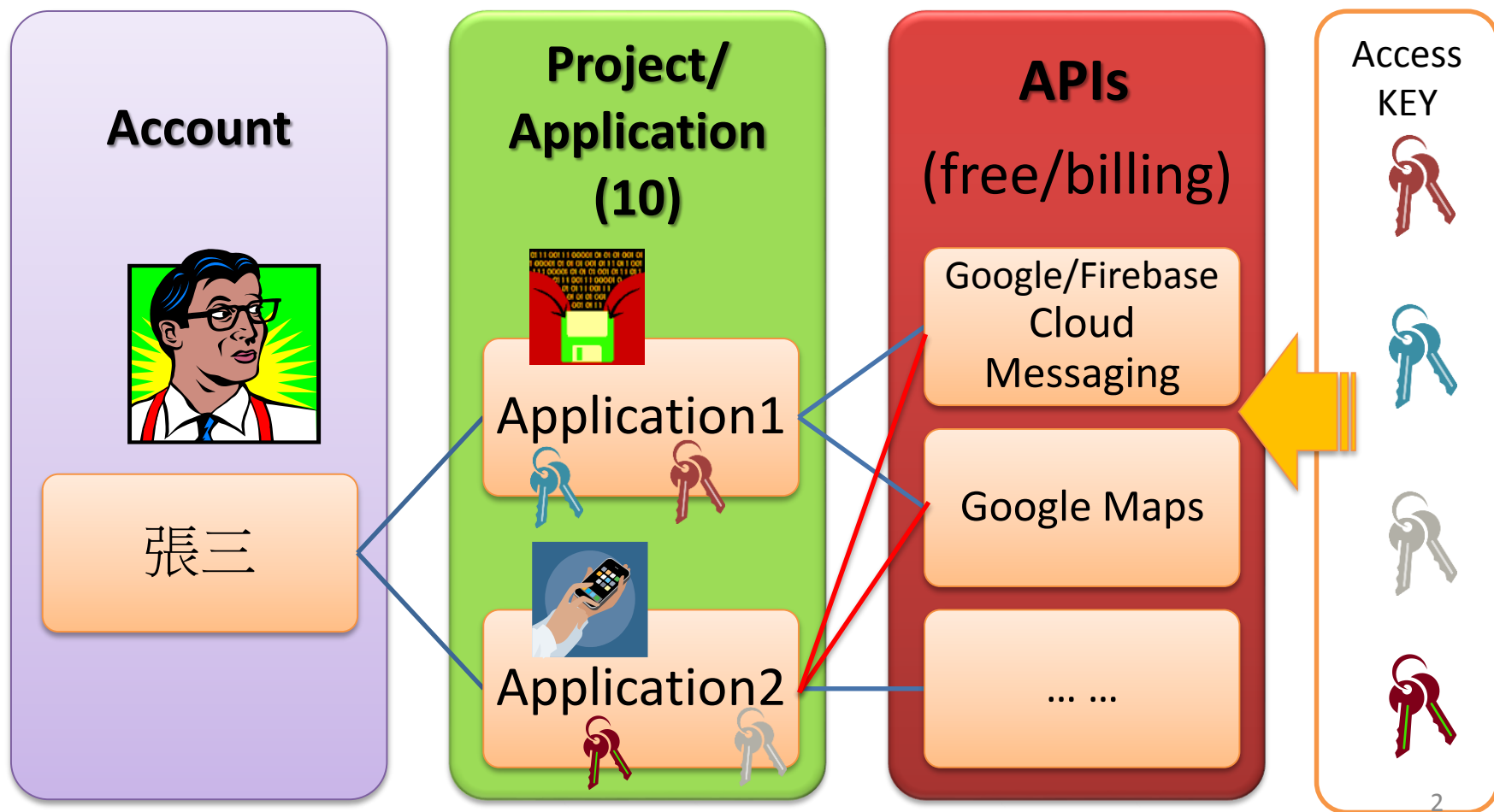




Google 雲端服務的基本觀念

- Account/Application(Project)/API Access

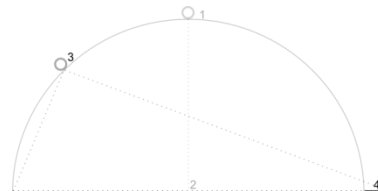
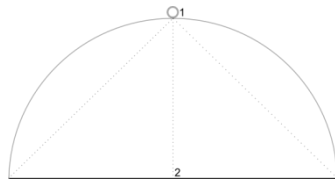


開發 Google 雲端服務之前置作業

- **Account: 申請 Google 帳號**
 - You should have!
- A. Application/Project: 註冊專案(Application ID)
 - <https://console.developers.google.com>
 - 取得**Project Number (= GCM Sender ID)**
- B. APIs : 選擇服務內容→啟用 APIs Access
 - **Google Cloud Messaging for Android**
- C. Keys: 取得存取API 的鑰匙(Register App)
 - 取得接取 **API 的Key (Server/Browser/Android/iOS)**
 - **App:必須對應到各別的套件名稱+SHA1 (Project)**

Utilize the Google Maps

- The **map view** is modeled as a **camera** looking down on a flat plane.
- The **position** of the camera is specified by the following properties:
 - **Target (latitude/longitude location)** : the Center of Map
 - **Zoom** : $256 * 2^n$ dp; $n = 0 \sim 21$.
 - **Bearing (Orientation)** : 垂直線與北極的夾角(方位角)
 - **Tilt (View Angle)** : 視角、3D 效果.



