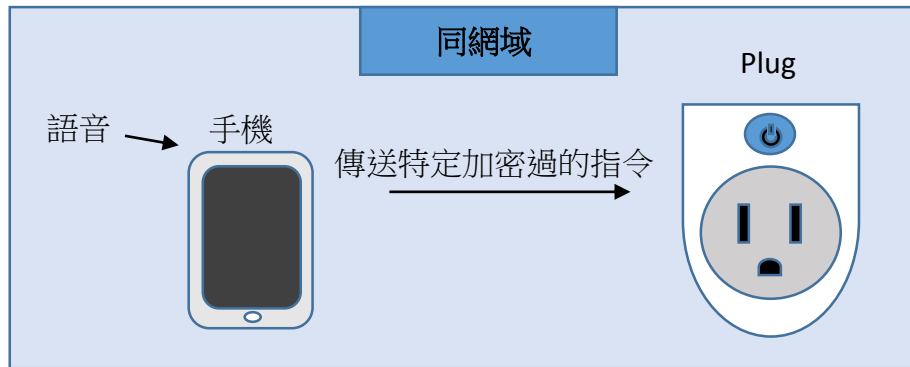


# Smart wifi plug HS100 以語音發送指令

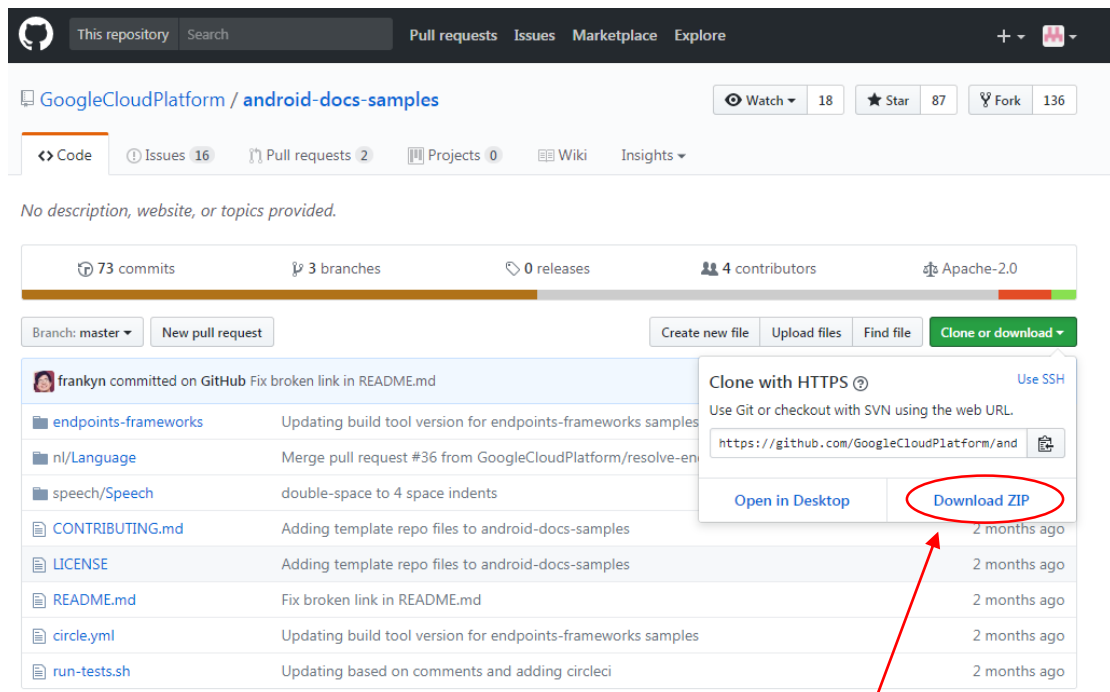
蕭偉銘 徐嘉昊 陳品翰

## 流程



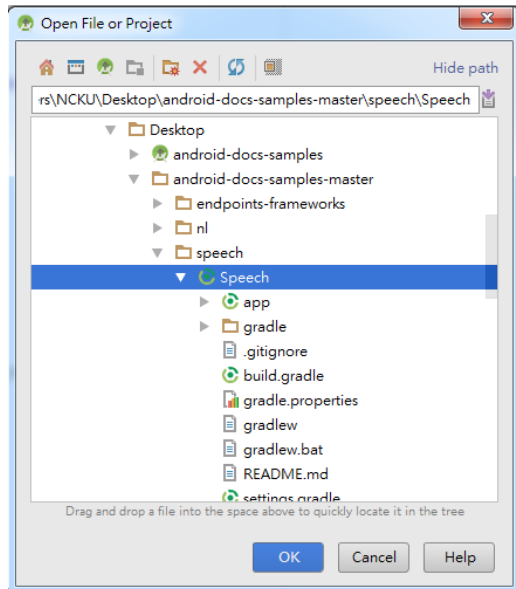
## 步驟:

- 1.使用 github 上的資源，下載已製作好具有語音接收能力的 android studio 檔案包: <https://github.com/GoogleCloudPlatform/android-docs-samples>

