตั้งแต่ Android 2.2 ขึ้นไป และติดตั้งแอพ Google Play Store แล้ว

อิมพอร์ต Library Project ของ Google Play services

- 1 ที่ Eclipse ให้คลิกเมนู File > Import



- 2 เลือก Android ▶ Existing Android Code Into Workspace และคลิก Next
- 3 คลิกปุ่ม Browse และเข้าไปเลือกโฟลเดอร์ <android-sdk-folder>\extras\google\google_play_services\libproject เมื่อเลือกแล้วจะปรากฏชื่อโปรเจ็ค google-play-services_lib ในช่อง Projects ก็ให้คลิกเลือกโปรเจ็คนี้
- 4 เลือก Copy projects into workspace และคลิก Finish

Library Project ของ Google Play services จะถูกอิมพอร์ตเข้ามาใน Eclipse หลังจากนั้น เมื่อต้องการสร้างโปรเจ็คแอนดรอยด์ที่เรียกว่า Maps API เราจะต้องตั้งค่าโปรเจ็คให้อ้างอิงไปยัง Library Project ดังกล่าว ซึ่งจะอธิบายต่อไปตอนสร้างแอพ MapDemo

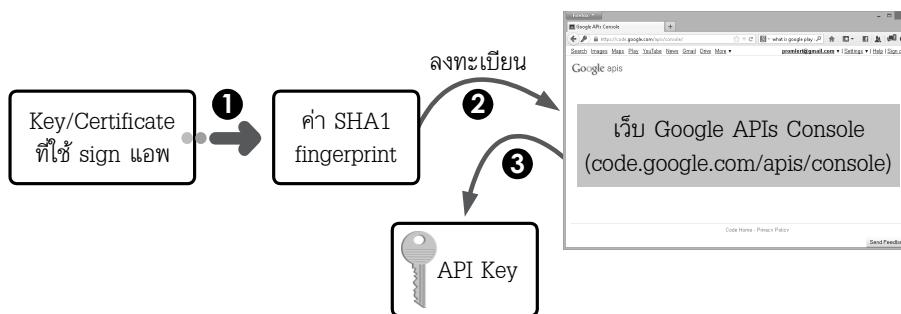
TIP»»

หลังจากอิมพอร์ตแล้ว ถ้ามีเครื่องหมายกากรบทสีแดงที่ชื่อโปรเจ็ค google-play-services_lib ให้ลองปิด และเปิด Eclipse ใหม่

การลงทะเบียน API Key

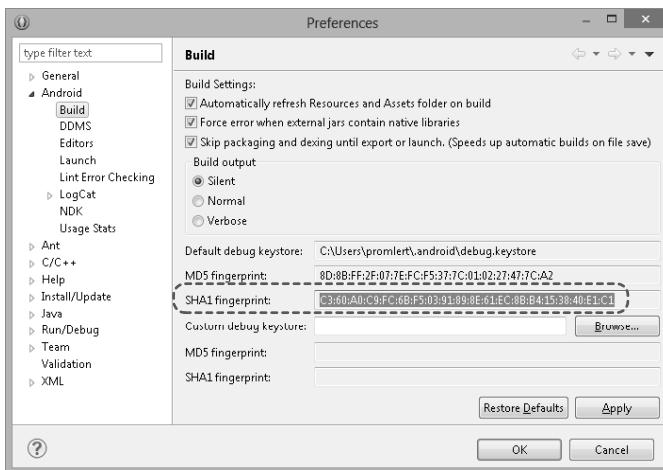
ถูกเก็บกำหนดให้นักพัฒนาต้องขอ API Key หากต้องการเข้าถึงข้อมูลแผนที่ Google Maps จากแอพแอนดรอยด์ ทั้งนี้เพื่อที่ถูกเก็บจะตรวจสอบได้ว่าแผนที่ถูกใช้งานโดยใครบ้าง (หมายถึงผู้พัฒนาแอพ) และถูกใช้งานอย่างไร

หลักการขอ API Key คือ ให้นำค่า SHA1 fingerprint ของคีย์หรือ Certificate ที่คุณใช้ในการ sign แอพ ไปลงทะเบียนที่เว็บ Google APIs Console (code.google.com/apis/console) คุณก็จะได้ API Key สำหรับนำระบุในโปรเจ็คแอนดรอยด์ที่คุณพัฒนา



เนื่องจากการทดสอบบนแอปบนอีเมลเตอร์นั้น Eclipse จะ sign แอพให้เราโดยใช้ Debug Key (Debug Certificate) ดังนั้นในที่นี่จะนำค่า SHA1 fingerprint ของ Debug Key ไปใช้ในการขอ API Key (แต่เมื่อจะส่งแอพขึ้น Google Play คุณต้องนำ Private Key ที่ใช้ sign แอพไปขอ API Key ใหม่)

ก า ร ห า ค า SHA1
 fingerprint ของ Debug Key
 ที่ใช้ sign แอพ สามารถทำได้
 ง่ายๆ โดยเปิด Preferences
 ของ Eclipse ขึ้นมา (เมนู
 Window ▶ Preferences)
 แล้วไปที่หัวข้อ Android ▶
 Build ทางซ้าย ก็จะพบค่า
 SHA1 fingerprint ทางขวา
 ดังรูป ให้ก็อบปี้ค่านี้เก็บไว้
 เพื่อนำไปลงทะเบียนที่เว็บ
 Google APIs Console ต่อไป



NOTE»»

ใน ADT รุ่นเก่าๆ จะไม่แสดงค่า SHA1 fingerprint ในหน้าจอ Preferences ของ Eclipse ซึ่งถ้าเป็นอย่างนั้น คุณจะต้องหา SHA1 fingerprint โดยพิมพ์คำสั่งที่ Command Prompt ดังนี้ (พิมพ์ต่อเนื่องในบรรทัดเดียว)

```
keytool -list -v -keystore "ชื่อพาธเต็มของไฟล์ debug.keystore"
-alias androiddebugkey -storepass android -keypass android
```

ซึ่งจะได้ SHA1 fingerprint ดังรูป

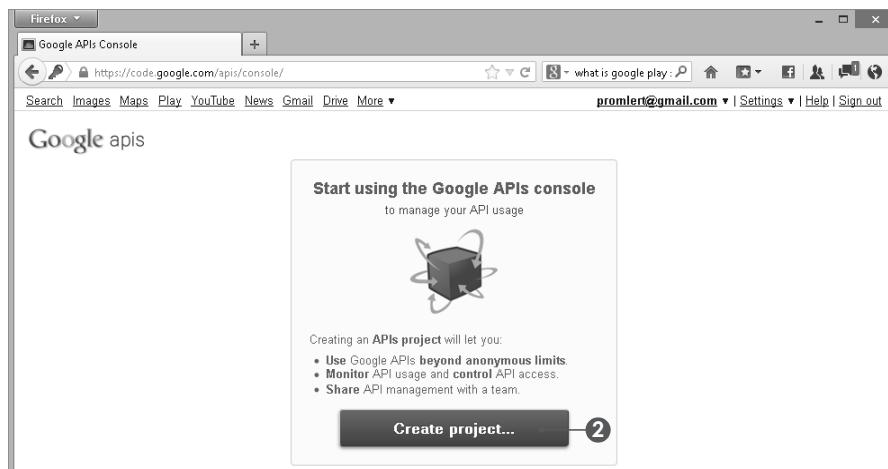
```
C:\Program Files\Java\jdk1.7.0_09\bin>keytool -list -v -keystore "C:\Users\Promlert\.android\debug.keystore" -alias androiddebugkey -storepass android -keypass android
Alias name: androiddebugkey
Creation date: 11 ๐. ๖. 2555
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=Android Debug, O=Android, C=US
Issuer: CN=Android Debug, O=Android, C=US
Serial number: 73967ed1
Valid from: Tue Dec 11 15:01:27 ICT 2012 until: Thu Dec 04 15:01:27 ICT 2042
Certificate fingerprints:
        MD5: 64:3E:CE:A4:22:CE:7C:8C:6A:59:25:95:E4:04:59:31
        SHA1: 03:60:A0:C9:FC:6B:F5:03:91:89:8E:61:EC:8B:B4:15:38:40:E1:C1
        SHA256: 89:R1:R8:R9:D2:26:23:DB:Rf:66:R4:4D:BD:40:5A:9F:1D:42:B6:60:9A:
A0:53:C8:A9:B7:8F:C7:71:4A:1D:69
        Signature algorithm name: SHA256withRSA
        Version: 3

Extensions:
```

อันนี้ คำสั่ง keytool เป็นส่วนหนึ่งของ JDK โดยอยู่ที่ไดร์ก็อกวี <JDK-folder>\bin เช่น C:\Program Files\Java\jdk1.7.0_09\bin เป็นต้น ดังนั้นหากไม่ได้กำหนด PATH ไว้ก็จะต้องเข้าไปยังไดร์ก็อกวีดังกล่าว ก่อนจะเรียกใช้คำสั่ง keytool ได้

หลังจากได้ SHA1 fingerprint มาแล้วให้นำไปลงที่เป็นที่เว็บ Google APIs Console ดังนี้

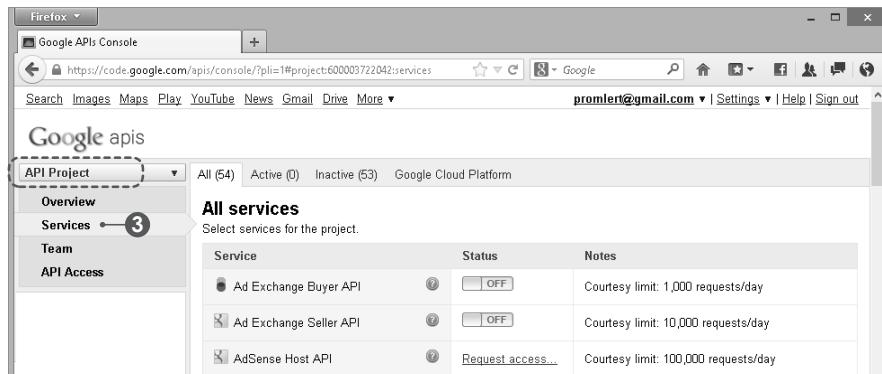
- 1 ให้บรรดาวเชอร์เข้าเว็บ Google APIs Console ที่ code.google.com/apis/console ถ้าคุณยังไม่ได้ล็อกอินด้วยบัญชี Google จะปรากฏหน้าจอให้ล็อกอิน หลังจากล็อกอินแล้วจะแสดงหน้าจอดังรูป



Google APIs Console คือเครื่องมือที่กูเกิลทำไว้ให้นักพัฒนาใช้บริหารจัดการการใช้งาน API ต่างๆ ของกูเกิล

2 คลิก Create project

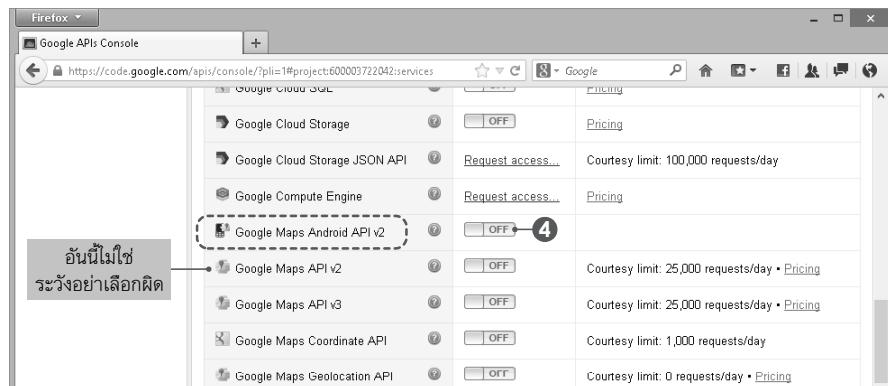
Google APIs Console จะสร้างโปรเจคใหม่ที่มีชื่อว่า API Project ขึ้นมา และจะแสดงชื่อโปรเจคที่มุ่งหมายบน ดังรูป



