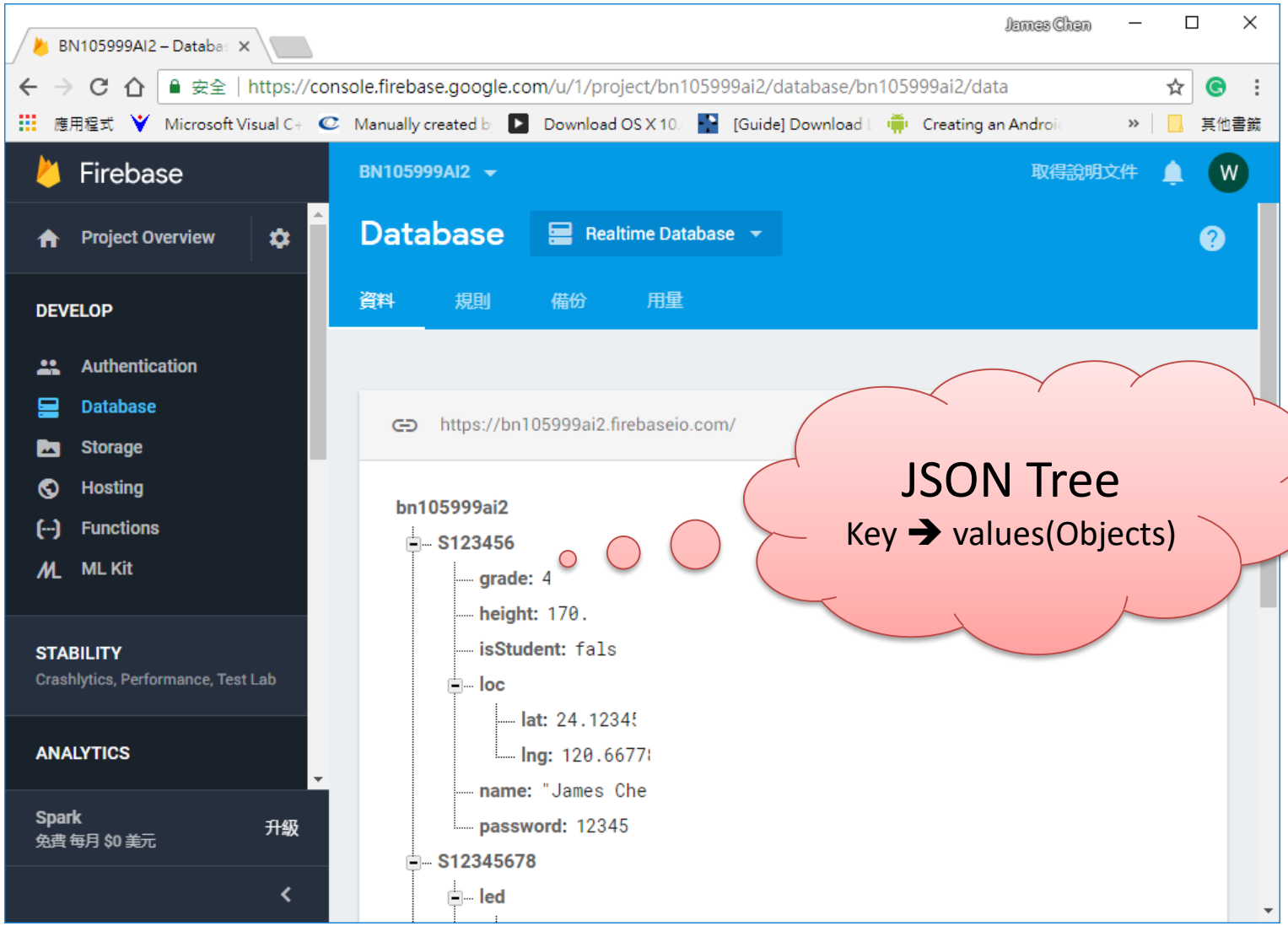
Firebase 雲端服務

James Chen
jameschen@hust.edu.tw

Firebase 主要服務

- **Cloud Messaging (FCM/GCM)**
  - Send/receive notifications
- **Realtime Database (DataStore)**
  - JSON tree, NoSQL
  - https://youtu.be/U5aeM5dvUpA
- **Authentication**
  - Integrate with identity providers or email
  - Ex: Google, Twitter, Facebook, … …