改變可視區域、縮放、視角

- 利用 **GoogleMap.moveCamera(cameraUpdate)** 來移動地圖; 或者 **GoogleMap.animationCamera(cameraUpdate, duration, callback)** : 動畫
- 利用 **CameraUpdateFactory** 可以產生多種 **CameraUpdate**

zoomIn() 、 zoomOut()	放大一倍、縮小 一倍	zoomBy(ratio)	縮放倍率
zoomTo(ratio)	指定倍率	zoomBy(ratio, point)	縮放倍率，指定 點的顯示位置保 持不變
newLatLng(lat, lng)	指定新的地圖中 心點	newLatLngZoom (lat, lng, ratio)	指定新的地圖中 心點與縮放率
newCameraPosition(CameraPosition cpos) : 多功能定位 cpos = new CameraPosition(LatLng target, float zoom, float tilt, float bearing);			
newLatLngBounds(LatLngBounds bounds, int padding) : 設定可見區域範圍 bounds = new LatLngBounds(西南點, 東北點); // ex: 方圓 5 KM			
scrollBy(float, float) : 往右、往下移動的像素(+,+)			

與使用者互動 – UI Controls

- 利用 **UiSettings** 取得使用者互動：
GoogleMap.getUiSettings()
 - *setZoomControlsEnabled(true)* : 出現控制 Bar
 - *setCompassEnabled(boolean)* : 顯示指北針
 - *setMyLocationButtonEnabled(boolean)* : 
 - *setIndoorLevelPickerEnabled(boolean)* : 設定室內縮放選單
 - *setMapToolbarEnabled(boolean)* : 顯示工具列 
 - *setXXXGesturesEnabled(boolean)* : 手勢控制

與使用者互動 - Event Handler

- 開啟 **Clickable**
 - *fragment.getView().setClickable(true);*
- 地圖 **Click** 事件
 - *setOnMapClickListener(...)*
 - *setOnMapLongClickListener(...)*
- 其他事件
 - *setOnCameraChangeListener()*
 - *setOnIndoorStateChangeListener(...)*

Demo01 - 關鍵步驟

- 申請Google 帳號 (+APIs +取得憑證Key)
 - Keytool → SHA1 (金鑰、數位簽章)
 - <https://console.developers.google.com> (Project)
- 增修清單檔案內容
 - 使用權限宣告 (v2 較為複雜)
 - meta-data 的定義 (API key)
- 布局檔內以 **fragment** 設定
 - android:name=“*com.google.android.gms.maps.MapFragment*”
 - 街景: *com.google.android.gms.maps.StreetViewPanoramaFragment*
- 需知道的重要類別
 - FragmentManager、MapFragment
 - GoogleMap、LatLng、CameraUpdate(CameraUpdateFactory)

定義新權限、使用權限

```
<permission
```

```
    android:name="tw.idv.jameschen.permission.MAPS_RECEIVE"
```

```
    android:protectionLevel="signature" />
```

```
<uses-permission android:name="tw.idv.jameschen.permission.MAPS_RECEIVE" />
```

4.3 之後版本，可以省略!

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

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

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

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

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

```
<uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
```

加入 meta-data

<meta-data

android:name="*com.google.android.maps.v2.API_KEY*"
android:value="*AlzaSyAf9TX4qCDEDH1Fa.....*" />

換成自己的 API Key

<meta-data

android:name="*com.google.android.gms.version*"
android:value="*@integer/google_play_services_version*" />

常常報錯!?

解法：完全移除現有工作區內的
Google Play Service Library.

➔ 再重新匯入!