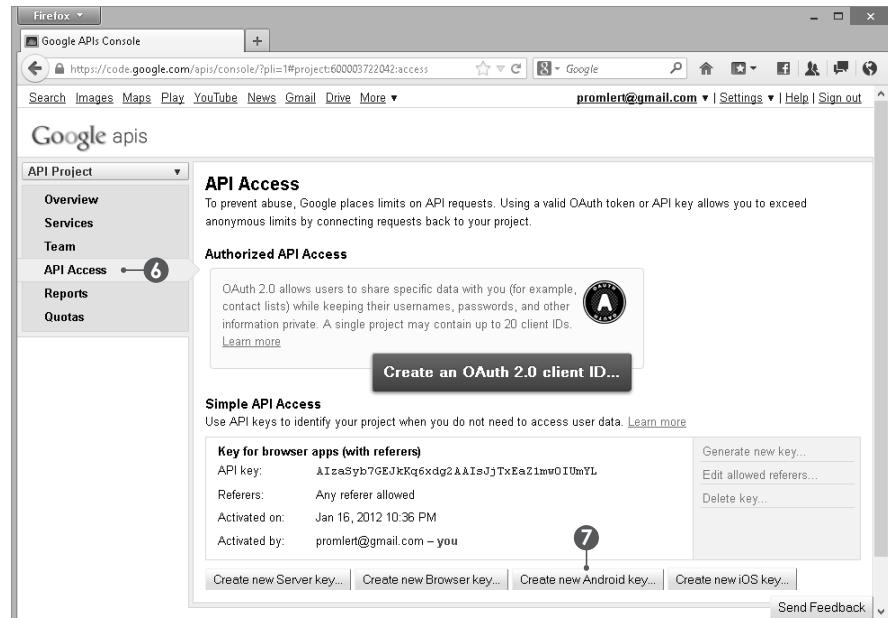
NOTE»»

คำว่า “โปรเจค” ในที่นี้หมายถึงโครงการใน Google APIs Console ซึ่งแต่ละโครงการจะเกี่ยวกับการใช้งาน Google API ต่างๆ

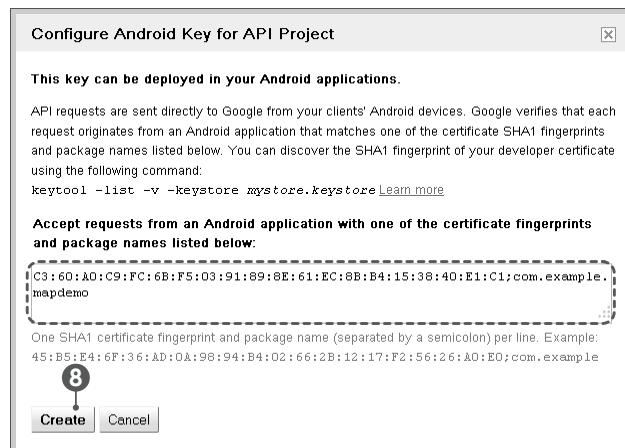
- 3 ตอนนี้คุณจะอยู่ที่หน้า All services ซึ่งแสดงรายชื่อ API ต่างๆของคุณเกลิที่เปิดให้บันทึกน้ำภายนอก เรียกใช้ได้ แต่ถ้าไม่ได้อยู่ที่หน้านี้ให้คลิกหัวข้อ Services ทางซ้าย



- 4 เลื่อนลงมาที่ Google Maps Android API v2 (จะมีชื่อคล้ายกับอันก่อนคือ Google Maps API v2 ต้องระวังอย่าเลือกพิสดั้น) แล้วคลิกสวิตช์ OFF เพื่อเปลี่ยนสถานะเป็น ON ซึ่งจะเข้าสู่หน้าจอที่แสดงเงื่อนไขการใช้งาน
- 5 เลื่อนลงมาข้างล่าง เลือก I agree to these terms แล้วคลิก Accept จะกลับไปหน้า All services



- 6 คลิกหัวข้อ API Access ทางซ้าย
- 7 คลิก Create new Android key
- 8 นำ SHA1 fingerprint ที่ ก็อปปี้มารอ粘ลงใน
ตามด้วยเครื่องหมาย
semicolon (;) และชื่อ
แพคเกจของแอพที่จะสร้าง
(ในที่นี่คือ com.example.
mapdemo) เสิร์ฟเวอร์คลิก
Create



- 9 จะมีกรอบ Key for Android apps (with certificates) เพิ่มขึ้นมา ซึ่งภายในจะมี API Key ที่ ประกอบด้วยตัวอักษรและตัวเลขต่างๆจำนวน 40 ตัวอยู่ ให้ก็อปปี้ API Key นี้เก็บไว้ เพื่อนำไประบุ ในโปรเจคแอนдрอยด์ต่อไป

The screenshot shows the Google APIs Console interface. The left sidebar has 'API Project' selected, with 'Overview', 'Services', 'Team', 'API Access' (selected), 'Reports', and 'Quotas' as options. The main content area is titled 'API Access' and contains the following information:

To prevent abuse, Google places limits on API requests. Using a valid OAuth token or API key allows you to exceed anonymous limits by connecting requests back to your project.

Authorized API Access

OAuth 2.0 allows users to share specific data with you (for example, contact lists) while keeping their usernames, passwords, and other information private. A single project may contain up to 20 client IDs.

[Learn more](#)

Create an OAuth 2.0 client ID...

Simple API Access

Use API keys to identify your project when you do not need to access user data. [Learn more](#)

Key for Android apps (with certificates)		Generate new key... Edit allowed Android apps... Delete key...
API key:	A1zasyb... API Key ของคุณ e3tM82gb4	
Android apps:	C3:60:A0:C9:FC:6B:F5:03:91:89:8E:61:EC:8B:B4:15:38:40:E1:C1;com.example.mapdemo	
Activated on:	Jun 17, 2013 12:22 AM	
Activated by:	promlert@gmail.com - you	

Key for browser apps (with referers)

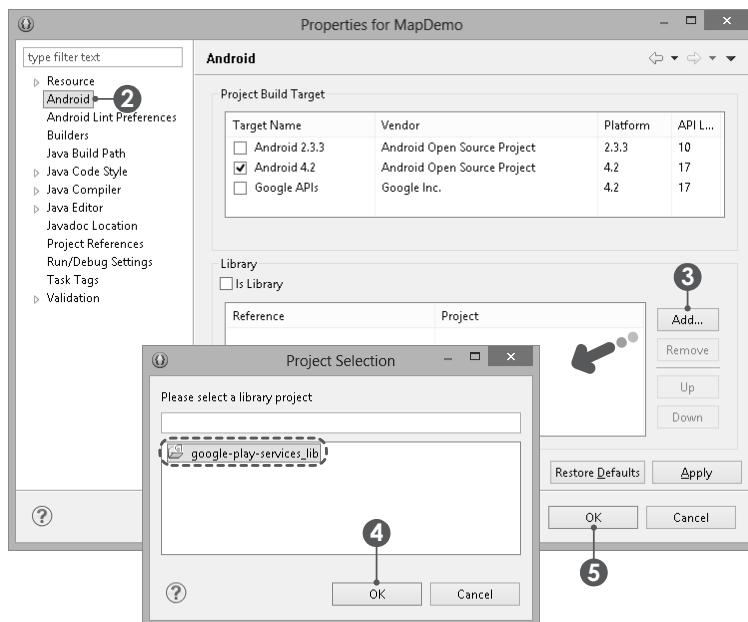
[Generate new...](#) [Send Feedback](#)

การแสดงแผนที่ Google Maps ด้วย MapFragment

ตัวอย่างนี้จะแสดงแผนที่ Google Maps โดยใช้ค่อนໂຫຣລ MapFragment ของ Google Maps API v2

ตัวอย่างและคำอธิบาย

- หลังจากสร้างโปรเจ็คแอนดรอยด์ใหม่แล้ว ก่อนอื่นต้องทำการอ้างอิง Library Project ของ Google Play services ที่เราอินพอร์ตเข้ามาใน Eclipse ก่อนหน้านี้ ให้คุณคลิกขวาที่ชื่อโปรเจ็คใน Package Explorer และเลือก Properties



- คลิกหัวข้อ Android ทางซ้าย
 - คลิก Add ในส่วนของ Library
 - เลือก google-play-services_lib และคลิก OK จะกลับมาที่หน้าจอ Properties ของ MapDemo และในกรอบ Library จะมีชื่อโปรเจ็ค google-play-services_lib เพิ่มเข้ามา
 - คลิก OK เพื่อปิดหน้าจอ Properties
- ต่อไปจะแก้ไขไฟล์ AndroidManifest.xml

- 6 เปิดไฟล์ AndroidManifest.xml และเพิ่มแท็กต่อไปนี้ไว้ก่อนแท็กปิด </application> (แท็กที่เพิ่มใหม่จะเป็น Child ของ <application>) โดยนำ API Key ที่ได้จาก Google APIs Console มาระบุ เป็นค่าของแอ็ตทริบิวต์ android:value

โปรเจก MapDemo, ไฟล์ AndroidManifest.xml

```
<meta-data
    android:name="com.google.android.maps.v2.API_KEY"
    android:value="API Key ของคุณ" />
```

- 7 เพิ่มแท็กต่อไปนี้ไว้ก่อนแท็ก </application>

โปรเจก MapDemo, ไฟล์ AndroidManifest.xml

```
<permission
    android:name="com.promlert.mapdemo.permission.MAPS_RECEIVE"
    android:protectionLevel="signature" />

<uses-permission
    android:name="com.promlert.mapdemo.permission.MAPS_RECEIVE" />

<uses-permission android:name="android.permission.INTERNET" /> ①
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" /> ②
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" /> ③
<uses-permission android:name=
    "com.google.android.providers.gsf.permission.READ_GSERVICES" /> ④

<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/> ⑤
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" /> ⑥

<uses-feature
    android:glEsVersion="0x00020000"
    android:required="true" /> ⑦
```

- ① ขอสิทธิ์การเข้าถึงอินเทอร์เน็ตเพื่อดาวน์โหลดข้อมูลแผนที่จากเซิร์ฟเวอร์ Google Maps
- ② ขอสิทธิ์การตรวจสอบสถานะการเชื่อมต่อ กับเซิร์ฟเวอร์ Google Maps
- ③ ขอสิทธิ์การแคลชข้อมูลแผนที่เก็บลง External Storage ของเครื่อง
- ④ ขอสิทธิ์การเข้าถึง web-based services ของภูเก็ต
- ⑤ ขอสิทธิ์การหาตำแหน่งของเครื่องโดยใช้ Wi-Fi หรือข้อมูลสัญญาณโทรศัพท์ (mobile cell data)
- ⑥ ขอสิทธิ์การหาตำแหน่งของเครื่องโดยใช้ GPS

- 7 เนื่องจาก Maps API ใช้ไลบรารี OpenGL ES เวอร์ชัน 2 ในการวาดรูปภาพแผนที่ เราจึงต้องระบุแท็กนี้เพื่อบอกให้บริการภายนอก เช่น Google Play Store รู้ เพื่อป้องกันไม่ให้แสดงซื่อแอบน็บนเครื่องที่ไม่สนับสนุน OpenGL ES เวอร์ชัน 2

NOTE»»

5 และ 6 นั้นไม่ต้องมีก็ได้ ถ้าหากแอพของคุณไม่ต้องการรูปตัวแทนของผู้ใช้

8 สังคลีนโปรเจค (เมนู Project ▶ Clean) เพื่อบวัดไฟล์ต่างๆของโปรเจคขึ้นมาใหม่

9 กำหนด Layout ของหน้าจอ

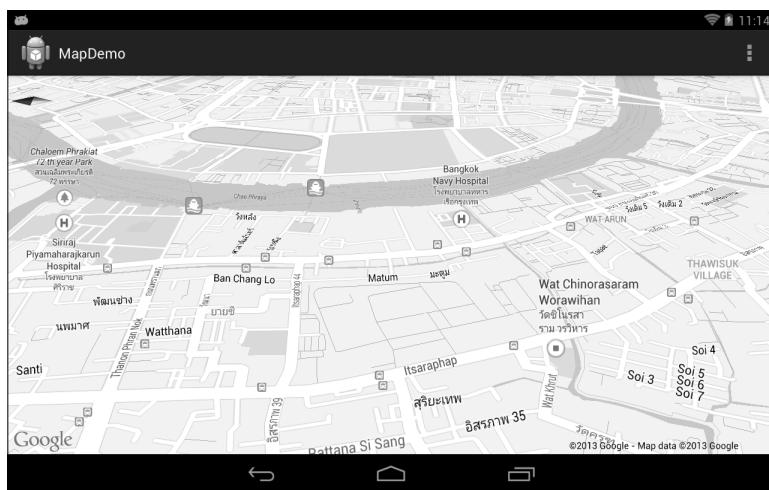
โปรเจค MapDemo, ไฟล์ activity_main.xml

```
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:map="http://schemas.android.com/apk/res-auto"
    android:id="@+id/map"
    android:name="com.google.android.gms.maps.MapFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    map:mapType="normal"
    map:cameraTargetLat="13.7500"
    map:cameraTargetLng="100.4833"
    map:cameraZoom="15"
    map:cameraBearing="90"
    map:cameraTilt="60"
    map:uiCompass="true"
    map:uiZoomControls="false"
    map:uiRotateGestures="true"
    map:uiScrollGestures="false"
    map:uiTiltGestures="true"
    map:uiZoomGestures="false" />
```

