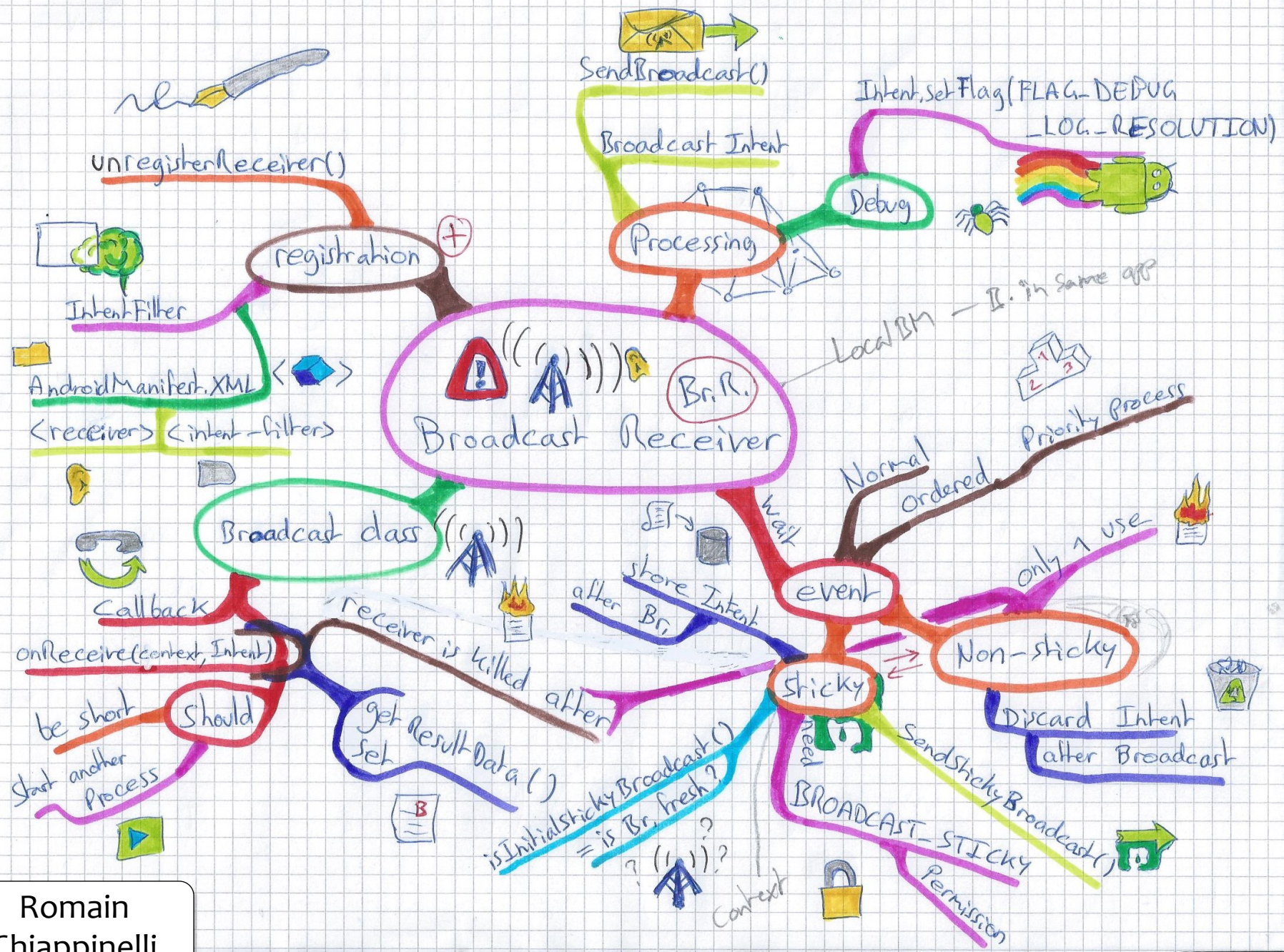


Richiapp

Android 5

18.02.14



PROGRAMMING HANDHELD SYSTEMS

ADAM PORTER

THE
BROADCASTRECEIVER
CLASS

TODAY'S TOPICS

THE BROADCAST CLASS

REGISTRATION



BROADCAST



PROCESSING

BROADCASTRECEIVER

BASE CLASS FOR COMPONENTS THAT
RECEIVE AND REACT TO EVENTS

BROADCASTRECEIVER

BROADCASTRECEIVERS REGISTER TO
RECEIVE EVENTS IN WHICH THEY ARE
INTERESTED

BROADCAST RECEIVER

WHEN EVENTS OCCUR THEY ARE
REPRESENTED AS INTENTS

THOSE INTENTS ARE THEN BROADCAST
TO THE SYSTEM

BROADCAST RECEIVER

ANDROID ROUTES THE INTENTS TO
BROADCASTRECEIVERS THAT HAVE
REGISTERED TO RECEIVE THEM

BROADCASTRECEIVERS RECEIVE THE
INTENT VIA A CALL TO `onReceive()`

TYPICAL USE CASE

REGISTER BROADCASTRECEIVERS

BROADCAST AN INTENT

ANDROID DELIVERS INTENT TO
REGISTERED RECIPIENTS BY CALLING
THEIR `onReceive()` METHOD

EVENT HANDLED IN `onReceive()`

REGISTERING FOR INTENTS

BROADCASTRECEIVERS CAN REGISTER IN
TWO WAYS

STATICALLY, IN ANDROIDMANIFEST.XML

DYNAMICALLY, BY CALLING A
registerReceiver() METHOD

STATIC REGISTRATION

PUT <receiver> AND <intent-filter>
TAGS IN ANDROIDMANIFEST.XML

<RECEIVER> FORMAT

<receiver

android:enabled=["true" | "false"]

android:exported=["true" | "false"]

android:icon="*drawable resource*"

android:label="*string resource*"

android:name="*string*"

android:permission="*string*"

android:process="*string*" >

...

</receiver>

INTENT FILTER

SPECIFY <intent-filter> TAG WITHIN THE
<receiver>

STATIC REGISTRATION

RECEIVERS ARE REGISTERED WITH THE
SYSTEM AT BOOT TIME OR WHEN THEIR
APPLICATION PACKAGE IS ADDED AT
RUNTIME



BroadcastReceiver.StaticRegistration

```
package course.examples.BroadcastReceiver.singleBroadcastStaticRegistration;

import android.app.Activity;

public class SimpleBroadcast extends Activity {

    private static final String CUSTOM_INTENT = "course.examples.BroadcastReceiver.show_toast";

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        Button button = (Button) findViewById(R.id.button);