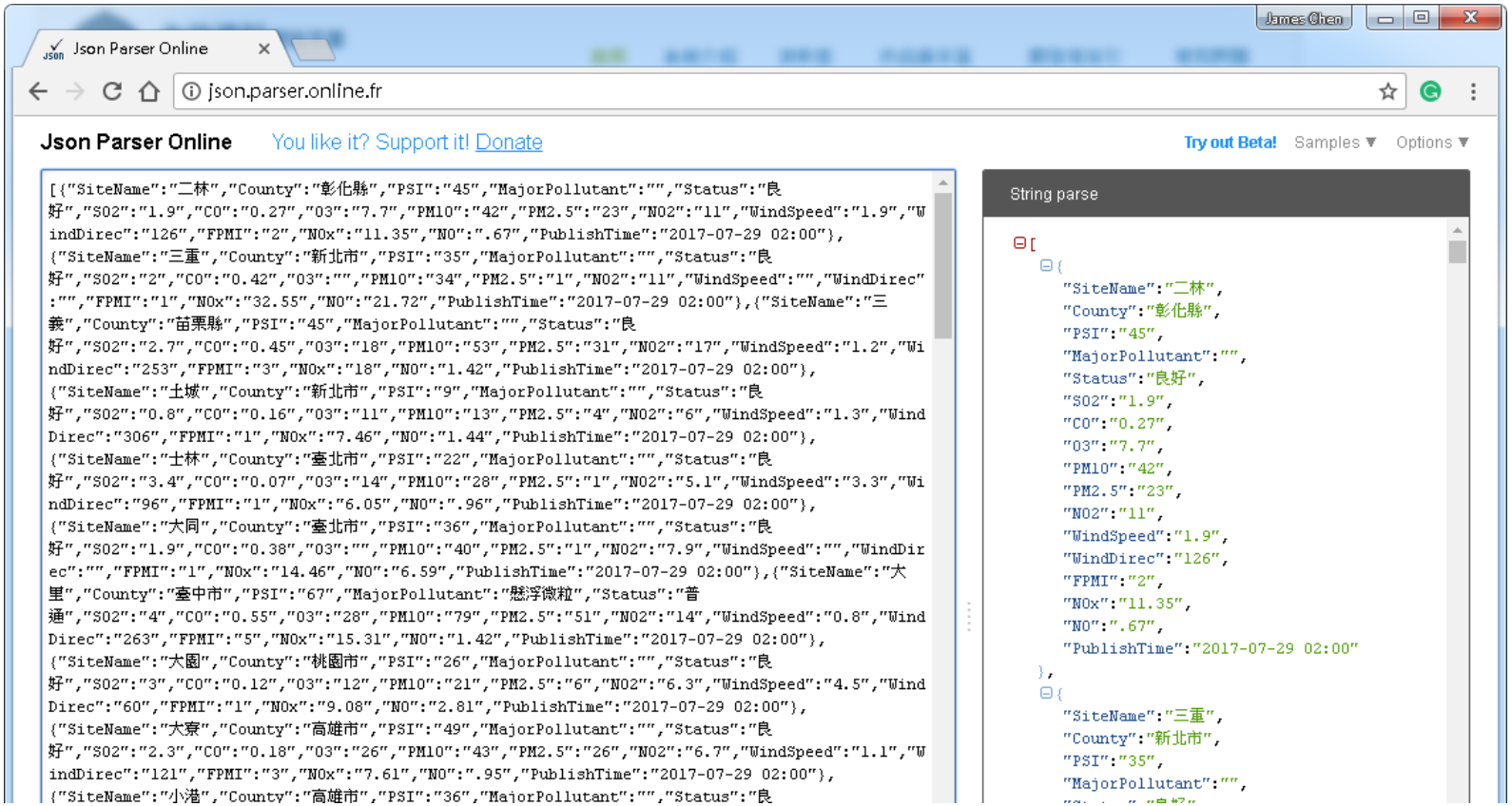
Firebase Realtime Database



https://firebase.google.com/docs/database/android/structure-data?utm_source=studio