修改布局檔：加入 **fragment**

<fragment

android:id="@+id/map"

android:layout_width="match_parent"

android:layout_height="match_parent"

class="com.google.android.gms.maps.MapFragment" />

使用 GoogleMap 的範例(指令)

- 取得 GoogleMap 物件

```
FragmentManager fmgr = getFragmentManager();
```

```
gmap = ((MapFragment)(fmgr.findFragmentById(R.id.map))).getMap();
```

- 產生定位點/中心點的座標變數(緯經度)

```
LatLng pos = new LatLng( 24.1456019, 120.6455316);
```

- 產生鏡頭更新物件(含縮放程度)

```
CameraUpdate camera1 =
```

```
CameraUpdateFactory.newLatLngZoom( pos, 17);
```

```
CameraUpdate camera2 = CameraUpdateFactory.zoomTo(17);
```

- 移動鏡頭(縮放或移動中心點)

```
gmap.animateCamera(camera1);
```

加入圖標 Markers

- 利用 *MarkerOptions* 加入地圖標籤
 - Marker* `marker = GoogleMap.addMarker(markerOptions)`
 - 圖式可利用 *BitmapDescriptorFactory* 取得 (from file, assets, bitmap, resources)
 - Marker可以定義為“可被拖曳” (long press)
 - `markerOptions.draggable(true)`
 - `marker.setDraggable(true)`
 - Marker被點擊之後會出現 information window
- Marker 相關事件處理
 - `setOnMarkerClickListener(...)`
 - `setOnMarkerDragListener(...)`
 - `setOnInfoWindowClickListener(...)`

Demo02 – 顯示地標

Marker myMarker1;

MarkerOptions opt1 = **new MarkerOptions()**

.position(myPos) // 經緯度

.title(myTitle) // info window 標題文字

.snippet("Test Snippet...") // title 下方文字

.anchor(0.5, 0.5) // 定位點

.rotation(90.0) // 旋轉角度

.icon(BitmapDescriptorFactory.fromResource(R.drawable.pointer_72));

myMarker1 = gmap.addMarker(opt1);

// 事件處理

gmap.setOnMarkerClickListener(this);

總表：GoogleMap's Interfaces

- GoogleMap.CancelableCallback
- GoogleMap.OnInfoWindowClickListener
 - GoogleMap.InfoWindowAdapter
- GoogleMap.OnCameraChangeListener
- GoogleMap.OnIndoorStateChangeListener
- GoogleMap.OnMapClickListener
- GoogleMap.OnMapLongClickListener
- GoogleMap.OnMapLoadedCallback
- **GoogleMap.OnMarkerClickListener**
- GoogleMap.OnMarkerDragListener
- GoogleMap.OnMyLocationButtonClickListener
- GoogleMap.OnMyLocationChangeListener

在Google地圖上繪圖

- 利用 Google Maps API 可以在 Google Maps 繪製客製化圖案: *Polyline*, *Polygon*, *Circle*
- 步驟：
 - 先產生必要物件(optObject : XXX)
 - 例：生成 ***PolylineOptions***物件，再利用 add(LatLng座標點) 或addAll(list) 產生一組連續點的集合.
 - 加入地圖 (畫出圖形)

GoogleMap.addXXX(optObject);

取得兩個座標點之間的路徑 (Google Direction)

- <http://maps.googleapis.com/maps/api/directions/xml?origin=24.095,120.713&destination=24.143993,120.731174&sensor=false&units=metric&mode=driving>
- <http://maps.googleapis.com/maps/api/directions/json?origin=24.095,120.713&destination=24.143993,120.731174&sensor=false&units=metric&mode=driving>

取得路徑的實際範例

```
<DirectionsResponse>
  <status>OK</status>
  <route>
    <summary>十九甲堤防</summary>
    <leg>
      <step>
        <travel_mode>DRIVING</travel_mode>
        <start_location>
          <lat>24.0949951</lat>
          <lng>120.7130006</lng>
        </start_location>
        <end_location>
          <lat>24.0954337</lat>
          <lng>120.7137313</lng>
        </end_location>
        <polyline>
          <points>w`arCgww_VAHaoAAC?CACAAACa@OYI</points>
        </polyline>
        <duration>
          <value>16</value>
          <text>1 分</text>
        </duration>
        <html_instructions>
          往<b>東</b><div style="font-size:0.9em">限制通行的道路</div>
        </html_instructions>
        <distance>
          <value>106</value>
          <text>0.1 公里</text>
        </distance>
      </step>
      <step>
        <travel_mode>DRIVING</travel_mode>
        <start_location>
          <lat>24.0954337</lat>
          <lng>120.7137313</lng>
        </start_location>
        <end_location>
          <lat>24.0953698</lat>
          <lng>120.7139689</lng>
        </end_location>
        <polyline>
          <points>mcarCy{w_VJo@</points>
        </polyline>
        <duration>
          <value>6</value>
          <text>1 分</text>
        </duration>
        <html_instructions>
          向<b>右</b>轉，朝<b>工業路</b>前進<div style="font-size:0.9em">限制通行的道路</div>
        </html_instructions>
      </step>
    </leg>
  </route>
</DirectionsResponse>
```

```
{
  "geocoded_waypoints" : [
    {
      "geocoder_status" : "OK",
      "place_id" : "ChIJGdYY4TQjaTQRcl0wDTBcAm8",
      "types" : [ "street_address" ]
    },
    {
      "geocoder_status" : "OK",
      "place_id" : "ChIJzW1HBqMiaTQR00596GxN93I",
      "types" : [ "route" ]
    }
  ],
  "routes" : [
    {
      "bounds" : {
        "northeast" : {
          "lat" : 24.1462894,
          "lng" : 120.7361193
        },
        "southwest" : {
          "lat" : 24.0949951,
          "lng" : 120.7130006
        }
      },
      "copyrights" : "地圖資料©2015 Google",
      "legs" : [
        {
          "distance" : {
            "text" : "8.0 公里",
            "value" : 7999
          },
          "duration" : {
            "text" : "19 分",
            "value" : 1160
          },
          "end_address" : "411台灣台中市太平區勤益大道",
          "end_location" : {
            "lat" : 24.143994,
            "lng" : 120.7311493
          },
          "start_address" : "412台灣台中市大里區工業路95號",
          "start_location" : {
            "lat" : 24.0949951,
            "lng" : 120.7130006
          },
          "steps" : [
            {
              "distance" : {
                "text" : "0.1 公里",
                "value" : 106
              },
              "duration" : {
```

