

# ANDROID 進階講義

吳柏翰

資料處理

# OUTLINE

- 資料格式
- Google雲端訊息
- 資料存取
- SQLite

# OUTLINE

- 資料格式
  - JSON格式
  - XML格式
- Google雲端訊息
- 資料存取
- SQLite

# JSON - 特性

- JSON ( JavaScript Object Notation ) 特性
  - 一種輕量級的資料交換語言
  - 以文字為基礎，且易於讓人閱讀
  - Javascript的一個子集，但為獨立於語言的文字格式，並且採用了類似於C語言家族的一些習慣

# JSON - 格式範例

- JSON格式範例

```
var employee = {  
  "fullname" : "Sam Kelly",  
  "telephones" : [  
    {"type" : "work" , "value" : "123-4567"},  
    {"type" : "home" , "value" : "987-6543"},  
  ],  
  "addresses" : [  
    {"type" : "work" , "value" : "11 1st Ave"},  
    {"type" : "home" , "value" : "22 Main St"},  
  ]  
}
```

# JSON - 格式 /1

- JSON格式
  - 物件 ( object )
    - 一個物件以「{」開始，並以「}」結束。
    - 一個物件包含一系列非排序的名稱 / 值對
    - 每個名稱 / 值對之間使用「,」分割
  - 名稱 / 值 ( collection )
    - 名稱和值之間使用「:」隔開
    - 名稱為一字串
    - 值為字串、數值、物件、布林值、陣列或是null
    - {name:value}
    - 多筆資料使用「,」分隔
  - 陣列 ( Array )
    - 使用中括弧「[ ]」將資料收集
    - 多筆資料使用「,」分隔
    - [collection, collection, collection]

# JSON - 格式 /2

- JSON格式
  - 字串
    - 以「"”」括起來的一串字元
  - 數值
    - 以0~9表示之正負數或小數
    - 「e / E」表示為指數形式
  - 布林值
    - 「true / false」表示
  - 範例
    - `{"status":"Fail","reason":"questType empty!"}`

# JSON – 解析

- 使用Android內建資料庫完成
  - 使用org.json.JSONObject類別
  - JSONObject(欲解析字串)
    - 將原始字串解析，並回傳JSONObject物件
  - get(name)
    - 取得name的value
  - getJSONArray(name)
    - 透過getJSONObject(index)取得JSONObject物件
  - getXXX(name), 如Int, Double, String...等。
    - 取得name對應的資料，並回完對應的資料格式



# JSON - 實作範例

- 解析

- <http://www.realtouchapp.com/test/ph/json.html>
- 範例請參考「SampleWebFormat」

```
try {
    String jsonString = HttpURLConnection.sendPost(
        "http://www.realtouchapp.com/test/ph/json.php", "UTF-8");
    //Log.e("JSONString", jsonString);

    //因為網站格式為以Object為開頭，因此將網站取得的資料進行JSONObject解析
    JSONObject jsonRoot = new JSONObject(jsonString);
    JSONArray phonesarr = jsonRoot.getJSONArray("telephones");
    Log.e("phonesarr", phonesarr.toString());

    JSONArray addressesarr = jsonRoot.getJSONArray("addresses");
    for (int i = 0; i < addressesarr.length(); i++) {
        JSONObject jsonObject = addressesarr.getJSONObject(i);
        Log.e("value-" + String.valueOf(i), jsonObject.getString("value"));
    }
} catch (JSONException e) {
    // TODO Auto-generated catch block
    //e.printStackTrace();
    Log.e("JSONException", e.toString());
}
```

# XML格式

- eXtensible Markup Language(XML)
  - 可延伸標記式語言
  - 一種標記式語言
- XML
  - 傳送及攜帶資料訊息
  - 用途為說明資料是什麼，以及攜帶資料訊息
  - HTML語言則用來表現資料
  - 不支援分影格 ( framing )
- XML定義
  - 結構
  - 儲存訊息
  - 傳送訊息

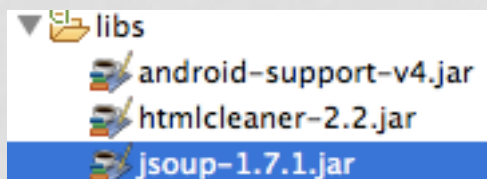
# XML - 格式範例

- XML格式範例

```
<?xml version='1.0' encoding='UTF-8'?>
<employee>
  <fullname> Sam Kelly </fullname>
  <telephones>
    <tel type="work"> 123-4567 </tel>
    <tel type="home"> 987-6543 </tel>
  </telephones>
  <addresses>
    <address type="work"> 11 1st Ave </address>
    <address type="home"> 22 Main St </address>
  </addresses>
</employee>
```

# XML WITH JSOUP

- 於專案libs/目錄下複製「jsoup-xxx.jar」
  - 可自行於網路上下載最新版



- 操作流程
  - 「Jsoup.connect().xxx()」取得Document資料
  - 透過「getxxx ("key")」取得Element資料

# XML WITH JSOUP - GET

- 基本架構
  - 透過GET的方式取得網頁HTML(XML)
    - Jsoup.connect("網址").get();

```
//解析淡江時報1
try {
    //Document doc = Jsoup.connect("http://gdc.tku.edu.tw/TodayNews/ashx/RssNews.ashx").get();
    Document doc = Jsoup.connect("http://tkutimes.tku.edu.tw/ashx/ChRss.ashx").get();
    |
} catch (MalformedURLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
```

# XML WITH JSOUP - POST

- 傳送POST
  - 透過POST傳送給網站參數與值
  - `.data("key", "value")`

```
try {  
    Document doc = Jsoup.connect("http://163.13.132.108/sample/post.php").  
        data("data1", "test1").  
        data("data2", "test2").  
        post();  
    String str = doc.html();  
    String strUTF8 = new String(str.getBytes("big5"), "UTF-8");  
    Log.e("POST", strUTF8);  
} catch (IOException e) {  
    // TODO Auto-generated catch block  
    e.printStackTrace();  
}
```

# XML WITH JSOUP - 取得資料

- 取得資料
  - Element: 元件資料
  - Elements: 元件列表
  - 取得元件中所有旗標名稱的元件列表
    - `getElementsByTag("旗標名稱")`

```
//Element body = doc.body();  
//Element channel = body.getElementsByTag("channel").get(0);  
//Elements items = channel.getElementsByTag("item");
```

- 直接取得資料中所有旗標的元件列表
  - `select("旗標名稱");`

```
Elements items = doc.select("item");  
Element item = items.get(0);
```

# XML WITH JSOUP – 解析資料

- 解析資料
  - tagName(): 取得旗標名稱

```
for (int i = 0; i < item.childNodes().size(); i++) {
    org.jsoup.nodes.Node node = item.childNodes(i);
    if(node.getClass() != Element.class) {
        continue;
    }
    Element el = (Element) node;
    if(el.tagName().compareTo("description") == 0) {
        //link如果資料為空，下一筆資料應為連結網址
        if(el.text().compareTo("") == 0) {
            if((i+1) < item.childNodes().size()) {
                org.jsoup.nodes.Node no = item.childNodes(i+1);
                //Toast.makeText(this, no.toString(), Toast.LENGTH_LONG).show();
            } else {
                continue;
            }
        } else {
            //Toast.makeText(this, el.text(), Toast.LENGTH_LONG).show();
        }
    }
}
```



# XML WITH JSOUP - 實作範例

- 解析

- <http://www.realtouchapp.com/test/ph/xml.html>
- 範例請參考「SampleWebFormat」

```
//XML sample of JSoup
try {
    Document doc = Jsoup
        .connect("http://www.realtouchapp.com/test/ph/xml.html")
        .get();
    Element body = doc.body();
    Elements employees = body.getElementsByTag("employee");
    for(int i = 0; i < employees.size(); i++) {
        //解析資料
    }
} catch (MalformedURLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
    Log.e("JSoup", e.toString());
} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
    Log.e("JSoup", e.toString());
}
catch (NullPointerException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
    Log.e("JSoup", e.toString());
}
```

```
//解析資料
DateFormat thisData = new DateFormat();
Element employee = employees.get(i);
Elements fullnames = employee.getElementsByTag("fullname");
if(fullnames.size() > 0) {
    Element fullname = fullnames.get(0);
    thisData.fullname = getElementData(fullname);
}

Elements telephones = employee.getElementsByTag("telephones");
for(int j = 0; j < telephones.size(); j++) {
    Element tel = telephones.get(j);
    thisData.telephones
        .put(tel.attr("type"), getElementData(tel));
}

Elements addresses = employee.getElementsByTag("addresses");
for(int j = 0; j < addresses.size(); j++) {
    Element address = addresses.get(j);
    thisData.addresses
        .put(address.attr("type"), getElementData(address));
}

data.add(thisData);
```

# XML WITH SAX

- 大量資料時，速度比較快
- Android內建，不需要複製任何函式庫
- 需自行設計資料結構
- 需自行設計處理類別
  - 需繼承DefaultHandler
  - 「startDocument()」開始解析XML文件
  - 「startElement()」讀取到元件開頭(<target>)
  - 「endElement()」讀取元件結尾(</target>)
  - 「endDocument()」結束解析XML文件

# XML WITH SAX - DEFAULTHANDLER

- 需 extends DefaultHandler

```
public class mySAXParserHelper extends DefaultHandler {
```

- 定義資料格式類別

```
//儲存資料  
public class format_road  
{  
    public String name;  
    public List<String> points = new ArrayList<String>();  
    public String distance_before;  
    public String distance;  
    public String through;  
}
```

- 宣告一儲存實體

```
public format_road data = new format_road();
```

# XML WITH SAX - DOCUMENT

- XML文件開始與結束
  - 直接呼叫父函式

```
//XML文件開始解析時呼叫此method  
public void startDocument() throws SAXException {  
    super.startDocument();  
}
```

```
//XML文件結束解析時呼叫此method  
public void endDocument() throws SAXException {  
    // TODO Auto-generated method stub  
    super.endDocument();  
}
```

- 取得旗標之間的資料

```
//取得Element的開頭結尾中間夾的字串  
private StringBuffer buffer=new StringBuffer();  
public void characters(char[] ch, int start, int length)throws SAXException {  
    buffer.append(ch,start,length);  
}
```

# XML WITH SAX – ELEMENT START

- 旗標開始
  - localName: 旗標名稱
  - attributes: 旗標屬性

```
//解析到Element的開頭時呼叫此method
public void startElement(String uri, String localName, String qName, Attributes attributes)
    throws SAXException {
    Log.e("start", localName);
    buffer.setLength(0);

    if(localName.compareTo("roadmap") != 0) return;
    for(int i = 0; i < attributes.getLength(); i++) {
        String key = attributes.getQName(i);
        if(key.compareTo("name") == 0) {
            data.name = attributes.getValue(i);
            break;
        }
    }
}
```

