

Lab12

Google Map

本節目的：

- 申請 Google Maps API 金鑰。
- 透過 Gradle 匯入 Maps Library。
- 使用 Google Maps API 。

12.1 Google Map

Google Maps 整合基本地圖、3D 建築、室內樓層平面圖、街景和衛星影像，以及自訂標記等功能。Google Maps Android API 可以根據「Google 地圖」的資料，將地圖新增至應用程式中。API 會自動處理對「Google 地圖」伺服器的存取、資料下載、地圖顯示，以及回應地圖手勢。也可以使用 API 呼叫，將標記、線段新增至基本地圖，以及變更使用者觀看的特定地圖區域。這些物件為地圖位置提供其他資訊，並允許使用者與地圖進行互動。

圖 12-1 是地圖畫面呈現，Google Maps 需要先完成專案配置、取得 API 金鑰，以下僅概略的介紹一些程式建置與功能用法。



圖 12-1 Google Map 畫面

12.1.1 新增地圖到 Android 應用程式

新增地圖的基本步驟如下，Google Maps API 第二版採用 SupportMapFragment 類別。MapFragment 是顯示地圖的畫面元件，他可以把地圖畫面當成一個元件放到 XML 畫面下呈現。我們直接在 activity_main.xml 中加入 <fragment> 標籤，如圖 12-2 所示。

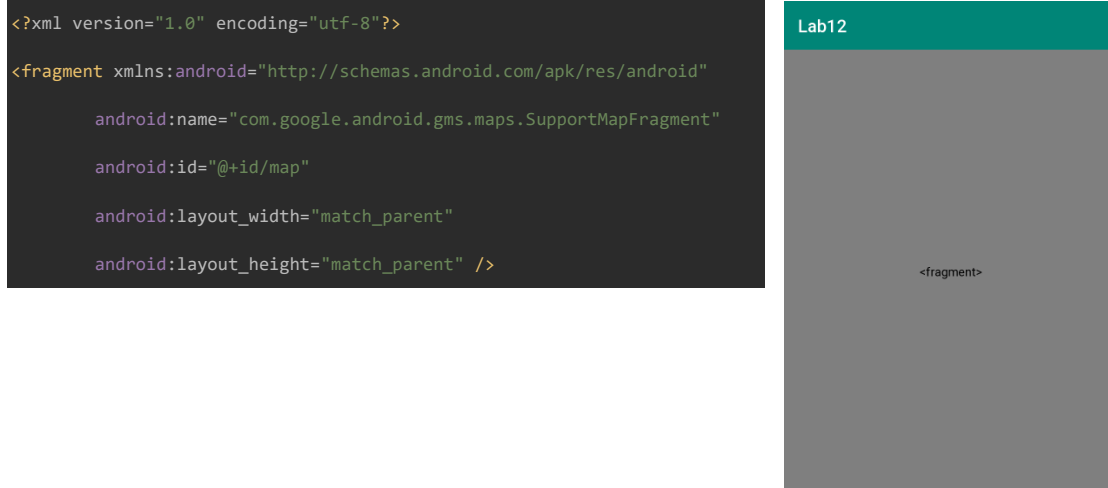


圖 12-2 MapFragment

而在 Kotlin 中，我們使用 supportFragmentManager.findFragmentById() 來連結 Xml 上的 SupportMapFragment 類別。我們需要使用 getMapAsync() 方法來啟動 GoogleMap，在其中我們需要實作 OnMapReadyCallback 介面，並且使用 onMapReady(map: GoogleMap) 來取得 GoogleMap 的執行程式。

```
class MainActivity : AppCompatActivity(), OnMapReadyCallback {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        //Step1：連接 MapFragment 元件
        val map = supportFragmentManager.findFragmentById(R.id.map)
                                                as SupportMapFragment

        //Step2：執行 map 的非同步方法
        map.getMapAsync(this)
    }
    //Step3：取得 GoogleMap
    override fun onMapReady(map: GoogleMap) {
    }
}
```

12.1.2 顯示目前位置

Google Maps 地圖本身就有內建顯示目前位置的按鈕，要顯示此按鈕需要加入 `googleMap.isMyLocationEnabled` 為 `true`。

```
override fun onMapReady(map: GoogleMap) {  
    map.isMyLocationEnabled = true //顯示目前位置與定位按鈕  
}
```