Demo03 – 繪製路徑規劃(Route)

1. 利用 **HttpClient**取得兩個座標點之路徑規劃
 - JSON, XML
2. 將重要內容抽取出來(**MapRoute**)
 - route → step
 - start_location (經緯度)
 - end_location(經緯度)
 - polyline (points: 編碼過的座標經緯度)
 - 找出所有轉折點的座標，並將其加入 `ArrayList<LatLng>`
3. 繪製出 **Polyline**

關鍵程式框架

```
MapRoute mroute = new MapRoute();  
Document doc = mroute.getDocument( myPos1, myPos2,  
MapRoute.MODE_DRIVING);  
//  
ArrayList<LatLng> routePoint = mroute.getDirection(doc);  
PolylineOptions polyLineOpt = new PolylineOptions();  
polyLineOpt.width(3).color(Color.RED);  
for(int i = 0; i < routePoint.size(); i++) {  
    polyLineOpt.add(routePoint.get(i));  
}  
// 繪製圖案  
Polyline polylin = gmap.addPolyline(polyLineOpt);
```

找地點 (Google Place Search)

- `https://maps.googleapis.com/maps/api/place/textsearch/xml?query=修平科大&sensor=false&key=<key>`
- 需開啟 Place API
- 在取得 reference 之後，可以利用 details (取代 textsearch)，並加入 reference 參數以取得該位置的更詳細資訊。

```
<PlaceSearchResponse>
  <status>OK</status>
  <result>
    <name>修平科技大學</name>
    <type>university</type>
    <type>point_of_interest</type>
    <type>establishment</type>
    <formatted_address>412台灣台中市大里區工業路11號</formatted_address>
    <geometry>
      <location>
        <lat>24.0956880</lat>
        <lng>120.7119170</lng>
      </location>
    </geometry>
    <rating>4.0</rating>
    <icon>
      https://maps.gstatic.com/mapfiles/place_api/icons/university-71.
    </icon>
    <reference>
      CnR1AAAA75EznJ2mIghWnMwS5pIt9q_80MXk03bY3J8j9PNb7iSVMy0UCIELgxVF
    </reference>
    <id>b9052a47d2f9833ad8281e27801f6dcf3d4b6e71</id>
    <photo>
      <photo_reference>
        CmRdAAAA1cJojs23TKREWLtFCAXDDXYCZ7W7yKo_WVq9qyc_ygxmrTdc27FHUS
      </photo_reference>
      <width>1080</width>
      <height>566</height>
      <html_attribution>
        <a href="https://maps.google.com/maps/contrib/1026838914050713
      </html_attribution>
    </photo>
    <place_id>ChIJeeSI4DQjaTQRsZsEG7WHADo</place_id>
  </result>
  <html_attribution>
    提供者: <a href="http://www.kingwaytek.com/">Kingway</a>
  </html_attribution>
</PlaceSearchResponse>
```

取得臨近的景點資訊 (Google Place Search)