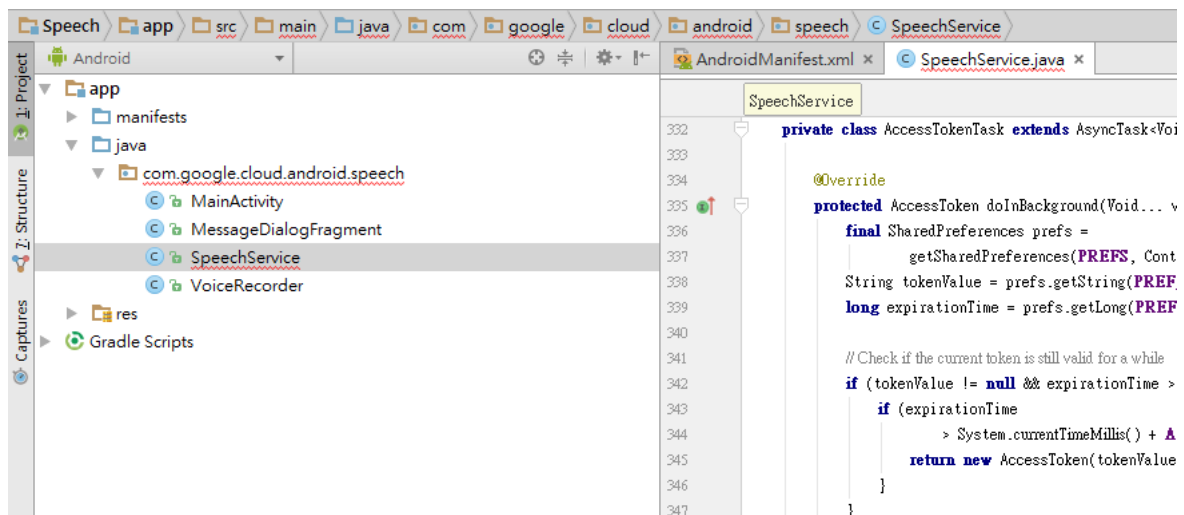


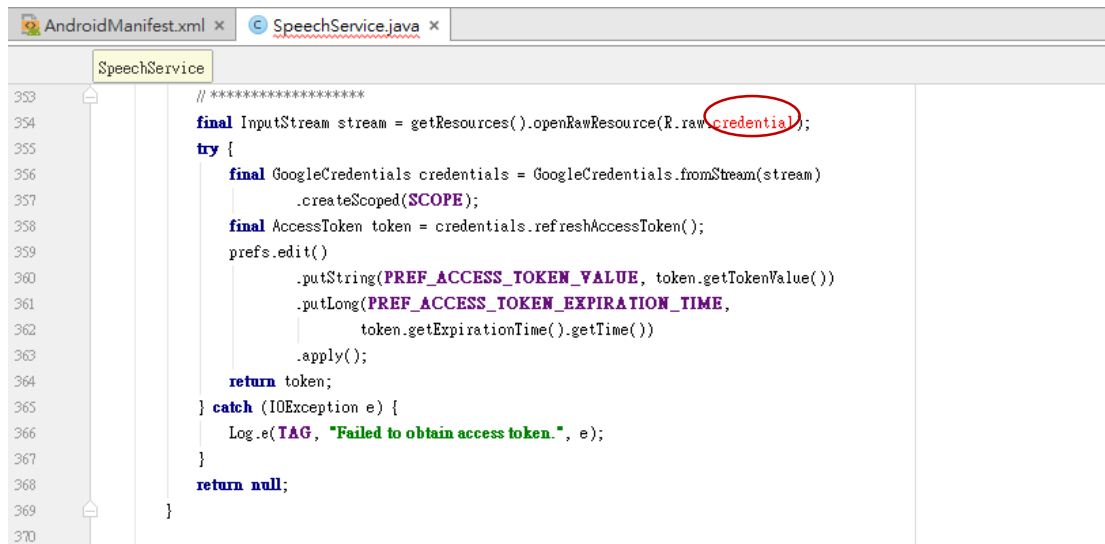
下載並解壓縮

2.使用 Android Studio 開啟該資料夾中的 Speech 資料夾(程式存放的位置)



3.直接執行將無法正常運作，發現 SpeechService.java 檔中有紅色底線 BUG。





```
353 // *****
354 final InputStream stream = getResources().openRawResource(R.raw.credential);
355 try {
356     final GoogleCredentials credentials = GoogleCredentials.fromStream(stream)
357         .createScoped(SCOPE);
358     final AccessToken token = credentials.refreshAccessToken();