不過如果只加入這一段程式碼，會發現此段程式碼被標記紅線，這是因為此功能需要有搭配 `ACCESS_FINE_LOCATION` 與 `ACCESS_COARSE_LOCATION` 權限，因此需要在 `AndroidManifest.xml` 加入權限宣告。

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="bluenet.com.lab12">  
    <!--允許程式使用網路權限-->  
    <uses-permission android:name="android.permission.INTERNET" />  
    <!--允許程式存取粗略的位置-->  
    <uses-permission android:name=  
        "android.permission.ACCESS_COARSE_LOCATION" />  
    <!--允許程式存取精確位置-->  
    <uses-permission android:name=  
        "android.permission.ACCESS_FINE_LOCATION" />
```

Android6.0 開始，程式碼中要加入對應危險權限的授權檢查，要求應用程式必須向使用者取得使用授權，才能存取資料。

```
class MainActivity : AppCompatActivity(), OnMapReadyCallback {  
    private val REQUEST_PERMISSIONS = 1  
    //權限要求結果  
    override fun onRequestPermissionsResult(requestCode: Int,  
        permissions: Array<String>, grantResults: IntArray) {  
        if (grantResults.isEmpty()) return  
        when (requestCode) {  
            REQUEST_PERMISSIONS -> {  
                for (result in grantResults)  
                    if (result != PackageManager.PERMISSION_GRANTED)  
                        finish() //若使用者拒絕給予權限則關閉 APP  
                else{ //連接 MapFragment 元件  
                    val map =
```

```

supportFragmentManager.findFragmentById(R.id.map) as SupportMapFragment
        map.getMapAsync(this)
    }
}
}

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)
    //檢查使用者是否已授權定位權限
    if (ActivityCompat.checkSelfPermission(this,
        android.Manifest.permission.ACCESS_COARSE_LOCATION) !=
        PackageManager.PERMISSION_GRANTED)

        //要求定位權限
        ActivityCompat.requestPermissions(this,
            arrayOf(android.Manifest.permission.ACCESS_FINE_LOCATION),
                REQUEST_PERMISSIONS)

    else{
        //連接 MapFragment 元件
        val map = supportFragmentManager.findFragmentById(R.id.map)
                                                as SupportMapFragment

        map.getMapAsync(this)
    }
}

override fun onMapReady(map: GoogleMap) {
    //檢查使用者是否已授權定位權限
    if (ActivityCompat.checkSelfPermission(this,
        android.Manifest.permission.ACCESS_FINE_LOCATION) !=
        PackageManager.PERMISSION_GRANTED &&
        ActivityCompat.checkSelfPermission(this,
            android.Manifest.permission.ACCESS_COARSE_LOCATION) !=
            PackageManager.PERMISSION_GRANTED)

        return
    ...
}
}

```

執行後，地圖右上角就會出現一個定位鈕，按下後地圖便會動畫切換至目前位置，如圖 12-3 所示。



圖 12-3 地圖右上角出現定位按鈕

12.1.3 標記地圖

要在地圖上新增一個標記需要透過 `addMarker()` 方法來新增一個標記，這邊需要使用到 `MarkerOptions` 類別，`MarkerOptions` 就是能產生一個地圖標籤物件，他具備幾種屬性可以設定如下：

```
override fun onMapReady(map: GoogleMap) {  
    //Step1：建立 MarkerOptions 物件  
    val marker = MarkerOptions()  
    //Step2：設定座標  
    marker.position(LatLng(25.033611, 121.565000))  
    //Step3：設定標記名稱  
    marker.title("台北 101")  
    //Step4：設定是否可拖曳  
    marker.draggable(true)  
    //Step5：插入 Marker 至 googleMap  
    map.addMarker(marker)  
}
```

其中經緯度需要使用 LatLng()類別來設定，LatLng()用法很簡單，只需要帶入經度與緯度參數即可定義其經緯度值。

執行後，地圖對應的經緯度上就會出現標記，點擊後便會出現名稱，如圖 12-4 所示。



圖 12-4 地圖上的標記

12.1.4 切換鏡頭

要切換鏡頭到指定的座標，需要透過 moveCamera()方法，這其中要透過 CameraUpdateFactory 的 newLatLngZoom()方法來設定位置，newLatLngZoom()中可以設定兩個參數，第一個是座標，而第二個是鏡頭深度，鏡頭深度的值越大，地圖就拉的越近，如圖 12-5 所示。