- <https://maps.googleapis.com/maps/api/place/search/xml?location=24.095,120.713&radius=1000&type=university&sensor=false&key=<key>>
- 需開啟 Place API, DistanceMatrix API

```
<PlaceSearchResponse>
  <status>OK</status>
  <result>
    <name>朝陽科技大學</name>
    <vicinity>霧峰區吉峰東路168號</vicinity>
    <type>university</type>
    <type>point_of_interest</type>
    <type>establishment</type>
    <geometry>
      <location>
        <lat>24.0685360</lat>
        <lng>120.7145170</lng>
      </location>
    </geometry>
    <rating>4.6</rating>
    <icon>
      https://maps.gstatic.com/mapfiles/place_api/icons/
    </icon>
    <reference>
      CnR1AAAA0TmBQAvgiCTzbTx3ohEY_cIFHsFWAAwsVxQWBcJZFv
    </reference>
    <id>5cd8b6735e4e5e21ad40b7895207a977d5df3caa</id>
    <opening_hours>
      <open_now>true</open_now>
    </opening_hours>
    <photo>
      <photo_reference>
        CmRdAAAAANV_9Di7X14h5ppi-e1wyzQqJc75XuhsK76UeQsBC
      </photo_reference>
      <width>816</width>
      <height>1088</height>
      <html_attribution>
        <a href="https://maps.google.com/maps/contrib/10
      </html_attribution>
    </photo>
    <place_id>ChIJJ874X6okaTQRBanKrGxJyAc</place_id>
    <scope>GOOGLE</scope>
  </result>
  <result>
    <name>修平科技大學</name>
    <vicinity>大里區工業路11號</vicinity>
    <type>university</type>
    <type>point_of_interest</type>
    <type>establishment</type>
    <geometry>
      <location>
        <lat>24.0956880</lat>
        <lng>120.7119170</lng>
      </location>
    </geometry>
```


Demo04 – 標示鄰近景點(Place Search)

1. 利用 Google Place Search API 功能
2. 利用 Github 套件 : `app.akexorcist.gdapllibrary`
 - GoogleDirection: 同 Demo03 的 MapRoute, 為非同步版本
 - GooglePlaceSearch: 非同步搜尋; 利用 Callback 取得結果
 - PlaceType: 提供預設的類別(分類); 例如 atm
3. 將取得的地標資訊抽出後，添加到地圖內; 並將 Marker 記錄到 List, 以利後續移除地標.

關鍵程式框架

```
1.  gp = new GooglePlaceSearch(<key>);
2.  gp.setOnPlaceResponseListener( new OnPlaceResponseListener() {
3.      public void onResponse(String status, ArrayList<ContentValues> arr_data, Document doc) {
4.          if (status.equals(GooglePlaceSearch.STATUS_OK)) {
5.              ArrayList<String> array = new ArrayList<String>();
6.              for(int i = 0 ; i < arr_data.size() ; i++) {
7.                  // 組裝 array 內容
8.                  // 建立地標參數、顯示地標並紀錄到 List
9.                  nearbyPointList.add(gmap.addMarker( opt1));
10.             }
11.             // 顯示景點清單 ListView (純文字)
12.             ArrayAdapter<String> adapter = new ArrayAdapter<String>(
13.                 MapPlaceSearch.this, R.layout.listview_text, array);
14.             listView.setAdapter(adapter);
15.         }
16.     }
17. });
```

顯示多個 Markers

```
// private ArrayList<Marker> allMarkers = new ArrayList<Marker>();  
// points : 需要標註地標的座標點物件陣列  
for ( int i = 0; i < points.size(); i++) {  
    LatLng loc = new LatLng( latitudeOfPointi, longitudeOfPointi);  
    MarkerOptions markerOpt = new MarkerOptions()  
        .position(loc)  
        .title( titleOfPointi )  
        .anchor(0.0f, 0.0f)  
        ;  
    Marker marker = googleMap.addMarker(markerOpt);  
    allMarkers.add( marker); // 儲存到集合內，以利後續運用。  
}
```

移除所有 Marker

```
// remove the OLD marker  
for (int i = allMarkers.size() - 1; i >= 0; i--) {  
    Marker marker = allMarkers.remove(i);  
    marker.remove();  
}
```

My Location



- Google 利用 [My Location] Layer 來標示使用當前在地圖上的位置
 - GoogleMap.setMyLocationEnabled(true);*
 - 地圖上會出現 My Location Button
 - 事件處理 *GoogleMap.setOnMyLocationButtonClickListener(...)*
- 若要顯示 My Location Layer, 也必須在 Manifest.xml 加入 uses permission (定位資訊)
 - ACCESS_COARSE_LOCATION
 - ACCESS_FINE_LOCATION

Demo05 - 使用 GPS 的關鍵步驟

- 宣告使用權限
 - `android.permission.INTERNET`
 - `android.permission.ACCESS_COARSE_LOCATION`
 - `android.permission.ACCESS_FINE_LOCATION`
- 定位設定 – 管理員 (**LocationManager**)
 - 必要時設定條件 (Criteria) 以取得最佳定位來源
 - 必要時，可以取得前一次已知位置(Location)
 - `locMgr.getLastKnownLocation(LocationManager.GPS_PROVIDER);`
- 註冊監聽(GPS)、解除監聽

定位設定

// 取得位置管理員

```
LocationManager locMgr = (LocationManager) getSystemService(LOCATION_SERVICE);;
```