359     prefs.edit()
360         .putString(PREF_ACCESS_TOKEN_VALUE, token.getTokenValue())
361         .putLong(PREF_ACCESS_TOKEN_EXPIRATION_TIME,
362             token.getExpirationTime().getTime())
363         .apply();
364     return token;
365 } catch (IOException e) {
366     Log.e(TAG, "Failed to obtain access token.", e);
367 }
368 return null;
369 }
370
```

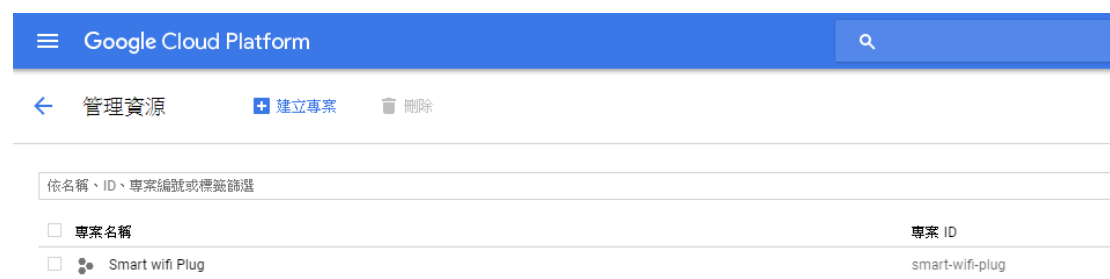
如上圖，在這段程式中 raw 資料夾中找不到這個憑證的檔案，代表  
需在 google 上申請使用 speech API 的憑證。