แท็ก `<fragment>` นี้ก็คือค้อนให้เรล MapFragment เนื่องจากกำหนดแอตทริบิวต์ `android:name` เป็น `com.google.android.gms.maps.MapFragment`

สำหรับแอตทริบิวต์ที่มีชื่อขึ้นต้นด้วย `map:` ใช้กำหนดสถานะเริ่มต้นของแผนที่ ซึ่งจะอธิบายต่อไป

ผลการรัน



NOTE>>

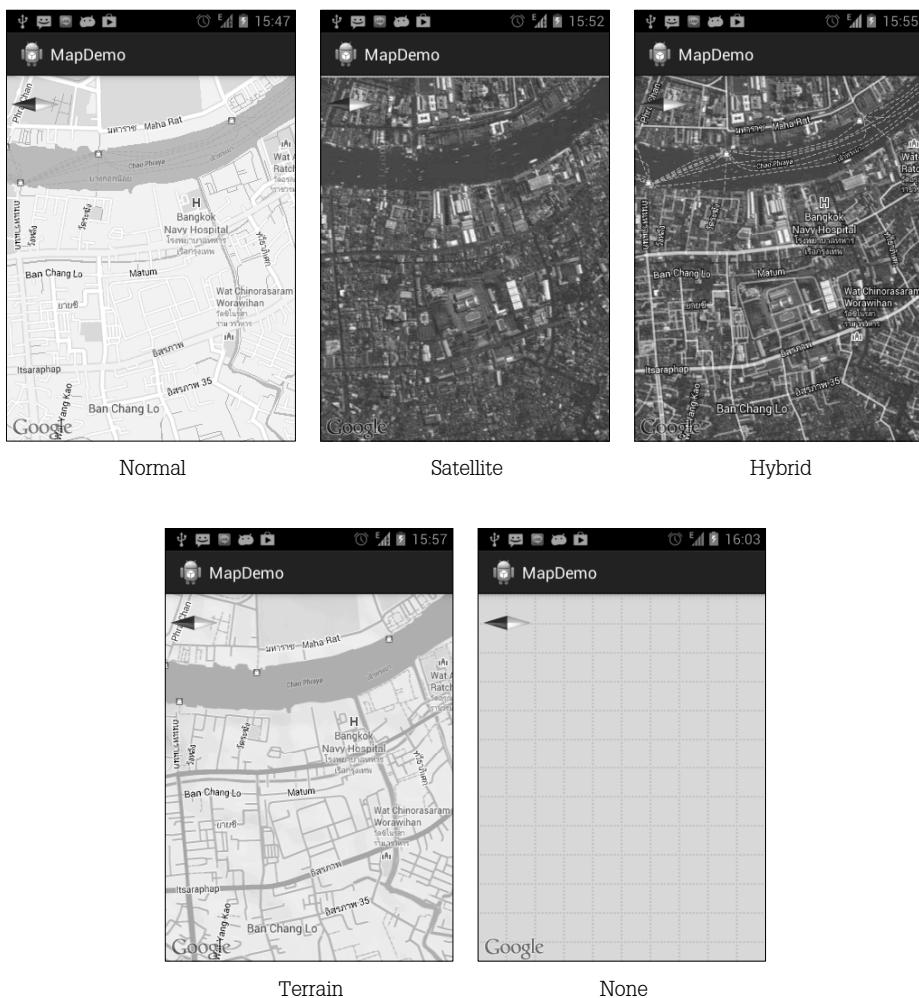
การใช้งาน MapFragment ต้องการแอนดรอยด์ 3.1 (API Level 12) ขึ้นไป แต่หากคุณต้องการรันแอปบนแอนดรอยด์ก่อน 3.1 ให้ใช้ SupportMapFragment ที่อยู่ใน Support Library แทน (ปกติ libraries ตัวนี้จะติดตั้งมาพร้อมกับ Android SDK อยู่แล้ว) ซึ่ง SupportMapFragment มีความสามารถเหมือนกับ MapFragment ทุกประการ

ธีมบายเพิ่มเติม

เอกสารที่บ่าว่าต์ของ MapFragment ที่ใช้กำหนดสถานะเริ่มต้นของแผนที่ มีความหมายดังนี้

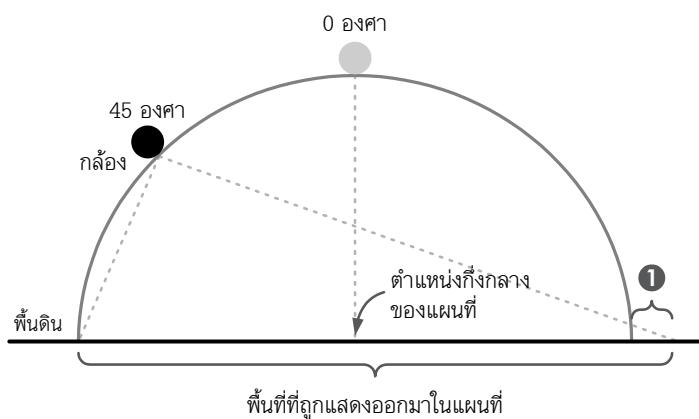
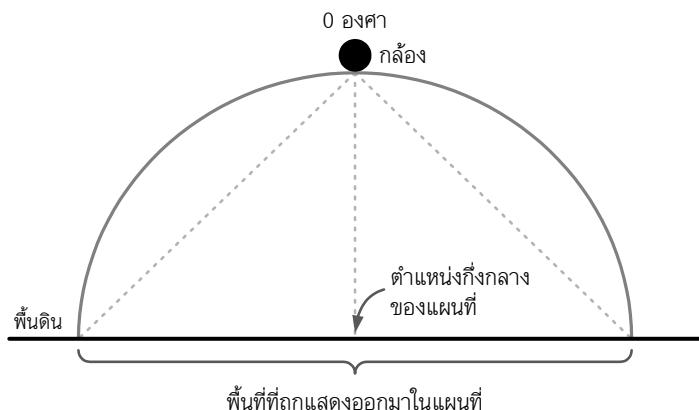
- ◆ **mapType**:mapType กำหนดรูปแบบของแผนที่ ซึ่งมี 5 รูปแบบคือ

Normal	แสดงเส้นถนนและลักษณะทางธรรมชาติที่สำคัญ เช่น แม่น้ำ
Satellite	แสดงภาพถ่ายดาวเทียม
Hybrid	แสดงภาพถ่ายดาวเทียมพร้อมเส้นถนน
Terrain	แสดงภาพที่บอกลักษณะภูมิประเทศ โดยใช้สีและเส้นชั้นความสูง (contour line) เป็นต้น นอกจากนี้อาจแสดงเส้นถนนด้วย
None	ไม่แสดงแผนที่ (map tiles) แต่จะแสดงเพียงเส้นตาราง (grid) ว่างๆ

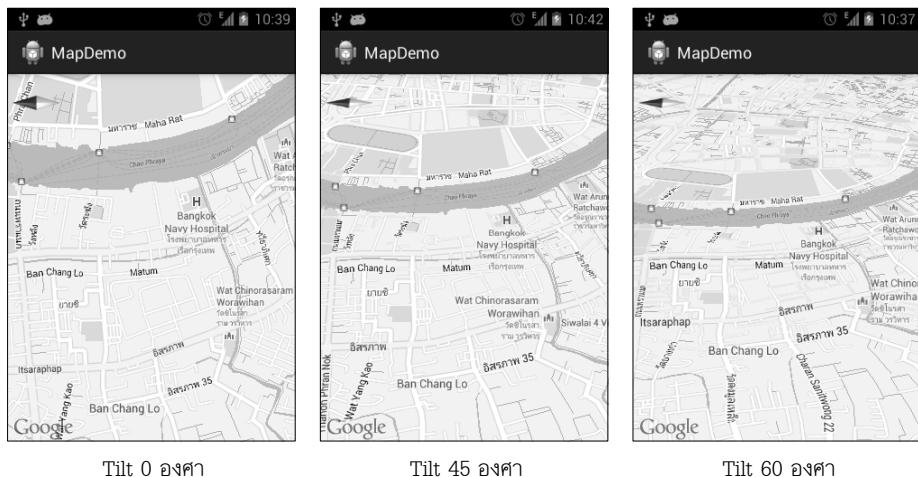


- ◆ `map:cameraTargetLat` และ `map:cameraTargetLng` กำหนดตำแหน่งเป้าหมายของแผนที่โดยระบุค่าละติจูดและลองจิจูดที่ต้องการให้แสดงที่จุดกึ่งกลางของแผนที่
- ◆ `map:cameraZoom` กำหนดระดับการซูมของกล้อง หรือค่ามาราส่วน (`scale`) ของแผนที่ โดยค่าiyึงมากจะยื่นรายละเอียดในแผนที่มาก (เหมือนชุมกล้องเข้าไปใกล้ๆ) และค่าiyึงน้อย จะทำให้เห็นพื้นที่ของโลกมากขึ้น แต่รายละเอียดน้อยลง (เหมือนชุมกล้องออกมาก) ทั้งนี้ค่า 0 จะแสดงภาพโลภทั้งใบที่มีความกว้างบนจอประมาณ 256dp หรือ 1.6 นิ้ว

- ◆ `map:cameraBearing` กำหนดองศาการหมุนของกล้อง หรือกึ่งคิวทิศทางของแผนที่ (orientation) โดยเทียบกับทิศเหนือเมื่อวัดตามเข็มนาฬิกา เช่น ในที่นี้กำหนดเป็น 90 จะทำให้เส้นแนวตั้งในแผนที่ซึ่งไปยังทิศตะวันออก (ทิศเหนือจะอยู่ทางซ้าย)
- ◆ `map:cameraTilt` กำหนดมุมในการมอง (viewing angle) ซึ่งหมายถึงตำแหน่งของกล้องบนเส้นครึ่งวงกลม ดังรูป โดยค่า 0 คือตำแหน่งบนสุดที่มองลงมาจั่งพื้นตรงๆ, ค่า 90 คือตำแหน่งที่พื้น และค่า 45 คือระยะครึ่งหนึ่งบนเส้นครึ่งวงกลมระหว่างข้างบนสุดกับพื้น เป็นต้น



การทำหน้า `cameraTilt` เป็นค่าอื่นที่ไม่ใช่ 0 จะทำให้ได้ภาพแผนที่ในลักษณะ Perspective (สิ่งที่อยู่ใกล้กับกล้องจะมีขนาดใหญ่กว่าสิ่งที่อยู่ไกลออกไป) โดยกล้องยังคงเล็งไปที่ตำแหน่งกึ่งกลางแผนที่เช่นเดียวกับที่มุ่ง 0 องศา แต่พื้นที่ด้านใกล้จะมีส่วนที่มองเห็นเพิ่มขึ้นมา (❶)



Tilt 0 องศา

Tilt 45 องศา

Tilt 60 องศา

- ◆ `map:uiCompass` กำหนดให้แสดงหรือไม่แสดงเข็มทิศในแผนที่
- ◆ `map:uiZoomControls` กำหนดให้แสดงหรือไม่แสดงปุ่มควบคุมการซูมกล้องในแผนที่
- ◆ `map:uiRotateGestures` กำหนดให้ใช้ Rotate Gesture ได้หรือไม่ได้ ซึ่ง Rotate Gesture คือการใช้นิ้ว 2 นิ้วแตะบนหน้าจอเลี้ยวหมุนเพื่อหมุนกล้อง
- ◆ `map:uiScrollGestures` กำหนดให้ใช้ Scroll Gesture ได้หรือไม่ได้ ซึ่ง Scroll Gesture คือ การใช้นิ้วปัดเพื่อเลื่อนแผนที่ (เลื่อนกล้องหรือแพนกล้อง)
- ◆ `map:uiTiltGestures` กำหนดให้ใช้ Tilt Gesture ได้หรือไม่ได้ ซึ่ง Tilt Gesture คือ การใช้นิ้ว 2 นิ้วปัดซึ้งลงในแนวตั้งของแผนที่เพื่อเอียงกล้อง (เปลี่ยน viewing angle ของกล้อง ดังที่อธิบายแล้ว)
- ◆ `map:uiZoomGestures` กำหนดให้ใช้ Zoom Gesture ได้หรือไม่ได้ ซึ่ง Zoom Gesture คือ การใช้นิ้วแตะเร็วๆ 2 ครั้ง (double tap) หรือการจับ 2 นิ้วเพื่อซูมเข้า/ออก

การเปลี่ยนตำแหน่งและมุมมองของแพนท์

ตอนเริ่มต้นเราอาจแสดงแผนที่ ณ ตำแหน่งหนึ่ง แต่หลังจากนั้นอาจย้ายไปแสดงตำแหน่งอื่น หรือเปลี่ยนมุมมอง เช่น ระดับการซูม ทิศทาง หรือการอธิบาย เป็นต้น ซึ่งทั้งหมดนี้จะทำได้โดยแก้ไขค่าต่างๆ ของกล้อง