// 定義定位條件

```
Criteria mCriteria01 = new Criteria();
```

```
mCriteria01.setAccuracy(Criteria.ACCURACY_FINE);
```

```
mCriteria01.setAltitudeRequired(false);
```

```
mCriteria01.setBearingRequired(false);
```

```
mCriteria01.setCostAllowed(true);
```

```
mCriteria01.setPowerRequirement(Criteria.POWER_LOW);
```

// 根據條件來取得最佳的位置服務來源

```
String strLocationPrvider = locMgr.getBestProvider( mCriteria01 , true);
```

```
//
```

```
Location location1, location2;
```

// 讀取上一次的已知位置

```
location1=locMgr.getLastKnownLocation(LocationManager.GPS_PROVIDER);
```

```
location2=locMgr.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
```


註冊與移除監聽(定位服務、GPS)

// 註冊: 位置更新事件

```
locMgr.requestLocationUpdates(LocationManager.GPS_PROVIDER, 100, 0, this);  
locMgr.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 1000, 0, this);
```

實作 LocationListener

// 註冊: GPS 狀態變動事件

```
locMgr.addGpsStatusListener(this);
```

實作 GpsStatusListener

// 設定接近指定地點的警示

```
locMgr.addProximityAlert(double lat, double lng, float r, long  
expiration, PendingIntent intent);
```

// 必要時，移除事件註冊

```
locMgr.removeUpdates(this);  
locMgr.removeGpsStatusListener(this);
```

LocationListener/GpsStatusListener

- 覆寫 LocationListener 主要方法

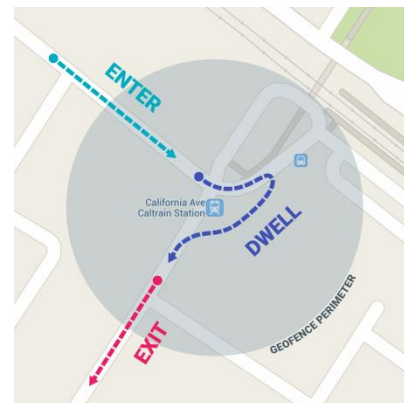
```
public void onLocationChanged(Location location) {  
    // location.getLongitude(), location.getLatitude()  
    // updateLocation(location);  
}
```

- 覆寫 GpsStatusListener 主要方法

```
public void onGpsStatusChanged(int event) {  
    // GpsStatus.GPS_EVENT_FIRST_FIX  
    // GpsStatus.GPS_EVENT_SATELLITE_STATUS  
    // GpsStatus.GPS_EVENT_STARTED  
    // GpsStatus.GPS_EVENT_STOPPED  
}
```

利用 Google Play services 存取Location

- 利用 Google Play Service 整合定位APIs 可以達到以下主要功能
 - 手持設備的定位
 - 傾聽位置的改變
 - 當設備移動時，得知移動的模式
 - 創建、監聽地理區域的事件(GeoFence)
- **FusedLocationProvider** API
 - 整合定位來源，提供方便存取方式



<https://developer.android.com/training/location/index.html>

<https://developer.android.com/training/building-location.html>

利用 GoogleApiClient 讀取定位

1. 建立 Google API Client

```
mGoogleApiClient = new GoogleApiClient.Builder(this)  
    .addConnectionCallbacks(this)  
    .addOnConnectionFailedListener(this)  
    .addApi(LocationServices.API)  
    .build();
```

2. 讀取資料(讀取最近一次的定位資訊) // **onConnected(...)**


```
mLastLoc =  
    LocationServices.FusedLocationApi.getLastLocation( mGoogleApiClient );  
if (mLastLoc != null) {  
    mLatitudeText.setText(String.valueOf( mLastLoc.getLatitude() ));  
    mLongitudeText.setText(String.valueOf( mLastLoc.getLongitude() ));  
}
```

註冊定位位置更新

- 進行位置更新註冊: 需先定義好 **LocationRequest**(更新參數)
LocationServices.FusedLocationApi.requestLocationUpdates(
mGoogleApiClient, **mLocationRequest**, *this*);

// 實作 **LocationListener**...

```
public void onLocationChanged(Location location) {  
    mCurrentLocation = location;  
    // ... updateUI();  
}
```



- 移除傾聽

LocationServices.FusedLocationApi.removeLocationUpdates(
mGoogleApiClient , *this*);

設定 **LocationRequest** 參數

- 更新頻率
 - *setInterval(max)* : 最慢不要慢於 max ms
 - *setFastestInterval(min)* : 最快不要快過 min ms