# XML WITH SAX – ELEMENT END

- 旗標結束
  - 儲存資料

```
//解析到Element的結尾時呼叫此method
public void endElement(String uri, String localName, String qName) throws SAXException {
    String str = buffer.toString();
    Log.e("end", localName+": "+str);

    str = str.replace("\t", "");
    str = str.replace("\n", "");

    if(localName.compareTo("vector3") == 0) {
        data.points.add(str);
    }
    else if(localName.compareTo("distance_before") == 0) {
        data.distance_before = str;
    }
    else if(localName.compareTo("distance") == 0) {
        data.distance = str;
    }
    else if(localName.compareTo("through") == 0) {
        data.through = str;
    }

    buffer.setLength(0);
}
```

# XML WITH SAX - PARSE

- 靜態函式，開啟與解析XML文件

```
public static format_road parseXML(InputStream inStream)
{
    try {
        SAXParserFactory factory=SAXParserFactory.newInstance();
        SAXParser parser = factory.newSAXParser();
        //實例化SAXParser對象，創建XMLReader對象，解析器
        XMLReader xmlReader=parser.getXMLReader();
        //實例化handler，事件處理器
        mySAXParserHelper helperHandler=new mySAXParserHelper();
        //解析器註冊事件
        xmlReader.setContentHandler(helperHandler);
        //解析文件
        xmlReader.parse(new InputSource(inStream));
        return helperHandler.data;
    } catch (SAXException e) {
        e.printStackTrace();
    } catch (ParserConfigurationException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
    return null;
}
```

# XML WITH SAX - 實作範例

- 解析

- <http://www.realtouchapp.com/test/ph/xml.html>
- 範例請參考「SampleWebFormat」

```
//XML sample of SAX
try {
    InputStream is = new
        URL("http://www.realtouchapp.com/test/ph/xml.html")
        .openStream();
    data.addAll(SAXPraserHelper.parseXML(is));
    //data = SAXPraserHelper.parseXML(is);
} catch (MalformedURLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
    Log.e("SAX", e.toString());
} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
    Log.e("SAX", e.toString());
}
catch (NullPointerException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
    Log.e("SAX", e.toString());
}
```



# 實作練習

- JSON
  - 解析台北市政府健康資訊
  - 使用ListView顯示所有標題
- XML
  - 解析淡江時報電子報
    - <http://tkutimes.tku.edu.tw/ashx/ChRss.ashx>
  - 使用ListView顯示所有標題

# OUTLINE

- 資料格式
- **Google**雲端訊息(GCM)
  - Client
  - Server
- 資料存取
- SQLite

# ANDROID (推播機制) /1

- Android Cloud to Device Messaging (C2DM )
  - 於2012-6-26正式廢除

**Important:** C2DM has been officially deprecated as of June 26, 2012. This means that C2DM has stopped accepting new users and quota requests. No new features will be added to C2DM. However, apps using C2DM will continue to work. Existing C2DM developers are encouraged to migrate to the new version of C2DM, called [Google Cloud Messaging for Android](#) (GCM). See the C2DM-to-GCM [Migration](#) document for more information. Developers must use GCM for new development.

GCM 取代了 C2DM (Android 雲端至裝置簡訊服務) 測試版。請注意，C2DM 即將終止服務，因此我們將不再受理任何新的 C2DM 人工申請。尚未切換至 GCM 的開發人員仍可繼續使用 C2DM，但其可用配額將不會增加。

# ANDROID (推播機制) /2

- Google Cloud Messaging for Android (GCM)
  - Google雲端通訊(GCM)
  - <https://support.google.com/googleplay/android-developer/answer/2663268?hl=zh-Hant>
  - 開發人員可透過該服務將伺服器中的資料傳送給 Android 應用程式使用
  - 可以是一個輕量型訊息，用來通知 Android 應用程式去擷取來自伺服器的新資料
  - 也可以是一個含有裝載資料 (最多 4 KB) 的訊息，以供即時通訊這類的應用程式可直接接收訊息內容
  - 訊息的存活最長時間為4週(2419200秒)

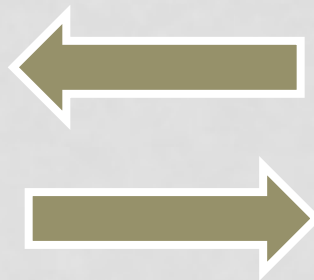
# GCM流程

- 流程

1 向Google GCM Server 註冊 Registration ID

Device

Google GCM Server

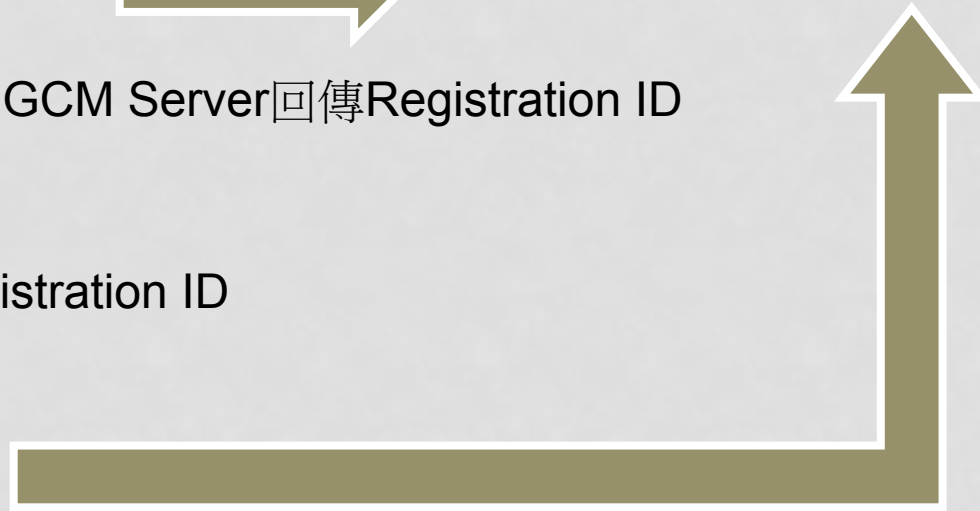


2 GCM Server回傳Registration ID



3 Registration ID

Web Service Server (開發者)



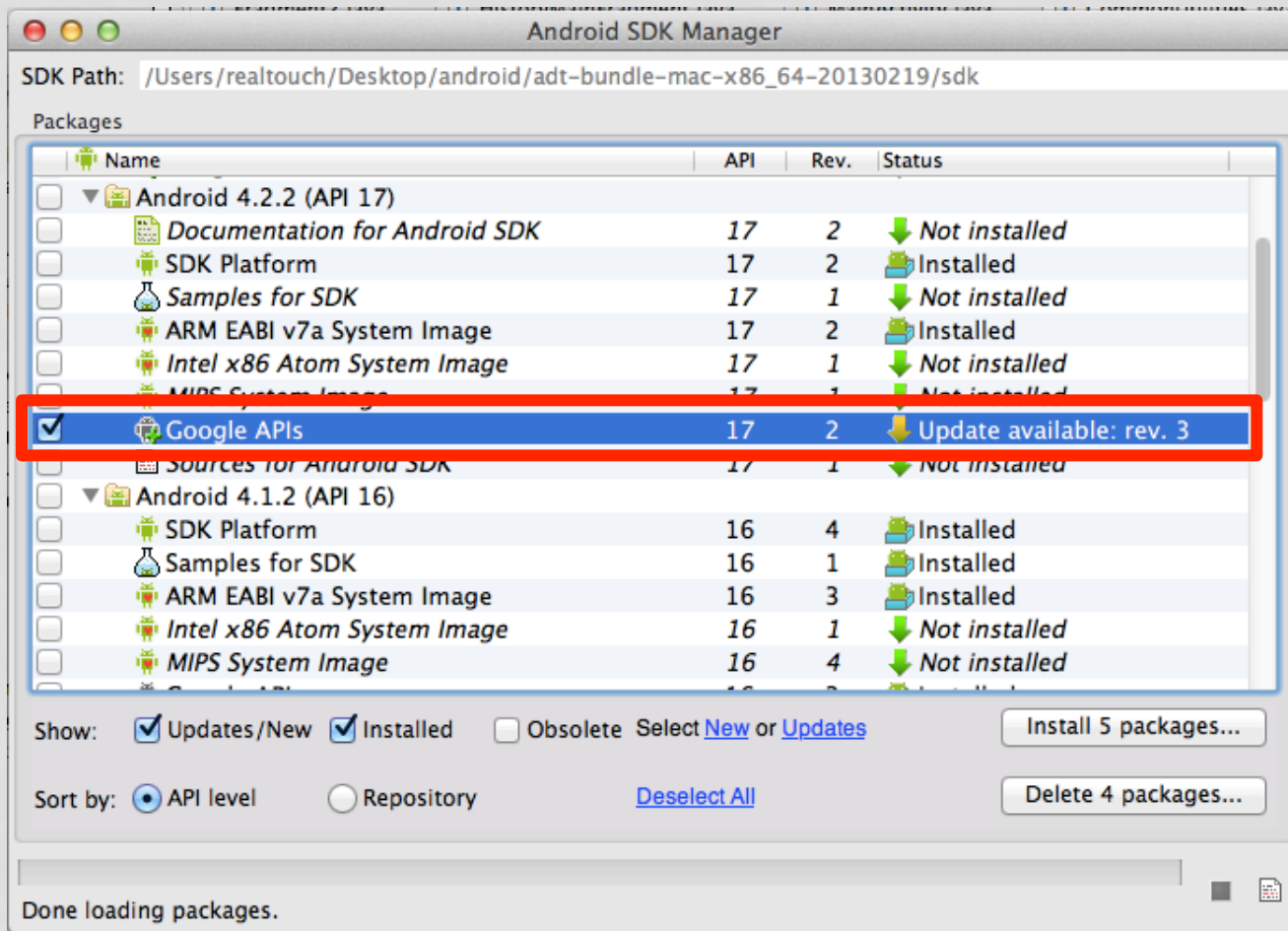
4 API Key, Registration ID, Message data

# CLIENT

- 下載支援套件
- 新增Google Apls模擬器
- 登入Google帳號
- GCM專案
  - 複製gcm.jar至專案中的libs/目錄下
  - 修改專案設定檔AndroidManifest.xml
  - 複製相關字串格式至「strings.xml」
  - 專案程式撰寫
  - 範例請參考「SampleGCM」

