# 5. 안드로이드 브로트캐트리시버(Android BroadcatReceiver)

**5.1 개요**

안드로이드 Application을 구성하는 4가지 컴포넌트 중에 하나인 BroadcatRecevier는 방송수신자 로 시스템, 앱등에서 이벤트가 발생할 때 수신할 수 있도록 방송을 해주는 컴폰넌트로 이벤트에 대한 처리를 리시버를 통해 할 수 있는 것이다. 다른게 표현하면 핸드폰에서 발생하는 특정 intent 신호를 받아주는 클래스로 보면 된다.

Android System에서 발생하는 이벤트를 항상 대기하고 있다가 이벤트 발생 시 이를 전달하는 역 할을 하는것으로 스마트 폰의 전원을 켰다 껐다 할 경우에도 Screen On / Off의 이벤트가 발생하 게 되는데 이를 BroadcastReceiver가 받아서 원하는 액션을 수행할 수 있도록 프로그래밍 할 수 있다.

Broadcast를 보내는 Sender와 이를 수신하는 Receiver로 구성되고 Sender와 Receiver는 하나의 Application에서 구현될 수도 있지만 서로 다른 Application에서 구현할 수도 있다. 또한 Sender와 Receiver의 관계는 1대1, 1대N의 관계가 될 수 있다. Broadcast 전달은 Binder를 통한 IPC로 이루 어지고 내용은 Intent를 통해 전달한다.

# - 브로드캐스트 전송방법

Broadcast는 Context class의 sendBroadcast()와 sendOrderedBroadcast() API를 통해서 전송할 수 있고 startBroadcast라는 Context에 포함된 함수를 통해 인텐트를 전달하는데 안텐트에 putExtra 를 통해 데이터를 전달할 수 있다.

Intent intent = new Intent("a.b.BroadCastService");

intent.putExtra("name", “JCLEE”); intent.putExtra("age", 22);

sendBroadcast(intent);

**sendBroadcast()** : 일반적인 전송방법으로 위 예제 코드에서는 Action name으로 "

kr.topcredu.www.BroadCastService"를 사용했는데 Broadcast를 수신하기 위해 Receiver에서는 Action name에 해당하는 Intent filter를 추가하면 된다.

같은 API name으로 Overload되어 있는 sendBroadcast(intent, permission)도 있는데, 두 번째 인자 는 해당 Broadcast를 수신하기 위해 필요한 Permission을 지정하게 된다. 만약 Receiver가 구현된 Package에 해당 Permission이 선언되어 있지 않다면 Broadcast가 전달되지 않는다.

**sendOrderedBroadcast()** : 일반적인 Broadcast는 Async로 동작하여 다수의 Receiver에 대해 수신

순서를 신경 쓰지 않는데 Ordered Broadcast도 Async이긴 하지만 Intent filer의 android:priority 속성이 지정하는 중요도에 따라 수신 순서가 정해지고 한 번에 하나씩 전달되는데 Broadcast를 수신한 Receiver의 onReceive()가 return 되어야 다음 Receiver에게 Broadcast가 전달된다.

Ordered Broadcast를 수신한 Receiver에서는 Abort flag와 Result code, data, extras를 Set/Get 할

수 있다.

# - 브로드캐스트 수신방법

BroadcastReceiver를 상속받아서 Receiver를 구현하고 등록하면 Broadcast intent를 수신할 수 있 다.

public class **HelloReceiver** extends BroadcastReceiver {

public void onReceive(Context context, Intent intent) { String action = intent.getAction();

if (action.equals("a.b.BroadCastService")) { int age = intent.getIntExtra("age", -1);

String name = intent.getStringExtra("name");

}

}

}

manifest file에 아래와 같이 등록한다.

