
MORE Wi-Fi Controller SDK

Version 0.16.08.15

2016.08.15

MORE Wi-Fi Controller SDK.....	1
<i>Version 0.16.08.15</i>	1
1. Introduction.....	3
1.1. Overview	3
2. Service.....	4
2.1. Wi-Fi Controller SDK.....	4
2.2. Use SDK Library	4
2.3. Wi-Fi Function	7
3. Parameter Definition	8
3.1. OnCallbackResult Function Parameter.....	8
3.2. Handler 與 Callback listener 參數對應.....	9
3.3. 加密模式 Type.....	9
3.4. WIFI MODE	9

1. Introduction

1.1. Overview

MORE WIFI Controller SDK 主要提供 Smart Mobile Device 更快速與簡潔的開發模組，APP 開發人員只要將 MORE Wi-Fi Controller SDK 加入到自己的 APP 專案裡，透過 API 的呼叫即可控制 Wi-Fi SSID 新增、移除、查詢等，以及當使用者關閉 Wi-Fi 時能再開啟 Wi-Fi，目前提供 Android 版本 SDK Library。

2. Service

2.1. Wi-Fi Controller SDK

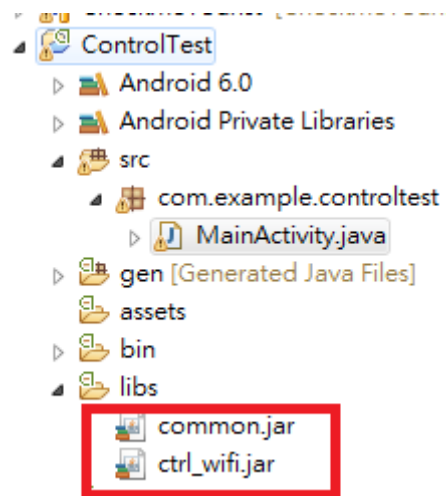
Wi-Fi Controller SDK 主要提供給開發者一個能新增移除 SSID、查看全部 SSID 列表、以及當使用者關閉 Wi-Fi 時能再開啟 Wi-Fi。

2.2. Use SDK Library

在使用 Wi-Fi Controller SDK 前，需要在 Android Manifest 加入網路、Wi-Fi 權限，此模組會透過 Handler 或 Callback Listener 回傳是否有錯誤產生。

使用步驟：

1. 先將以下 jar 檔加入到 Android 開發專案的 libs 資料夾



2. 在 `<manifest` 加入以下權限

```
<uses-permission android:name="android.permission.CHANGE_NETWORK_STATE" ></uses-permission>
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" ></uses-permission>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" ></uses-permission>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" ></uses-permission>
<uses-permission android:name="android.permission.WAKE_LOCK"></uses-permission>
<!--Android 6.0 need to add-->
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" > </uses-permission>
```

3. 匯入以下 library

```
import java.util.ArrayList;
import java.util.HashMap;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.util.Log;
import sdk.ideas.common.OnCallbackResult;
import sdk.ideas.ctrl.wifi.WifiHandler;
```

4. 設定Wi-Fi Handler

```
WifiHandler mWifiHandler = new WifiHandler(this);
```

5. 本SDK 提供Handler 和 Callback Listener方法來去知道程式是否有成功執行，開發者可以以自己喜好去做設定，以下是Handler 方法範例

1) 先宣告Handler

```
private Handler theHandler = new Handler()
{
    @Override
    public void handleMessage(Message msg)
    {
        switch (msg.what)
        {
            case CtrlType.MSG_RESPONSE_WIFI_HANDLER:
                Log.d("handler response: ", "Result: " + String.valueOf(msg.arg1) + " From: "
                    + String.valueOf(msg.arg2) + " Message: " +
                    ((HashMap<String,String>)msg.obj) );
                break;
            default:
                break;
        }
    }
};
```

