

Android Tutorial using Kotlin 第四堂

（2）設計地圖應用程式 - Google Maps

Android API

Android Tutorial using Kotlin 第四堂（1）使用照相機與麥克風 << 前情

地圖已經是非常普遍的使用在各種應用程式，Google在2013年2月，為Android平台推出Google Maps Android API v2，使用設計方式，改良地圖元件的畫面與效率，增加3D建築物，還有簡化的繪圖API。Android已經在2013年4月停止舊的API，所以會說明Google Maps Android API v2的設計方式。

Google Maps Android API v2包含在「Google Play Service SDK」，目前不能在模擬裝置中測試地圖應用程式，所以必需準備裝置，接下來的應用程式必須安裝在實體裝置執行測試。

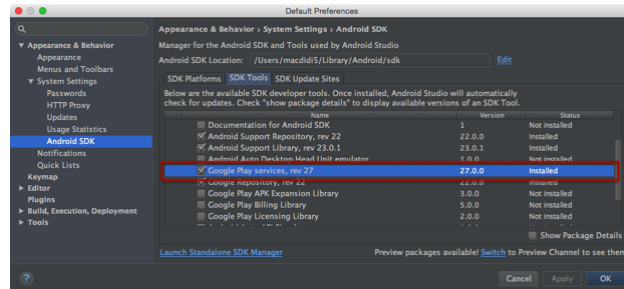
這一章為記事資料加入地圖元件的應用，如果使用者在記事資料儲存座標的資料，就可以啟動地圖元件顯示儲存的地點：



13-1 建立地圖元件

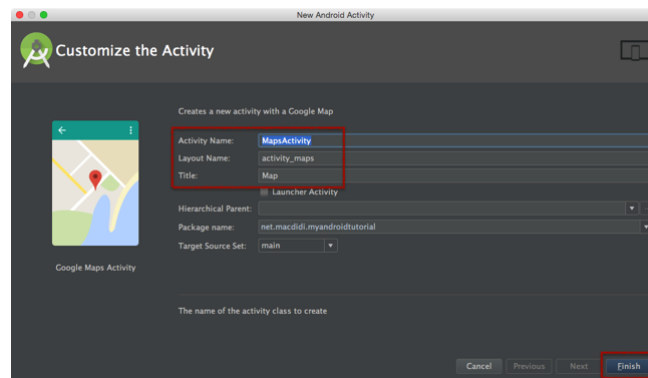
地圖元件從網路下載所有的資料，包含地圖、接道名稱、衛星空照與地形。應用程式必須申請專用的「Google Maps API Key」可以正確的下載資料。使用Android Studio可以簡化地圖應用程式的開發，在開始設計地圖應用程式之前，先依照下列的步驟確認已經安裝好「Google Play services」SDK：

1. 啟動Android Studio並開啟記事應用程式。
2. 選擇功能表「Tools -> Android -> SDK Manager」。
3. 在Android SDK Manager視窗，檢查「Extras -> Google Play services」是否已經安裝。如果還沒有安裝的話，勾選並執行安裝工作：

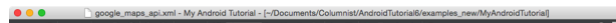


依照下列的步驟，為記事應用程式新增一的地圖元件：

1. 啟動Android Studio並開啟記事應用程式。
2. 在應用程式目錄（App）按滑鼠右鍵，選擇「New -> Google -> Google Maps Activity」。
3. 使用預設的設定，選擇「Finish」：



4. Android Studio建立好需要的元件以後，會自動開啟「google_maps_api.xml」，找到畫面中一個內容很長的網址，選擇以所有的內容：



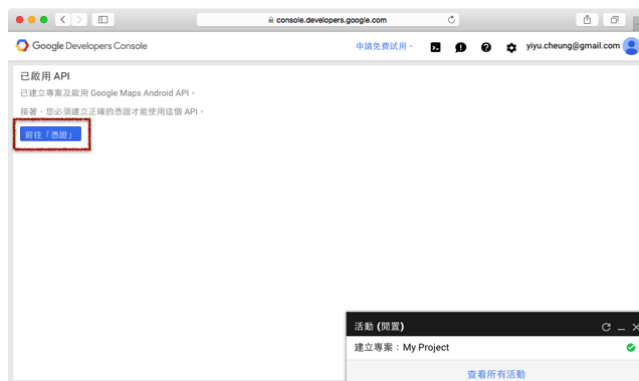
5. 啟動瀏覽器，在網址列貼上前一個步驟複製的網址，使用Google的帳號登入以後，勾選同意服務條款，選擇「同意並繼續」