```
override fun onMapReady(map: GoogleMap) {  
    map.moveCamera(CameraUpdateFactory.newLatLngZoom(  
        LatLng(25.033611, 121.565000), 15f))  
})
```

↓ ↓
座標 深度



圖 12-5 移動畫面到指定地點

12.1.5 畫線

要在地圖上繪製線段要透過 `addPolyline()` 方法來新增線段，需要使用到 `PolylineOptions` 類別。`PolylineOptions` 可以用 `PolylineOptions.add(經緯度)` 的方式，定義線段的每一個要走訪的經緯度，會依照 `add` 的先後順序來連接。線段可以透過 `PolylineOptions.Color` 的方式設定顏色值。

```
override fun onMapReady(map: GoogleMap) {  
    //Step1：建立 PolylineOptions 物件  
    val polylineOpt = PolylineOptions()  
    //Step2：加入三個線段通過的座標  
    polylineOpt.add(LatLng(25.033611, 121.565000))  
    polylineOpt.add(LatLng(25.032728, 121.565137))  
    polylineOpt.add(LatLng(25.047924, 121.517081))  
    //Step3：將線段設為藍色  
    polylineOpt.color(Color.BLUE)  
    //Step4：加入 PolylineOptions 到 googleMap，並產生 polyline  
    val polyline = map.addPolyline(polylineOpt)
```



