

應用軟體設計實習

班級：二子陸生一

學號：108AE4003

學生：劉曉敏

Lab12:在 Android 中想要透過網路取得資料，或是與他人交換訊息，就需要使用Http 通訊機制，Http 通訊協定被廣泛應用於網路環境，除了常見的瀏覽器外，Android 應用程式也能使用 Http 協定存取伺服器的資料。

Http(Hypertext Transfer Protocol)是一種網路通訊協定，用於客戶端發出請求(Request)給伺服器，伺服器將要給予的資料回傳

(Response)給客戶端使用，一般我們稱之為 API(Application Programming Interface)，手機送出 Request與伺服器建立連線，伺服器回傳 Response 資料，資料通常為 Xml 或 JSON...等格式的字串，需要再透過翻譯器轉換成 Android 看得懂的資料型態。

kotlin:

```
package com.example.lab12_kt

import xxx

class MainActivity : AppCompatActivity() {

    private val receiver: BroadcastReceiver = object : BroadcastReceiver() {
        override fun onReceive(context: Context, intent: Intent) {
            intent.extras?.getString( key: "json")?.let{ it: String?
                val data = Gson().fromJson(it, Data::class.java)
                val items = arrayOfNulls<String>(data.result.results.size)
                for (i in 0 until data.result.results.size)
                    items[i] = "\n列車即將進入 : ${data.result.results[i].Station}\n列車行駛目的地: ${data.result.results[i].Destination}"
                this@MainActivity.runOnUiThread {
                    AlertDialog.Builder( context: this@MainActivity).setTitle("台北捷運列車到站站名").setItems(items, listener: null).show()
                }
            }
        }
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        val intentfilter = IntentFilter( action: "MyMessage")
        registerReceiver(receiver, intentfilter)
    }
}
```

```

        btn_query.setOnClickListener(){ // View
            val req = Request.Builder().url( url: "https://data.taipexi/opendata/datalist/apiAccess?scope=resourceAquire&rid=55ec6d6e-dc5c-4268-a7
            OkHttpClient().newCall(req).enqueue(object: Callback{
                override fun onResponse(call: Call, response: Response) {
                    sendBroadcast(Intent( action: "MyMessage").putExtra( name: "json", response.body?.string()))
                }

                override fun onFailure(call: Call, e: IOException) {
                    Log.e( tag: "查詢失敗", msg: "$e")
                }
            })
        }
    }

    override fun onDestroy() {
        super.onDestroy()

        unregisterReceiver(receiver)
    }
}

class Data{
    lateinit var result: Result

    class Result{
        lateinit var results : Array<Results>

        class Results{

```

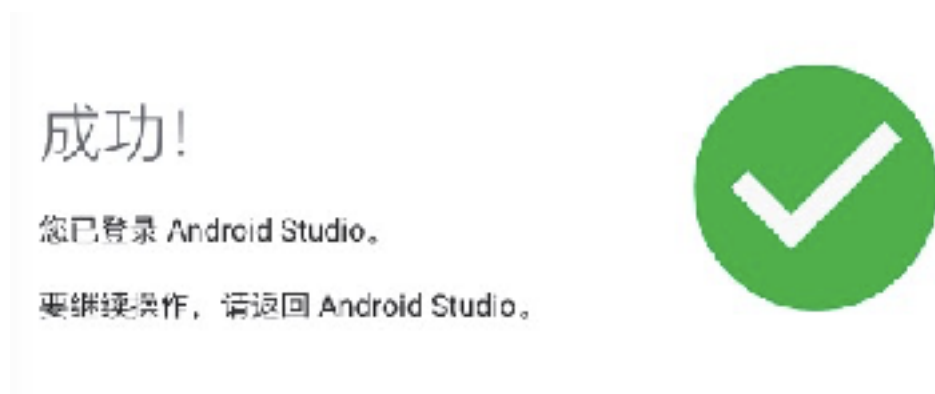
Lab13:

在早期叫做 Android Cloud to Device Messaging (C2DM)，後來改成使用 Google Cloud Messaging (GCM)，現在因為 Firebase 而推出了 Firebase Cloud Messaging (FCM)，FCM 簡單操作而且方便，只

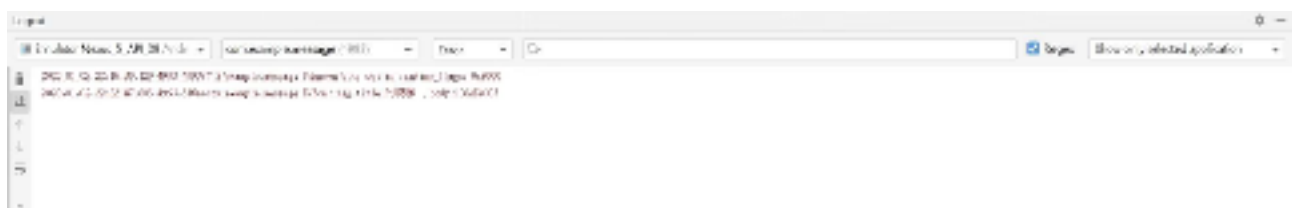
需要幾個簡單的設定，就可以完成推播的功能。
kotlin：

```
1 package com.example.myapplication
2
3 import android.util.Log
4 import com.google.firebase.messaging.FirebaseMessagingService
5 import com.google.firebase.messaging.RemoteMessage
6
7
8 class MyMessagingService : FirebaseMessagingService() {
9     companion object {
10         const val TAG = "MessagingService"
11     }
12     override fun onNewToken(token: String) {
13         super.onNewToken(token)
14         Log.e(TAG, "onNewToken: $token")
15     }
16     override fun onMessageReceived(msg: RemoteMessage) {
17         super.onMessageReceived(msg)
18         Log.e(TAG, "onMessageReceived: ${msg.from}?: """)
19         for (entry in msg.data.entries)
20             Log.e("tag: message", "msg: ${entry.key}/${entry.value}")
21     }
22 }
```

Firebase set up:



run成功頁面：



1 通知

通知標題 ②

刘晓敏

通知文字

108AE40J3

通知圖片 (選填) ③

範例: <https://yourapp.com/image.png> 查

通知名稱 (選填) ④

輸入名稱 (選用)

表面預覽

這個預覽畫面可讓您大致瞭解訊息在行動裝置上的呈現方式。系統會根據不同裝置調整資訊的顯示方式。如要取得資訊樣本，請使用書籍放置進行測試。

傳送測試訊息

初始狀態

已展開的快速畫面

刘晓敏

108AE40J3

AA

心得報告：

最後一次作業讓我學習到了新的東西，在日常生活中其實我們還蠻常使用到的，只是我們不知道那個是怎麼去實現的，透過這一次作業我了解到了原來手機的頁面是這麼實現的，引入JSON以及使用推播。我們在生活中的應用如果留心觀察都能看見我們的在lab中學過的知識。