WifiManager

陳健文

什麼是 WifiManager ?

WifiManager 是一組用來管理 WiFi 連線所有面向的 API

運用 WifiManager，你可以：

列出設定好的無線網路

取得目前無線網路的資料

取得存取點(Access Point，AP)掃描結果

依照 WiFi 的連線狀況發出對應的廣播

操作流程

取得 WifiManager

檢查 WiFi 開關

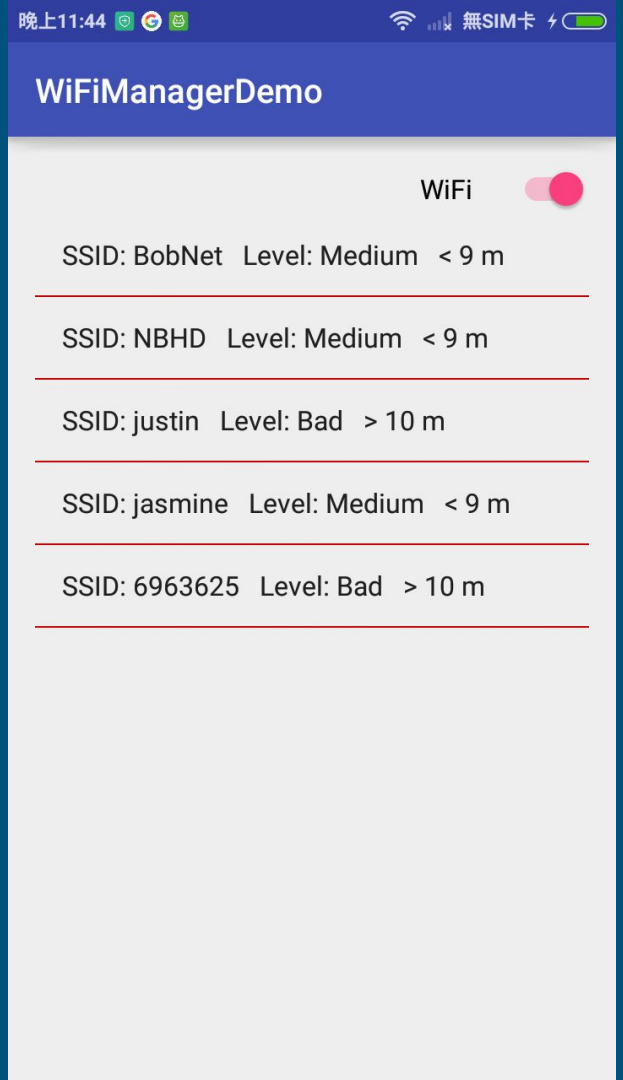
註冊廣播

SCAN_RESULTS_AVAILABLE_ACTION

WIFI_STATE_CHANGED_ACTION

版面配置

- 加入 Switch
  - 設定寬度
  - 設定文字大小
- 加入 ListView
  - 設定分隔線顏色
  - 設定分隔線寬度


晚上11:44
無SIM卡
WiFiManagerDemo
WiFi
SSID: BobNet Level: Medium < 9 m
SSID: NBHD Level: Medium < 9 m
SSID: justin Level: Bad > 10 m
SSID: jasmine Level: Medium < 9 m
SSID: 6963625 Level: Bad > 10 m

Layout

```
<Switch
    android:layout_width="100dp"
    android:layout_height="30dp"
    android:text="WiFi"
    android:id="@+id/wifi_switch"
    android:textSize="16dp"
    android:layout_alignParentTop="true"
    android:layout_alignParentRight="true"
    />
<android.support.v7.widget.ListViewCompat
    android:id="@+id/ssid_listview"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@id/wifi_switch"
    android:divider="@android:color/holo_red_dark"
    android:dividerHeight="1dp"/>
```