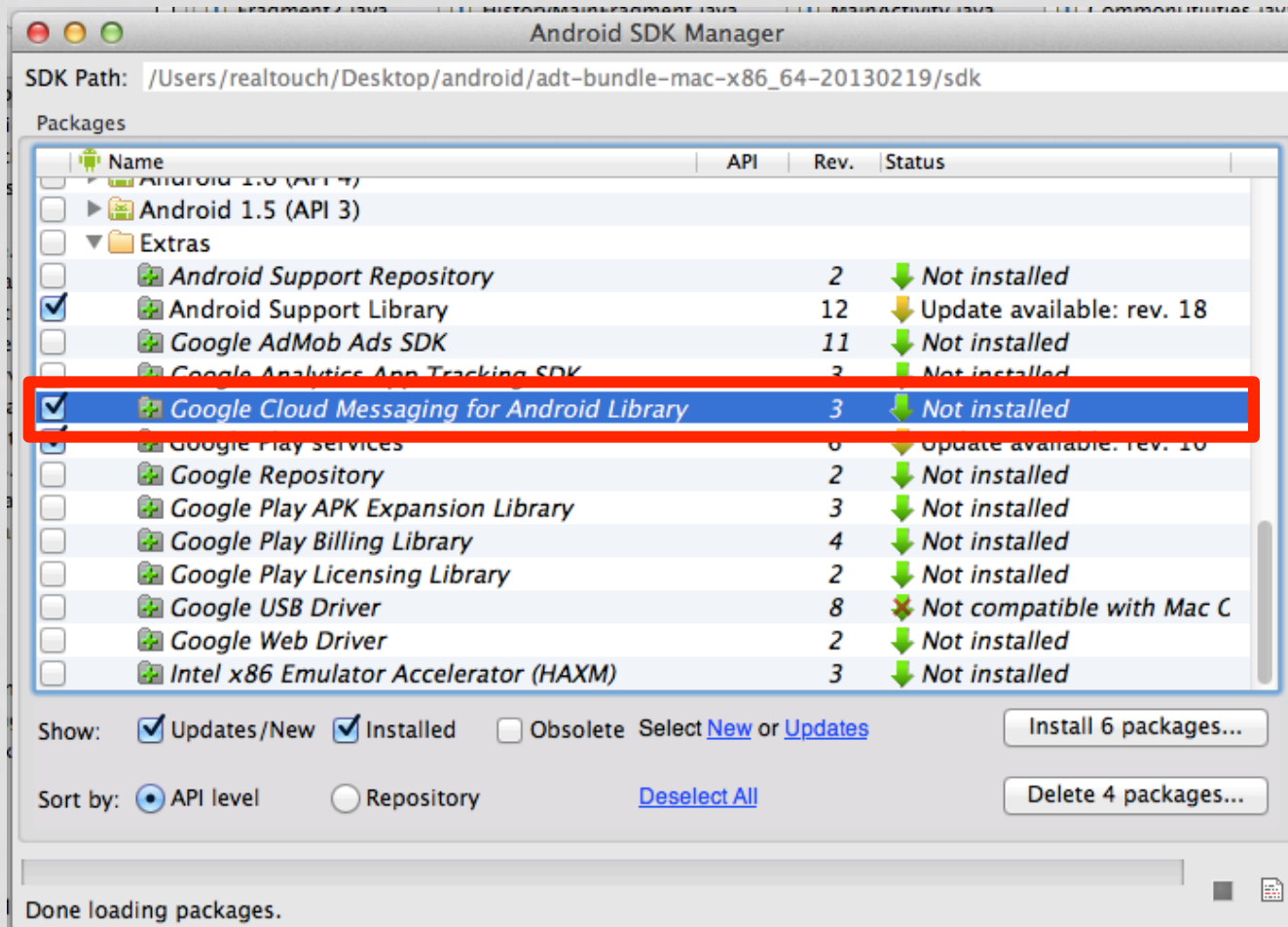
# 下載支援套件 /1

- 欲測試版本的「Google APIs」



# 下載支援套件 /2

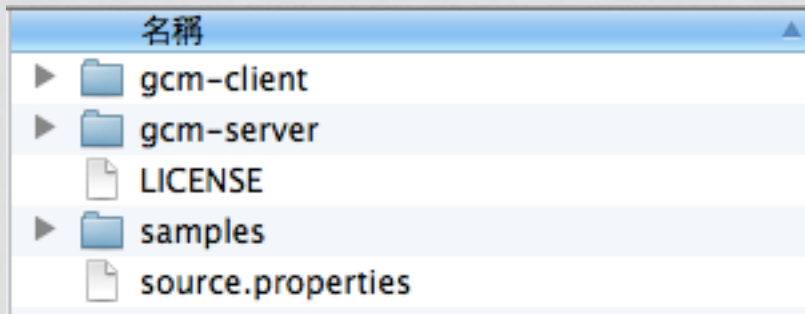
- Extras-> 「Google Cloud Messaging for Android Library」





## 下載支援套件 /3

- SDK Manager會在您的/android sdk安裝目錄/extras/google/下建立gcm/目錄
  - 包含子目錄gcm-client, gcm-server, samples/gcm-demo-client, samples/gcm-demo-server, samples/gcm-demo-appengine

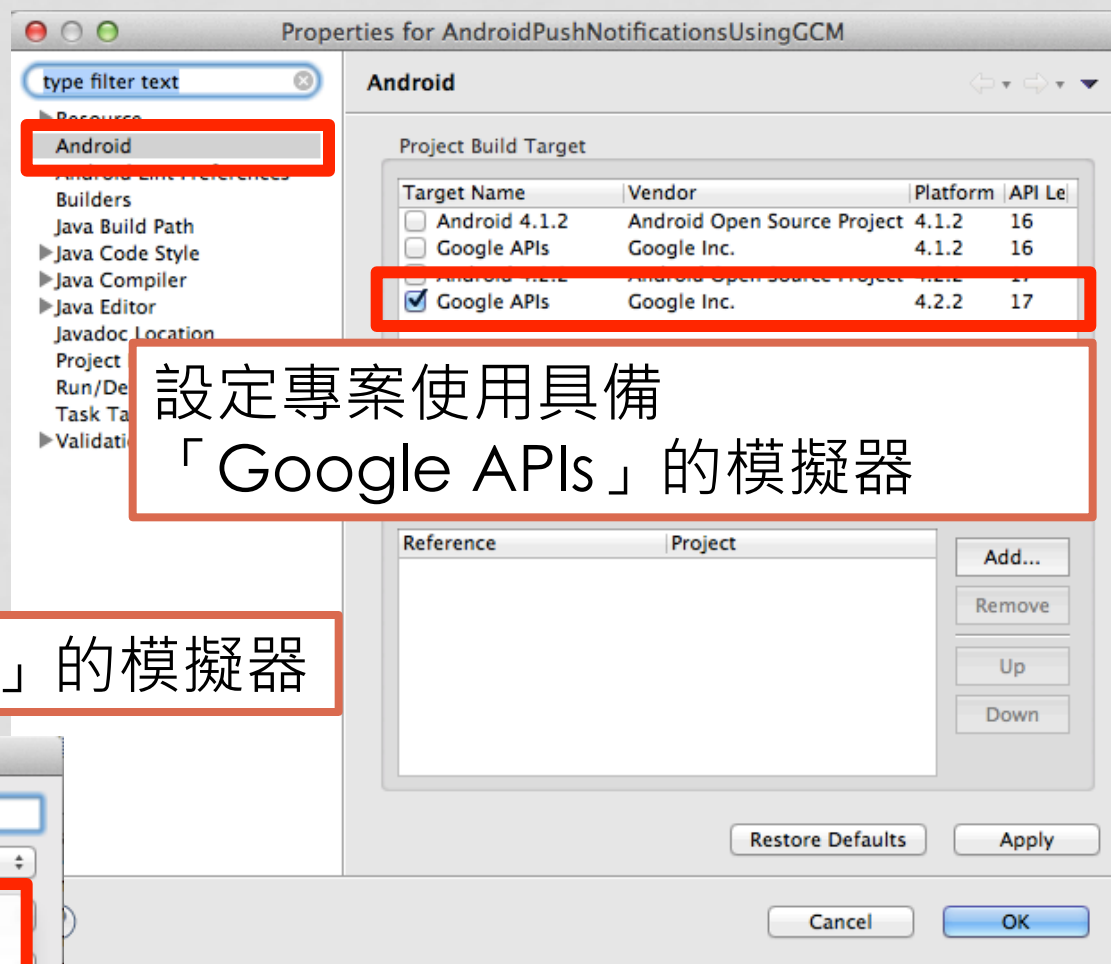


# 下載支援套件 /4

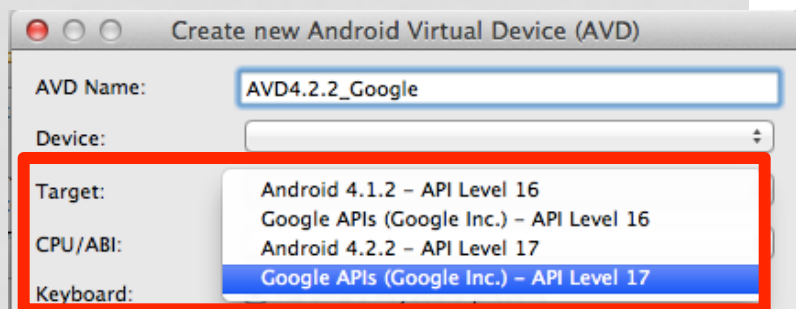
- gcm/gcm-client/dist/目錄
  - 找到gcm.jar，並複製到您app專案中的的/libs/