#### 4.取得 google speech API

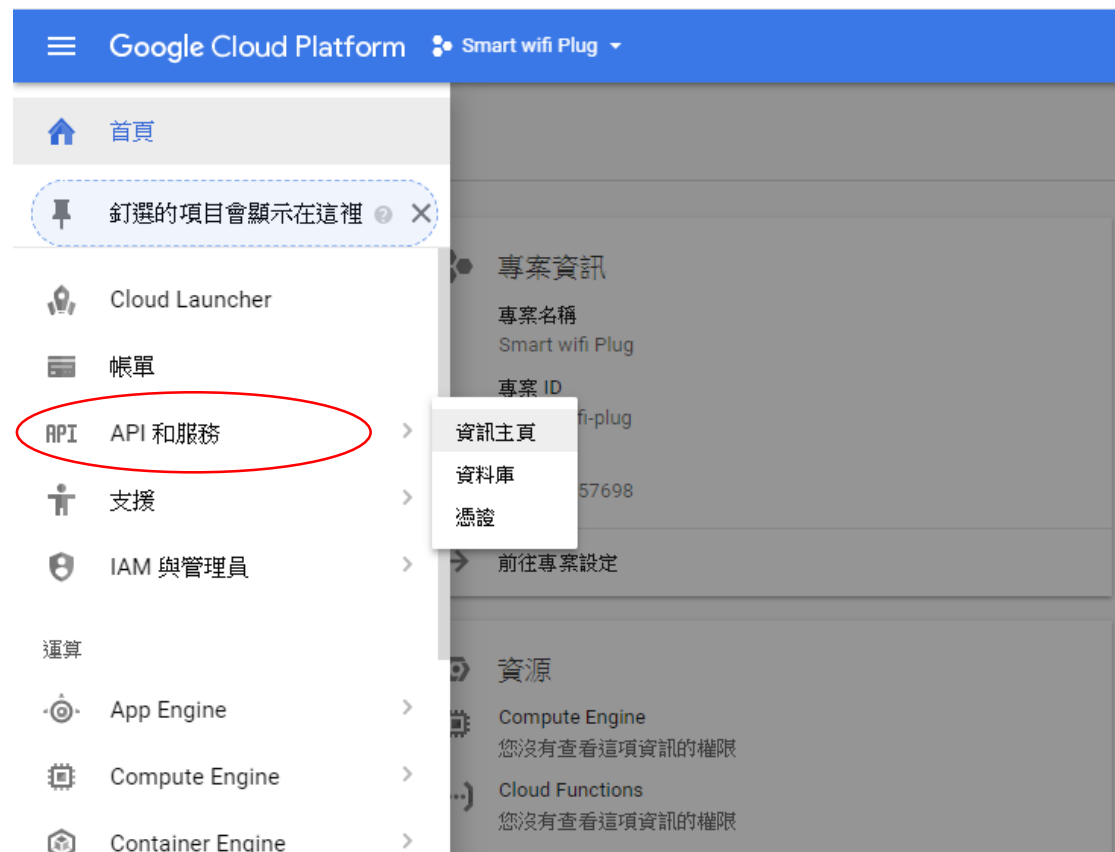
##### a.先進入此網站

<https://groups.google.com/a/chromium.org/forum/?fromgroups#!forum/chromium-dev>

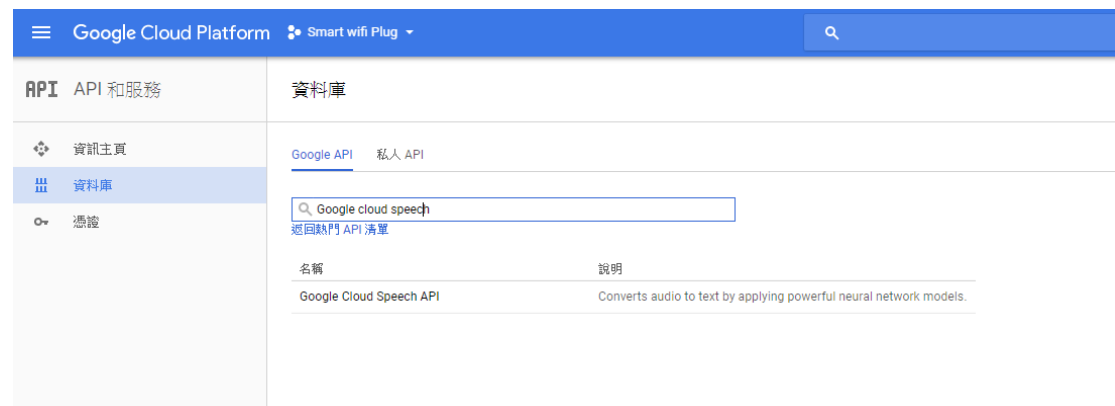
登入 google 並訂閱此網站群組，使你的 google 帳號擁有開法者身分，再  
到此網站建立一個專案 <https://console.cloud.google.com/cloud-resource-manager>，我的專案命名為 Smart Wifi Plug



b. 建立好專案後進入此專案，點選左邊列表中的 **API 服務**



進入後點擊「+啟用 API 和服務」並搜尋 Google Cloud Speech 的 API



c. 進入後點擊「啟用」，並完成付費申請，再來到左邊 **API 和驗證** 底下的“憑證”去建立 OAuth 憑證(credential)



點選「建立新的用戶端 ID」，會跳出一個小視窗，選擇「網路應用程式」，然後選「設定同意畫面」，在新畫面中，填入商品名稱，我填 smartplug

## 憑證

憑證

OAuth 同意畫面

網路驗證

電子郵件地址 ?