利用線上工具來解析 JSON 格式

- JSON Parser, JSON Editor, …



Firebase **即時**資料庫的特點

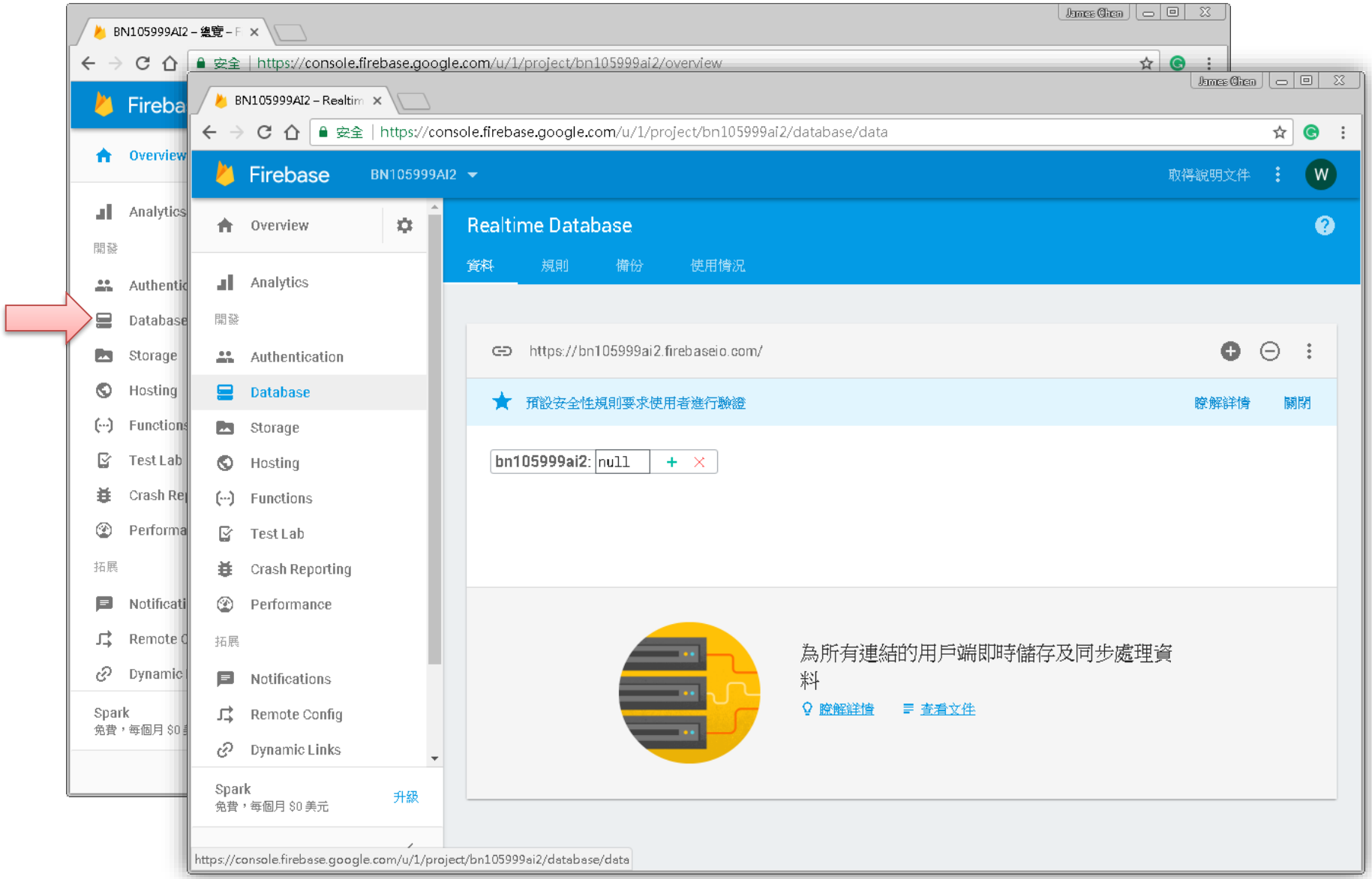
1. An GUI based Management Console;
2. On the cloud for any devices;
3. Smart **authentication** provides secure access;
4. Being a real-time database;
5. … …

https://console.firebase.google.com

Firebase 主控台 – 建立專案

- https://console.firebase.google.com (Gmail帳號)