        button.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {

                sendBroadcast(new Intent(CUSTOM_INTENT),
                    android.Manifest.permission.VIBRATE);
            }
        });
    }
}
```

BroadcastReceiverSingleBroadcastStaticRegistration

```
package course.examples.BroadcastReceiver.singleBroadcastStaticRegistration;

import android.content.BroadcastReceiver;

public class Receiver extends BroadcastReceiver {

    private final String TAG = "Receiver";

    @Override
    public void onReceive(Context context, Intent intent) {

        Log.i(TAG, "INTENT RECEIVED");

        Vibrator v = (Vibrator) context
            .getSystemService(Context.VIBRATOR_SERVICE);
        v.vibrate(500);

        Toast.makeText(context, "INTENT RECEIVED by Receiver", Toast.LENGTH_LONG).show();

    }
}
```

DYNAMIC REGISTRATION

CREATE AN INTENTFILTER

CREATE A BROADCASTRECEIVER

REGISTER BROADCASTRECEIVER USING
registerReceiver()


LOCALBROADCASTMANAGER


CONTEXT

CALL unregisterReceiver() TO
UNREGISTER BROADCASTRECEIVER

BroadcastReceiver

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    mBroadcastMgr = LocalBroadcastManager
        .getInstance(getApplicationContext());
    mBroadcastMgr.registerReceiver(receiver, intentFilter);

    setContentView(R.layout.main);

    Button button = (Button) findViewById(R.id.button);
    button.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
            mBroadcastMgr.sendBroadcast(new Intent(CUSTOM_INTENT));
        }
    });
}
```

EVENT BROADCAST

SEVERAL BROADCAST METHODS
SUPPORTED

NORMAL VS. ORDERED

NORMAL: PROCESSING ORDER UNDEFINED

ORDERED: SEQUENTIAL PROCESSING IN
PRIORITY ORDER

EVENT BROADCAST

STICKY VS. NON-STICKY

STICKY: STORE INTENT AFTER INITIAL
BROADCAST

NON-STICKY: DISCARD INTENT AFTER
INITIAL BROADCAST

WITH OR WITHOUT RECEIVER
PERMISSIONS

SOME DEBUGGING TIPS

LOG EXTRA INTENT RESOLUTION INFORMATION