```
//Step5：設定線段寬度
polyline.width = 10f
}
```

透過 addPolyline() 加入 PolylineOptions 後，會回傳一個 Polyline 的物件，此物件就是在地圖上繪製完成的線段物件，取得後我們可以進一步使用 Polyline.width 控制線段的寬度。

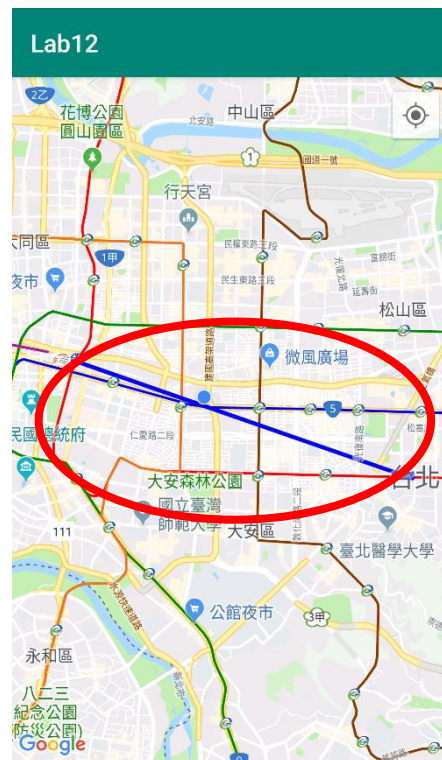


圖 12-6 在地圖上加入 polyline

12.2 Google Map 實戰演練

- 申請 Google Maps API 金鑰。
- 安裝 Google Maps API。
- 實際練習在 Google Map 上顯示自己的定位，透過 marker 標記地點，以及兩點畫線，如圖 12-7 所示。

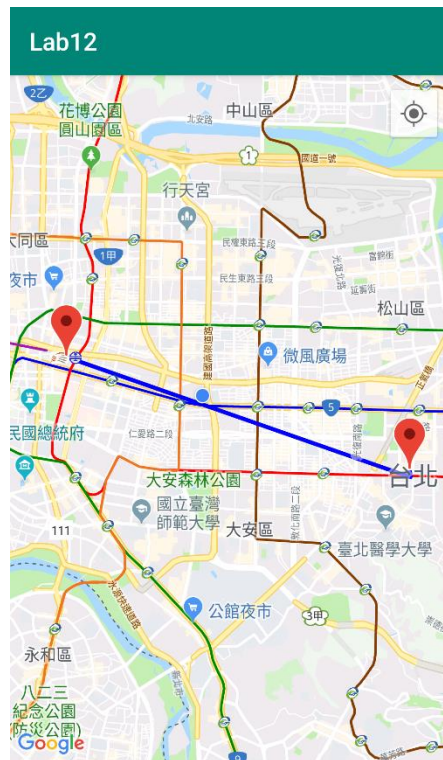


圖 12-7 APP 實作畫面

12.2.1 Google API 申請

要在應用程式使用 Google Maps 前，需先申請 Google Maps API Key 才能使用 Google Maps 服務。

Step1 到 Google APIs Console 網頁：<https://console.developers.google.com/apis/dashboard> 並登入 google 帳戶，接著按下「選取專案」，並按下「新增專案」來創建一個新的專案，如圖 12-8 所示。

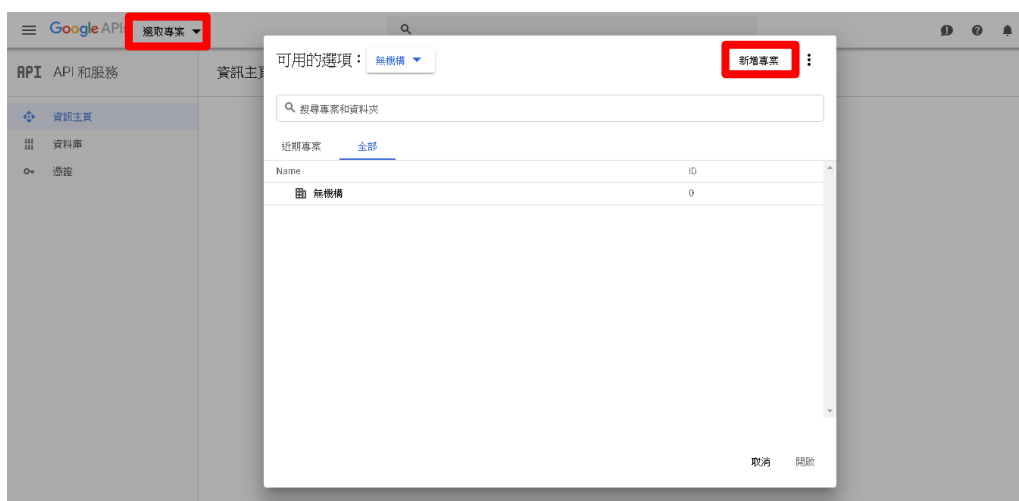


圖 12-8 新增 Google API 專案

Step2 輸入專案名稱後按下「建立」，如圖 12-9 所示。

新增專案

您的配額尚可供建立 10 projects。您可以要求 增加配額，或請考慮刪除專案。
[瞭解詳情](#)
[MANAGE QUOTAS](#)

專案名稱 *
My Project

專案 ID：eng-copilot-222405。專案 ID 設定完成後即無法變更。 [編輯](#)

位置 *
無機構 [瀏覽](#)

上層機構或資料夾

[建立](#) [取消](#)

圖 12-9 設定專案名稱