ตัวอย่างและคำอธิบาย

ตัวอย่างนี้จะทำต่อจากตัวอย่างที่แล้ว โดยเพิ่มคำสั่ง 2 คำสั่งในเมนู เพื่อให้แสดงแผนที่ของจังหวัดเชียงใหม่และภูเก็ต

- สร้างไฟล์ main.xml ในโฟลเดอร์ res\menu และพิมพ์โค้ด XML ดังนี้

โปรเจก MapDemo, ไฟล์ res\menu\main.xml

```
<menu xmlns:android="http://schemas.android.com/apk/res/android" >

    <item
        android:id="@+id/chiangmai_menu"
        android:showAsAction="never"
        android:title="Chiangmai"/>
    <item
        android:id="@+id/phuket_menu"
        android:showAsAction="never"
        android:title="Phuket"/>

</menu>
```

- ที่แอคทิวิตี้ ให้เพิ่มค่าคงที่และตัวแปรคลาส

โปรเจก MapDemo, ไฟล์ MainActivity.java

```
private static final LatLng CHIANGMAI = new LatLng(18.701224, 98.789770);
private static final LatLng PHUKET = new LatLng(7.966598, 98.359929);
private GoogleMap map;
```

ค่าคงที่ CHIANGMAI และ PHUKET จะเก็บพิกัดละติจูด/ลองจิจูดของจังหวัดเชียงใหม่และภูเก็ต ส่วนตัวแปร map จะเอาไว้เก็บออบเจ็ค GoogleMap ซึ่งเป็นตัวแทนของแผนที่ Google Maps ใน MapFragment

- เพิ่มโค้ดในเมธอด onCreate ของแอคทิวิตี้

โปรเจก MapDemo, ไฟล์ MainActivity.java

```
MapFragment mapFragment = (MapFragment) getSupportFragmentManager()
    .findFragmentById(R.id.map); ①
map = mapFragment.getMap(); ②
```

เราใช้เมธอด `findFragmentById` เข้าถึง `MapFragment` ใน Layout ❶ จากนั้นจึงใช้เมธอด `getMap` เข้าถึงออบเจ็ค `GoogleMap` ใน `MapFragment` อีกที ❷ ซึ่งการดำเนินการต่างๆกับแผนที่ จะกระทำผ่านออบเจ็ค `GoogleMap` นี้ (ตัวแปร `map`)

4 เพิ่มเมธอด `onCreateOptionsMenu` และเมธอด `onOptionsItemSelected` ในแอคทิวิตี้

โปรเจ็ค `MapDemo`, ไฟล์ `MainActivity.java`

```
@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
    MenuInflater inflater = getMenuInflater();  
    inflater.inflate(R.menu.main, menu);  
    return true;  
}  
  
@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
    switch (item.getItemId()) {  
  
        case R.id.chiangmai_menu:  
            // ย้ายกล้องไปที่จังหวัดเชียงใหม่ และกำหนดการซูมระดับ 15  
            map.moveCamera(CameraUpdateFactory.newLatLngZoom(CHIANGMAI, 15));  
  
            // ซูมกล้องแบบมีแอนิเมชั่นมาที่ระดับ 10 ในเวลา 2 วินาที  
            map.animateCamera(CameraUpdateFactory.zoomTo(10), 2000, null);  
  
            return true;  
  
        case R.id.phuket_menu:  
            // สร้างออบเจ็ค CameraPosition สำหรับระบุตำแหน่งและมุมมองของกล้อง  
            CameraPosition cameraPosition = new CameraPosition.Builder()  
                .target(PHUKET) // ย้ายกล้องไปที่จังหวัดภูเก็ต  
                .zoom(15) // กำหนดการซูมระดับ 15  
                .bearing(270) // หันกล้องไปทางทิศตะวันตก  
                .tilt(30) // เอียงกล้อง 30 องศา  
                .build(); // สร้าง CameraPosition ขึ้นมาจาก Builder  
  
            // ย้ายกล้องและปรับมุมมองของกล้องตามที่ระบุในออบเจ็ค CameraPosition  
            map.animateCamera(CameraUpdateFactory.newCameraPosition(  
                cameraPosition));  
  
            return true;  
  
        default:  
            return super.onOptionsItemSelected(item);  
    }  
}
```

ขอบเจ๊ค GoogleMap มีเมธอดที่ใช้ปรับเปลี่ยนตำแหน่งและ/หรือมุมมองของแผนที่ (กล้อง) อยู่ 2 เมธอด คือ

- ◆ moveCamera เปลี่ยนตำแหน่งและ/หรือมุมมองไปตามออบเจ็ค CameraUpdate ที่ระบุทันที
- ◆ animateCamera เปลี่ยนตำแหน่งและ/หรือมุมมองไปตามออบเจ็ค CameraUpdate ที่ระบุ โดยแสดงแอนิเมชั่นของการเปลี่ยนแปลงด้วย

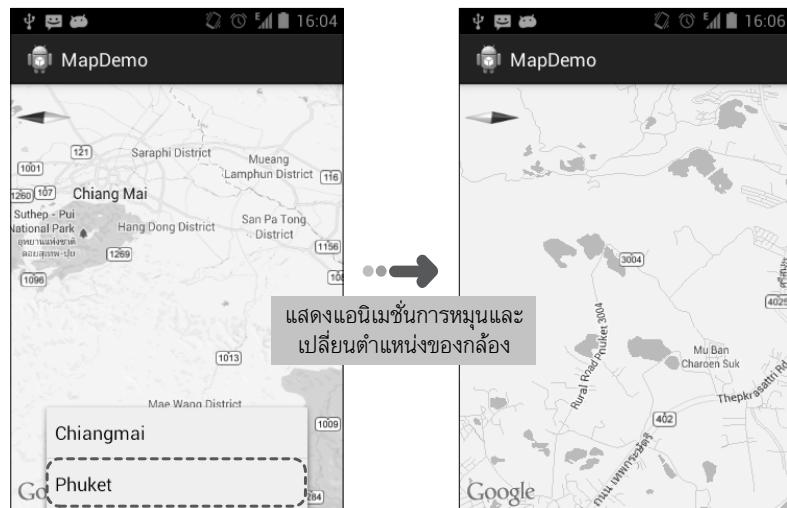
สำหรับการสร้างออบเจ็ค CameraUpdate ที่กำหนดตำแหน่งหรือมุมมองใหม่นั้นจะใช้เมธอดต่างๆ ของคลาส CameraUpdateFactory เช่น

- ◆ เมธอด newLatLng ระบุตำแหน่งใหม่
- ◆ เมธอด newLatLngZoom ระบุตำแหน่งและระดับการซูมใหม่
- ◆ เมธอด zoomTo เปลี่ยนระดับการซูมไปเป็นค่าที่ระบุ
- ◆ เมธอด newCameraPosition เปลี่ยนตำแหน่งและมุมมองไปตามออบเจ็ค CameraPosition ที่ระบุ

โดยสรุปเกือบ การเปลี่ยนตำแหน่งและ/หรือมุมมองของแผนที่ อันดับแรกจะต้องเรียกเมธอดของ CameraUpdateFactory เพื่อสร้างออบเจ็ค CameraUpdate ขึ้นก่อน จากนั้นนำออบเจ็ค CameraUpdate ไปประบู๊ฟให้เมธอด moveCamera หรือ animateCamera ขึ้นอยู่กับว่าต้องการให้เปลี่ยนทันทีหรือเปลี่ยนแบบมีแอนิเมชั่น

ผลการรับ





การเพิ่มหมุด (Marker) ลงบนแผนที่

หมุดหรือ Marker คือสิ่งที่ใช้ระบุตำแหน่งในแผนที่ ซึ่ง Google Maps Android API v2 ไม่เพียงให้เราเพิ่มหมุดลงในแผนที่ได้เท่านั้น แต่ยังให้เราสามารถเปลี่ยนสีหมุด, กำหนดภาพหมุดโดยใช้รูปภาพของเราเอง และรายละเอียดอื่นๆได้ด้วย

เมื่อต้องการเพิ่มหมุด ให้สร้างออบเจ็ค `MarkerOptions` โดยกำหนดคุณสมบัติต่างๆของหมุดตามต้องการ แล้วจึงเรียกเมธอด `addMarker` ของออบเจ็ค `GoogleMap` โดยระบุออบเจ็ค `MarkerOptions` นั้น เป็นพารามิเตอร์ เช่น

```
MarkerOptions options = new MarkerOptions();
options.position(new LatLng(0, 0));
options.title("This is a title");
map.addMarker(options);
```

หรือเขียนแบบนี้ก็ได้เช่นกัน

```
map.addMarker(new MarkerOptions()
    .position(new LatLng(0, 0))
    .title("This is a title")
);
```

คุณสมบัติที่สำคัญของหมุด

- ◆ **Position** กำหนดตำแหน่งของหมุดบนแผนที่ โดยกำหนดด้วยออบเจ็ค LatLng
- ◆ **Title** กำหนดข้อความที่แสดงใน Info window ซึ่งเป็นหน้าต่างที่จะปรากฏขึ้นเมื่อผู้ใช้คลิก (แตะ) ที่หมุด
- ◆ **Snippet** กำหนดข้อความเพิ่มเติมที่แสดงใน Info window
- ◆ **Icon** กำหนดภาพไอคอนของหมุด
- ◆ **Draggable** กำหนดว่าผู้ใช้สามารถย้ายหมุดได้หรือไม่ ดีฟอลต์คือ false (ย้ายไม่ได้)
- ◆ **Visible** กำหนดให้แสดงหรือซ่อนหมุด ดีฟอลต์คือ true (แสดงหมุด)

ตัวอย่างและคำอธิบาย

ตัวอย่างนี้จะทำการตั้งค่าให้หมุด Chonburi สามารถแสดงรายละเอียดของหมุดได้ ต้องเพิ่มคำสั่งในไฟล์ main.xml ของแอปพลิเคชัน ดังนี้

1 เพิ่มคำสั่ง Chonburi ในเมนูอีกคำสั่งหนึ่ง

```
โปรเจ็ค MapDemo, ไฟล์ res\menu\main.xml
<item
    android:id="@+id/chonburi_menu"
    android:showAsAction="never"
    android:title="Chonburi"/>
```

2 เพิ่มค่าสตริงในไฟล์ strings.xml ซึ่งเป็นข้อความที่เราจะแสดงใน Info window ของหมุดแต่ละอัน

```
โปรเจ็ค MapDemo, ไฟล์ res\value\strings.xml
<string name="chonburi_title">ชลบุรี</string>
<string name="chonburi_snippet">อำเภอเมือง จังหวัดชลบุรี</string>

<string name="bangsaen_title">บางแสน</string>
<string name="bangsaen_snippet">ตำบลแสนสุข อำเภอเมือง จังหวัดชลบุรี</string>

<string name="kk_zoo_title">สวนสัตว์เปิดเขาเขียว</string>
<string name="kk_zoo_snippet">เปิด 8.00-18.00 น., โทร. 038-318-444</string>

<string name="tg_zoo_title">สวนเสือศรีราชา</string>
<string name="tg_zoo_snippet">เปิด 8.00-18.00 น., โทร. 038-296-556</string>
```

- 3 ที่แอคทิวิตี้ ให้เพิ่มค่าคงที่ระดับคลาส ซึ่งเป็นพิกัดของสถานที่ต่างๆที่เราจะแสดงหมุดในแผนที่

โปรเจ็ค MapDemo, ไฟล์ MainActivity.java

```
private static final LatLng CHONBURI = new LatLng(13.36114, 100.98467);
private static final LatLng BANGSAEN_BEACH =
        new LatLng(13.29466, 100.90582);
private static final LatLng KHAO_KHEOW_OPEN_ZOO =
        new LatLng(13.21498, 101.05598);
private static final LatLng SIRACHA_TIGER_ZOO =
        new LatLng(13.14873, 101.01256);
```

- 4 เพิ่มโค้ดในเมธอด onOptionsItemSelected เพื่อจัดการกรณีเลือกคำสั่ง Chonburi จากเมนู

โปรเจ็ค MapDemo, ไฟล์ MainActivity.java

```
case R.id.chonburi_menu:
    CameraPosition cameraChonburi = new CameraPosition.Builder()
        .target(CHONBURI)
        .zoom(10)
        .bearing(0)
        .tilt(0)
        .build();

    map.moveCamera(CameraUpdateFactory.newCameraPosition(cameraChonburi));

    addMarker();
    return true;
```

เมื่อผู้ใช้เลือกคำสั่ง Chonburi เราจะเลื่อนกล้องไปที่อำเภอเมืองชลบุรี แล้วเพิ่มหมุดลงในแผนที่