```
Intent.setFlag(FLAG_DEBUG_LOG_RESOLUTION)
```

LIST REGISTERED BROADCASTRECEIVERS

DYNAMICALLY REGISTERED

```
% adb shell dumpsys activity b
```

STATICALLY REGISTERED

```
% adb shell dumpsys package
```


EVENT DELIVERY

INTENTS DELIVERED BY CALLING
onReceive(), PASSING IN:

THE CONTEXT IN WHICH THE RECEIVER IS
RUNNING

THE INTENT THAT WAS BROADCAST

EVENT HANDLING IN onReceive()

HOSTING PROCESS HAS HIGH PRIORITY
WHILE onReceive() IS EXECUTING

EVENT HANDLING IN ONRECEIVE()

onReceive() RUNS ON THE MAIN
THREAD, SO IT SHOULD BE SHORT-LIVED

IF EVENT HANDLING IS LENGTHY,
CONSIDER STARTING A SERVICE, RATHER
THAN PERFORMING COMPLETE
OPERATION IN onReceive()

EVENT HANDLING IN ONRECEIVE()



RECEIVER IS NOT CONSIDERED VALID
ONCE `onReceive()` RETURNS

NORMALLY BROADCASTRECEIVERS CAN'T
START ASYNCHRONOUS OPERATIONS

E.G., SHOWING A DIALOG, STARTING AN
ACTIVITY VIA `startActivityForResult()`



ORDERED BROADCASTS

// send Intent to BroadcastReceivers in priority order

```
void sendOrderedBroadcast (Intent intent,  
                           String receiverPermission)
```

// send Intent to BroadcastReceivers in priority order

// includes multiple parameters for greater control

```
void sendOrderedBroadcast (Intent intent,  
                           String receiverPermission,  
                           BroadcastReceiver resultReceiver,  
                           Handler scheduler,  
                           int initialCode,  
                           String initialData,  
                           Bundle initialExtras)
```

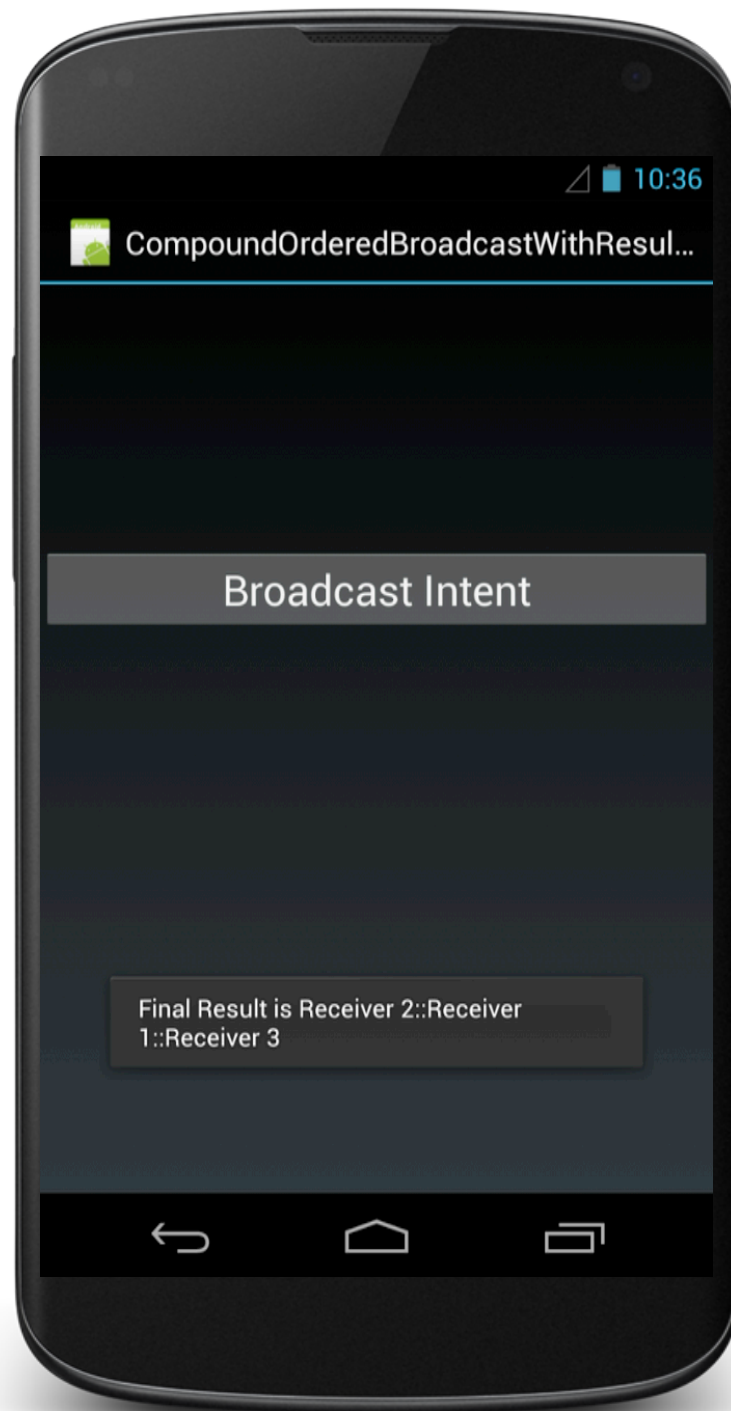


BroadcastReceiver

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    IntentFilter intentFilter = new IntentFilter(CUSTOM_INTENT);
    intentFilter.setPriority(3);
    registerReceiver(mReceiver, intentFilter);

    Button button = (Button) findViewById(R.id.button);
    button.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
            sendOrderedBroadcast(new Intent(CUSTOM_INTENT),
                android.Manifest.permission.VIBRATE);
        }
    });
}
```