Step3 專案建立完成後在左上角的選取專案中找到已建立的專案，並選擇「Maps SDK for Android」，如圖 12-10 所示。

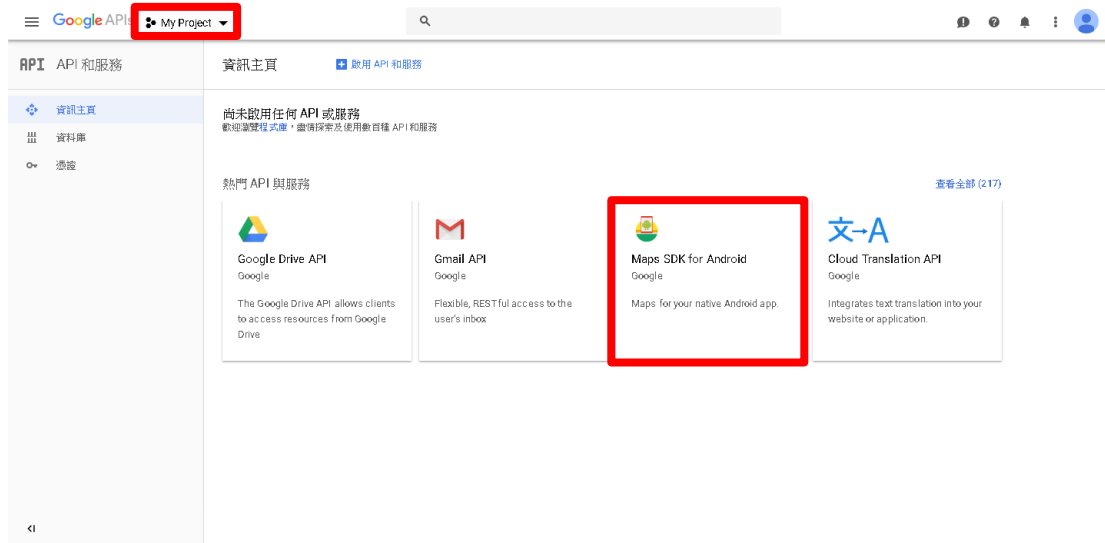


圖 12-10 選取專案並點選 Map SDK

Step4 按下「啟用」按鈕，以啟動 Maps SDK for Android，如圖 12-11 所示。

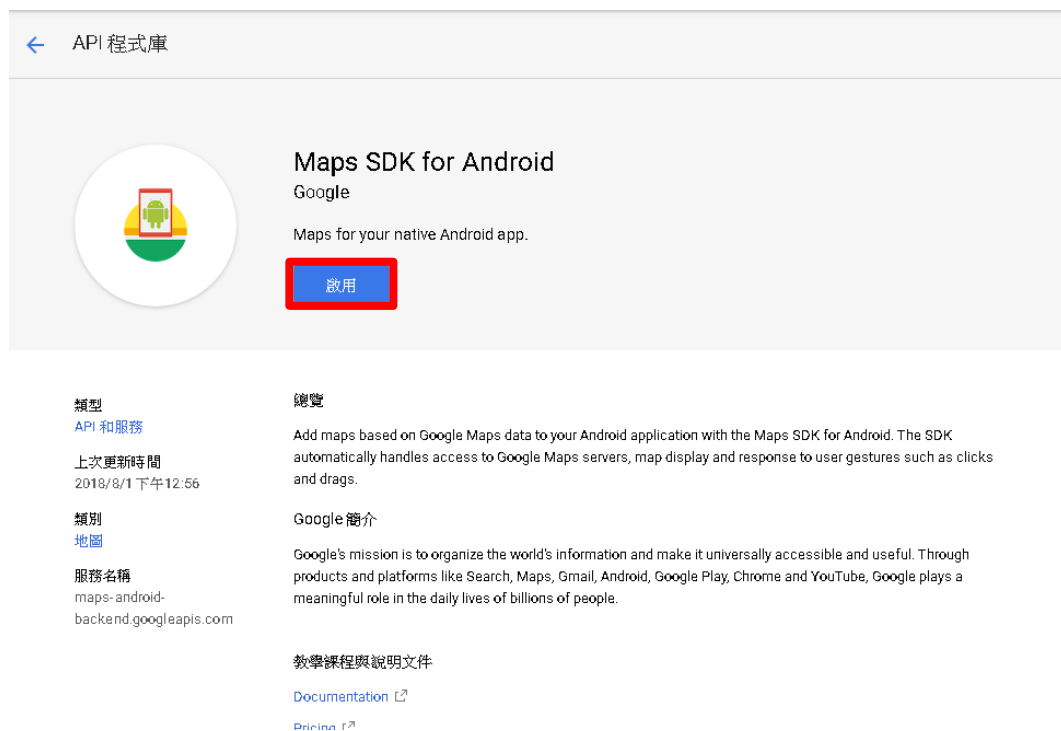


圖 12-11 啟動 SDK

Step5 啟用成功後，會看到如圖 12-12 的畫面，SDK 監控後台可以用來監控 API 的使用流量，觀察 API 的使用狀況。

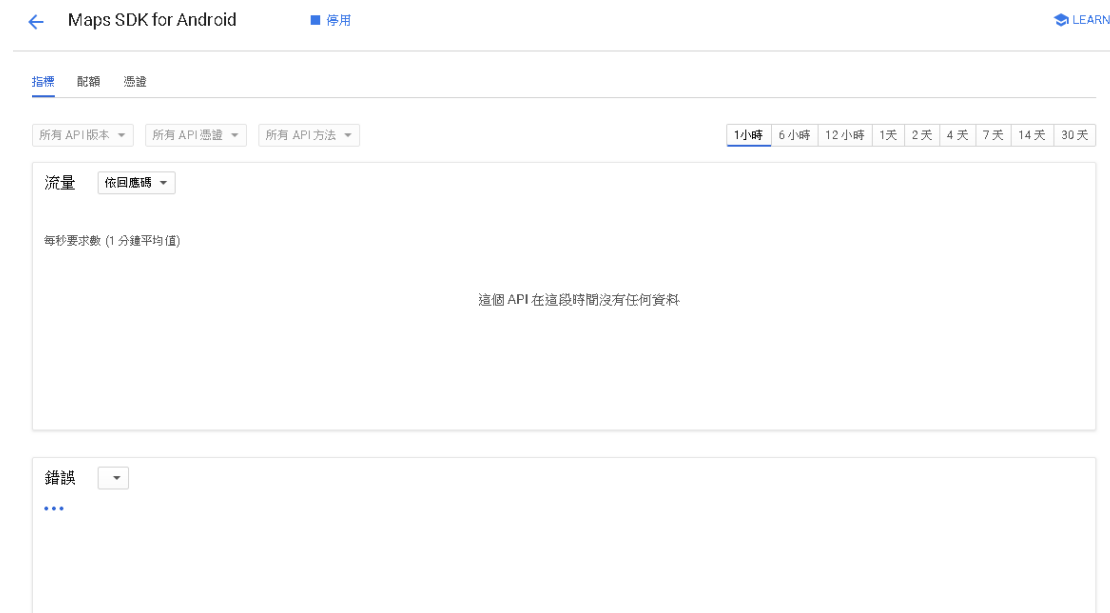


圖 12-12 SDK 監控後台

Step6 點選左上角的「憑證」，並選取「建立憑證」，如圖 12-13 所示。



圖 12-13 建立憑證

Step7 選擇「API 金鑰」建立 API 鑰匙，如圖 12-14 所示。



圖 12-14 建立 API 金鑰

Step8 完成後下可以得到一組金鑰，之後需要放到專案程式中使用，如圖 12-5 所示。

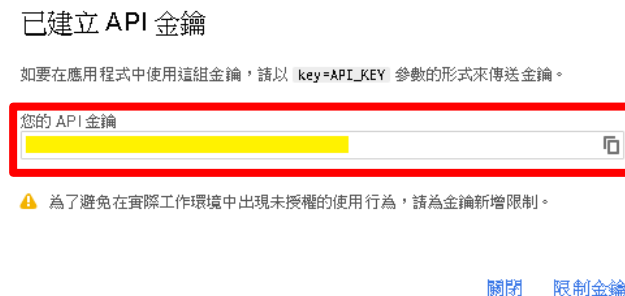


圖 12-15 取得金鑰匙

12.2.2 安裝 Google Maps API

Step1 開啟位於 Android Studio 工具列的「SDK Manager」，如圖 12-16 所示。



圖 12-16 SDK Manager

Step2 選擇安裝 SDK Tools 的「Android Support Library」及「Google Play Services」，如圖 12-17 所示。

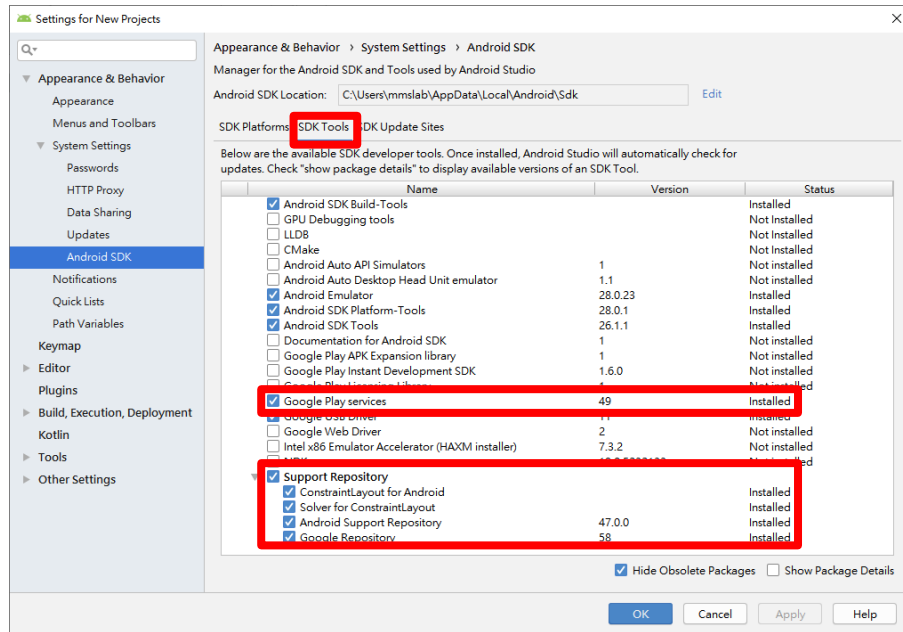


圖 12-17 下載 SDK Tools

Step3 將「google-play-services」library 匯入專案中，開啟 build.gradle 加入 implementation 'com.google.android.gms:play-services-maps:16.0.0'。