- 5 เพิ่มเมธอด addMarker ในแอคทิวิตี้

โปรเจ็ค MapDemo, ไฟล์ MainActivity.java

```
private void addMarker() {
    // เพิ่มหมุดที่ อ.เมือง จ.ชลบุรี
    map.addMarker(new MarkerOptions()
        .position(CHONBURI)
        .title(getString(R.string.chonburi_title))
        .snippet(getString(R.string.chonburi_snippet))
    );

    // เพิ่มหมุดที่บางแสน และเปลี่ยนลักษณะเป็นลีเชียว
    map.addMarker(new MarkerOptions()
        .position(BANGSAEN_BEACH)
        .title(getString(R.string.bangsaen_title))
        .snippet(getString(R.string.bangsaen_snippet))
```

```
        .icon(BitmapDescriptorFactory.defaultMarker(
            BitmapDescriptorFactory.HUE_GREEN))
    );

    // เพิ่มหมุดที่สวนสัตว์เปิดเขาเขียว และให้ใช้ภาพ kk_zoo_logo.png เป็นหมุด
    map.addMarker(new MarkerOptions()
        .position(KHAO_KHEOW_OPEN_ZOO)
        .title(getString(R.string.kk_zoo_title))
        .snippet(getString(
            R.string.kk_zoo_snippet))
        .icon(BitmapDescriptorFactory
            .fromResource(R.drawable.kk_zoo_logo)))
    );

    // เพิ่มหมุดที่สวนเสือศรีราชา และเปลี่ยนลักษณะเป็นลีน้ำเงิน
    map.addMarker(new MarkerOptions()
        .position(SIRACHA_TIGER_ZOO)
        .title(getString(R.string.tg_zoo_title))
        .snippet(getString(R.string.tg_zoo_snippet))
        .icon(BitmapDescriptorFactory.defaultMarker(
            BitmapDescriptorFactory.HUE_BLUE)))
    );
}
```

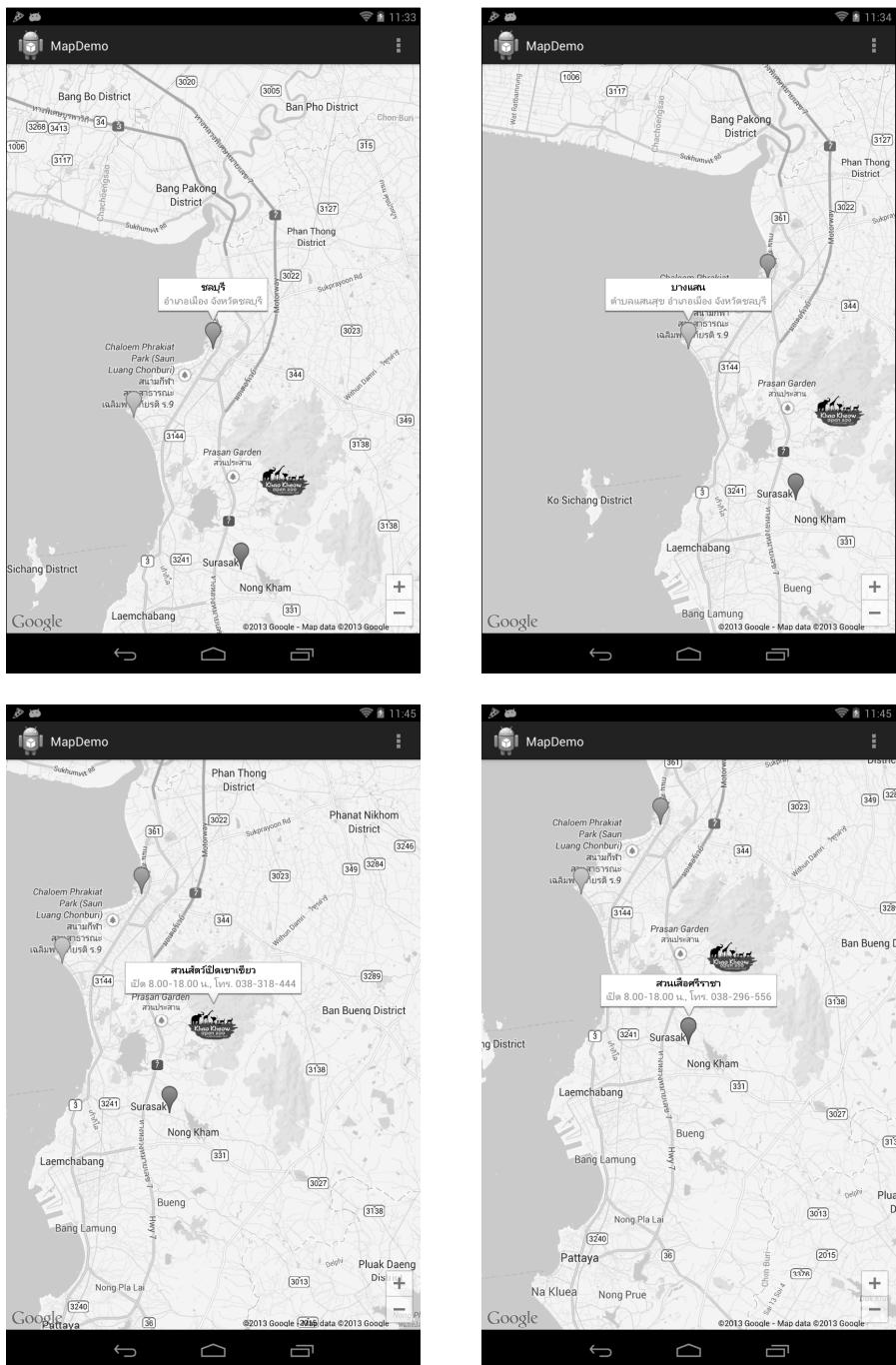


kk_zoo_logo.png

ผลการรัน

เมื่อรันแล้ว ให้เลือกคำสั่ง Chonburi จากเมนู กล้องจะเลื่อนไปที่จังหวัดชลบุรี โดยมีอำเภอเมือง ชลบุรีอยู่กึ่งกลางแผนที่ และหมุดสีแดงปักที่ตำแหน่งนี้ นอกจากนี้จะมีหมุดสีเขียวปักที่บางแสน หมุดสีน้ำเงินปักที่สวนเสือศรีราชา และหมุดรูปภาพ kk_zoo_logo.png ปักที่สวนสัตว์เปิดเขาเขียว

หลังจากนั้นให้คลิกที่หมุดแต่ละอันจะปรากฏ Info window ที่แสดงข้อมูลเพิ่มเติม และตำแหน่งของหมุดนั้นจะขยับมาอยู่กึ่งกลางหน้าจอแทน ดังรูปหน้าถัดไป



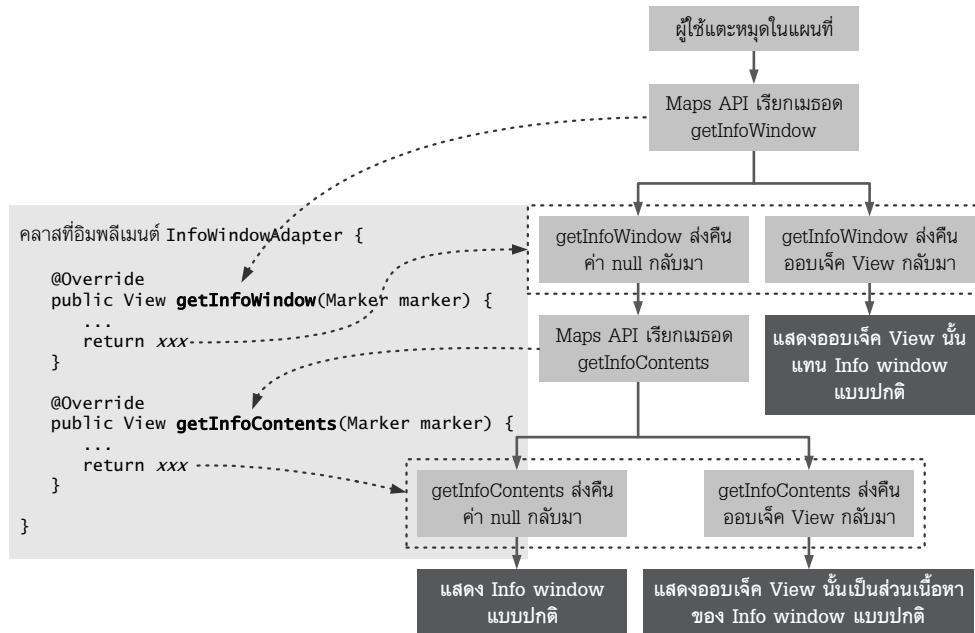
การปรับแต่ง Info Window ของหมุด

เมื่อผู้ใช้แตะหมุดจะปรากฏ Info window ซึ่งแสดงข้อความ Title และ Snippet ตามรูปแบบที่ Maps API กำหนดไว้ อย่างไรก็ตาม เราสามารถปรับแต่งหน้าตาของ Info window ให้น่าสนใจยิ่งขึ้น (เช่น การแสดงรูปภาพ หรือพื้นหลังที่มีสีสัน) โดยใช้ Layout ที่เราออกแบบเองได้

หลักการคือ ให้สร้างคลาสที่ Implement อินเทอร์เฟซ **InfoWindowAdapter** แล้ว Override เมธอด **getInfoWindow** และเมธอด **getInfoContents**

- ◆ เมธอด **getInfoWindow** ใช้กรณีที่เราต้องการกำหนด Layout ใหม่แทน Info window เดิม ทั้งอัน
- ◆ เมธอด **getInfoContents** ใช้กรณีที่เราต้องการกำหนด Layout ใหม่แทนที่ส่วนเนื้อหาของ Info window เดิม แต่ยังคงใช้กรอบและพื้นหลังของ Info window เดิม

Maps API จะเรียกมาใช้เมธอด **getInfoWindow** ก่อน แต่หากเราส่งคืนค่า null ก็จะมาเรียก **getInfoContents** ต่อไป และหากเราส่งคืนค่า null อีก ก็จะแสดง Info window ในรูปแบบปกติ ดังรายละเอียดในแผนผังต่อไปนี้



ตัวอย่างและคำอธิบาย

เราจะปรับปรุงตัวอย่างที่แล้วให้แสดง Info window ในแบบของเราเองสำหรับหมุด “สวนสีอุตราราชา” และหมุด “สวนสัตว์เปิดเขาเขียว” โดยหมุด “สวนสีอุตราราชา” นั้นจะปรับแต่งเฉพาะส่วนเนื้อหาของ Info window ในขณะที่หมุด “สวนสัตว์เปิดเขาเขียว” จะแทนที่ Info window เดิมทั้งอัน

- สร้าง Layout File ชื่อ info_window.xml แล้วพิมพ์โค้ดดังนี้

ไปริจิค MapDemo, ไฟล์ res\layout\info_window.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/layout"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal" >

    <ImageView
        android:id="@+id/icon"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_vertical"
        android:padding="3dp" />

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical" >

        <TextView
            android:id="@+id/title"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="25sp"
            android:textStyle="bold" />

        <TextView
            android:id="@+id/snippet"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="15sp" />
    </LinearLayout>
</LinearLayout>
```

- 2 สร้างไฟล์ myshape.xml ที่โฟลเดอร์ res\drawable แล้วพิมพ์โค้ดดังนี้ (ถ้ายังไม่มีโฟลเดอร์ drawable ให้สร้างขึ้นมาก่อน)

โปรเจ็ค MapDemo, ไฟล์ res\drawable\myshape.xml

```
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle" >

    <gradient
        android:endColor="#003300"
        android:gradientRadius="300"
        android:startColor="#ffffffff"
        android:type="radial"
        android:useLevel="false" />

    <corners
        android:bottomLeftRadius="8dp"
        android:bottomRightRadius="8dp"
        android:topLeftRadius="8dp"
        android:topRightRadius="8dp" />

    <padding
        android:bottom="4dp"
        android:left="4dp"
        android:right="4dp"
        android:top="4dp" />

</shape>
```

ไฟล์นี้คือ Shape Drawable (ดูบทที่ 10) ที่จะนำมากำหนดเป็นพื้นหลังของ Info window ของหมุด “สวนสัตว์เปิดเขาเขียว” เพราะสำหรับหมุดนี้ เราจะแทนที่ Info window เดิมทั้งอันด้วย Layout ของเราเอง

- 3 กำหนดให้แอคทิวิตี้ทำการ Implement อินเทอร์เฟซ InfoWindowAdapter เพื่อที่เราจะเพิ่มเมธอด getInfoWindow และ getInfoContents ไว้ในแอคทิวิตี้เลย

โปรเจ็ค MapDemo, ไฟล์ MainActivity.java