6. 網頁右下角顯示建立專案的活動訊息：



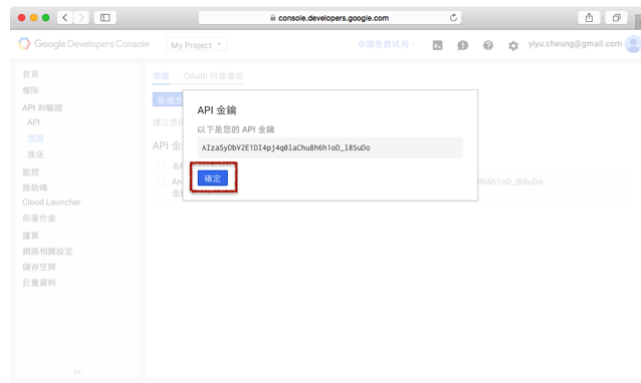
7. 專案建立完成後，選擇「前往憑證」：



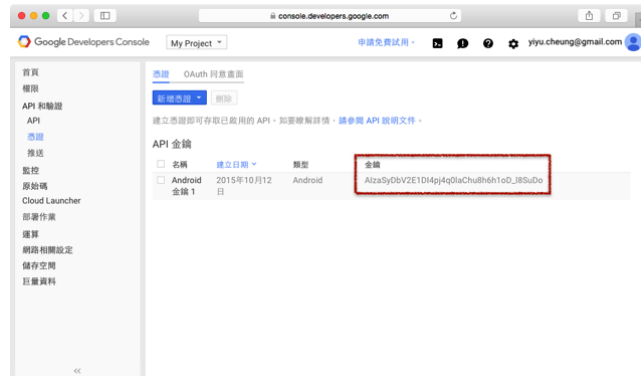
8. 檢視應用程式的套件名稱與SHA-1憑證指紋，選擇「建立」：



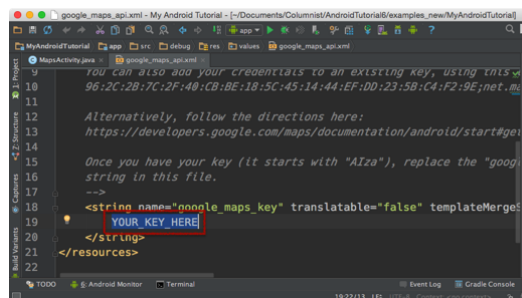
9. 建立完成後在對話框顯示API金鑰，選擇「確定」：



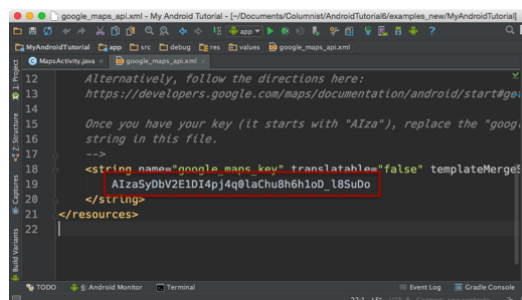
10. 會到憑證畫面後，「金鑰」的內容就是為已經申請好的Google Maps API Key。選擇並複製API金鑰的內容：



11. 回到Android Studio，在google_maps_api.xml檔案，找到「YOURKEYHERE」的位置：



12. 把第八個步驟複製的API金鑰貼上與覆蓋「YOUR_KEY_HERE」：



完成上面的步驟以後，就已經建立好一個預設的地圖元件。依照下列的說明，檢查Android Studio為應用程式加入的內容：

- google_maps_api.xml：儲存與設定Google Maps API Key的資源檔案。
- MapsActivity.kt：地圖元件。
- activity_maps.xml：地圖元件使用的畫面資源檔。
- Gradle Scripts -> build.gradle(Module:app)：在「dependencies」區塊加入Google Play Service SDK的設定。

```

...
android {
    ...
}

dependencies {
    ...
    implementation 'com.google.android.gms:play-services-maps:11.6.2'
    ...
}

```

- **AndroidManifest.xml**：在**manifest**標籤下加入讀取位置的授權，目前還沒有使用讀取位置的功能。

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="net.macdidi.atk">

    ...