```
android {
    compileSdkVersion 28
    defaultConfig {
        applicationId "bluenet.com.lab12"
        minSdkVersion 22
        targetSdkVersion 28
        versionCode 1
        versionName "1.0"
    }
    ...
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"
    implementation 'com.android.support:appcompat-v7:28.0.0'
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'
    ...
    implementation 'com.google.android.gms:play-services-maps:16.0.0'
}
```

Step4 按下同步按鈕匯入 gradle 加入的 library，如圖 12-18 所示。

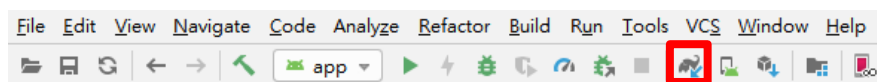


圖 12-18 同步專案完成 library 匯入

Step5 Google Maps 使用時需要透過網路、以及會需要在 AndroidManifest.xml 宣告對應的權限，同時在 AndroidManifest.xml 也需要加入 Google API 金鑰。(若要運行在 API 28 的裝置上，則需額外設定 Apache HTTP)

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="bluenet.com.lab12">

    <!-- 允許程式使用網路權限 -->
    <uses-permission android:name="android.permission.INTERNET" />

    <uses-permission <!-- 允許程式存取粗略位置 -->
        android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission <!-- 允許程式存取精確位置 -->
        android:name="android.permission.ACCESS_FINE_LOCATION" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">

        <!-- 若目標版本在 Android 28 或以上需額外設定 Apache -->
        <uses-library
            android:name="org.apache.http.legacy"
            android:required="false" />

        <!-- 此處要放入在 Google API Console 取得的 API 金鑰 -->
        <meta-data
            android:name="com.google.android.geo.API_KEY"
            android:value="YOUR_API_KEY" />

    </application>
</manifest>
```