<receiver android:name="a.b.**HelloReceiver**">

<intent-filter>

<action android:name=" a.b.BroadCastService" />

</intent-filter>

</receiver>

Context class의 registerReceiver()와 unregisterReceiver() API로 동적으로 Receiver를 등록, 해제할

수도 있는데 하나의 Receiver에는 다수의 Intent filter를 등록할 수 있고 Android platform에서 전 송하는 ACTION\_LOCALE\_CHANGED나 ACTION\_TIME\_CHANGED와 같은 Broadcast도 수신할 수 있 다.

# - 제약사항

전달하는 Data에는 크기 제약이 있는데 Broadcast는 같은 Process에 속하는 Broadcast Queue에

전달되기 때문에 Binder transaction이 일어나고 Broadcast queue에서 다시 Receiver Process로

Broadcast를 전달하는 시점에도 Binder transaction이 일어난다. 1대1로 Broadcast가 전달되는 경 우에 총 2번의 Binder transaction이 일어난다.

(Sender process) -> (Android process) -> (Receiver process)

인자와 return 값을 저장하기 위한 용도로 Binder transaction buffer가 사용되는데, 각 Process마다 기본적으로 1MB가 할당된다.

Intent는 Extra를 포함하면서 다른 정보도 담고 있기 때문에 우리가 Extra에 담을 수 있는 크기는 1MB에 못 미치게 된다. Binder transaction buffer는 한 Process 내에서 모든 Binder transaction이 공유하는 것이기 때문에 각각의 크기가 작더라도 동시에 사용되는 경우에는 Buffer를 초과할 수 있다.

이 때 아래와 같은 Log 메시지를 볼 수 있다. 여기서 12234는 Sender의 Process ID이다. E/JavaBinder(12234): FAILED BINDER TRANSACTION

Android의 Broadcast queue가 속한 Process는 우리가 전달하는 Broadcast 뿐만 아니라 다른 서비 스를 위해 Binder transaction이 빈번하게 일어나고 있기 때문에 Broadcast queue에 전달하는 것 이 성공했다고 해도 아래와 같이 Broadcast queue에서 Receiver로 전달하는 것이 실패할 수도 있 다.

결론적으로 Broadcast로 전달하는 Data는 작게 유지해야 한다. 큰 데이터는 파일로 저장해서 그 경로만 공유하는 것도 한 방법이다.

- 브로드캐스트에서 받을수 있는 액션 예

ACTION\_BOOT\_COMPLETED : 부팅이 끝났을 때 (RECEIVE\_BOOT\_COMPLETED 권한등록 필요)

ACTION\_CAMERA\_BUTTON : 카메라 버튼이 눌렸을 때

ACTION\_DATE\_CHANGED, ACTION\_TIME\_CHANGED : 휴대폰의 날짜, 시간이 수동으로 변했을때 (설정에서 수정했을때)

ACTION\_SCREEN\_OFF, ACTION\_SCREEN\_ON : 화면 on, off ACTION\_AIRPLANE\_MODE\_CHANGED : 비행기 모드

ACTION\_BATTERY\_CHANGED, ACTION\_BATTERY\_LOW, ACTION\_BATTERY\_OKAY : 배터리 상태변화

ACTION\_PACKAGE\_ADDED,

ACTION\_PACKAGE\_CHANGED,ACTION\_PACKAGE\_DATA\_CLEARED,ACTION\_PACKAGE\_INSTALL,ACTIO N\_PACKAGE\_REMOVED,ACTION\_PACKAGE\_REPLACED,ACTION\_PACKAGE\_RESTARTED : 어플 설치/제

거와 관련된 것

ACTION\_POWER\_CONNECTED, ACTION\_POWER\_DISCONNECTED : 충전 관련 ACTION\_REBOOT, ACTION\_SHUTDOWN : 재부팅/종료

ACTION\_TIME\_TICK : 매분마다 수신

android.provider.Telephony.SMS\_RECEIVED : sms 수신 (RECEIVE\_SMS 권한 필요)

[예제]

문자메시지를 BroadcastReceiver를 통해 받는 예제를 작성해 보자.