```
public class MainActivity extends Activity implements InfoWindowAdapter {
    ...
}
```

- 4 เพิ่มเมธอด getInfoWindow และ getInfoContents ในแอคทิวิตี้

โปรเจ็ค MapDemo, ไฟล์ MainActivity.java

```
@Override
public View getInfoContents(Marker marker) {
    // ตรวจสอบว่าใช่หมุด “สวนเลือเครื่อง” หรือไม่
```

```

if (marker.getTitle().equals(getString(R.string.tg_zoo_title))) {
    // ทำการ Inflate Layout ให้เป็นออบเจ็ค View
    LayoutInflater inflater = getLayoutInflater();
    View infoWindow = inflater.inflate(R.layout.info_window, null);
    // กำหนดภาพไอคอนใน Layout โดยใช้ไฟล์ภาพ tg_zoo_logo.png
    ImageView icon = (ImageView) infoWindow.findViewById(R.id.icon);
    icon.setImageResource(R.drawable.tg_zoo_logo);

    // กำหนดชื่อความ Title ใน Layout โดยใช้ Title ของหมุด
    TextView tv = (TextView) infoWindow.findViewById(R.id.title);
    tv.setText(marker.getTitle());

    // กำหนดชื่อความ Snippet ใน Layout โดยใช้ Snippet ของหมุด
    tv = (TextView) infoWindow.findViewById(R.id.snippet);
    tv.setText(marker.getSnippet());

    return infoWindow;
} else {
    return null; // สำหรับหมุดอื่นๆที่ไม่ใช่ “สวนเสือศรีราชา”
}
}

@Override
public View getInfoWindow(Marker marker) {
    // ตรวจสอบว่าใช่หมุด “สวนสัตว์เปิดเขาเขียว” หรือไม่
    if (marker.getTitle().equals(getString(R.string.kk_zoo_title))) {
        // ทำการ Inflate Layout ให้เป็นออบเจ็ค View
        LayoutInflater inflater = getLayoutInflater();
        View infoWindow = inflater.inflate(R.layout.info_window, null);

        // กำหนดพื้นหลังของ Layout โดยใช้ Shape Drawable (ไฟล์ myshape.xml)
        LinearLayout layout = (LinearLayout) infoWindow.findViewById(
            R.id.layout);
        layout.setBackgroundResource(R.drawable.myshape);

        // กำหนดภาพไอคอนใน Layout โดยใช้ไฟล์ภาพ kk_zoo_logo.png
        ImageView icon = (ImageView) infoWindow.findViewById(R.id.icon);
        icon.setImageResource(R.drawable.kk_zoo_logo);

        // กำหนดชื่อความ Title ใน Layout โดยใช้ Title ของหมุด
        TextView tv = (TextView) infoWindow.findViewById(R.id.title);
        tv.setText(marker.getTitle());

        // กำหนดชื่อความ Snippet ใน Layout โดยใช้ Snippet ของหมุด
        tv = (TextView) infoWindow.findViewById(R.id.snippet);
    }
}

```



tg_zoo_logo.png

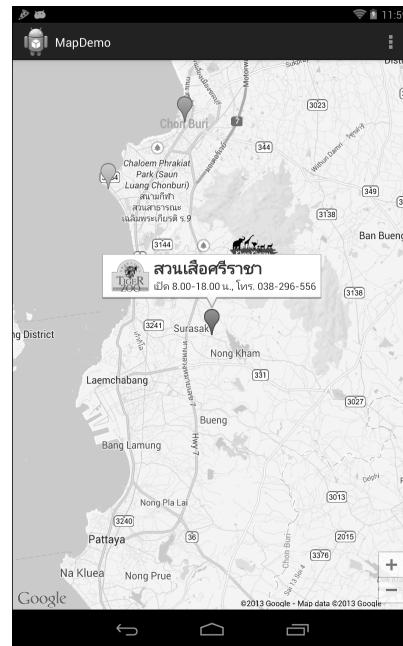
```
tv.setText(marker.getSnippet());  
  
        return infoWindow;  
    } else {  
        return null; // สำหรับหมุดอื่นๆไม่ใช่ “สวนสัตว์เปิดเขาเขียว”  
    }  
}
```

- 5 เพิ่มโค้ดในเมธอด addMarker เพื่อเชื่อมโยงการทำงานเมื่อผู้ใช้แตะหมุดเข้ากับเมธอดทั้งสองข้างต้น

โปรเจ็ค MapDemo, ไฟล์ MainActivity.java

```
map.setInfoWindowAdapter(this);
```

ผลการรัน

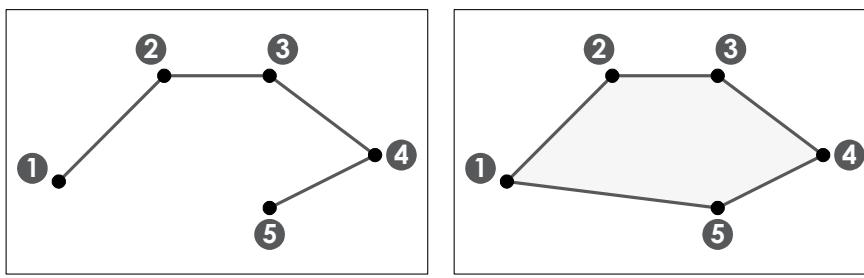


การวาดเส้นและรูปทรงลจบแนพบ์

การวาดเส้นและรูปทรงบนแผนที่จะทำให้เราสร้างแอพที่น่าสนใจได้มาก many ยกตัวอย่างเช่น เราอาจสร้างแอพที่บันทึกเส้นทางการเดินทางของผู้ใช้ (โดยใช้ Location API ที่จะอธิบายต่อไป) และ วาดเส้นทางนั้นออกมาในแผนที่ หรือเราราอาจดูรูปทรงเพื่อแสดงขอบเขตของสถานที่หรืออาณาบริเวณ หนึ่งๆในแผนที่ เป็นต้น

ความสามารถสรุปรูปทรงบนแผนที่ได้ 3 รูปแบบ

- ◆ **Polyline** คือเส้นตรงที่เชื่อมต่อจุดต่างๆตั้งแต่ 2 จุดขึ้นไป ซึ่งแต่ละจุดกำหนดด้วยออบเจ็ค **LatLng**
- ◆ **Polygon** คือรูปหลายเหลี่ยมที่เกิดจากการวาดเส้นตรงเชื่อมต่อจุดต่างๆเข้าด้วยกัน Polyline แต่จะขาดเส้นตรงจากจุดสุดท้ายมายังจุดแรกด้วย ทำให้เกิดเป็นรูปทรงปิด
- ◆ **Circle** คือรูปวงกลม



การสรุปต่างๆจะเขียนโค้ดในลักษณะเดียวกับการเพิ่มหมุด โดยก่อนอื่นให้สร้างออบเจ็ค **XxxOptions** เพื่อกำหนดคุณสมบัติของรูปที่จะวาด และส่งออบเจ็ค **XxxOptions** นั้นให้เมธอด **addXxx** ของออบเจ็ค **GoogleMap** เพื่อวาดอุปกรณ์บนแผนที่ เช่น

```
PolylineOptions line = new PolylineOptions()
    .add(
        new LatLng(0, 0),
        new LatLng(0, 1),
        new LatLng(1, 1),
        new LatLng(1, 0)) // วาดเส้นตรงจาก (0,0) ไปยัง (0,1), (1,1) และ (1,0) ตามลำดับ
    .width(3)           // กำหนดความหนาของเส้นเป็น 3 พิกเซล
    .color(Color.RED); // กำหนดสีของเส้นเป็นสีแดง

map.addPolyline(line);
```

ตัวอย่างและคำอธิบาย

เราจะทำต่อจากตัวอย่างที่แล้ว โดยเพิ่มคำสั่งในเมนูที่จะเลื่อนกล้องไปยังมหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตบางเขน แล้ววัดเส้นตรงต่อๆกัน (Polyline) ซึ่งแสดงเส้นทางจากประตูฝั่งถนนวิภาวดีไปยังอาคารสารนิเทศ 50 ปี และรูปหลากริม (Polygon) ที่แสดงสถานที่ต่ออย่างคร่าวๆของโรงเรียนสาขิตแห่งมหาวิทยาลัยเกษตรศาสตร์ ซึ่งตั้งอยู่ภายในมหาวิทยาลัย

- 1 เพิ่มคำสั่งในไฟล์ strings.xml

โปรเจ็ค MapDemo, ไฟล์ res\value\strings.xml

```
<string name="line_begin">ประตูฝั่งถนนวิภาวดี</string>
<string name="line_end">อาคารสารนิเทศ 50 ปี</string>
```

- 2 เพิ่มคำสั่ง Kasetsart University ในเมนูอีกคำสั่งหนึ่ง

โปรเจ็ค MapDemo, ไฟล์ res\menu\main.xml

```
<item
    android:id="@+id/kasetsart_menu"
    android:showAsAction="never"
    android:title="Kasetsart University"/>
```

- 3 ที่แอคทิวิตี้ ให้เพิ่มคำคงที่ระดับคลาส ซึ่งเป็นพิกัดของมหาวิทยาลัยเกษตรศาสตร์ บางเขน

โปรเจ็ค MapDemo, ไฟล์ MainActivity.java

```
private static final LatLng KASETSART = new LatLng(13.85187, 100.56752);
```

- 4 เพิ่มโค้ดในเมธอด onOptionsItemSelected เพื่อจัดการกรณีเลือกคำสั่ง Kasetsart University จากเมนู

โปรเจ็ค MapDemo, ไฟล์ MainActivity.java

```
case R.id.kasetsart_menu:
    CameraPosition cameraKasetsart = new CameraPosition.Builder()
        .target(KASETSART)
        .zoom(16)
        .bearing(0)
        .tilt(0)
        .build();

    map.moveCamera(CameraUpdateFactory.newCameraPosition(cameraKasetsart));

    drawOnMap();
    return true;
```

เมื่อผู้ใช้เลือกคำสั่ง Kasetsart University เราจะเลื่อนกล้องไปยังมหาวิทยาลัยเกษตรศาสตร์ที่ระดับการซูม 16 แล้วเรียกเมธอด drawOnMap เพื่อวัดเส้นและรูปทรงลงบนแผนที่

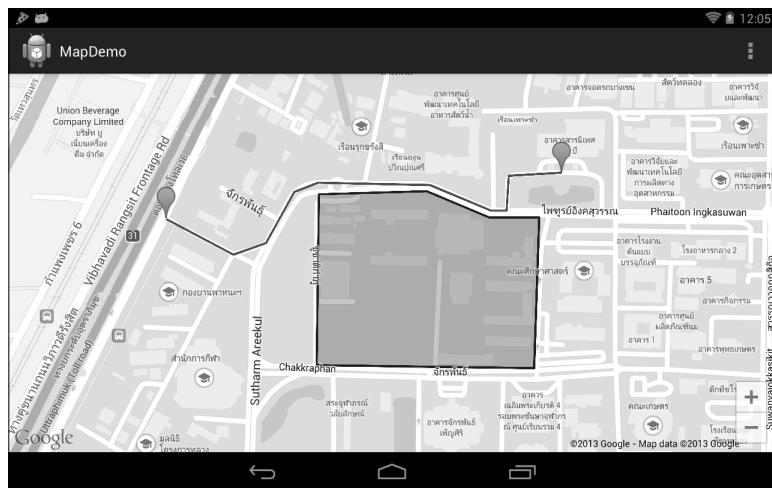
5 เพิ่มเมร์ด drawOnMap ในแอคทิวิตี้

โปรเจ็ค MapDemo, ไฟล์ MainActivity.java

```
private void drawOnMap() {
    // วาดเส้นตรงที่แสดงเส้นทางจากปะตูผ่านนวีกาดีไปยังอาคารสารนิเทศ 50 ปี
    PolylineOptions line = new PolylineOptions()
        .add(
            new LatLng(13.85243, 100.56449),
            new LatLng(13.85197, 100.56539),
            new LatLng(13.85212, 100.56583),
            new LatLng(13.85278, 100.56621),
            new LatLng(13.85290, 100.56653),
            new LatLng(13.85287, 100.56803),
            new LatLng(13.85254, 100.56885),
            new LatLng(13.85254, 100.56906),
            new LatLng(13.85299, 100.56909),
            new LatLng(13.85303, 100.56980))
        .width(3)          // ความหนาของเส้น
        .color(Color.RED); // สีเส้น
    map.addPolyline(line);