Step6 在 activity_main.xml 中加入地圖頁面，MainActivity 暫時不做修改，如圖 12-19 所示。

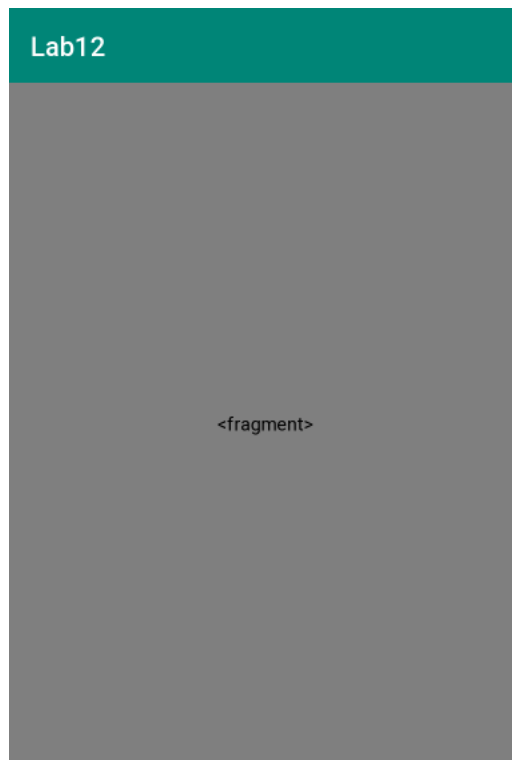


圖 12-19 SupportMapFragment

對應的 xml 如下：

```
<?xml version="1.0" encoding="utf-8"?>
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
    android:name="com.google.android.gms.maps.SupportMapFragment"
    android:id="@+id/map"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
```