1. SMSApplication 이라는 이름의 프로젝트 생성, EmptyActivity, SmsActivity 라는 액티비티 명으로 생성
2. activity\_sms.xml

*<?***xml version="1.0" encoding="utf-8"***?>*

<**android.support.constraint.ConstraintLayout xmlns:android=**[**"http:/**](http://schemas.android.com/apk/res/android)**/**[**schemas.android.com/apk/res/android**](http://schemas.android.com/apk/res/android)**"**

**xmlns:app=**[**"http://schemas.android.com/apk/res**](http://schemas.android.com/apk/res-auto)**-**[**auto"**](http://schemas.android.com/apk/res-auto) **xmlns:tools=**[**"http://schemas.android.com/tools**](http://schemas.android.com/tools)**" android:layout\_width="match\_parent"**

**android:layout\_height="match\_parent" tools:context="kr.co.topcredu.smsapplication.SmsActivity"**>

<**LinearLayout xmlns:android=**[**"http://schemas.android.com/apk/res/androi**](http://schemas.android.com/apk/res/android)**d" android:orientation="vertical" android:layout\_width="fill\_parent" android:layout\_height="fill\_parent" android:id="@+id/MainLayout"**

>

<**TextView**

**android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:textAppearance="?android:attr/textAppearanceMedium" android:text="SMS Inbox"**

**android:id="@+id/textView" android:layout\_gravity="center\_horizontal"** />

<**ListView android:id="@+id/SMSList"**

**android:layout\_height="wrap\_content" android:layout\_width="match\_parent" android:layout\_margin="5dp"** />

</**LinearLayout**>

</**android.support.constraint.ConstraintLayout**>

1. SmsBroadcastReceiver.java

**package** kr.co.topcredu.smsapplication;

**import** android.content.BroadcastReceiver; **import** android.content.Context;

**import** android.content.Intent; **import** android.os.Bundle;

**import** android.telephony.SmsMessage; **import** android.widget.Toast;

**public class** SmsBroadcastReceiver **extends** BroadcastReceiver { **public static final** String ***SMS\_BUNDLE*** = **"pdus"**;

**public void** onReceive(Context context, Intent intent) { Bundle intentExtras = intent.getExtras();

**if** (intentExtras != **null**) {

Object[] sms = (Object[]) intentExtras.get(***SMS\_BUNDLE***); String smsMessageStr = **""**;

**for** (**int** i = 0; i < sms.**length**; ++i) {

SmsMessage smsMessage = SmsMessage.*createFromPdu*((**byte**[]) sms[i]);

String smsBody = smsMessage.getMessageBody().toString(); String address = smsMessage.getOriginatingAddress();

smsMessageStr += **"SMS From: "** + address + **"\n"**; smsMessageStr += smsBody + **"\n"**;

}

Toast.*makeText*(context, smsMessageStr, Toast.***LENGTH\_SHORT***).show();

*//this will update the UI with message* SmsActivity inst = SmsActivity.*instance*(); inst.updateList(smsMessageStr);

}

}

}

1. SmsActivity.java

**package** kr.co.topcredu.smsapplication;

**import** android.Manifest; **import** android.app.Activity;

**import** android.content.ContentResolver;

**import** android.content.pm.PackageManager; **import** android.database.Cursor;

**import** android.net.Uri; **import** android.os.Bundle;

**import** android.support.v4.app.ActivityCompat;

**import** android.support.v4.content.ContextCompat; **import** android.view.View;

**import** android.widget.AdapterView;

**import** android.widget.AdapterView.OnItemClickListener; **import** android.widget.ArrayAdapter;

**import** android.widget.ListView; **import** android.widget.Toast;

**import** java.util.ArrayList;

**public class** SmsActivity **extends** Activity **implements** OnItemClickListener {

**private static** SmsActivity *inst*;

ArrayList<String> **smsMessagesList** = **new** ArrayList<String>(); ListView **smsListView**;

ArrayAdapter **arrayAdapter**;

**public static** SmsActivity instance() { **return** *inst*;

}

@Override

**public void** onStart() { **super**.onStart(); *inst* = **this**;

}

**private void** requestSmsPermission() {

String permission = Manifest.permission.***RECEIVE\_SMS***;