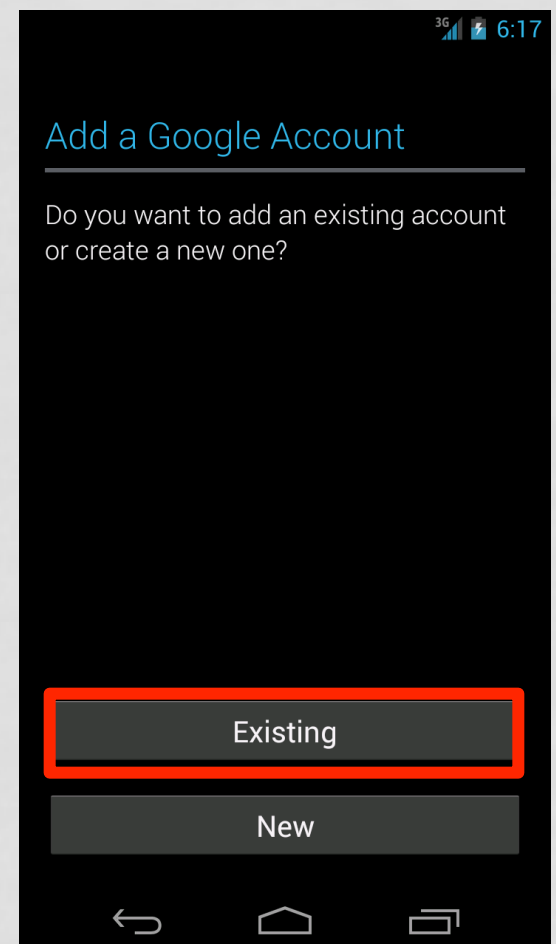
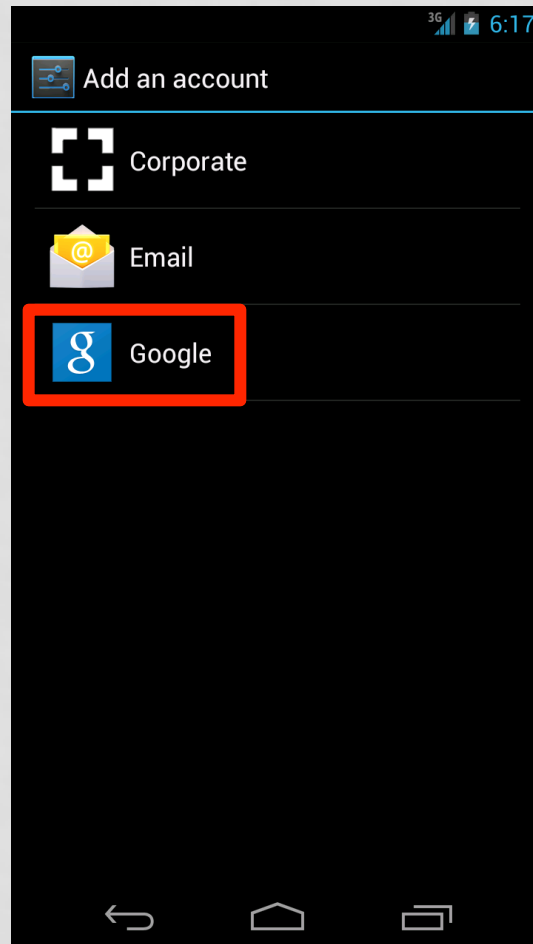
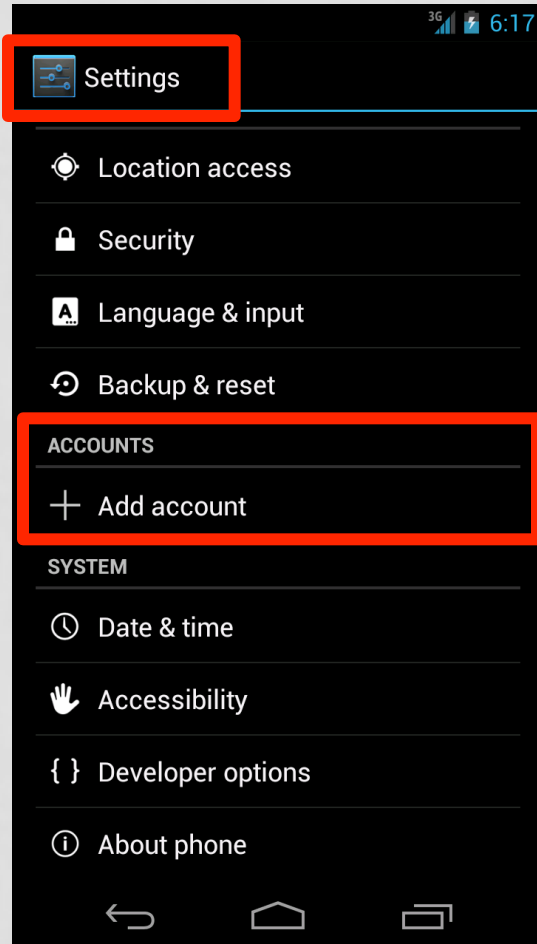
# 新增GOOGLE APIS模擬器



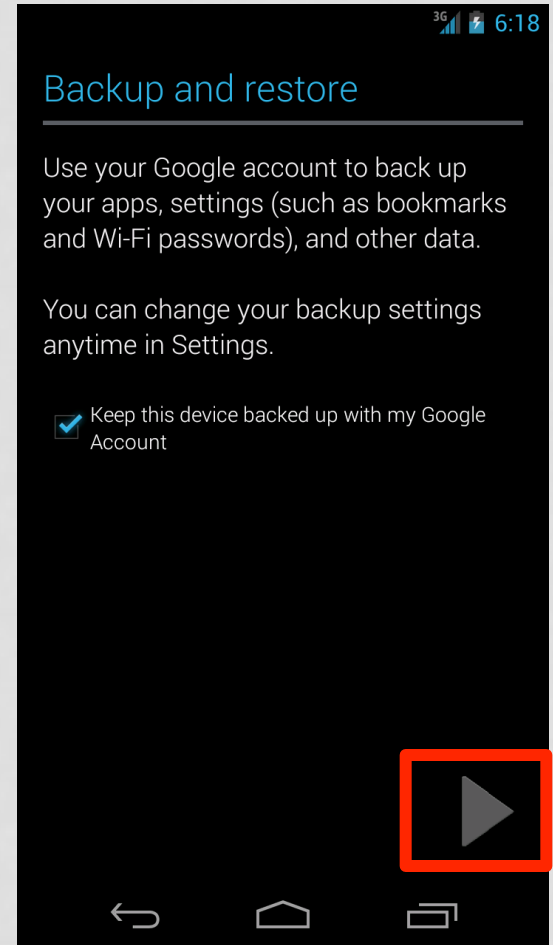
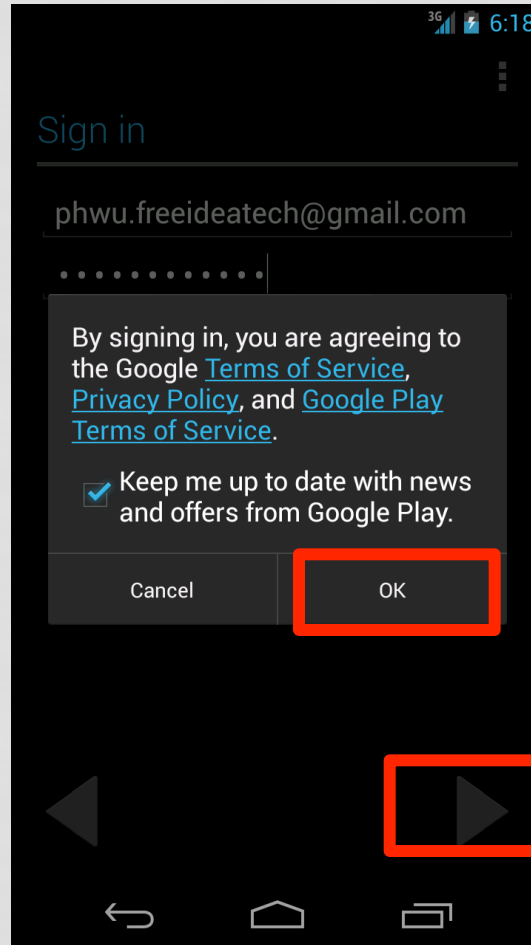
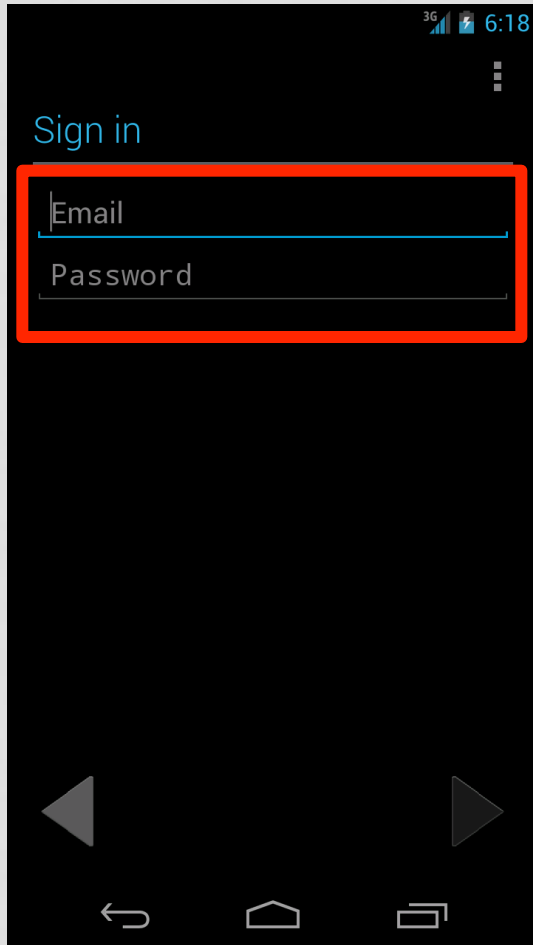
新增具備「Google APIs」的模擬器



# 登入GOOGLE帳號 /1

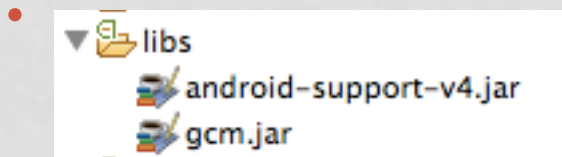


# 登入GOOGLE帳號 / 2



# GCM專案製作

- 複製gcm.jar至專案中的libs/目錄下



- 修改專案設定檔AndroidManifest.xml
  - 「package」為專案package名稱

```
<!-- 注意 minSdkVersion 要 8 以上 -->
<!-- GCM requires Android SDK version 2.2 (API level 8) or above. -->
<uses-sdk
    android:minSdkVersion="8"
    android:targetSdkVersion="17" />

<!-- Creates a custom permission so only this app can receive its messages. -->
<permission
    android:name="package.permission.C2D_MESSAGE"
    android:protectionLevel="signature" />
<!-- 使用GCM -->
<uses-permission android:name="package.permission.C2D_MESSAGE" />
```

# GCM專案設定檔 /1

- 修改專案設定檔AndroidManifest.xml
  - uses-permission設定

```
<!-- 以下設定固定不變 -->
<!-- App receives GCM messages. -->
<!-- This app has permission to register and receive data message. -->
<uses-permission android:name="com.google.android.c2dm.permission.RECEIVE" />

<!-- GCM requires a Google account. -->
<uses-permission android:name="android.permission.GET_ACCOUNTS" />

<!-- 存取 internet -->
<!-- GCM connects to Internet Services. -->
<uses-permission android:name="android.permission.INTERNET" />

<!-- Network State Permissions to detect Internet status -->
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

<!-- Keeps the processor from sleeping when a message is received. -->
<uses-permission android:name="android.permission.WAKE_LOCK" />

<!-- 手機震動 -->
<!-- Permission to vibrate -->
<uses-permission android:name="android.permission.VIBRATE" />
```

# GCM專案設定檔 /2

- 修改專案設定檔AndroidManifest.xml
  - Application設定，設定於「</activity>」下方

```
<!-- 以下設定固定不變 -->
<!-- 接收 GCM 的 receiver -->
<receiver
    android:name="com.google.android.gcm.GCMBroadcastReceiver"
    android:permission="com.google.android.c2dm.permission.SEND" >
    <intent-filter>

        <!-- Receives the actual messages. -->
        <action android:name="com.google.android.c2dm.intent.RECEIVE" />
        <!-- Receives the registration id. -->
        <action android:name="com.google.android.c2dm.intent.REGISTRATION" />

        <category android:name="com.androidhive.pushnotifications" />
    </intent-filter>
</receiver>

<!-- GCM service -->
<!-- 處理GCM訊息 -->
<!-- 名稱須為GCMIntentService，起該Class需繼承GCMBaseIntentService -->
<service android:name=".GCMIntentService" />
```