Firebase 主控台 – 專案總覽



線上維護 Firebase 的資料

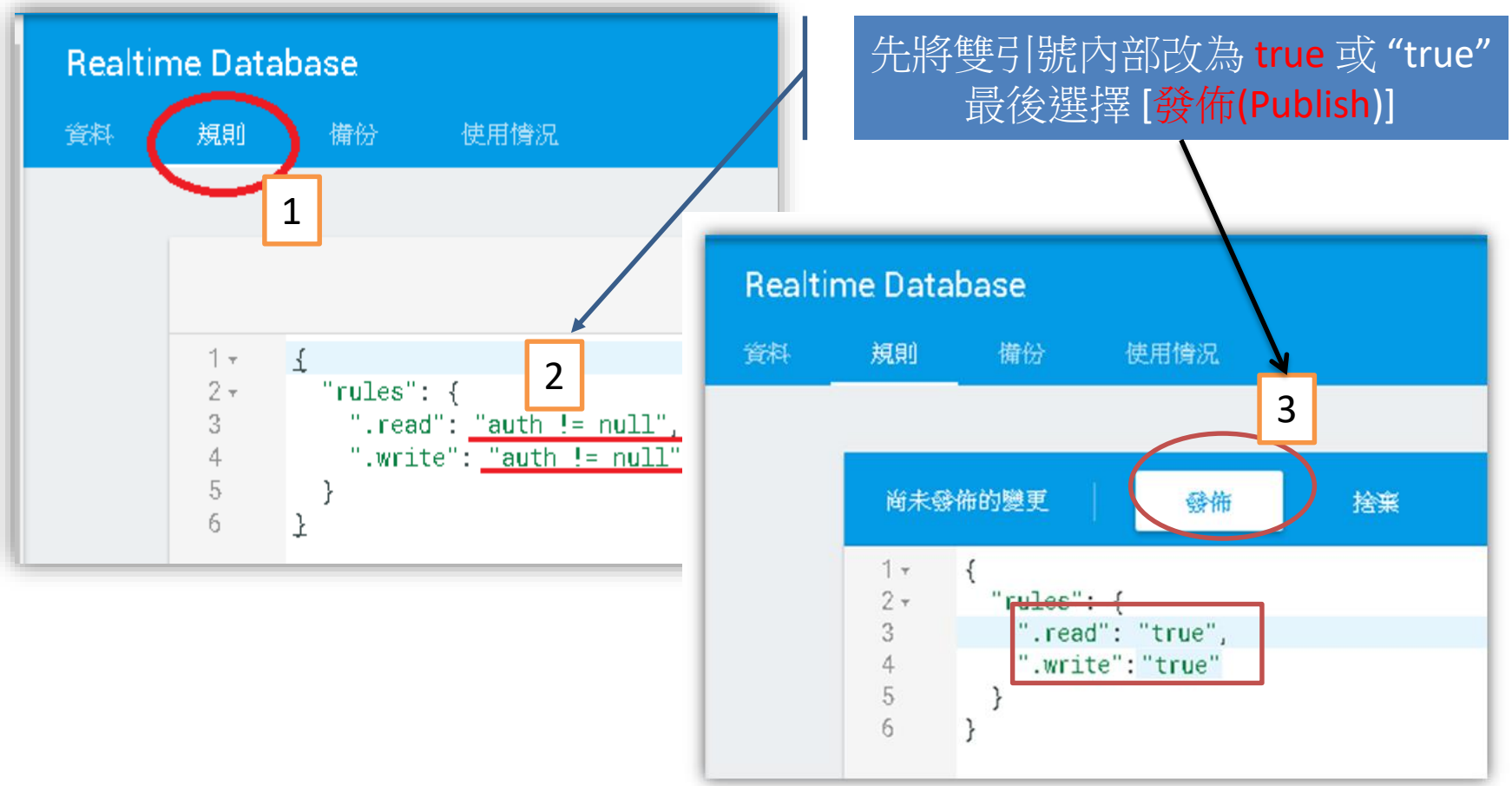
- **Firebase 資料是以 key, value 方式儲存 (JSON)**
- **新增**: 點擊 [+] → 輸入對應的 key與 value , 再按【新增】
- 例如：加入你自己的個人資料；類似以下內容：
  - name(姓名), id(學號), grade(年級), age(年紀), …

請**手動**輸入**測試資料** …

```
1.  {
2.    "S123456":
3.      {
4.            "name": "S123456",
5.            "password": "123456",
6.            "grade": 4,
7.            "height": 170.5,
8.            "weight": 75.5,
9.            "isStudent": false,
10.           "loc": { "lat":24.123456, "lng":120.667788}
11.     }
12. }
```