Step7 選擇有支援 google-play-services 服務的模擬器或是選擇在實機上執行，如圖 12-20 所示。

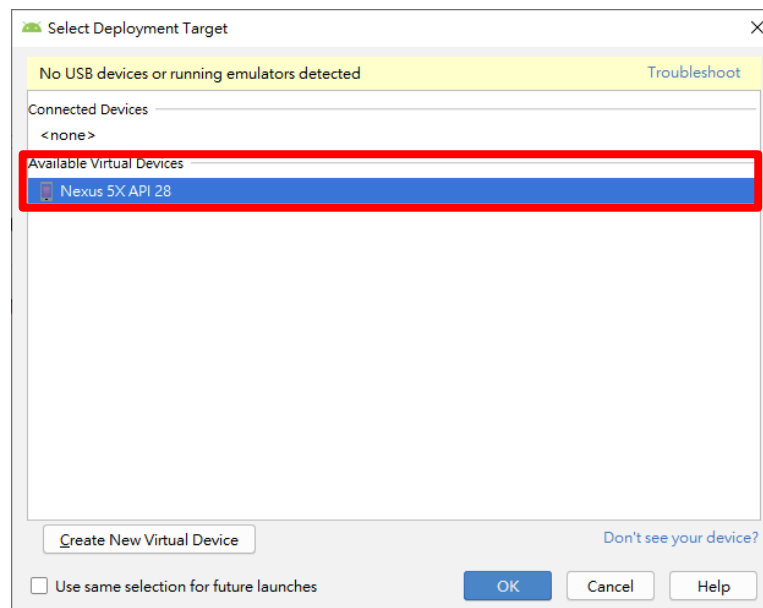


圖 12-20 選擇燒錄裝置

Step8 如果看到圖 12-21 所示的地圖畫面，就表示成功安裝好 Google Maps API。



圖 12-21 Google Map 畫面

12.2.3 Google Map 程式設計

在 MainActivity 中加入以下程式碼：

```
class MainActivity : AppCompatActivity(), OnMapReadyCallback {
    private val REQUEST_PERMISSIONS = 1
    // 權限要求結果
    override fun onRequestPermissionsResult(requestCode: Int,
        permissions: Array<String>, grantResults: IntArray) {
        if (grantResults.isEmpty()) return
        when (requestCode) {
            REQUEST_PERMISSIONS -> {
                for (result in grantResults)
                    if (result != PackageManager.PERMISSION_GRANTED)
                        finish() // 若使用者拒絕給予權限則關閉 APP
                    else{ // 連接 MapFragment 元件
                        val map =
supportFragmentManager.findFragmentById(R.id.map) as SupportMapFragment
                        map.getMapAsync(this)
                    }
                }
        }
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        // 檢查使用者是否已授權定位權限
        if (ActivityCompat.checkSelfPermission(this,
            android.Manifest.permission.ACCESS_COARSE_LOCATION) !=
            PackageManager.PERMISSION_GRANTED)
            // 檢查使用者是否已授權定位權限
            ActivityCompat.requestPermissions(this,
                arrayOf(android.Manifest.permission.ACCESS_FINE_LOCATION),
                REQUEST_PERMISSIONS)
        else{ // 連接 MapFragment 元件
            val map = supportFragmentManager.findFragmentById(R.id.map)
                as SupportMapFragment
            map.getMapAsync(this)
        }
    }
}
```

```

    }
}

override fun onMapReady(map: GoogleMap) {
    //檢查使用者是否已授權定位權限
    if (ActivityCompat.checkSelfPermission(this,
        android.Manifest.permission.ACCESS_FINE_LOCATION) !=
        PackageManager.PERMISSION_GRANTED &&
        ActivityCompat.checkSelfPermission(this,
            android.Manifest.permission.ACCESS_COARSE_LOCATION) !=
            PackageManager.PERMISSION_GRANTED)

        return

    //顯示目前位置與目前位置的按鈕
    map.isMyLocationEnabled = true
    //建立 MarkerOptions 物件
    val marker = MarkerOptions()
    marker.position(LatLng(25.033611, 121.565000))
    marker.title("台北 101")
    marker.draggable(true)
    map.addMarker(marker)
    marker.position(LatLng(25.047924, 121.517081))
    marker.title("台北車站")
    marker.draggable(true)
    map.addMarker(marker)
    //加入 PolylineOptions 到 googleMap
    val polylineOpt = PolylineOptions()
    polylineOpt.add(LatLng(25.033611, 121.565000))
    polylineOpt.add(LatLng(25.032728, 121.565137))
    polylineOpt.add(LatLng(25.047924, 121.517081))
    polylineOpt.color(Color.BLUE)
    val polyline = map.addPolyline(polylineOpt)
    polyline.width = 10f
    //移動鏡頭
    map.moveCamera(CameraUpdateFactory.newLatLngZoom(
        LatLng(25.034, 121.545), 13f))
}
}

```