# GCM專案字串檔

- 複製相關字串格式至「strings.xml」

```
<!-- GCM(Google Cloud Messaging) -->
<string name="error_config">Please set the %1$s constant and recompile the app.</string>
<string name="already_registered">Device is already registered on server.</string>
<string name="gcm_registered">From GCM: device successfully registered!</string>
<string name="gcm_unregistered">From GCM: device successfully unregistered!</string>
<string name="gcm_message">From GCM: you got message!</string>
<string name="gcm_error">From GCM: error (%1$s).</string>
<string name="gcm_recoverable_error">From GCM: recoverable error (%1$s).</string>
<string name="gcm_deleted">From GCM: server deleted %1$d pending messages!</string>
<string name="server_registering">Trying (attempt %1$d/%2$d) to register device on Demo Server.</string>
<string name="server_registered">From Demo Server: successfully added device!</string>
<string name="server_unregistered">From Demo Server: successfully removed device!</string>
<string name="server_register_error">Could not register device on Demo Server after %1$d attempts.</string>
<string name="server_unregister_error">Could not unregister device on Demo Server (%1$s).</string>
<string name="options_register">Register</string>
<string name="options_unregister">Unregister</string>
<string name="options_clear">Clear</string>
<string name="options_exit">Exit</string>
```

# GCM專案程式 / 1

- 複製並改寫
  - GCMIntentService.java
    - 接收GCM後的相關動作
    - 改寫「onRegistered()」、「onUnregistered()」、「onMessage」、「generateNotification()」
  - GCMServerUtilities.java
    - 與Web Service相關動作
    - 「SENDER\_ID」為傳送者ID，可於「Google主控台->Overview->Project Number」查看
  - GCMCommonUtilities.java
    - GCM相關設定與共同函式

# GCM專案程式 /2

- 改寫「GCMIntentService.java」
  - 處理取得手機註冊ID(傳送至主機儲存)

```
/**
 * Method called on device registered
 **/
@Override
protected void onRegistered(Context context, String registrationId) {
    Log.i(TAG, "Device registered: regId = " + registrationId);
    showMessage(context, "Device registered: regId = " + registrationId);
    //註冊後·與Web Service連線
    GCMServerUtilities.register(context, registrationId);
}
```

- 處理取得手機註銷ID(傳送至主機移除)

```
/**
 * Method called on receiving a deleted message
 * */
@Override
protected void onDeletedMessages(Context context, int total) {
    Log.i(TAG, "Received deleted messages notification");
    String message = getString(R.string.gcm_deleted, total);
    showMessage(context, message);
}
```

# GCM專案程式 /3

- 改寫「GCMIntentService.java」
  - 處理收到GCM訊息後的動作

```
/**
 * Method called on Receiving a new message
 * */
@Override
protected void onMessage(Context context, Intent intent) {
    // 接收 GCM server 傳來的訊息
    Bundle bData = intent.getExtras();

    // 處理 bData 內含的訊息
    // 在本例中, 我的 server 端程式 gcm_send.php 傳來了 message, title, number 三項資料
    String message = bData.getString("message");
    String title = bData.getString("title");
    String number = bData.getString("number");
    Log.i(TAG, "Received message: " + message + ", " + title + ", " + number);

    showMessage(context, "Received message: " + message + ", " + title);

    // 通知 user
    generateNotification(context, bData);
}
```

# GCM專案程式 / 4

- 改寫「GCMIntentService.java」
  - 傳送至手機推播佇列中(設定圖示、聲音、訊息等資料)

```
/**
 * Issues a notification to inform the user that server has sent a message.
 */
// 處理訊息並顯示於「推播列」中
private static void generateNotification(Context context, Bundle data) {
    //設定推播列中顯示的圖示
    int icon = R.drawable.ic_action_search;

    long when = System.currentTimeMillis();
    NotificationManager nm = (NotificationManager) context.getSystemService(Context.NOTIFICATION_SERVICE);
    Intent ni = new Intent(context, SampleGCMActivity.class);
    ni.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP | Intent.FLAG_ACTIVITY_SINGLE_TOP);
    PendingIntent intent = PendingIntent.getActivity(context, 0, ni, 0);
    // 如果您想讓通知的內容有動態的變化
    // 就可以運用傳進來的參數 -- Bundle 型別 data
    // 取出您要的欄位填入 setContentTitle() 和 setContentText()
    Notification noti = new NotificationCompat.Builder(context)
        //只會顯示在推播列中
        .setContentTitle(data.getString("title"))
        .setContentText(data.getString("message"))
        .setNumber(Integer.getInteger(data.getString("number")))
        .setContentIntent(intent)
        .setDefaults(Notification.DEFAULT_ALL)
        .setSmallIcon(icon)
        .setWhen(when)
        .build();

    nm.notify(0, noti);
}
```

# GCM專案程式 /5

- 修改「SampleGCMActivity.java」
  - 新增程式碼於「onCreate()」

```
//註冊訊息接收器
registerReceiver(mHandleMessageReceiver, new IntentFilter(DISPLAY_MESSAGE_ACTION));

//GCM
// 檢查裝置是否支援 GCM
// Make sure the device has the proper dependencies.
GCMRegistrar.checkDevice(SampleGCMActivity.this);

// Make sure the manifest was properly set - comment out this line
// while developing the app, then uncomment it when it's ready.
GCMRegistrar.checkManifest(SampleGCMActivity.this);

// Get GCM registration id
final String regId = GCMRegistrar.getRegistrationId(SampleGCMActivity.this);
if (regId.equals(""))
{
    Log.d(TAG, "尚未註冊 Google GCM, 進行註冊");
    GCMRegistrar.register(SampleGCMActivity.this, SENDER_ID);
} else {
    Log.e(TAG, "regId="+regId);
}
```



# GCM專案程式 / 6

- 修改「SampleGCMActivity.java」

- 設定接收Broadcast訊息時的動作

```
//設定接收Broadcast訊息時的動作
//CommonUtilities->displayMessage
private final BroadcastReceiver mHandleMessageReceiver =
    new BroadcastReceiver() {
        @Override
        public void onReceive(Context context, Intent intent) {
            String StrMSG = intent.getExtras().getString(EXTRA_MESSAGE);
            GCMRegID.setText(StrMSG);
            setClipboard(StrMSG);
        }
    };
```

- 實作練習

- 透過POST傳送資料至以下網站，並接收推播
- 「[http://www.realtouchapp.com/gcm\\_send.php](http://www.realtouchapp.com/gcm_send.php)」
- POST Key: regID, title, message, number

# SERVER


- 建立API Project
- 取得 Sender ID(Project Number)
- 開啓GCM服務
- 建立新的Server API Key
- Server程式說明(PHP)



# 建立API Project /1

- 在取得 GCM Server Key之前，我們必須到 Google APIs Console 建立一個 API Project
- <https://code.google.com/apis/console/>

**Start using the Google APIs console**  
to manage your API usage



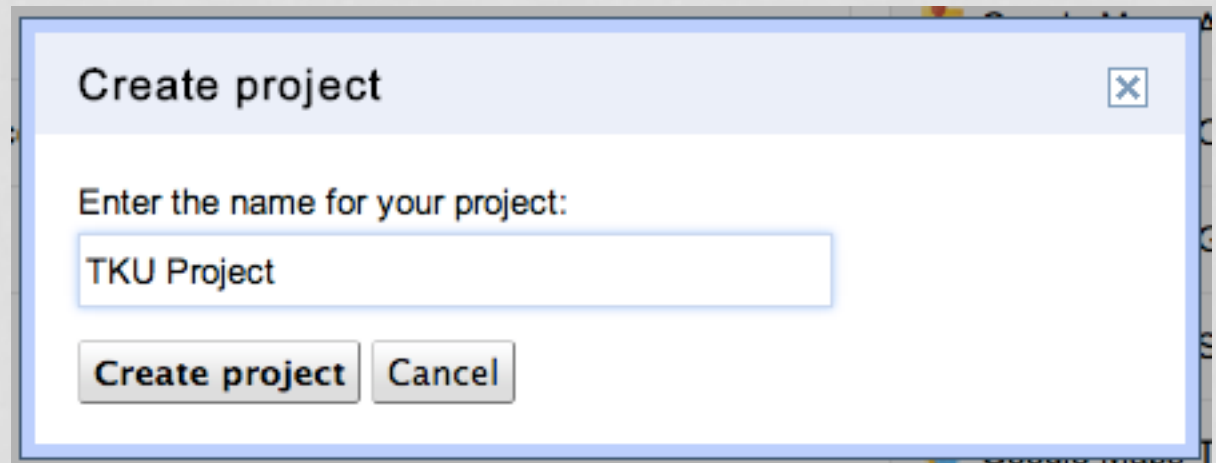
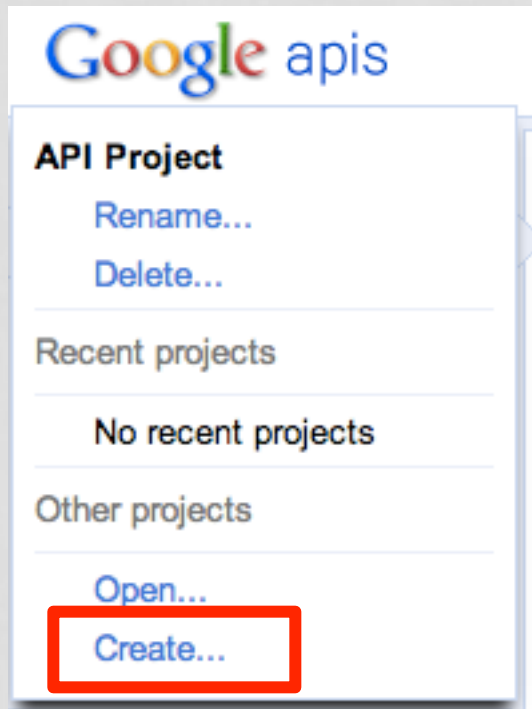
Creating an **APIs project** will let you:

- **Use** Google APIs **beyond anonymous limits.**
- **Monitor** API usage and **control** API access.
- **Share** API management with a team.