    <!-- 新增地圖元件的時候，自動加入的設定 -->
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

    <application ...>
        ...
    </application>

</manifest>

```

- **AndroidManifest.xml**：在**application**標籤下加入Google Service版本、Google Map API key與地圖元件的設定。

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="net.macdidi.atk">

    ...

    <application ...>
        ...

        <!-- Google Map API key -->
        <meta-data
            android:name="com.google.android.geo.API_KEY"
            android:value="@string/google_maps_key" />

        <!-- 地圖元件 -->

```

```

        <activity
            android:name=".MapsActivity"
            android:label="@string/title_activity_maps"></activity>
    </application>

</manifest>

```

13-2 啟動地圖元件

建立好的地圖元件，可以使用一般的方式啟動它。開啟「ItemActivity.kt」，參考下列的內容，加入啟動地圖元件的程式碼：

```

package net.macdidi.atk

...

class ItemActivity : AppCompatActivity() {

    ...

    fun clickFunction(view: View) {
        when (view.id) {
            ...
            R.id.set_location -> {
                // 啟動地圖元件用的Intent物件
                val intentMap = Intent(this, MapsActivity::class.java)
                // 啟動地圖元件
                startActivityForResult(intentMap, ItemAction.LOCATION.ordinal)
            }
            ...
        }
    }

    ...

}

```

完成上面的程式碼，連接實體裝置到電腦，執行應用程式後選擇安裝到實體裝置。在應用程式的主畫面選擇任何一個記事資料擇位置按鈕元件：



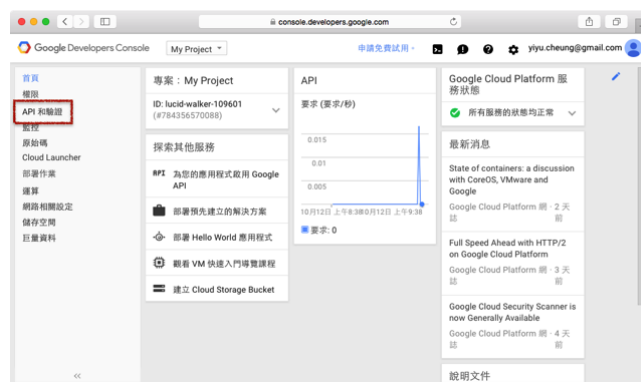
應用程式啟動地圖元件以後，如果出現下列的畫面，表示地圖元件可以正確的運作：



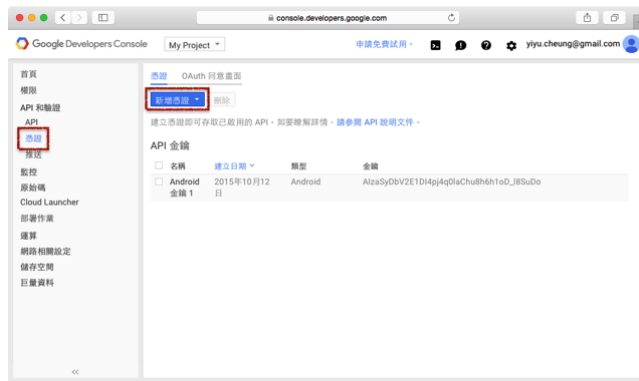
13-3 管理與新增Google Maps API Key

每一個使用地圖元件的應用程式，都需要依照上列的方式建立Google Maps API Key，地圖才可以正確下載與顯示。如果開發用地圖元件的應用程式，可以繼續使用同一個Google Maps API Key。參考下列的步驟，瞭解執行設定的作法：

1. 在瀏覽器開啟這個網頁：<https://console.developers.google.com/>。
2. 選擇「API和憑證」：



3. 選擇「憑證」目錄後選擇「新增憑證」：



4. 一個API金鑰可以讓多個應用程式使用，選擇「新增套件名稱和指紋」：



5. 在套件名稱輸入應用程式的主套件名稱，在同一台電腦開發應用程式，SHA-1憑證指紋輸入同樣的內容，最後選擇「儲存」



13-4 設定地圖顯示的位置

目前地圖元件顯示的畫面是預設的位置，以記事應用程式來說，在啟動地圖元件以後，需要將位置移動到儲存的位置，或是將位置資訊顯示目前的位置。目前的記事資料還沒有儲存位置資訊，下一章就會說明讀取與儲存位置資訊的作法，現在先瞭解設定顯示位置的作法。