- 更新優先順序、條件
 - *setPriority(...)*

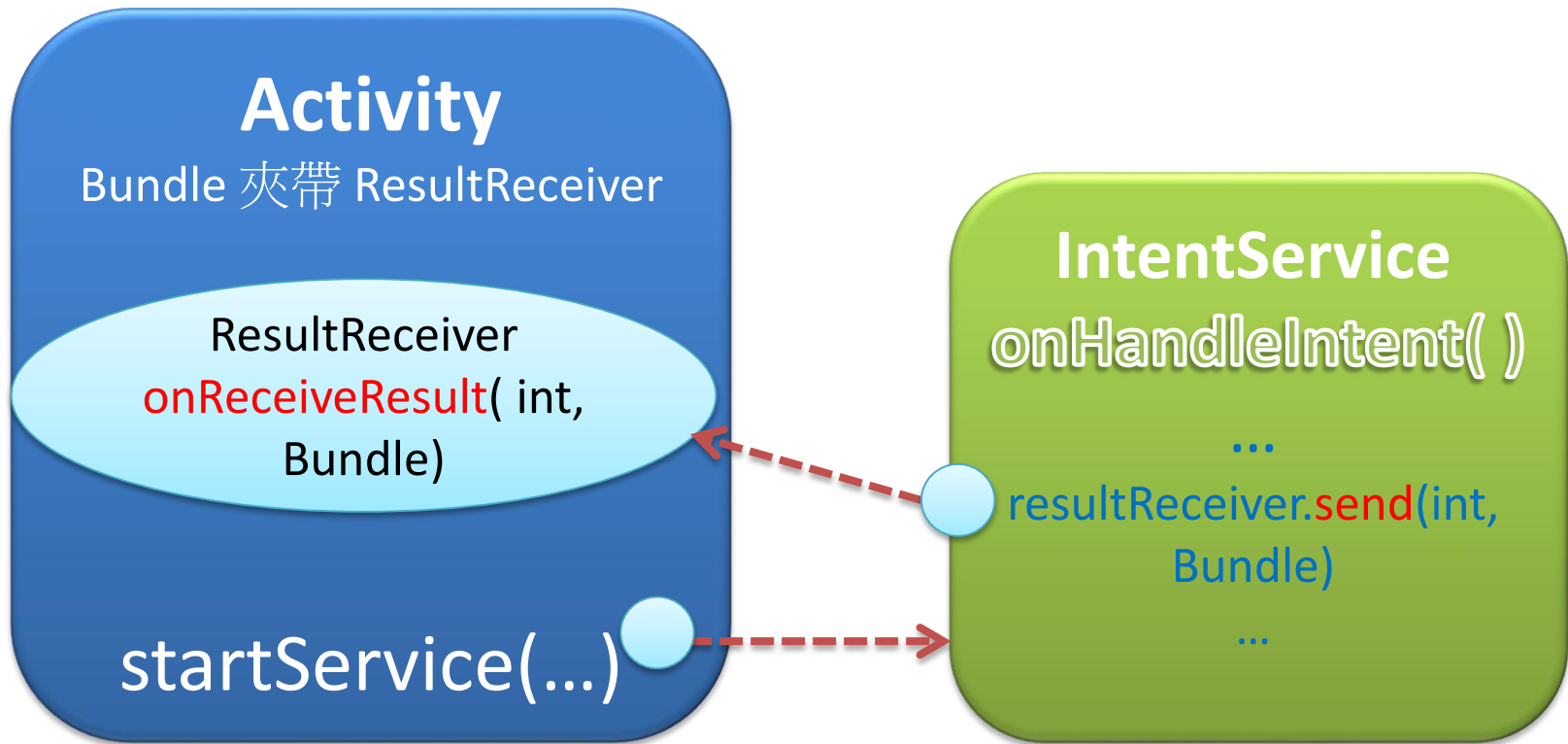
- 範例：

```
LocationRequest mLocReq = new LocationRequest();  
mLocReq.setInterval(10000);  
mLocReq.setFastestInterval(5000);  
mLocReq.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
```

由位置找街道、地址

- 主要方法 **Geocoder**.*getFromLocation(location)*是一個同步方法呼叫，所以不適合直接放在 UI Thread.
 - 必須以非同步方式取得地址資料; 例如：可利用 **IntentService** + **ResultReceiver** (IPC), ...
- 使用步驟
 - 在 *IntentService.onHandleIntent(Intent)* 處理
Geocoder geocoder = new **Geocoder**(this, Locale.getDefault());
List<Address> addresses = geocoder.getFromLocation(location);
Address addr = addresses.get(i); // 多筆住址符合
String addrSegment = addr.getAddressLine(k); // 多行資料
// 利用 ResultReceiver 將結果回傳
resultReceiver.send(RESULT_OK, bundle);

結合 IntentService, ResultReceiver



(*1) 螢幕座標與地理座標轉換

Projection proj = googleMap.getProjection();

// 假設貼全螢幕圖片 → 對應的實際地理座標

Point p1 = new Point(0, 0);

LatLng ul = proj.fromScreenLocation(p1);//左上角

Point p2 = new Point(img_width-1, img_height-1);