[Create project...](#)

# 建立API Project / 2

- 點擊下拉式選單->Create



# 取得 SENDER ID

- 查看Overview中的Project Number

TKU Project ▼

**Overview**

Services

Team

API Access

Reports

Quotas

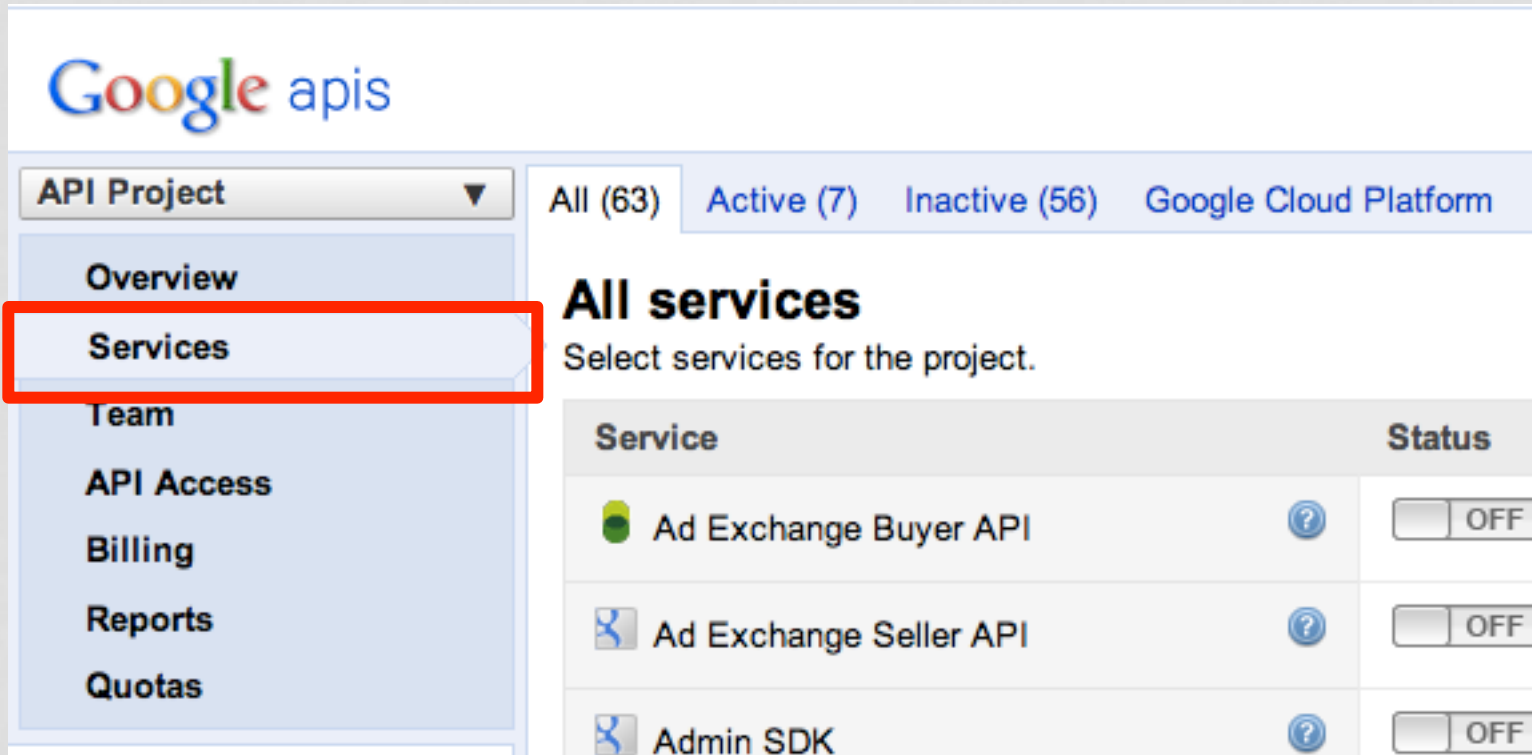
## Dashboard

### Project Summary

Name	TKU Project
Project Number	[REDACTED]
Project ID	<a href="#">Register...</a>
Google+ Page	<a href="#">Request connection</a>
Owners	phwu.freeideatech@gmail.com - you

# 開啓GCM服務 /1

- 進入Google API主控台，點擊Services









Google apis

API Project ▼


All (63) Active (7) Inactive (56) Google Cloud Platform

**All services**  
Select services for the project.

Service	Status
 Ad Exchange Buyer API 	<input type="checkbox"/> OFF
 Ad Exchange Seller API 	<input type="checkbox"/> OFF
 Admin SDK 	<input type="checkbox"/> OFF

# 開啓GCM服務 /2

- 開啓GCM服務

 Google Cloud Messaging for Android



OFF

Google apis

Review terms of service

[Print](#)

## Terms of Service

By using this API, you consent to be bound by the Google APIs Terms of Service ("API ToS") at <https://developers.google.com/terms>.

0 of 1 terms of service accepted.

Google Cloud Messaging for Android

I agree to these terms.

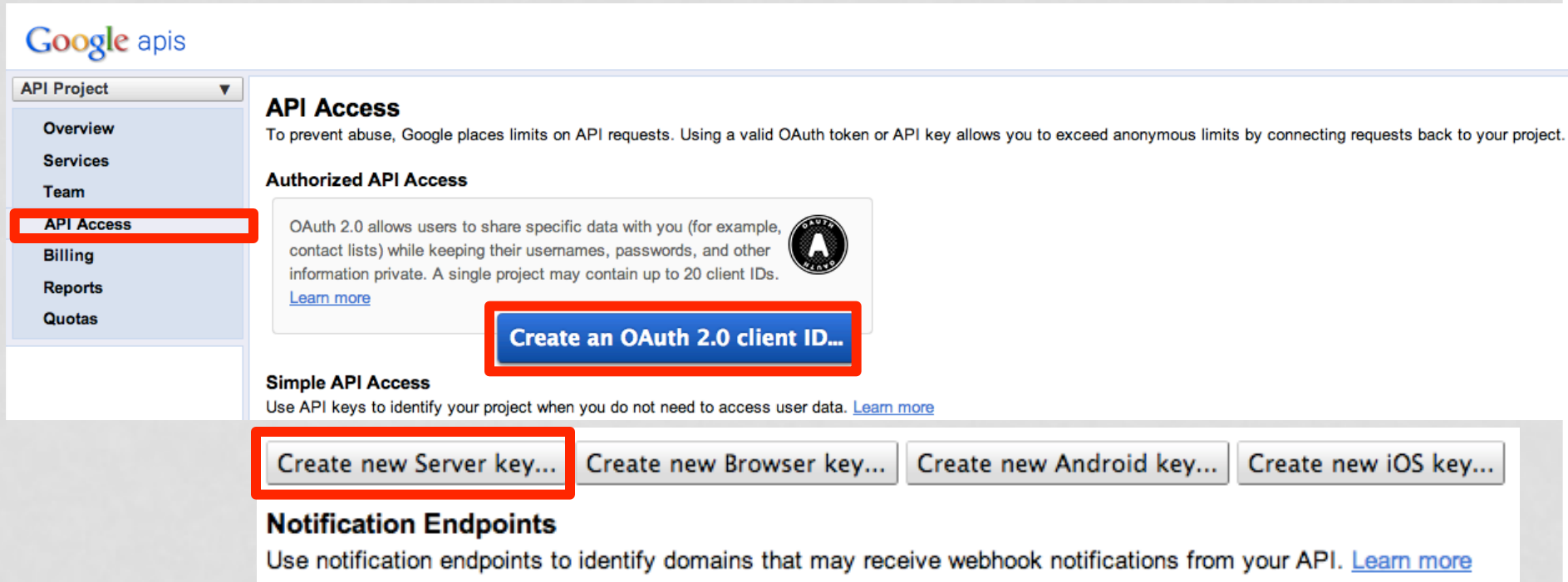
 Google Cloud Messaging for Android



ON

# 建立新的SERVER API KEY /1

- 進入Google API主控台，點擊API Access



Google apis

API Project ▾

- Overview
- Services
- Team
- API Access**
- Billing
- 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](#)

**Create new Server key...** Create new Browser key... Create new Android key... Create new iOS key...

### Notification Endpoints

Use notification endpoints to identify domains that may receive webhook notifications from your API. [Learn more](#)

# 建立新的SERVER API KEY /2

- 點擊「Create new Server Key...」
- 直接點擊「Create」

### Configure Server Key for API Project ✕

This key should be kept secret on your server.

Every API request is generated by software running on a machine that you control. Per-user limits will be enforced using the address found in each request's `userIp` parameter, (if specified). If the `userIp` parameter is missing, your machine's IP address will be used instead. [Learn more](#)

Accept requests from these server IP addresses:

Example: 192.168.12.0/23. One IP address or subnet per line.

**Create** Cancel

# 建立新的SERVER API KEY /3

- API Key

## Key for server apps (with IP locking)

API key: AIzaSyCyXufHa5VawclnG-FDMLsKp-9USD1416Y

IPs: Any IP allowed

Activated on: Aug 12, 2013 1:28 AM

Activated by: phwu.freeideatech@gmail.com – you



# 建立新的SERVER API KEY /4

- 可以透過「Create new key...」重新產生，但不會馬上生效

## Key for server apps (with IP locking)

API key: AIzaSyCyXufHa5Vawc1nG-FDMLsKp-9USD1416Y  
IPs: Any IP allowed  
Activated on: Aug 12, 2013 1:28 AM  
Activated by: phwu.freeideatech@gmail.com - you

[Generate new key...](#)

[Edit allowed IPs...](#)

[Delete key...](#)

Do you want to replace the current key with a new one?

The new key will be available immediately. The current key will be deactivated in 24 hours.

[Generate new key](#)

Cancel

## Key for server apps (with IP locking)

API key: AIzaSyD2eJ6GSBfRB00jk7wkV1ajEvQB-g00kSE  
IPs: Any IP allowed  
Activated on: Aug 12, 2013 1:40 AM  
Activated by: phwu.freeideatech@gmail.com - you  
Obsolete key: AIzaSyCyXufHa5Vawc1nG-FDMLsKp-9USD1416Y

Status: **Active until Aug 13, 2013 1:40 AM**

[Generate new key...](#)

[Revert to obsolete key...](#)

[Edit allowed IPs...](#)

[Delete key...](#)

# SERVER程式說明(PHP) /1

- 設定傳送者ID
  - 主控台->Overview->Project Number
  - \$SenderId =  
"AlzaSyCi3CBoMDhd0S6ppLoTPfLaBou7DsxZE3M";
- 傳送網址
  - \$url = 'https://android.googleapis.com/gcm/send';
- 設定傳送標頭
  - \$headers = array('Content-Type: application/json',  
'Authorization: key='.\$SenderId);
- 設定欲傳送的手機
  - ```
$regID = array();  
//將裝置ID存進去$regID array  
array_push($regID, "傳送手機1");
```

# SERVER程式說明(PHP) / 2

- 設定傳送內容

```
// 發送的訊息內容轉成 JSON 格式
/*
{
  "registration_ids": ["手機裝置ID"],
  "data": {"message": "your message"} <-裡面為要傳送給User的資訊(4kb以內)
}
*/
// 要發送的訊息內容
// 例如我要發送 message, title, number 三樣資訊
// 就將這 3 個組成陣列
// 您可依您自己的需求修改
$fields = array( 'registration_ids' => $regID,
                'data'              => array( 'title' => 'gcm_title',
   'message' => 'gcm_message',
   'number' => '2'
   )
                );
```

# SERVER程式說明(PHP) /3

- 傳送流程(PHP)

```
// Open connection
$ch = curl_init();

// Set the url, number of POST vars, POST data
curl_setopt( $ch, CURLOPT_URL, $url );
curl_setopt( $ch, CURLOPT_POST, true );
curl_setopt( $ch, CURLOPT_HTTPHEADER, $headers);
curl_setopt( $ch, CURLOPT_RETURNTRANSFER, true );
// Disabling SSL Certificate support temporarily
// curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
curl_setopt( $ch, CURLOPT_POSTFIELDS, json_encode( $fields ) );

// 傳送到 Google GCM server,
// 並接收回傳結果
$result = curl_exec($ch);

// Close connection
curl_close($ch);

echo 'result -->'.$result.'<br />';
echo json_encode( $fields );
```