2) 設定此 handler

```
mWifiHandler.setHandler(theHandler);
```

6. Callback Listener 方式

```
mWifiHandler.setOnCallbackResultListener(new OnCallbackResult()
{
    @Override
    public void onCallbackResult(int result, int what, int from, HashMap<String,
String> message)
    {
        Log.d("Listener response: ", "Result: " + String.valueOf(result) + " What:
" + String.valueOf(what)
            + " From: " + String.valueOf(from) + " " + message);
    }
});
```

7. 儲存 Wi-Fi SSID

```
mWifiHandler.saveWIFIConfig("SSID", "password", 3, false);
```

8. 掃描取得附近 SSID 資訊

```
ArrayList<String >scanResult = mWifiHandler.getScanResult();
```

9. 開始去監聽 Wi-Fi 是否有變更，當使用者有關閉 Wi-Fi 時， 會嘗試去再
開啟 Wi-Fi。

```
mWifiHandler.startListenAction();
```

10. 讓 Wi-Fi 即使是在螢幕變黑休眠時，依然是連接網路的模式。

```
mWifiHandler.createWIFILockAndLockNow("test", 1);
```

2.3. Wi-Fi Function

Wi-Fi Controller API	
Function	Description
WifiHandler(Context context)	建構元
void saveWIFIConfig(String ssid, String password, int type, boolean linkThisSSIDNow)	儲存 SSID 跟密碼，type 為加密模式，參考 3.3，linkThisSSIDNow 當 false 時僅儲存 SSID 跟密碼，當 true 時會嘗試去連接此組 SSID，注意會先斷開目前 WIFI 連線。
void removeWIFIConfig(String ssid)	移除 SSID 跟密碼。
void createWIFILockAndLockNow(String lockName, int lockType)	此 function 能夠阻止 wifi 進入睡眠狀態，使 wifi 一直處於活躍狀態，lockType 參考 3.4。
void releaseWIFILock()	解除 Wi-Fi Lock。
void disconnectWIFI(String ssid)	停止使用這組 SSID 且不能再連線此組 SSID，如果此 SSID 正在連線中，則會中斷連線。
ArrayList<String> getScanResult()	取得掃描 SSID 結果，其中把包括：BSSID、SSID、capabilities、frequency、level 等資訊
void setOpenWifiAuto(boolean openWifiAuto)	設定當使用此 SDK 的 APP 運行中時，使用者關閉 wifi 是否要自動開啟

void startListenAction()	開始監聽網路與 wifi 是否關閉
void stopListenAction()	停止監聽網路與 wifi 是否關閉
void setHandler(Handler handler)	設定 Handler，參數對應可參考 3.2
void setOnCallbackResultListener(OnCallb ackResult listener)	設定 callback listener，其參數參考 3.1

3. Parameter Definition

3.1. OnCallbackResult Function Parameter

result :

Value	Description
1	SUCCESS
0	UNKNOWN ERROR
-2	ILLEGAL STRING LENGTH OR NULL ERROR
-21	SAVE WIFI SSID CONFIG OPERATION ERROR
-22	WIFI SSID NOT FOUND ERROR

what :

Value	Description
1041	WIFI CONTROLLER HANDLER

from :

Value	Description
0	Wi-Fi Lock Function
1	Remove Wi-Fi SSID Config Function
2	Save Wi-Fi SSID Config Function
3	Disconnect Wi-Fi SSID Config Function

message :

key	Description
“message”	敘述執行成功或錯誤訊息

3.2. Handler 與 Callback listener 參數對應

Handler	OnCallbackResult
msg.what	int what
msg.arg1	int result
msg.arg2	int from
msg.obj	HashMap<String, String> message

3.3. 加密模式 Type

Value	Description
1	NO_PASSWORD
2	WEP
3	WPA /WPA2 PSK

3.4. WIFI MODE

Value	Description
1	WIFI MODE FULL , 掃描並自動的嘗試去連接一個曾經批 配過的 SSID
2	WIFI MODE SCAN ONLY , 僅掃描
3	WIFI MODE FULL HIGH PERF , 在 WIFI MODE FULL 模式的基礎上 , 保持最佳性能