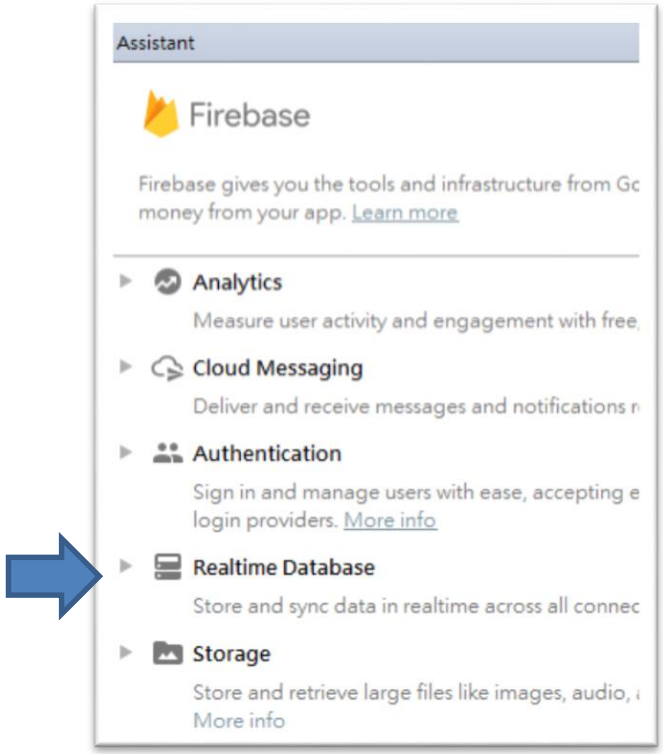
一件重要的事情 … 設定權限

- 至少開放 read 權限：改為 true ➔ 無須登入授權



終端設備的存取 …

- **Arduino/ESP8266/…**
  - https://github.com/firebase/firebase-arduino
- **Android App …**
  - Studio : [Tools] > [Firebase]



設定 Android Studio專案

- 修改專案 **build.gradle**

```
dependencies {
    classpath 'com.android.tools.build:gradle:3.1.4'
    // ...
    classpath 'com.google.gms:google-services:4.2.0'
}
```

- 修改 **app/build.gradle**
  - 加入外掛 **Firebase plugin** for Gradle

```
apply plugin: 'com.google.gms.google-services'
```

  - 加入程式庫相關 (dependencies):

```
compile 'com.google.firebase:firebase-database:17.0.0'
```

存取 Firebase 基本指令

- **取得資料庫參考物件 (DatabaseReference)**

```
FirebaseDatabase database = FirebaseDatabase.getInstance();
DatabaseReference myRef = database.getReference("theDataKey");
```

- **儲存資料**到Firebase 資料庫 : ***setValue(…), updateChildren(…)***

```
// 單純 key(ref) → value
myRef.setValue("theDataValue");
// 有下一階資料 (內含多組 key:value → HashMap )
myRef.updateChildren( hashmap, aCompletionListener );
```

- **讀取資料(異動)** : 使用 ***ValueEventListener*** , ***ChildEventListener***

```
myRef.addValueEventListener(new ValueEventListener() {
    public void onDataChange(DataSnapshot dataSnapshot) {
        String value = dataSnapshot.getValue(String.class);
    }
    public void onCancelled(DatabaseError error) {
        Log.w(TAG, "Failed to read value.", error.toException( ));
    }
});
```

https://firebase.google.com/docs/database/android/read-and-write

寫入資料

// datatype: String, Long, Double, Boolean,

// datatype: Map<String, Object>, List<Object>

1. ***FirebaseDatabase database =***

   FirebaseDatabase.getInstance( );

2. ***DatabaseReference myRef =***

   database.getReference("S123456/regid");

3. *myRef.setValue(“1234567890");*

讀取資料

```
1.  DatabaseReference myRef = database.getReference("S123456");
2.  myRef.addValueEventListener(new ValueEventListener( ) {
3.      @Override
4.      public void onDataChange(DataSnapshot dataSnapshot) {
5.          // Called once: (1) initialize ; (2) data at this location is updated.
6.          String value = dataSnapshot.getValue(String.class);
7.          Log.d(TAG, "Value is: " + value);
8.      }

9.      @Override
10.     public void onCancelled(DatabaseError error) {
11.         // Failed to read value
12.         Log.w(TAG, "Failed to read value.", error.toException());
13.     }
14. });
```

即時監看Firebase 資料變化

(**ChildEventListener**)

```
Firebase.setAndroidContext(this);
String url = "https://yourUrl.firebaseio.com/user";
new Firebase(url).addChildEventListener(…);
```

```
DatabaseReference myRef = FirebaseDatabase.getInstance( ).getReference();
myRef.addChildEventListener ( new ChildEventListener() {
    public void onChildAdded(DataSnapshot dataSnapshot, String s) {
        //  (String) dataSnapshot.child("name").getValue();
    }
    public void onChildRemoved(DataSnapshot dataSnapshot) {
        // (String)  dataSnapshot.child("name").getValue());
    }
     public void onChildMoved(DataSnapshot dataSnapshot, String s) {  /* … */ }
     public void onCancelled(FirebaseError firebaseError) {    }
     public void onChildChanged(DataSnapshot dataSnapshot, String s) { }
  }
);
```