sky84003@gmail.com

向使用者顯示的產品名稱 ?

smartplug

首頁網址 (選填)

https:// 或 http://

產品標誌網址 (選填) ?

http://www.example.com/logo.png



這張圖片顯示使用者所看到的標誌樣式  
尺寸上限：120x120 像素

隱私權政策網址  
部署應用程式之後即必須設定

https:// 或 http://

服務條款網址 (選填)

https:// 或 http://

儲存

取消



當您要求使用您的用戶端 ID 存取使用者的私人資料時，系統會向使用者顯示同意畫面。凡是在這項專案中註冊的應用程式都會顯示這個畫面。

您必須提供電子郵件地址和產品名稱，OAuth 才能正常運作。

填完後，憑證頁面上會出現網路用戶 1

憑證

憑證

OAuth 同意畫面

網路驗證

建立憑證

刪除

建立憑證即可存取您啟用的 API，詳情請參閱 API 說明文件。

OAuth 2.0 用戶端 ID

名稱	建立日期	類型	用戶端 ID
網路用戶端 1	2017年9月15日	網路應用程式	942717457698-cfptbrf2mglvnjurm7p9a89v4jok7u6t.apps.googleusercontent.com

d.最後在「建立憑證」下選擇「建立服務帳戶金鑰」，服務帳戶選擇剛剛的產品名稱 smartplug，並選 json 格式建立

## ← 建立服務帳戶金鑰

服務帳戶

smartplug

金鑰類型

下載內含私密金鑰的檔案。請妥善保存這個檔案，金鑰一旦遺失即無法重新取得。

☒ JSON

建議選項

☐ P12

能與使用 P12 格式的程序碼向下相容

建立

取消

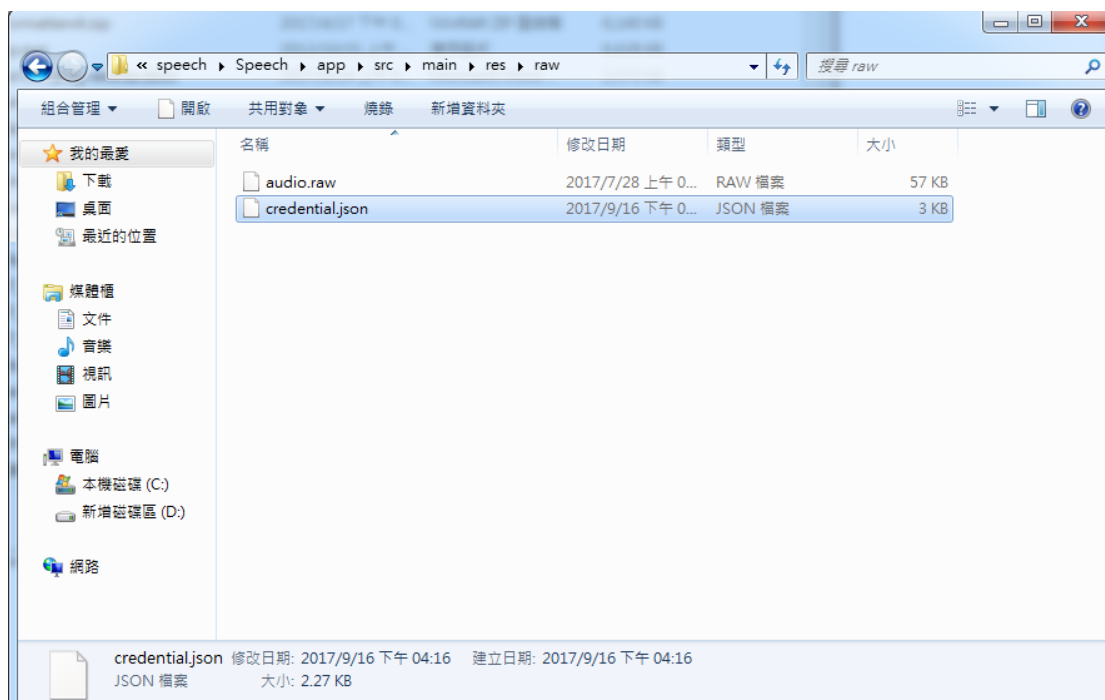
得到一個 json 格式檔，為可使用 google cloud speech API 權限的金鑰檔案。