BCASTRECCOMPORDBCASTWITHRESREC

```
private final Receiver1 mReceiver1 = new Receiver1();

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    setContentView(R.layout.main);

    IntentFilter intentFilter = new IntentFilter(CUSTOM_INTENT);
    intentFilter.setPriority(3);
    registerReceiver(mReceiver1, intentFilter);

    Button button = (Button) findViewById(R.id.button);
    button.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
            sendOrderedBroadcast(new Intent(CUSTOM_INTENT), null,
                new BroadcastReceiver() {
                    @Override
                    public void onReceive(Context context, Intent intent) {
                        Toast.makeText(context,
                            "Final Result is " + getResultData(),
                            Toast.LENGTH_LONG).show();
                    }
                }, null, 0, null, null);
        }
    });
}
```

STICKY BROADCASTS



STICKY INTENTS ARE CACHED BY
ANDROID

NEW INTENTS OVERWRITE OLDER INTENTS
THEY MATCH

STICKY BROADCASTS

WHEN BROADCASTRECEIVERS ARE
DYNAMICALLY REGISTERED

CACHED STICKY INTENTS MATCHING THE
SPECIFIED INTENTFILTER ARE BROADCAST TO
THE BROADCASTRECEIVER

ONE MATCHING STICKY INTENT IS RETURNED
TO THE CALLER

STICKY BROADCASTS

```
//public abstract class Context ...
```

```
// send sticky Intent to interested BroadcastReceivers
```

```
void sendStickyBroadcast (Intent intent)
```

```
// send sticky Intent to interested BroadcastReceivers in priority order
```

```
// sender can provide various parameters for greater control
```

```
void sendStickyOrderedBroadcast (Intent intent,  
                                BroadcastReceiver resultReceiver,  
                                Handler scheduler,  
                                int initialCode,  
                                String initialData,  
                                Bundle initialExtras)
```

BROADCASTER MUST HAVE BROADCAST_STICKY
PERMISSION TO SEND STICKY INTENTS

BCASTRECSTICKYINT

```
registerReceiver(new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {

        if (intent.getAction().equals(Intent.ACTION_BATTERY_CHANGED)) {

            String age = "Reading taken recently";
            if (isInitialStickyBroadcast()) {
                age = "Reading may be stale";
            }

            currentStateView.setText("Current Battery Level:"
                + String.valueOf(intent.getIntExtra(
                    BatteryManager.EXTRA_LEVEL, -1))
                + System.getProperty("line.separator") + age);
        }
    }, new IntentFilter(Intent.ACTION_BATTERY_CHANGED));
```


NEXT TIME

THREADS, ASYNCTASKS & HANDLERS