LatLng lr = proj.fromScreenLocation(p2);//右下角

(*2) 圖層處理



- **TileOverlay**: A tile overlay, sometimes called a tile layer, is a collection of images that are displayed on top of the base map tiles.
 - Ex: use tile overlays to add extra features to the map by providing **transparent tile images**.
 - Need to provide the **tiles for each zoom level** that you want to support.
 - Tile **coordinates** and **zoom levels** → Level 2: 4*4 tiles
 - Tile: 512 * 512 pixel
- **GroundOverlay**

<https://developers.google.com/maps/documentation/android/tileoverlay>
<http://www.maptiler.com/>

圖層處理 - GroundOverlay



- **TileOverlay**
- **GroundOverlay** are image overlays that are tied to latitude/longitude coordinates, so they move when you drag or zoom the map.
 - An image that is fixed to a map;
 - Rotating, tilting or zooming the map will change the orientation of the image;

範例：GroundOverlay 圖層處理

GroundOverlay = GroundOverlayOptions, LatLngBounds, BitmapDescriptor

GroundOverlay *imageOverlay*;

// 貼上圖層 -- 範圍、圖片、透明度

LatLngBounds *bounds* = new *LatLngBounds*(
 new *LatLng*(this.lr.latitude, this.ul.longitude), /*西南角*/
 new *LatLng*(this.ul.latitude, this.lr.longitude)); /*東北角*/

BitmapDescriptor *imageDesc* =
BitmapDescriptorFactory.fromBitmap(image);

GroundOverlayOptions *groundOverlay* = new *GroundOverlayOptions*()
 .image(*imageDesc*)
 .positionFromBounds(*bounds*)
 .transparency(alpha / 100f);

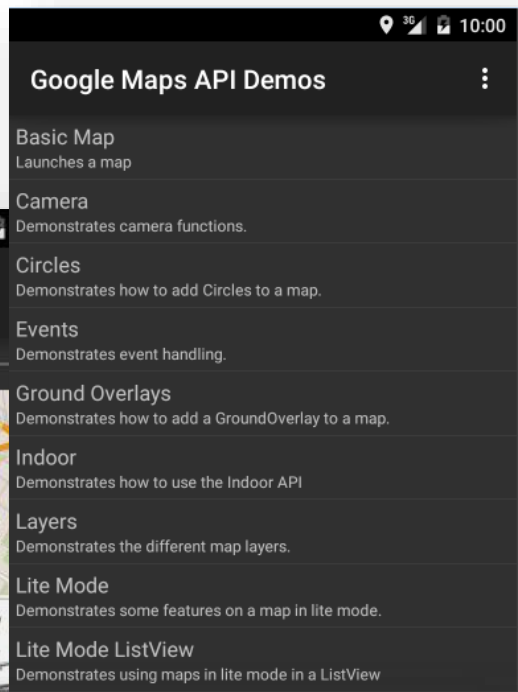
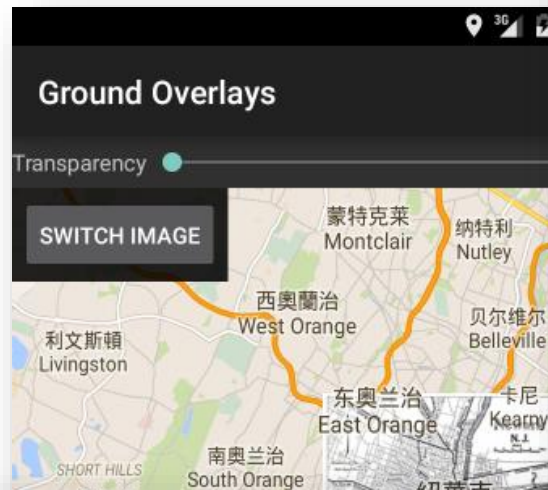
groundOverlay.visible(true);

imageOverlay = *googleMap*.addGroundOverlay(*groundOverlay*);

// 適時移除圖層
imageOverlay.remove();

無法顯示地圖 (空白)?!

- Key ?
- 套件名稱?
- 註冊?



Google Maps Android API Authorization failure. Please see <https://developers.google.com/maps/documentation/android/start> for how to correctly set up the map.

Google Maps Android API In the Google Developer Console (<https://console.developers.google.com>) Ensure that the "Google Maps Android API v2" is enabled.

Google Maps Android API Ensure that the following Android Key exists:

Google Maps Android API API Key: AIzaSyBsg6bQMmzwjICDVluvoGx24wsFM_du3kI

Google Maps Android API Android Application (<cert_fingerprint>;<package_name>): C3:66:EC:53:AF 0 :04:C9:C7:75:43:51:DE:C3:B9:88:B2:47:66:23:73;tw.idv.jameschen.mapv2demo

Google Maps Android API Failed to contact Google servers. Another attempt will be made when conn 0 activity is established.

Google Maps Android API Failed to load map. Error contacting Google servers. This is probably an 0 authentication issue (but could be due to network errors).