    // เพิ่มหมุดที่จุดเริ่มต้น (ปะตูผ่านนวีกาดี) และจุดสุดท้าย (อาคารสารนิเทศ 50 ปี) ของเส้นตรง
    map.addMarker(new MarkerOptions()
        .position(new LatLng(13.85243, 100.56449))
        .title(getString(R.string.line_begin)))
    ;
    map.addMarker(new MarkerOptions()
        .position(new LatLng(13.85303, 100.56980))
        .title(getString(R.string.line_end)))
    ;
    // วาดรูปหลายเหลี่ยมที่แสดงอาณาเขตของโรงเรียนสาวีดีแห่งมหาวิทยาลัยเกษตรศาสตร์
    PolygonOptions area = new PolygonOptions()
        .add(
            new LatLng(13.85275, 100.56654),
            new LatLng(13.85054, 100.56651),
            new LatLng(13.85050, 100.56943),
            new LatLng(13.85245, 100.56950),
            new LatLng(13.85245, 100.56883),
            new LatLng(13.85279, 100.56799))
        .strokeWidth(3)          // ความหนาของเส้น
        .strokeColor(Color.BLUE) // สีเส้น
        .fillColor(Color.argb(64, 0, 0, 255)); // สีภายในรูปหลายเหลี่ยม
    map.addPolygon(area);
}
```

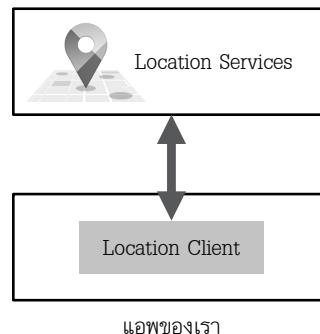
ผลการรัน



การหาตำแหน่งของผู้ใช้

แผนที่มักถูกใช้ร่วมกับการระบุตำแหน่งผู้ใช้ ซึ่งแอนดรอยด์มี Location API ไว้ให้เราทำงานเกี่ยวกับตำแหน่ง เช่น การหาตำแหน่งปัจจุบันของผู้ใช้ (ตำแหน่งของเครื่องแอนดรอยด์ที่ผู้ใช้ใช้งานอยู่), ติดตามการเปลี่ยนตำแหน่ง และอื่นๆ

แอนดรอยด์มี Location Services ซึ่งรับอยู่เบื้องหลัง และจะคอยตรวจสอบตำแหน่งปัจจุบันของผู้ใช้เป็นระยะๆ และส่งข้อมูลมา�ังแอพของเราโดยผ่านทาง Location Client (ออบเจ็ค LocationClient) ที่เราสร้างและเชื่อมต่อเข้ากับ Location Services ดังรูป



แอพของเรา

NOTE>>>

Location API ที่ใช้งานในบทนี้เป็นส่วนหนึ่งของ Google Play services SDK เช่นเดียวกับ Google Maps API v2 ดังนั้นคุณจะต้องติดตั้ง SDK ดังกล่าวจาก Android SDK Manager ก่อน ตามรายละเอียด ในหัวข้อแรกของบทนี้

ขั้นตอนการเขียนโค้ดเพื่อหาต่าฯหนังบังจุับของผู้ใช้

- 1 กำหนด Service Callback โดยสร้างคลาสที่ Implement อินเทอร์เฟซ `GooglePlayServicesClient.ConnectionCallbacks` และอินเทอร์เฟซ `GooglePlayServicesClient.OnConnectionFailedListener` แล้วทำการ Override เมธอด `onConnected`, `onDisconnected` และ `onConnectionFailed`

ตัวอย่างนี้จะกำหนดให้แอคทิวิตี้ Implement อินเทอร์เฟซทั้งสอง เพื่อกำหนดเมธอดที่จำเป็นไว้ในแอคทิวิตี้เลย

```
public class MainActivity extends Activity implements
    GooglePlayServicesClient.ConnectionCallbacks,
    GooglePlayServicesClient.OnConnectionFailedListener {

    @Override
    public void onConnected(Bundle data) {
        // ได้ดีที่จะทำงานเมื่อเชื่อมต่อ กับ Location Services
    }

    @Override
    public void onDisconnected() {
        // ได้ดีที่จะทำงานเมื่อยกเลิกการเชื่อมต่อ กับ Location Services
    }

    @Override
    public void onConnectionFailed(ConnectionResult result) {
        // ได้สำหรับข้อความการแจ้งเตือนต่อ Location Services ไม่สำเร็จ
    }
}
```

- 2 สร้าง Location Client พร้อมทั้งกำหนดออบเจ็คที่ทำหน้าที่จัดการ Callback ของอินเทอร์เฟซ ทั้งสอง (ในที่นี้ออบเจ็คดังกล่าวก็คือแอคทิวิตี้) โดยทว่าไปจะสร้าง Location Client ที่เมธอด `onCreate` ของแอคทิวิตี้

```
private LocationClient mLocationClient;

@Override
protected void onCreate(Bundle savedInstanceState) {
    ...
    /* paramiko เราก็ต้องมีการตั้งค่าอีกด้วย */
    mLocationClient = new LocationClient(this, this, this);
    ...
}
```

- 3 สั่ง Location Client ให้เชื่อมต่อและยกเลิกการเชื่อมต่อ กับ Location Services เมื่อถึงเวลาที่เหมาะสม ซึ่งเอกสารของแอนดรอยด์แนะนำให้เชื่อมต่อที่เมธอด onStart ของแอคทิวิตี้ และยกเลิกการเชื่อมต่อที่เมธอด onStop

```
@Override
protected void onStart() {
    super.onStart();
    mLocationClient.connect();
}

@Override
protected void onStop() {
    mLocationClient.disconnect();
    super.onStop();
}
```

Location Services จะค่อยตรวจสอบตำแหน่งของผู้ใช้เมื่อมีการเชื่อมต่อมาจาก Location Client (ถึงแม้ Location Client จะไม่ได้ขอตำแหน่งปัจจุบันก็ตาม) ดังนั้นเมื่อแอคทิวิตี้ของเรามาได้แสดงผลอยู่เบื้องหน้า (อยู่ในสถานะ Stop) เราจึงควรยกเลิกการเชื่อมต่อ กับ Location Services เพื่อประหยัดแบตเตอรี่

- 4 ขอตำแหน่งปัจจุบันจาก Location Services โดยใช้เมธอด getLastLocation ของ Location Client

```
private Location mCurrentLocation;
...
mCurrentLocation = mLocationClient.getLastLocation();
```

ตัวอย่าง

ตัวอย่างนี้จะมีปุ่มและแผนที่ (MapFragment) ซึ่งเมื่อคลิกปุ่มจะแสดงตำแหน่งปัจจุบันของผู้ใช้ที่ กีบกลางแผนที่ และปักหมุดที่ตำแหน่งนั้นด้วย

- 1 กำหนด Layout ของหน้าจอ

โปรเจ็ค LocationDemo, ไฟล์ activity_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:map="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <Button
        android:id="@+id/current_location_button"
        android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
        android:layout_margin="6dp"
        android:text="Current Location" />

<fragment
    android:id="@+id/map"
    android:name="com.google.android.gms.maps.MapFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    map:cameraBearing="0"
    map:cameraTargetLat="13.7500"
    map:cameraTargetLng="100.4833"
    map:cameraTilt="0"
    map:cameraZoom="15"
    map:mapType="normal"
    map:uiCompass="true"
    map:uiRotateGestures="true"
    map:uiScrollGestures="true"
    map:uiTiltGestures="true"
    map:uiZoomControls="false"
    map:uiZoomGestures="true" />

</LinearLayout>
```

2 เพิ่มโค้ดในแอคทิวิตี้จนเป็นตั้งนี้ (ขอลบบรรทัดการ import ต่างๆ)

โปรเจ็ค LocationDemo, ไฟล์ MainActivity.java

```
package com.example.locationdemo;

...
public class MainActivity extends Activity implements
        GooglePlayServicesClient.ConnectionCallbacks,
        GooglePlayServicesClient.OnConnectionFailedListener {

    private LocationClient mLocationClient; // Location Client
    private Location mCurrentLocation; // ตำแหน่งปัจจุบันของผู้ใช้
    private GoogleMap map; // แผนที่ Google Maps ใน MapFragment

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // เชื่อม MapFragment
        MapFragment mapFragment = (MapFragment) getSupportFragmentManager()
                .findFragmentById(R.id.map);
```

```
// เข้าถึงออบเจ็ค GoogleMap ใน MapFragment
map = mapFragment.getMap();

// สร้าง Location Client
mLocationClient = new LocationClient(this, this, this);

// ระบุการทำงานเมื่อปุ่มถูกคลิก
Button btnCurrentLocation = (Button) findViewById(
    R.id.current_location_button);
btnCurrentLocation.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // ตรวจสอบเพื่อความแน่ใจว่าเชื่อมต่อ Location Services แล้ว
        if (mLocationClient.isConnected()) {
            map.clear(); // ลบหมุดที่มีอยู่เดิม

            // ขอตำแหน่งปัจจุบันจาก Location Services
            mCurrentLocation = mLocationClient.getLastLocation();
            // หาพิกัดละติจูด/ลองจิจูด
            double lat = mCurrentLocation.getLatitude();
            double lng = mCurrentLocation.getLongitude();

            // สร้างออบเจ็ค LatLng ที่เก็บพิกัดละติจูด/ลองจิจูด
            LatLng currentLatLng = new LatLng(lat, lng);
            // ย้ายกล้องไปที่ตำแหน่งนั้นแบบมีแอนิเมชั่น และปักหมุด
            map.animateCamera(CameraUpdateFactory
                .newLatLng(currentLatLng));
            map.addMarker(new MarkerOptions()
                .position(currentLatLng)
            );

            // แสดง Toast บอกพิกัดของตำแหน่งปัจจุบัน
            String msg = "Current location: "
                + String.valueOf(lat) + ", "
                + String.valueOf(lng);
            Toast.makeText(MainActivity.this, msg,
                Toast.LENGTH_LONG).show();
        } else {
            String msg = "Can't connect to location services,
                try again later.";
            Toast.makeText(MainActivity.this, msg,
                Toast.LENGTH_LONG).show();
        }
    }
});
```

```
@Override
protected void onStart() {
    super.onStart();
    mLocationClient.connect(); // เชื่อมต่อ Location Services
}

@Override
protected void onStop() {
    mLocationClient.disconnect(); // ยกเลิกการเชื่อมต่อ Location Services
    super.onStop();
}

@Override
public void onConnected(Bundle data) {
    Toast.makeText(this, "Connected", Toast.LENGTH_SHORT).show();
}

@Override
public void onDisconnected() {
    Toast.makeText(this, "Disconnected", Toast.LENGTH_SHORT).show();
}

@Override
public void onConnectionFailed(ConnectionResult result) {
    Toast.makeText(this, "Connection Failed", Toast.LENGTH_SHORT)
        .show();
}
}
```

- 3 ตั้งค่าโปรเจ็คให้อ้างอิง Library Project ของ Google Play services และเพิ่มข้อมูลในไฟล์ AndroidManifest.xml ตามรายละเอียดในหัวข้อ “การแสดงแผนที่ Google Maps ด้วย MapFragment”
- 4 ใช้บราวเซอร์เข้าไปที่ Google APIs Console (code.google.com/apis/console) และเพิ่มข้อมูล SHA1 fingerprint คู่กับชื่อแพคเกจของแอพนี้ (ในกรณีของผู้เขียนคือ com.example.locationdemo) เข้าไปอีกชุดหนึ่ง โดยใช้ลิงค์ Edit allowed Android apps ทางขวา ตั้งรูปหน้าถัดไป

Google apis

API Project

- [Overview](#)
- [Services](#)
- [Team](#)
- API Access**
- [Reports](#)
- [Quotas](#)

API Access
To prevent abuse, Google places limits on API requests. Using a valid OAuth token or API key allows you to exceed anonymous limits by connecting requests back to your project.

Authorized API Access

OAuth 2.0 allows users to share specific data with you (for example, contact lists) while keeping their usernames, passwords, and other information private. A single project may contain up to 20 client IDs.

[Learn more](#)

Create an OAuth 2.0 client ID...