Configure ProGuard (*)

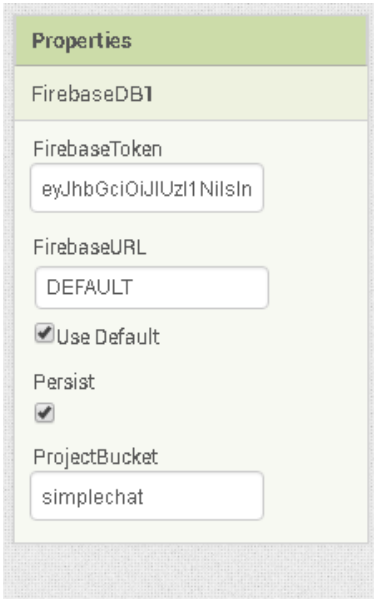
```
# Add this global rule
-keepattributes Signature


# This rule will properly ProGuard all the model classes in
# the package com.yourcompany.models. Modify to fit the structure
# of your app.
-keepclassmembers class com.yourcompany.models.** {
    *;
}
```

利用 App Inventor 存取 Firebase 資料

- **實驗元件 FirebaseDB**：方便AI2 存取 Firebase 即時資料庫
  - **模擬器無法支援!! 必須安裝到實機、進行測試。**
- **屬性**
  - FirebaseToken：自動產生，不需更改
  - FirebaseURL：必須在設計時就給定網址; **預設的測試網址: DEFAULT**
  - FirebaseBucket：一般是**第一層名稱(key)**

https://xxx.firebaseio.com

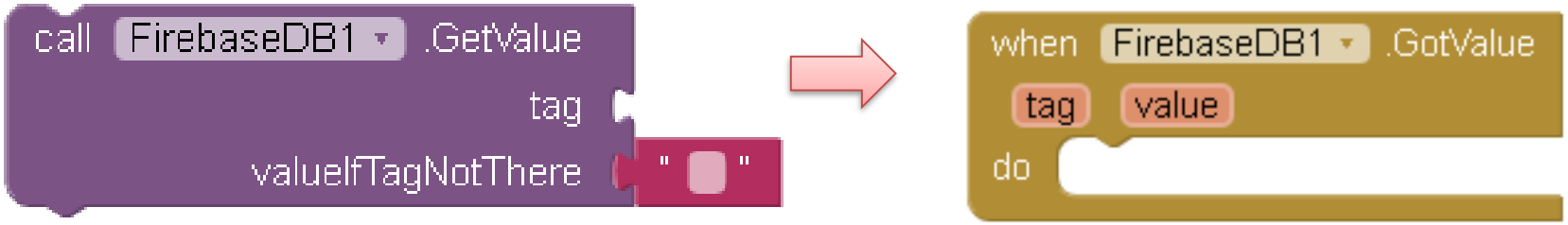
- **方法**
  - 讀取**標籤**值(GetValue)、儲存資料(StoreValue)、增加標籤資料(AppendValue)
  - 讀取並移除首項資料(RemoveFirst)
  - 讀取標籤清單(GetTagList)、清除指定標籤(ClearTag)
- **事件**
  - 取得數值(GotValue)
  - 資料改變(DataChanged)
  - 資料庫錯誤(FirebaseError)
  - 取得首項資料(FirstRemoved)
  - 取得標籤清單(TagList)



