



BroadcastReceiver



陳健文



什麼是 BroadcastReceiver ?

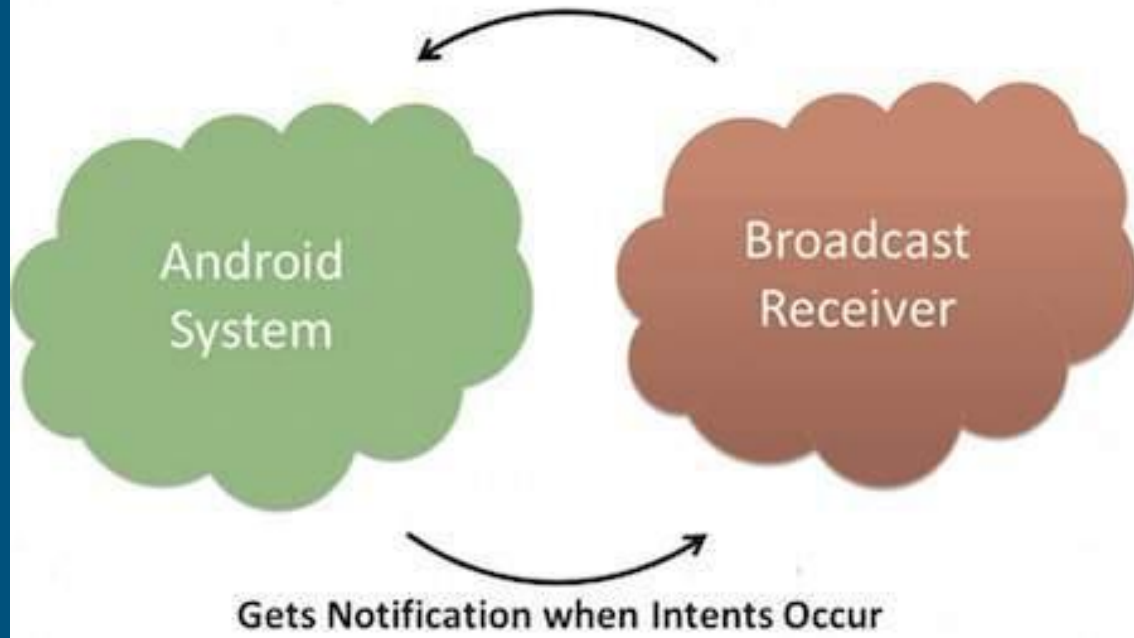
廣播接收器。用來註冊接收 Android 系統中各種廣播訊息(Intent)的接收器。

Android 中的廣播是指跨行程通訊(corss-process communication)，本質上是將系統偵測到的事件，轉成訊息發出，以讓註冊接收這種事件的程式能收到。

另有一種行程內的廣播，可透過 LocalBroadcastManager 來進行。

廣播有一般廣播(無法確認哪一個接收器先收到)與有序廣播(接收器依序接到廣播)。

Registers for Intents to Observe



實作重點

製作廣播接收器 - 製作繼承 BroadcastReceiver 的類別

在架構檔中加入 <receiver> 標籤, 在其中的 <intent-filter> 中指定要接收的廣播類型。

接收器只需實作 onReceive() 即可接收所指定的廣播



Configure Component

Android Studio

Creates a new broadcast receiver component and adds it to your Android manifest.

Class Name:

☒ Exported

☒ Enabled

是否能接收app外的廣播

是否能產生實例

Cancel

Previous

Next

Finish

靜態式註冊接收

app 安裝後就自動註冊接收該廣播。

即使 app 沒有執行過，依然可接收所註冊的廣播。

靜態式註冊接收

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:supportsRtl="true"
    android:theme="@style/SplashScreen">
    <receiver
        android:name=".BCReceiver"> <!-- name of the receiver class-->
        <intent-filter>
            <!-- name of the action to receive -->
            <action android:name="android.intent.action.ACTION_POWER_CONNECTED" />
        </intent-filter>
    </receiver>

    <activity android:name=".MainActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>
```

動態式註冊接收

```
public class MainActivity extends AppCompatActivity {  
  
    private BCReceiver bcr;  
    private IntentFilter inf;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        setTheme(R.style.AppTheme);  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        bcr = new BCReceiver();  
        inf = new IntentFilter();  
        inf.addAction(Intent.ACTION_AIRPLANE_MODE_CHANGED);  
        registerReceiver(bcr, inf);  
    }  
  
    @Override  
    protected void onDestroy(){  
        super.onDestroy();  
        unregisterReceiver(bcr);  
    }  
}
```


實作 onReceive 方法

```
public class BCReceiver extends BroadcastReceiver {  
    public BCReceiver() {  
    }  
  
    @Override  
    public void onReceive(Context context, Intent intent) {  
        switch(intent.getAction()){  
            case Intent.ACTION_POWER_CONNECTED:  
                Log.i("Omomys", "External power is connected.");  
                Toast.makeText(context, "External power is connected.", Toast.LENGTH_SHORT).show();  
                break;  
            case Intent.ACTION_AIRPLANE_MODE_CHANGED:  
                Log.i("Omomys", "Airplane mode is changed");  
                Toast.makeText(context, "Airplane mode is changed.", Toast.LENGTH_SHORT).show();  
                break;  
        }  
    }  
}
```

自定廣播

```
public void onSendBroadcast(View view){  
    Intent intent = new Intent();  
    intent.setAction("BobAction");  
    sendBroadcast(intent);  
}
```

```
<receiver  
    android:name=".BCReceiver"> <!-- name of the receiver class-->  
    <intent-filter>  
        <!-- name of the action to receive -->  
        <action android:name="BobAction" />  
        <action android:name="android.intent.action.ACTION_POWER_CONNECTED" />  
    </intent-filter>  
</receiver>
```

sendOrderedBroadcast

依照優先序叫用不用 receiver 的 onReceive

優先序可指定為 -1000 ~ 1000 的範圍, 優先序愈大者先叫用。

優先序以 `<intent-filter android:priority="500">` 作靜態設定

優先序以 `intentFilter.setPriority("")` 作動態設定

以 `sendOrderedBroadcast(Intent, null)` 發送

sendStickyBroadcast

一般廣播在執行 `sendBroadcast()` 前需完成註冊

Sticky Broadcast - 滯留廣播, 廣播發送後不會消失, 讓後來註冊的接收者可以再
接到最近的一則廣播。

在 `sendBroadcast()` 前或後註冊都能運作

需開啟 `android.permission.BROADCAST_STICKY` 權限

以 `sendStickyBroadcast(Intent)` 發送

已在 API Level 21 中停用

Demo