Simple API Access
Use API keys to identify your project when you do not need to access user data. [Learn more](#)

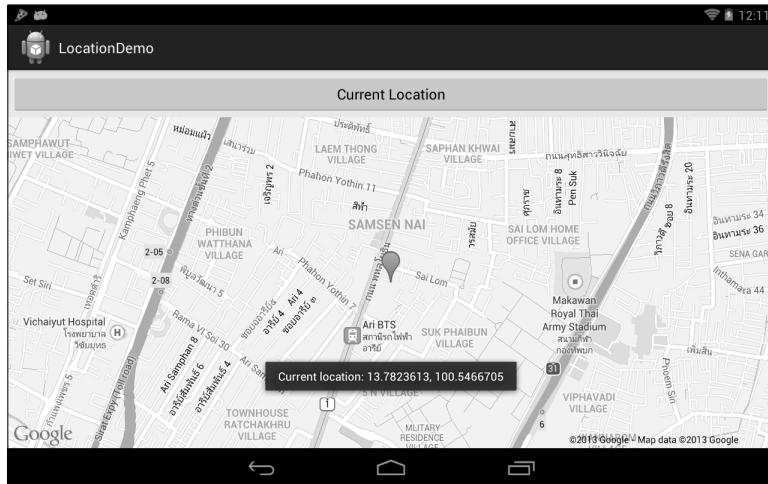
Key for Android apps (with certificates)	
API key:	A1ca5yDgzcR9fHtIXyUqdvNOhw9XfYe3tN82gb4
Android apps:	C0:GO:AO:C9:FC:6D:F5:03:91:09:0E:61:EC:0D:D4:15: 38:40:FI:C1:com.example.mapdemo C3:60:AO:C9:FC:6B:F5:03:91:89:8E:61:EC:8B:B4:15: 38:40:FI:C1:com.example.locationondemo
Activated on:	Jun 17, 2013 12:22 AM
Activated by:	promlert@gmail.com - you

[Generate new key...](#)
[Edit allowed Android app...](#)
[Delete key...](#)

ถ้าหากไม่เพิ่มข้อมูลนี้ เวลารันแอพจะไม่ปรากฏแผนที่ เนื่องจากกูเกิลจะอนุญาตเฉพาะแอพและค่า SHA1 fingerprint ตามที่ระบุใน Google APIs Console เท่านั้นที่สามารถเข้าถึงบริการแผนที่ Google Maps ได้ การเพิ่มข้อมูลนี้ไม่เกี่ยวข้องกับการใช้งาน Location API แต่อย่างใด

ผลการรัน

เมื่อรันแอพแล้วให้คลิกปุ่ม Current Location



การติดตามการเปลี่ยนแปลงตำแหน่งของผู้ใช้

สามารถส่งคำขอ (request) ไปยัง Location Services เพื่อให้แจ้งตำแหน่งของผู้ใช้ทุกๆ 5 วินาที ของเราราบุกทุกๆ ช่วงเวลาที่กำหนดได้

ขั้นตอนการเขียนโค้ดเพื่อติดตามการเปลี่ยนแปลงตำแหน่งของผู้ใช้

- 1 ทำขั้นตอนที่ 1-3 ของหัวข้อที่แล้ว ซึ่งประกอบด้วยการกำหนด Service Callback (เมธอดที่จะถูกเรียกเมื่อเข้ามต่อ กับ Location Services, เมื่อยกเลิกการเข้ามต่อ และเมื่อการเข้ามต่อล้มเหลว), สร้าง Location Client และสั่งให้ Location Client เข้ามต่อหรือยกเลิกการเข้ามต่อ กับ Location Services ในเวลาที่ต้องการ
- 2 กำหนด Update Callback (เมธอดที่ Location Services จะเรียกกลับมาเป็นระยะๆ เพื่ออัพเดท ตำแหน่งให้เรา) โดยสร้างคลาสที่ Implement อินเทอร์เฟซ LocationListener และทำการ Override เมธอด onLocationChanged

ตัวอย่างนี้จะกำหนดให้แอคทิวิตี้ Implement อินเทอร์เฟซดังกล่าว เพื่อใส่เมธอด onLocationChanged ไว้ในแอคทิวิตี้เลย

```
public class MainActivity extends Activity implements
    GooglePlayServicesClient.ConnectionCallbacks,
    GooglePlayServicesClient.OnConnectionFailedListener,
    LocationListener {

    ...

    @Override
    public void onLocationChanged(Location location) {
        // อ่านตำแหน่งปัจจุบันของผู้ใช้จากตัวแปร location มาใช้งาน
    }
}
```

Location Services จะส่งค่าพารามิเตอร์เป็นออบเจ็ค Location มาให้เรา (ตัวแปร location ในโค้ดข้างต้น) ซึ่งตำแหน่งปัจจุบันของผู้ใช้ รวมถึงข้อมูลอื่นๆ จะเก็บอยู่ในออบเจ็คนี้

- 3 กำหนดช่วงเวลาในการอัพเดทตำแหน่ง หรือพูดง่ายๆ คือ จะให้ Location Services แจ้งตำแหน่ง ปัจจุบันให้เราทุกๆ 5 วินาที โดยสร้างออบเจ็ค LocationRequest และเรียกเมธอด setInterval ของมัน

```
LocationRequest request = LocationRequest.create();
request.setInterval(5000); // ให้อัพเดทตำแหน่งทุกๆ 5 วินาที
request.setFastestInterval(1000);
request.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
```

Location Services อาจจะอัพเดทตำแหน่งให้เราทุกๆ วินาที ซึ่งเราต้องกำหนดโดยเมื่อตัด `setInterval` ก็ได้ ขึ้นอยู่กับว่าตอนนั้นมีแอปฯ ไหนๆ ติดตามการเปลี่ยนแปลงตำแหน่งจาก Location Services ด้วยหรือไม่ รวมถึงการกำหนดช่วงเวลาอัพเดทของแอปเหล่านั้น

การอัพเดทตำแหน่งบ่อยเกินไปจะสิ้นเปลืองแบตเตอรี่โดยไม่จำเป็น ดังนั้น `LocationRequest` จึงมี เมื่อตัด `setFastestInterval` ไว้ให้กำหนดช่วงเวลาการอัพเดทที่เร็วที่สุด (บ่อยที่สุด) ที่เราคิดว่า เหมาะสมสำหรับแอปของเราหรือแอปของสาธารณะกรณีการได้

สำหรับเมื่อตัด `setPriority` จะกำหนดว่าเราต้องการรู้ตำแหน่งจากแหล่ง (location source) ใด ซึ่งค่าคงที่ `PRIORITY_HIGH_ACCURACY` นั้นหมายถึงการหาตำแหน่งโดยใช้ GPS

- 4 เริ่มการอัพเดทตำแหน่ง โดยเรียกเมื่อตัด `requestLocationUpdates` ของ `LocationClient` พร้อมทั้งระบุ `LocationRequest` และออบเจ็คที่ทำหน้าที่เป็น `Location Listener` (ในที่นี้คือ แอคทิวิตี้) เป็นพารามิเตอร์

```
mLocationClient.requestLocationUpdates(request, MainActivity.this);
```

- 5 เมื่อต้องการยกเลิกการอัพเดทตำแหน่ง ให้เรียกเมื่อตัด `removeLocationUpdate` โดยระบุ `Location Listener` ที่จะยกเลิกเป็นพารามิเตอร์

```
mLocationClient.removeLocationUpdates(MainActivity.this);
```

ตัวอย่าง

ตัวอย่างนี้จะมีปุ่ม Start Update, Stop Update และแผนที่ (MapFragment) ซึ่งเมื่อคลิก Start Update จะแสดงตำแหน่งปัจจุบันของผู้ใช้ที่ก่อกลางแผนที่ และอัพเดทตำแหน่งทุก 5 วินาที

- 1 กำหนด Layout ของหน้าจอ

โปรเจ็ค LocationUpdateDemo, ไฟล์ activity_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:map="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="6dp" >

        <Button
            android:id="@+id/start_update_button"
            android:layout_width="0dp"
```

```
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Start Update" />

    <Button
        android:id="@+id/stop_update_button"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Stop Update" />
</LinearLayout>

<fragment
    android:id="@+id/map"
    android:name="com.google.android.gms.maps.MapFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    map:cameraBearing="0"
    map:cameraTargetLat="13.7500"
    map:cameraTargetLng="100.4833"
    map:cameraTilt="0"
    map:cameraZoom="15"
    map:mapType="normal"
    map:uiCompass="true"
    map:uiRotateGestures="true"
    map:uiScrollGestures="true"
    map:uiTiltGestures="true"
    map:uiZoomControls="false"
    map:uiZoomGestures="true" />
</LinearLayout>
```

2 เพิ่มโค้ดในแอคทิวิตี้เป็นดังนี้ (ขอลบบรรทัดการ import ต่างๆ)

โปรเจ็ค LocationUpdateDemo, ไฟล์ MainActivity.java

```
package com.example.locationupdatedemo;

...
public class MainActivity extends Activity implements
    GooglePlayServicesClient.ConnectionCallbacks,
    GooglePlayServicesClient.OnConnectionFailedListener,
    LocationListener {

    private LocationClient mLocationClient; // Location Client
    private GoogleMap map; // แผนที่ Google Maps ใน MapFragment
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    // เข้าถึง MapFragment
    MapFragment mapFragment = (MapFragment) getSupportFragmentManager()
        .findFragmentById(R.id.map);
    // เข้าถึงออบเจ็ค GoogleMap ใน MapFragment
    map = mapFragment.getMap();

    // สร้าง Location Client
    mLocationClient = new LocationClient(this, this, this);

    // สร้าง Location Request แล้วกำหนดช่วงเวลาการอัปเดท
    final LocationRequest request = LocationRequest.create();
    request.setInterval(5000);
    request.setFastestInterval(1000);
    request.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);

    // เข้าถึงปุ่มทั้งสองใน Layout
    final Button btnStartUpdate = (Button) findViewById(
        R.id.start_update_button);
    final Button btnStopUpdate = (Button) findViewById(
        R.id.stop_update_button);

    // ตอนเริ่มต้นกำหนดให้ปุ่ม Start Update ใช้งานได้ แต่ปุ่ม Stop Update ใช้งานไม่ได้
    btnStartUpdate.setEnabled(true);
    btnStopUpdate.setEnabled(false);

    // ระบุการทำงานของปุ่ม Start Update
    btnStartUpdate.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            // ตรวจสอบเพื่อความแน่ใจว่าเชื่อมต่อ Location Services แล้ว
            if (mLocationClient.isConnected()) {
                // ขออัปเดตตำแหน่ง
                mLocationClient.requestLocationUpdates(
                    request, MainActivity.this);
                // กำหนดให้ปุ่ม Start Update ใช้งานไม่ได้ แต่ Stop Update ใช้งานได้
                btnStartUpdate.setEnabled(false);
                btnStopUpdate.setEnabled(true);
            } else {
                String msg = "Can't connect to location services,
                           try again later.";
            }
        }
    });
}
```

```
        Toast.makeText(MainActivity.this, msg,
                      Toast.LENGTH_LONG).show();
    }
}
});

// ระบุการทำงานของปุ่ม Stop Update
btnStopUpdate.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        if (mLocationClient.isConnected()) {
            // ยกเลิกการอัพเดทตำแหน่ง
            mLocationClient.removeLocationUpdates(
                MainActivity.this);
            // กำหนดให้ปุ่ม Start Update ใช้งานได้ และ Stop Update ใช้งานไม่ได้
            btnStartUpdate.setEnabled(true);
            btnStopUpdate.setEnabled(false);
        }
    }
});
}

@Override
protected void onStart() {
    super.onStart();
    mLocationClient.connect();
}

@Override
protected void onStop() {
    mLocationClient.disconnect();
    super.onStop();
}

@Override
public void onConnected(Bundle data) {
    Toast.makeText(this, "Connected", Toast.LENGTH_SHORT).show();
}

@Override
public void onDisconnected() {
    Toast.makeText(this, "Disconnected", Toast.LENGTH_SHORT).show();
}

@Override
public void onConnectionFailed(ConnectionResult result) {
```

```

        Toast.makeText(this, "Connection Failed", Toast.LENGTH_SHORT)
            .show();
    }

    @Override
    public void onLocationChanged(Location location) {
        map.clear(); // ลบหมุดที่มีอยู่เดิม

        // หาพิกัดละติจูด/ลองจิจูดของตำแหน่งปัจจุบัน
        double lat = location.getLatitude();
        double lng = location.getLongitude();

        // สร้างออบเจ็ค LatLng ที่เก็บค่าละติจูด/ลองจิจูด
        LatLng currentLatLng = new LatLng(lat, lng);
        // ย้ายกล้องไปที่ตำแหน่งนั้นแบบมีเส้นimeชั่ว และปักหมุด
        map.animateCamera(CameraUpdateFactory.newLatLng(currentLatLng));
        map.addMarker(new MarkerOptions()
            .position(currentLatLng)
        );
    }
}

```

ผลการรับ

ทดลองรันแอปบนเครื่องจริง จากนั้นคลิกปุ่ม Start Update ในแอป แล้วถือเครื่องเดินไปเรื่อยๆ จะเห็นว่าตำแหน่งของหมุดเปลี่ยนไปตามตำแหน่งปัจจุบันของเรา (สังเกตว่าแอนดรอยด์แสดงไอคอน ✧ บนแถบสถานะ บอกให้ผู้ใช้รู้ว่าตอนนี้กำลังติดตามตำแหน่งของผู้ใช้อยู่ และเมื่อคลิก Stop Update ไอคอนนี้จะหายไป)