**int** grant = ContextCompat.*checkSelfPermission*(**this**, permission); **if** ( grant != PackageManager.***PERMISSION\_GRANTED***) {

String[] permission\_list = **new** String[1];

permission\_list[0] = permission; ActivityCompat.*requestPermissions*(**this**, permission\_list, 1);

}

}

@Override

**protected void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState); setContentView(R.layout.***activity\_sms***);

**smsListView** = (ListView) findViewById(R.id.***SMSList***);

**arrayAdapter** = **new** ArrayAdapter<String>(**this**, android.R.layout.***simple\_list\_item\_1***, **smsMessagesList**);

**smsListView**.setAdapter(**arrayAdapter**); **smsListView**.setOnItemClickListener(**this**);

refreshSmsInbox();

}

**public void** refreshSmsInbox() { requestSmsPermission();

ContentResolver contentResolver = getContentResolver();

Cursor smsInboxCursor = contentResolver.query(Uri.*parse*(**"content://sms/inbox"**), **null**, **null**, **null**, **null**);

**int** indexBody = smsInboxCursor.getColumnIndex(**"body"**);

**int** indexAddress = smsInboxCursor.getColumnIndex(**"address"**); **if** (indexBody < 0 || !smsInboxCursor.moveToFirst()) **return**; **arrayAdapter**.clear();

**do** {

String str = **"SMS From: "** + smsInboxCursor.getString(indexAddress) + **"\n"** + smsInboxCursor.getString(indexBody) + **"\n"**;

**arrayAdapter**.add(str);

} **while** (smsInboxCursor.moveToNext());

}

**public void** updateList(**final** String smsMessage) { **arrayAdapter**.insert(smsMessage, 0); **arrayAdapter**.notifyDataSetChanged();

}

**public void** onItemClick(AdapterView<?> parent, View view, **int** pos, **long** id) { **try** {

String[] smsMessages = **smsMessagesList**.get(pos).split(**"\n"**);

String address = smsMessages[0]; String smsMessage = **""**;

**for** (**int** i = 1; i < smsMessages.**length**; ++i) { smsMessage += smsMessages[i];

}

String smsMessageStr = address + **"\n"**; smsMessageStr += smsMessage;

Toast.*makeText*(**this**, smsMessageStr, Toast.***LENGTH\_SHORT***).show();

} **catch** (Exception e) { e.printStackTrace();

}

}

}

1. AndroidManifest.xml

*<?***xml version="1.0" encoding="utf-8"***?>*

<**manifest xmlns:android=**[**"http://schemas.android.com/apk/res/android"**](http://schemas.android.com/apk/res/android) **package="kr.co.topcredu.smsapplication"**>

<**uses-permission android:name="android.permission.WRITE\_SMS"** />

<**uses-permission android:name="android.permission.READ\_SMS"** />

<**uses-permission android:name="android.permission.RECEIVE\_SMS"** />

<**uses-permission android:name="android.permission.PROCESS\_OUTGOING\_CALLS"** />

<**uses-permission android:name="android.permission.INTERNET"** />

<**application**

**android:allowBackup="true" android:icon="@mipmap/ic\_launcher" android:label="@string/app\_name" android:roundIcon="@mipmap/ic\_launcher\_round" android:supportsRtl="true" android:theme="@style/AppTheme"**

**android:enabled="true" android:exported="true"**>

<**activity**

**android:name=".SmsActivity" android:label="@string/app\_name"** >

<**intent-filter**>

<**action android:name="android.intent.action.MAIN"** />

<**category android:name="android.intent.category.LAUNCHER"** />

</**intent-filter**>

</**activity**>

<**receiver android:name=".SmsBroadcastReceiver" android:exported="true"**>

<**intent-filter android:priority="999"** >

<**action android:name="android.provider.Telephony.SMS\_RECEIVED"** />

</**intent-filter**>

</**receiver**>

</**application**>

</**manifest**>