# OUTLINE

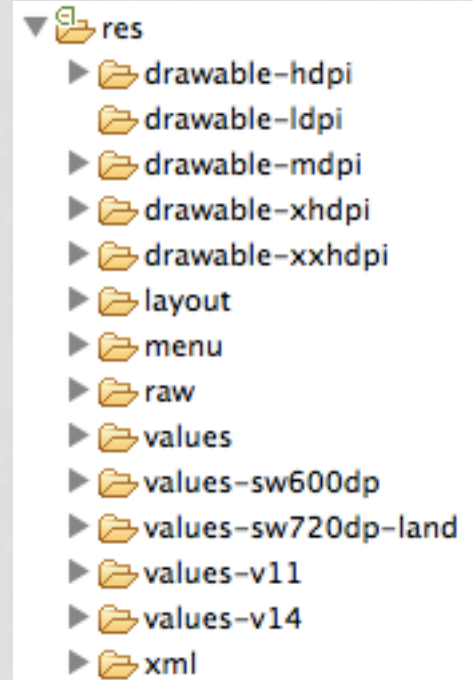
- 資料格式
- Google雲端訊息
- **資料存取**
- SQLite

# 資料存取

- Resources
- Assets
- Shared Preferences(分享式喜好設定)
  - 儲存私有的簡單資料在鍵-值配對
- Internal Storage
  - 儲存私有的資料在裝置的記憶體(手機中)
- External Storage
  - 儲存公開資料在外部的儲存空間(如SD卡)
- SQLite Databases
  - 儲存結構化的資料在私有的資料庫
- 全部範例請參考「DataAccessSample」

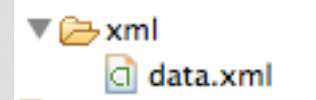
# RESOURCE (/RES)

- 存放應用程式的資源
  - 如圖示、GUI佈局等
- 將被打包到編譯後的Java中
- 不支援深度子目錄(不能自行建立子目錄)
- 大部分使用的資源目錄
  - /res/drawable
  - /res/values
  - /res/layout
- res/menu
  - 存放基於XML的功能表描述
- res/values
  - 存放字串、尺寸值



# RESOURCES (/RES/XML/) /1

- /res/xml/
  - 存放通用的XML檔
  - 成二進位格式放到最終的安裝包裡的
  - 通過R類的資源ID，取得檔案串流，並且解析裡面的內容 (data.xml)
    - 使用getXml方法，取得xml解析器
    - 解析器的工作原理和SAX方式差不多





# RESOURCES (/RES/XML/) /2

- 實作範例

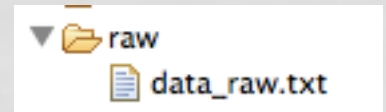
```
//sample of xml
try {
    XmlResourceParser xml = getResources().getXml(R.xml.data);
    xml.next();
    int eventType = xml.getEventType();

    boolean inTitle = false;
    while(eventType != XmlPullParser.END_DOCUMENT) {
        //到達title節點時標記一下
        if(eventType == XmlPullParser.START_TAG) {
            if(xml.getName().equals("title")) {
                inTitle = true;
            }
        }

        //如過到達標記的節點則取出內容
        if(eventType == XmlPullParser.TEXT && inTitle) {
            Log.e("PHDEBUG", "XML="+xml.getText());
        }
        xml.next();
        eventType = xml.getEventType();
    }
} catch (XmlPullParserException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
```

# RESOURCES (/RES/RAW/)

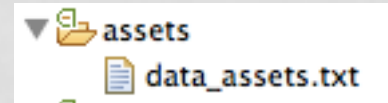
- /res/raw/
  - 存放通用的檔案
  - 按原樣複製到設備上
  - 該資料夾內的檔將不會被編譯成二進位檔案
  - 通過R類的資源ID，取得檔案串流，並且解析裡面的內容 (data\_raw.txt)
    - 使用openRawResource，取檔案串流



```
//sample of raw in resources  
InputStream is = getResources().openRawResource(R.raw.data_raw);  
String str = readStream(is);  
Log.e("PHDEBUG", "RAW="+str);
```

# ASSETS (/ ASSETS)

- /assets
  - 支援任意深度的子目錄(可任意的建立子目錄)
  - 不會生成任何資源ID
  - 高自由度，不受Android平臺的約束
  - 不會被編譯成二進位形式



```
//sample of assets
try {
    AssetManager assets = getAssets();
    InputStream isa = assets.open("data_assets.txt");
    String stra = readStream(isa);
    Log.e("PHDEBUG", "ASSETS="+stra);
} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
```

# SHARED PREFERENCES /1

- 記錄的檔案位於/data/data/[package name]/shared\_prefs/[preferences filename].xml，可以用DDMS觀察它的大小，就知道有沒有記錄進入或被清除。
- 如果要改成讓其他的Activity能方便使用的Prefs，或是多個Prefs檔案
  - settingsActivity=getSharedPreferences("MyCustomSharedPreferences", 0);
  - 記錄檔案會變成MyCustomSharedPreferences.xml，其他的Activity可以用定義的名稱來存取這個檔案

# SHARED PREFERENCES /2

The screenshot displays the Android Studio interface with the File Explorer view open. The 'DDMS' tab is highlighted in red. The file explorer shows a list of application packages, with 'com.example.dataaccesssample' and its subfolders 'cache', 'lib', and 'shared\_prefs' highlighted in red. The 'shared\_prefs' folder is expanded, showing 'DataAccessActivity.xml' selected. A red box highlights the XML content of this file, which contains a single key-value pair: 'mykey' with value 'myValue'.

| Name                             | Size | Date       | Time  | Permissions | Info |
|----------------------------------|------|------------|-------|-------------|------|
| com.android.soundrecorder        |      | 2013-04-14 | 09:16 | drwxr-x--x  |      |
| com.android.speechrecorder       |      | 2013-04-14 | 09:16 | drwxr-x--x  |      |
| com.android.systemui             |      | 2013-04-14 | 09:16 | drwxr-x--x  |      |
| com.android.vending              |      | 2013-04-19 | 11:52 | drwxr-x--x  |      |
| com.android.vpndialogs           |      | 2013-04-14 | 09:16 | drwxr-x--x  |      |
| com.android.wallpaper.livepicker |      | 2013-04-20 | 12:22 | drwxr-x--x  |      |
| com.android.widgetpreview        |      | 2013-04-14 | 09:16 | drwxr-x--x  |      |
| com.example.android.apis         |      | 2013-04-14 | 09:16 | drwxr-x--x  |      |
| com.example.android.livecubes    |      | 2013-04-14 | 09:16 | drwxr-x--x  |      |
| com.example.android.softkeyboard |      | 2013-04-14 | 09:16 | drwxr-x--x  |      |
| com.example.dataaccesssample     |      | 2013-06-08 | 20:38 | drwxr-x--x  |      |
| cache                            |      | 2013-06-08 | 20:06 | drwxrwx--x  |      |
| lib                              |      | 2013-06-08 | 20:06 | drwxr-xr-x  |      |
| shared_prefs                     |      | 2013-06-08 | 20:38 | drwxrwx--x  |      |
| DataAccessActivity.xml           | 108  | 2013-06-08 | 20:38 | -rw-rw----  |      |
| com.example.realtouchjar         |      | 2013-04-20 | 17:17 | drwxr-x--x  |      |
| com.example.samplefragments      |      | 2013-04-14 | 10:10 | drwxr-x--x  |      |
| com.google.android.apps.maps     |      |            |       |             |      |
| com.google.android.gms           |      |            |       |             |      |
| com.hans.fragment                |      |            |       |             |      |
| com.hans.tab                     |      |            |       |             |      |
| com.realtouch.googlemap          |      |            |       |             |      |
| com.realtouch.myloveapp          |      | 2013-05-25 | 00:22 | drwxr-x--x  |      |
| com.svox.pico                    |      | 2013-04-14 | 09:28 | drwxr-x--x  |      |
| com.tamkangapp                   |      | 2013-05-25 | 08:13 | drwxr-x--x  |      |

```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
  <string name="mykey">myValue</string>
</map>
```

# SHARED PREFERENCES /3

- 實作範例

```
//sample of SharedPreferences
//取得SharedPreferences物件
SharedPreferences settingsActivity = getPreferences(MODE_PRIVATE);
//如果要改成讓其他的Activity能方便使用的Prefs，或是多個Prefs檔案
//SharedPreferences settingsActivity =
//| getSharedPreferences("DataAccessSharedPreferences", 0);

//抓取字串key=mykey的值，預設值為空字串
String myValue = settingsActivity.getString("mykey", "");
Log.e("PHDEBUG", "myKey="+myValue);
```

```
//儲存資料
//取得到SharedPreferences.Editor物件
SharedPreferences.Editor editor = settingsActivity.edit();
//設定key=mystring的值
editor.putString("mykey", "myValue");
//最後要提交commit
editor.commit();
```

```
//清除資料
//取得到SharedPreferences.Editor物件
SharedPreferences.Editor editor = settingsActivity.edit();
//清除Preference中的數值，如果用remove可以移除某鍵的值
editor.clear();
//最後要提交commit
editor.commit();
```

# INTERNAL STORAGE /1

- 不需要額外設定權限
- 直接在設備上的內部操作檔案
- 預設情況為該App私有的檔案，因此其他App無法取得該檔案
- 儲存在/data/data/package\_name/files
- 由於是儲存在手機的記憶體中，所有會有空間的限制，不建議在這裡儲存容量太大的檔案
- 當App被移除，所有的資料也會被移除
- Context.MODE\_PRIVATE
  - 該檔案為私有的



# INTERNAL STORAGE /2

The screenshot shows the Android Studio interface with the File Explorer tab active. The DDMS icon in the top right corner is highlighted with a red box. The File Explorer shows a list of application packages with columns for Name, Size, Date, Time, Permissions, and Info. The package 'com.example.dataaccesssample' is expanded, showing its internal structure. The file 'file\_internal.txt' is highlighted with a blue selection bar and a red box.