http://ai2.appinventor.mit.edu/reference/other/firebaseIntro.html

Firebase主要積木使用時機

- **讀取資料**：非同步取得資料

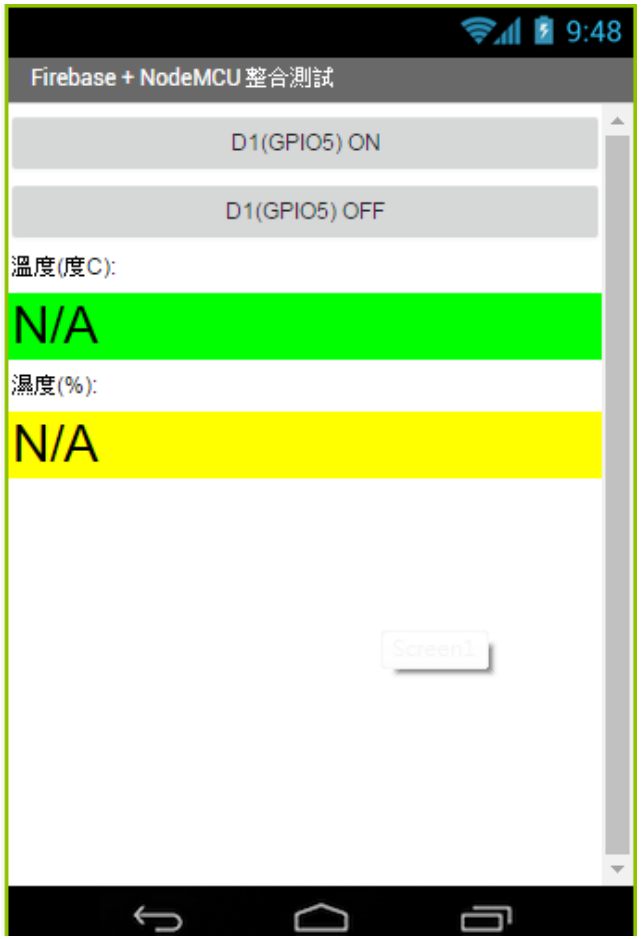


- **儲存資料**:

- **當資料被改變、或錯誤發生**：主動通知

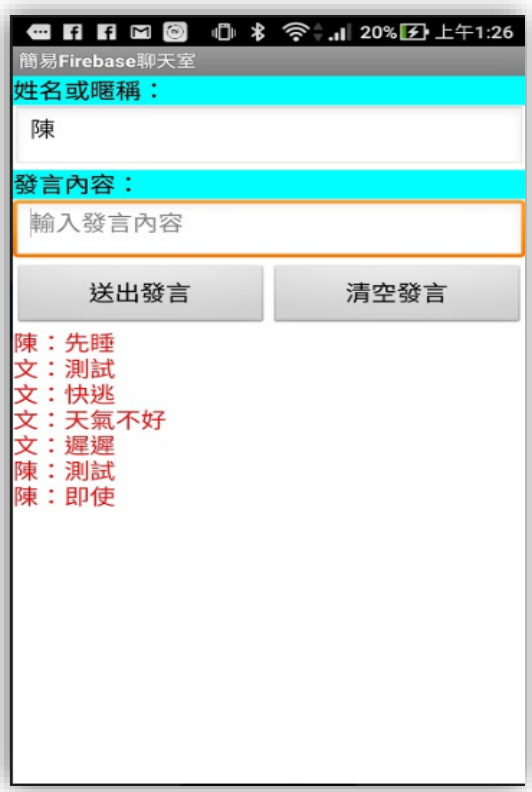
FirebaseTest 即時資料庫體驗

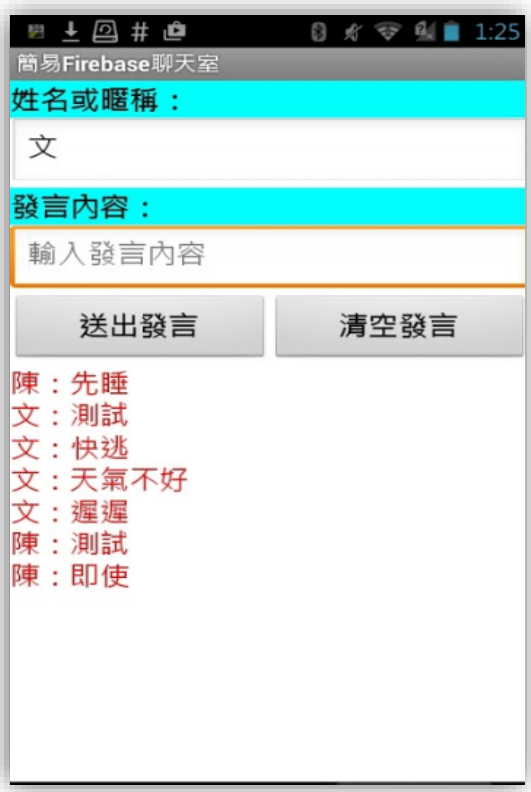
- 匯入專案：Lab0308_FirebaseTest.aia



FirebaseChatroom 聊天室體驗

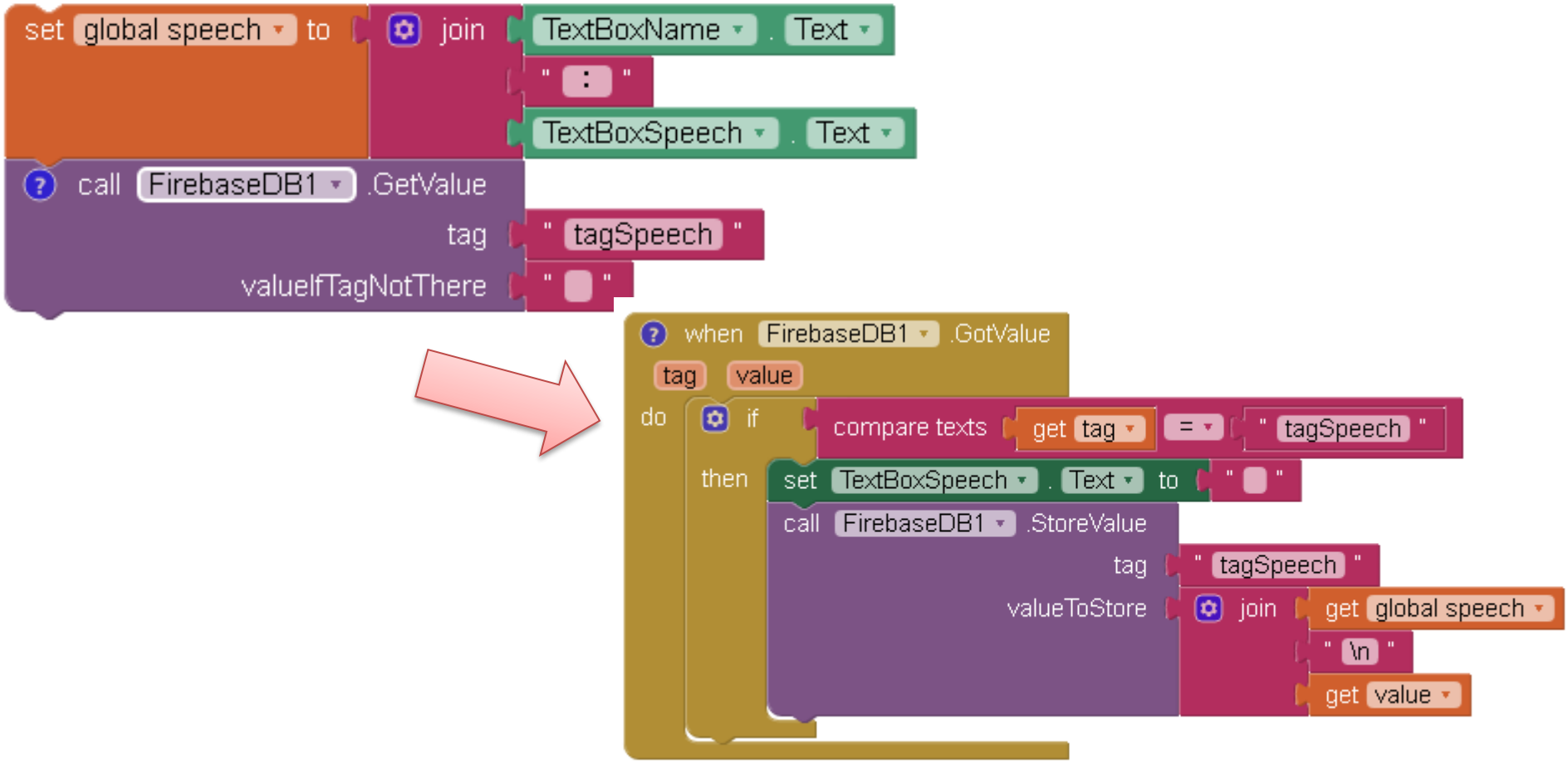
- 匯入專案 ：Lab0308_FirebaseChatroom.aia
- 一起聊天的使用者必須安裝**同一App**





Firebase 聊天室的基本架構-1

- **發言時**：
  - 確認必要資料都輸入完成後，**先讀取遠端Firebase的聊天室內容**
  - 取得最新的聊天室內容後，加上自己的發言，再回寫到遠端Firebase



Firebase 聊天室的基本架構-2

- **聊天室內容有更新時(會主動通知)：**
  - 判斷為聊天室內容之後，更新到畫面顯示區