新的私密金鑰

Smart wifi Plug-e50304d5a710.json已儲存至您的電腦。這是金鑰的唯一副本，請務必妥善保管。

[關閉](#)

5.取得此 API 後，將該 json 檔案改名為 SpeechService.java 上要求的檔案名稱“credential.json”並存放在目錄\android-docs-samples-master\speech\Speech\app\src\main\res\raw 下

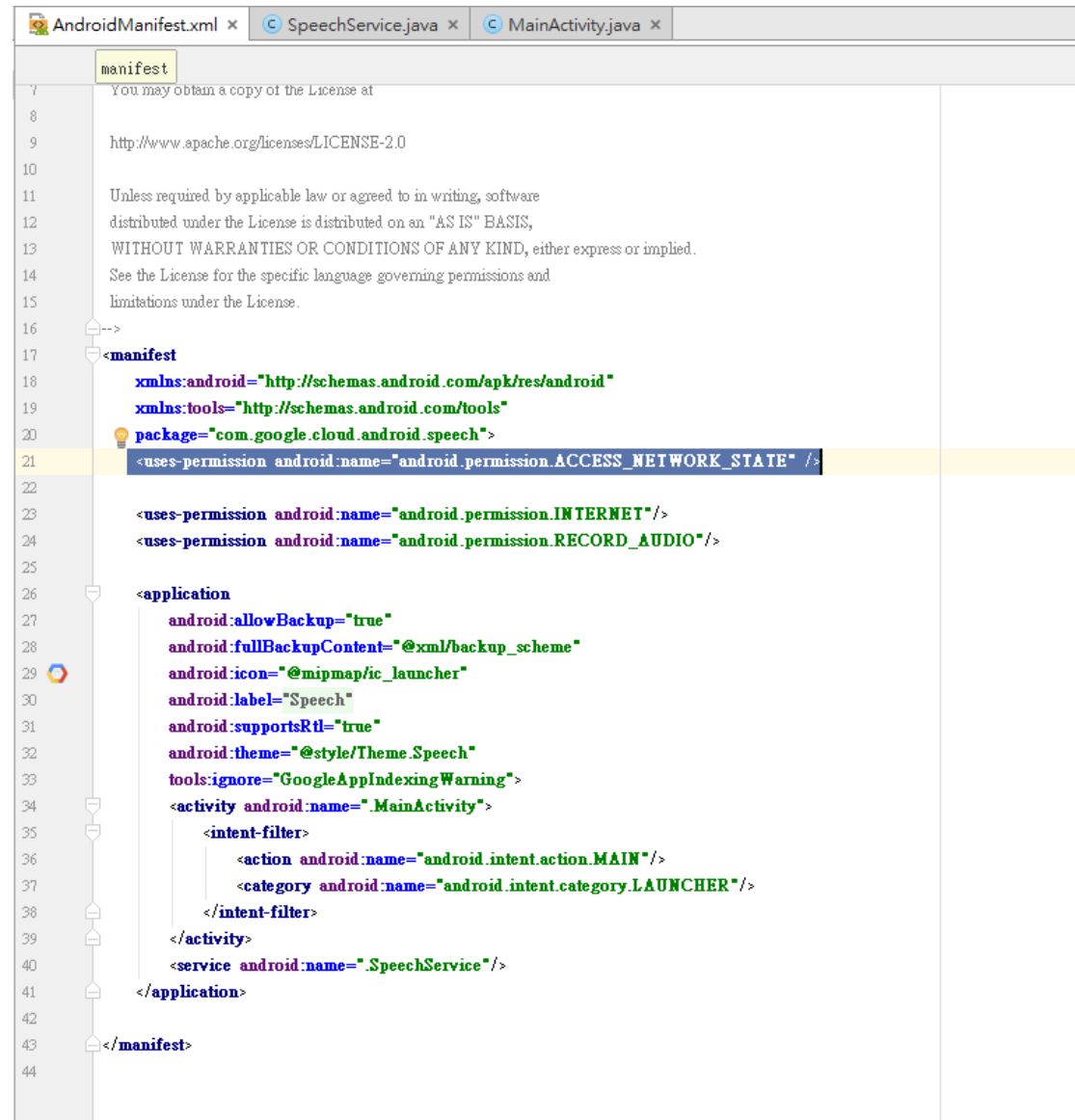


6. 做到這邊，已經可以使手機接收語音功能，我們需要將發送指令給開關的 `code` 加入。

a. 在 manifests 檔案中加入

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

使程式有接收存取網路資料的權限



```
manifest
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
<!-->
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.google.cloud.android.speech">
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.RECORD_AUDIO"/>
    <application
        android:allowBackup="true"
        android:fullBackupContent="@xml/backup_scheme"
        android:icon="@mipmap/ic_launcher"
        android:label="Speech"
        android:supportRtl="true"
        android:theme="@style/Theme.Speech"
        tools:ignore="GoogleAppIndexingWarning">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
        <service android:name=".SpeechService"/>
    </application>
</manifest>
```

b. 在 MainActivity 檔案下 SpeechService.Listener 中 `text` 變數即存放接受到的語音，我們對此變數內容做判斷，若內容為「開」則以 `socket` 傳送使開關打開的加密訊息，若內容為「關」則以 `socket` 傳送使開關關閉的加密訊息

```

250     }
251     if (mText != null && !TextUtils.isEmpty(text)) {
252         runOnUiThread() → {
253             if (isFinal) {
254                 mText.setText(null);
255                 mAdapter.addResult(text);
256                 mRecyclerView.smoothScrollToPosition(0);
257             } else {
258                 mText.setText(text);
259                 if (text.equals("開燈")){
260                     Thread thread; //執行緒
261                     thread= new Thread((Runnable) () → {
262                         Socket socket=new Socket();
263                         try {
264                             InetAddress addr = new InetAddress("192.168.72.133", 9999);
265                             socket.connect(addr);
266                             String s = "{\"system\":{\"set_relay_state\":{\"state\":1}}}\";
267                             int key = 171;
268                             int a;
269                             byte[] r2=new byte[s.length()+4];
270                             r2[0]=0;
271                             r2[1]=0;
272                             r2[2]=0;