取得 WifiManager 與 Switch 並設定事件傾聽器

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    lv = (ListViewCompat) findViewById(R.id.ssid_listview);
    receiver = new WifiScanReceiver();
    wifiManager = (WifiManager) getSystemService(Context.WIFI_SERVICE);
    sw = (Switch) findViewById(R.id.wifi_switch);
    sw.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
        @Override
        public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
            if (isChecked) {
                if (!wifiManager.isWifiEnabled()){
                    //enable wifi
                    wifiManager.setWifiEnabled(true);
                    wifiManager.startScan();
                }
            } else {
                if (wifiManager.isWifiEnabled()) {
                    //disable wifi
                    wifiManager.setWifiEnabled(false);
                }
            }
        }
    });

    if(wifiManager.isWifiEnabled()){
        sw.setChecked(true);
        wifiManager.startScan();
    }
    else
        sw.setChecked(false);
}
```

以動態方式註冊廣播

```
@Override
protected void onResume(){
    registerReceiver(receiver, new IntentFilter(WifiManager.SCAN_RESULTS_AVAILABLE_ACTION));
    registerReceiver(receiver, new IntentFilter(WifiManager.WIFI_STATE_CHANGED_ACTION));
    super.onResume();
}

@Override
protected void onPause(){
    unregisterReceiver(receiver);
    super.onPause();
}
```

編寫廣播接收器 - 處理 SCAN_RESULTS... 廣播

```
private class WifiScanReceiver extends BroadcastReceiver{
    @Override
    public void onReceive(Context c, Intent intent){
        switch(intent.getAction()) {
            case WifiManager.SCAN_RESULTS_AVAILABLE_ACTION:
                List<ScanResult> wifiList = wifiManager.getScanResults();
                ssids = new String[wifiList.size()];
                StringBuilder sb;
                for (int i = 0; i < wifiList.size(); i++) {
                    sb = new StringBuilder();
                    sb.append("SSID: ");
                    sb.append(wifiList.get(i).SSID);
                    sb.append("   Level: ");
                    // >-35dBm 100%, =-65dBm 50%, =-95dBm 1%
                    if (Integer.valueOf(wifiList.get(i).level) >= -35)
                        sb.append("Excellent   < 3 m");

                    else if (Integer.valueOf(wifiList.get(i).level) < -35 &&
                            Integer.valueOf(wifiList.get(i).level) >= -50)
                        sb.append("Good   < 6 m");

                    else if(Integer.valueOf(wifiList.get(i).level)<-50 &&
                            Integer.valueOf(wifiList.get(i).level)>=-70)
                        sb.append("Medium   < 9 m");

                    else
                        sb.append("Bad   > 10 m");

                    ssids[i] = sb.toString();
                }
                ssidAdapter = new ArrayAdapter<String>(c, android.R.layout.simple_list_item_1, ssids);
                lv.setAdapter(ssidAdapter);
                break;
```

編寫廣播接收器 - 處理 SCAN_RESULTS... 廣播

```
case WifiManager.WIFI_STATE_CHANGED_ACTION:
    int state = wifiManager.getWifiState();
    switch(state){
        case WifiManager.WIFI_STATE_DISABLED:
            ssids = null;
            ssidAdapter = null;
            ssids = new String[0];
            ssidAdapter = new ArrayAdapter<String>(c, android.R.layout.simple_list_item_1, ssids);
            lv.setAdapter(ssidAdapter);
            break;

        case WifiManager.WIFI_STATE_DISABLING:
        case WifiManager.WIFI_STATE_ENABLING:
            break;

        case WifiManager.WIFI_STATE_ENABLED:
            wifiManager.startScan();
            break;

    }
```

關於 WiFi 訊息的強度

ScanResults 中的 level 值

level 值為訊號強度與 1 mW 的比值取 log 乘上 10 即 10 log(訊號強度/1mW)

WiFi 訊號強度會因距離而變小故 level 值為負值 (不會大於 1mW)

level 值大於 -35 的話，WiFi 訊號強度通常會標示成滿格(100%)

level 值小於 -95 的話，WiFi 訊號強度通常會標示成幾無訊號(1%)

以訊號強度來估算手機與 AP 的距離需要以機率模型來估測

別忘了要開權限啊！

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

Demo