開啟「MapsActivity.kt」，加入提供移動地圖功能的函式：

```
package net.macdidi.atk

...

class MapsActivity : AppCompatActivity(), OnMapReadyCallback {

    ...
```



```

// 移動地圖到參數指定的位置
private fun moveMap(place: LatLng) {
    // 建立地圖攝影機的位置物件
    val cameraPosition = CameraPosition.Builder()
        .target(place)
        .zoom(17f)
        .build()

    // 使用動畫的效果移動地圖
    mMap.animateCamera(CameraUpdateFactory.newCameraPosition(cameraPosition))
}

...

}

```

同樣在「MapsActivity.kt」，找到預設的「onMapReady」函式，參考下列的內容修改原有的程式碼：

```

package net.macdidi.atk

...

class MapsActivity : AppCompatActivity(), OnMapReadyCallback {

    ...

    // 覆寫OnMapReadyCallback的函式
    override fun onMapReady(googleMap: GoogleMap) {
        mMap = googleMap

        // 刪除原來預設的內容
        //val sydney = LatLng(-34.0, 151.0)
        //mMap.addMarker(MarkerOptions().position(sydney).title("Marker in Sydney"))
        //mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney))

        // 建立位置的座標物件
        val place = LatLng(25.033408, 121.564099)

        // 移動地圖
        moveMap(place)
    }

    ...

}

```

完成上面的程式碼，執行應用程式後選擇安裝到實體裝置。在應用程式的主畫面選擇任何一個記事資料，選擇位置按鈕元件，程式啟動地圖元件以後，地圖畫面就會移動到指定的位置：



13-5 新增地圖標記

地圖元件可以依照應用程式的需求，在地圖指定的位置加入標記。在「com.google.android.gms.maps.model」套件提供下列元件：

- **Marker** – 在地圖上的一個標記就是一個Marker物件。
- **MarkerOptions** – 在加入Marker到地圖前，先使用它設定好Marker的所有資訊，例如位置、圖示與標題。
- **BitmapDescriptor**與**BitmapDescriptorFactory** – 需要設定Marker圖示的時候會用到它們。

開啟「MapsActivity.kt」，加入設定地圖標記的函式：

```
package net.macdidi.atk

...

class MapsActivity : AppCompatActivity(), OnMapReadyCallback {

    ...

    // 在地圖加入指定位置與標題的標記
    private fun addMarker(place: LatLng, title: String, context: String) {
        val icon : BitmapDescriptor =
            BitmapDescriptorFactory.fromResource(R.drawable.atk_launcher)

        val markerOptions = MarkerOptions()
        markerOptions.position(place)
            .title(title)
            .snippet(context)
            .icon(icon)

        mMap.addMarker(markerOptions)
    }

}
```

選擇這個連結https://github.com/macdidi5/Android-Tutorial-Kotlin/blob/master/images/launcher/mipmap-hdpi/atk_launcher.png下載標記圖示，複製下載的圖示檔案到「res/drawable」。

同樣在「MapsActivity.kt」，找到「onMapReady」函式，參考下列的內容修改原有的程式碼：

```
package net.macdidi.atk

...

class MapsActivity : AppCompatActivity(), OnMapReadyCallback {

    ...

    // 覆寫OnMapReadyCallback的函式
    override fun onMapReady(googleMap: GoogleMap) {
        mMap = googleMap

        val place = LatLng(25.033408, 121.564099)
        moveMap(place)

        // 加入地圖標記
        addMarker(place, "Hello!", " Google Maps v2!");
    }

    ...

}
```

完成上面的程式碼，執行應用程式後選擇安裝到實體裝置。在應用程式的主畫面選擇任何一個記事資料，選擇位置按鈕元件，程式啟動地圖元件以後，地圖畫面在指定的位置顯示標記：



點選標記以後，標記上方就會顯示設定的說明：



完成這一章關於地圖的應用，下一章加入位置資訊的讀取與儲存以後，就可以在地圖元件顯示記事儲存的位置，或是根據裝置顯示目前的位置。

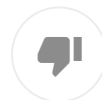
相關的檔案都可以在GitHub瀏覽與下載：

GitHub


<https://github.com/macdidi5/Android-Tutorial-Kotlin>

後續 >> Android Tutorial using Kotlin 第四堂（3）讀取裝置目前的位置 – Google Services Location

Does Clearly work fine?



Shortcuts: **SHIFT+CTRL+C** to Toggle, **ESC** to Close.

 Give us feedback

Build upon ♥ with Clearly