273                             r2[3]=0;
274                             for (int i = 0; i < s.length(); i++) {
275                                 a = key ^ s.charAt(i);
276                                 key = a;
277                                 r2[i+4]=(byte) a;
278                             }
279                             DataOutputStream output = new DataOutputStream(socket.getOutputStream());
280                             output.write(r2);
281                             output.flush();
282                         } catch (Exception e) {
283                             e.printStackTrace();
284                         }
285                         try {
286                             socket.close();
287                         } catch (IOException e) {
288                             e.printStackTrace();
289                         }
290                     });
291                     thread.start();
292                 } else if (text.equals("關燈")){

```

## Code:

```

if(text.equals("開燈")){           //判斷 text 內容

    Thread thread;                 //新增執行緒

    thread= new Thread(new Runnable() {           //執行緒功能

        @Override

        public void run() {

            Socket socket=new Socket();;           //建立 socket

            try {

                InetAddress addr = new InetAddress("192.168.72.133", 9999);

                socket.connect(addr);

                String s = "{\"system\":{\"set_relay_state\":{\"state\":1}}}\"; //要傳送的指令

                int key = 171;           //駭客得到的 key

                int a;                   //加密過程用來與下個字符的 unicode 作 XOR 的運算元

                byte[] r2=new byte[s.length()+4]; //存放最後加密完的訊息

                r2[0]=0;

                r2[1]=0;

                r2[2]=0;

```



```

        r2[3]=0;           //規定前四項為 0
        for (int i = 0; i < s.length(); i++) {
            a = key ^ s.charAt(i);
            key = a;
            r2[i+4]=(byte) a;
        }                 //加密過程

        DataOutputStream output = new DataOutputStream(socket.getOutputStream());
        output.write(r2);
        output.flush();
    } catch (Exception e) {
    }

    try {
        socket.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

});

thread.start();           //啟用 thread
}

```

最後製成 **APK** 檔案即可在手機上以語音方式控制智慧開關。