Name	Size	Date	Time	Permissions	Info
com.android.smoketest.tests		2013-04-14	09:16	drwxr-x--x	
com.android.soundrecorder		2013-04-14	09:16	drwxr-x--x	
com.android.speechrecorder		2013-04-14	09:16	drwxr-x--x	
com.android.systemui		2013-04-14	09:16	drwxr-x--x	
com.android.vending		2013-04-19	11:52	drwxr-x--x	
com.android.vpndialogs		2013-04-14	09:16	drwxr-x--x	
com.android.wallpaper.livepicker		2013-04-20	12:22	drwxr-x--x	
com.android.widgetpreview		2013-04-14	09:16	drwxr-x--x	
com.example.android.apis		2013-04-14	09:16	drwxr-x--x	
com.example.android.livecubes		2013-04-14	09:16	drwxr-x--x	
com.example.android.softkeyboard		2013-04-14	09:16	drwxr-x--x	
com.example.dataaccesssample		2013-06-09	23:27	drwxr-x--x	
cache		2013-06-08	20:06	drwxrwx--x	
files		2013-06-09	23:27	drwxrwx--x	
file_internal.txt	13	2013-06-09	23:27	-rw-rw----	
shared_prefs		2013-06-08	20:06	drwxrwx--x	
com.example.realtouchjar		2013-04-20	17:17	drwxr-x--x	
com.example.samplefragments		2013-04-14	10:10	drwxr-x--x	
com.google.android.apps.maps		2013-04-14	10:02	drwxr-x--x	
com.google.android.gms		2013-04-14	10:04	drwxr-x--x	
com.hans.fragment		2013-04-19	18:10	drwxr-x--x	
com.hans.tab		2013-04-19	18:13	drwxr-x--x	
com.realtouch.googlemap		2013-04-14	09:28	drwxr-x--x	
com.realtouch.mvloveapp		2013-05-25	00:22	drwxr-x--x	



# INTERNAL STORAGE /3

- 實作範例

```
//sample of Internal Storage Write
try {
    FileOutputStream fos =
        openFileOutput("file_internal.txt", Context.MODE_PRIVATE);
    fos.write("internal data".getBytes());
    fos.close();
} catch (FileNotFoundException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
```

```
//sample of Internal Storage Read
try {
    FileInputStream fis = openFileInput("file_internal.txt");
    byte[] input = new byte[fis.available()];
    while (fis.read(input) != -1) {}
    Log.e("PHDEBUG", "Internal="+new String(input));
} catch (FileNotFoundException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
```

# EXTERNAL STORAGE (/SDCARD) /1

- 儲存目錄為/sdcard
- 必須設定讀取SDCard的權限

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

- 容量則取決於SDcard本身的容量
- 不會因為App被移除而消失

# EXTERNAL STORAGE (/SDCARD) /2

- DIRECTORY\_MUSIC (Music/)
  - Media scanner classifies all media found here as user music.
- DIRECTORY\_PODCASTS (Podcasts/)
  - Media scanner classifies all media found here as a podcast.
- DIRECTORY\_RINGTONES (Ringtones/)
  - Media scanner classifies all media found here as a ringtone.
- DIRECTORY\_ALARMS (Alarms/)
  - Media scanner classifies all media found here as an alarm sound.
- DIRECTORY\_NOTIFICATIONS (Notifications/)
  - Media scanner classifies all media found here as a notification sound.
- DIRECTORY\_PICTURES (Pictures/)
  - All photos (excluding those taken with the camera).
- DIRECTORY\_MOVIES (Movies/)
  - All movies (excluding those taken with the camcorder).
- DIRECTORY\_DOWNLOADS (Download/)
  - Miscellaneous downloads.

# EXTERNAL STORAGE (/SDCARD) /3

- 實作範例

- 「writeToSdcard()」及「readFromSdcard()」請參考原始碼

```
//sample of External storage
//取得外部儲存媒體的狀態
String state = Environment.getExternalStorageState();
//判斷狀態
if (Environment.MEDIA_MOUNTED.equals(state)) {
    //可以讀寫
    writeToSdcard("file_external.txt", "external data");
    String data_ext = readFromSdcard("file_external.txt");
    Log.e("PHDEBUG", "External="+data_ext);
} else if (Environment.MEDIA_MOUNTED_READ_ONLY.equals(state)) {
    //只可以讀取，無法寫入
    String data_ext = readFromSdcard("file_external.txt");
    Log.e("PHDEBUG", "External(READ_ONLY)="+data_ext);
} else {
    //無法讀寫
}
}
```

# OUTLINE

- 資料格式
- Google雲端訊息
- 資料存取
- **SQLite**

# SQLITE

- 需自行設計資料類別
- 需自行設計處理類別
  - 需繼承SQLiteOpenHelper
- 建立/刪除SQLite資料表
- 操作SQLite資料庫
  - SQLite新增功能
  - SQLite查詢功能
  - SQLite修改與刪除功能
  - 範例請參考「SampleSQLite」

# SQLITE - OPENHELPER /1

- 需 extends SQLiteOpenHelper

```
public class SQLiteUtil extends SQLiteOpenHelper {
```

- 資料設定

- DATABASE\_VERSION: 版本
- DATABASE\_NAME 資料庫名稱
- TABLE\_NAME: 表格名稱
- COLUMN\_NAMES: 欄位名稱

```
//資料庫版本關係到App更新時，資料庫是否要調用onUpgrade()  
public final static int DATABASE_VERSION = 1; //資料庫版本  
public final static String DATABASE_NAME = "SQLiteUtil.db";  
public static final String TABLE_NAME = "SQLiteUtil";  
  
public static final String[] COLUMN_NAMES = {"data", "update_time"};
```

# SQLITE - OPENHELPER /2

- 複寫建構子
- onCreate: 第一次開啟時動作
- onUpgrade: 版本更新時動作

```
public SQLiteUtil(Context context) {  
    super(context, DATABASE_NAME, null, DATABASE_VERSION);  
}  
  
@Override  
public void onCreate(SQLiteDatabase db) {  
    // TODO Auto-generated method stub  
    Create(db, TABLE_NAME);  
}  
  
@Override  
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
    // TODO Auto-generated method stub  
    Drop(db, TABLE_NAME);  
    Create(db, TABLE_NAME);  
}
```



# SQLITE - OPENHELPER /3

- 開啟與關閉
  - 直接呼叫父函式

```
@Override
public void onOpen(SQLiteDatabase db) {
    super.onOpen(db);
    // TODO 每次成功打開數據庫後首先被執行
}

@Override
public synchronized void close() {
    super.close();
}
```

# SQLITE - OPENHELPER /4

- 重新建立資料庫

```
public void rebuild(String table) {  
    SQLiteDatabase db = this.getReadableDatabase();  
    Drop(db, table);  
    Create(db, table);  
    db.close();  
}
```

# SQLITE – 執行SQL語法

- 執行SQL語法

```
public void execSQL(String strSQL) {  
    SQLiteDatabase db = this.getReadableDatabase();  
    db.execSQL(strSQL);  
    db.close();  
}
```

# SQLITE - CREATE & DROP

- 建立與刪除資料表

```
public void Create(SQLiteDatabase db, String table) {
    String []columns = getColumns(table);
    if(columns == null) return;

    String DATABASE_CREATE_TABLE =
        "create table " + table + " (" + "id INTEGER PRIMARY KEY
        AUTOINCREMENT NOT NULL,";
    for(String column : columns) {
        DATABASE_CREATE_TABLE += column + " TEXT,";
    }
    DATABASE_CREATE_TABLE = DATABASE_CREATE_TABLE.substring(0,
        DATABASE_CREATE_TABLE.length()-1)+ ");";
    db.execSQL(DATABASE_CREATE_TABLE);
}

public void Drop(SQLiteDatabase db, String table) {
    // TODO Auto-generated method stub
    db.execSQL("DROP TABLE IF EXISTS " + table); //刪除舊有的資料表
}
```

# SQLITE - INSERT

- Insert
  - 插入(加入)新資料

```
//回傳目前新增資料的ID
//return the row ID of the newly inserted row, or -1 if an error occurred
public long Insert(String table, ContentValues cv) {
    SQLiteDatabase db = this.getReadableDatabase();
    long row = db.insert(table, null, cv);
    db.close();
    return row;
}
```

# SQLITE - UPDATE

- Update
  - 更新資料

```
//constrain don't need 'where'  
//回傳資料影響數量  
//return the number of rows affected  
public long Update(String table, String constrain, ContentValues cv) {  
    SQLiteDatabase db = this.getWritableDatabase();  
    long row = db.update(table, cv, constrain, null);  
    db.close();  
    return row;  
}
```

# SQLITE - DELETE

- Delete
  - 刪除資料

```
//constrain don't need 'where'  
//return the number of rows affected if a whereClause is passed in,  
//0 otherwise. To remove all rows and get a count pass "1" as the whereClause.  
public int Delete(String table, String constrain) {  
    SQLiteDatabase db = this.getWritableDatabase();  
    int r = db.delete(table, constrain, null);  
    db.close();  
    return r;  
}
```

# SQLITE - SELECT

- Select
  - 取得全部的資料

```
public List< Map<String, String> > SelectAll(String table) {
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor cursor = db.rawQuery("SELECT * from " + table, null);
    List< Map<String, String> > list = new ArrayList< Map<String, String> >();
    if(cursor.moveToFirst()) {
        do {
            Map<String, String> map = new HashMap<String, String>();
            map.put("id", String.valueOf(cursor.getInt(cursor.getColumnIndex("id"))));
            for(String column : COLUMN_NAMES) {
                map.put(column, cursor.getString(cursor.getColumnIndex(column)));
            }
            list.add(map);
        } while(cursor.moveToNext());
    }
    cursor.close();
    db.close();
    return list;
}
```



# SQLITE - SAMPLE /1

- 操作範例

- 建立元件，並重新建立資料庫(依情況做)

```
SQLiteUtil db = new SQLiteUtil(ChangeActByListActivity.this);  
db.reBuild(SQLiteUtil.TABLE_NAME);  
...
```

- 加入新資料

```
/**  
//加入資料庫1  
ContentValues cv = new ContentValues();  
cv.put(SQLiteUtil.COLUMN_NAMES[0], "data1");  
cv.put(SQLiteUtil.COLUMN_NAMES[1], "1981-09-14 13:00:01");  
long row = db.Insert(SQLiteUtil.TABLE_NAME, cv);  
Toast.makeText(this, "row="+row, Toast.LENGTH_LONG).show();  
/**/  
//加入資料庫2, 第一個欄位為null  
db.execSQL("Insert into " + SQLiteUtil.TABLE_NAME  
+ " values (null, 'data2', '1988-04-27 13:00:01')");  
db.execSQL("Insert into " + SQLiteUtil.TABLE_NAME  
+ " ('data', 'update_time') values ('data3', '1988-04-27 13:00:01')");
```

# SQLITE - SAMPLE /2

- 操作範例
  - 更新資料

```
/*  
//更新資料1  
ContentValues cv = new ContentValues();  
cv.put(SQLiteUtil.COLUMN_NAMES[0], "data2");  
cv.put(SQLiteUtil.COLUMN_NAMES[1], "1988-04-27 13:00:01");  
db.Update(SQLiteUtil.TABLE_NAME, " id=2", cv);  
/**/  
//更新資料2  
db.execSQL("Update " + SQLiteUtil.TABLE_NAME  
    + " set data='data1', update_time='1981-09-14 13:00:01' where id=1");
```

- 刪除資料

```
//刪除資料  
//db.Delete(SQLiteUtil.TABLE_NAME, " id=1");  
//刪除資料2  
//db.execSQL("Delete from " + SQLiteUtil.TABLE_NAME + " where id=2");
```

# SQLITE - 實作練習

- 解析淡江時報電子報
  - <http://tkutimes.tku.edu.tw/ashx/ChRss.ashx>
- 儲存資料至SQLite中
- 使用ListView，顯示資料庫中所有儲存的資料