Mobile Application Develpment

Broadcast Receiver

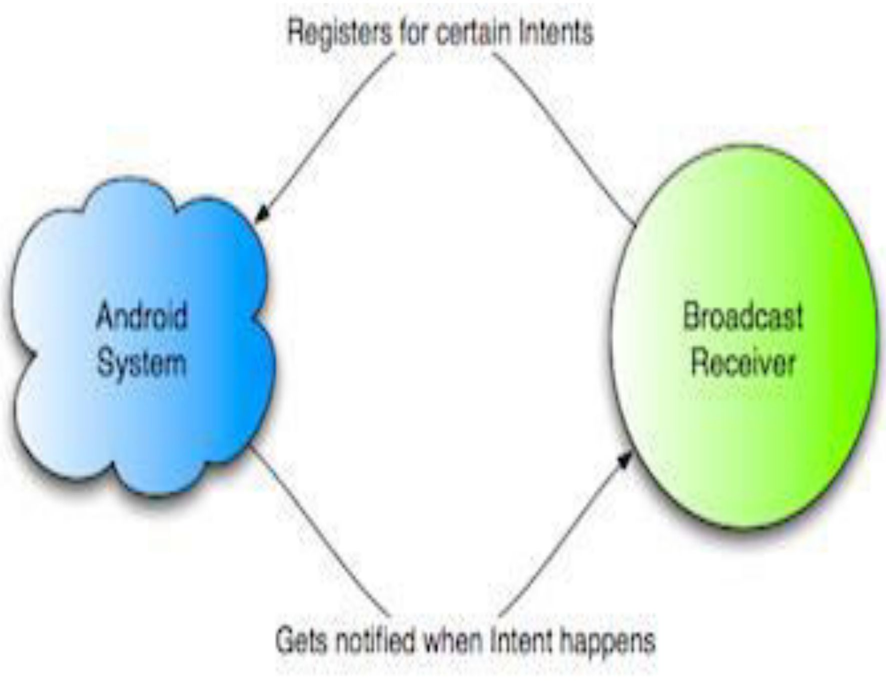
What is Broadcast Receivers

- ***A broadcast receiver (receiver)*** is an Android component which allows you to register for **system** or **application** **events** (system-wide publish/subscribe mechanism).

- All **registered receivers** for an event are notified by the Android runtime once this **event** happens.

- The **receiver** is simply a inactive code that gets activated once an event it is subscribed to happens.

- The **Android system** itself **broadcasts** events **all the time**.

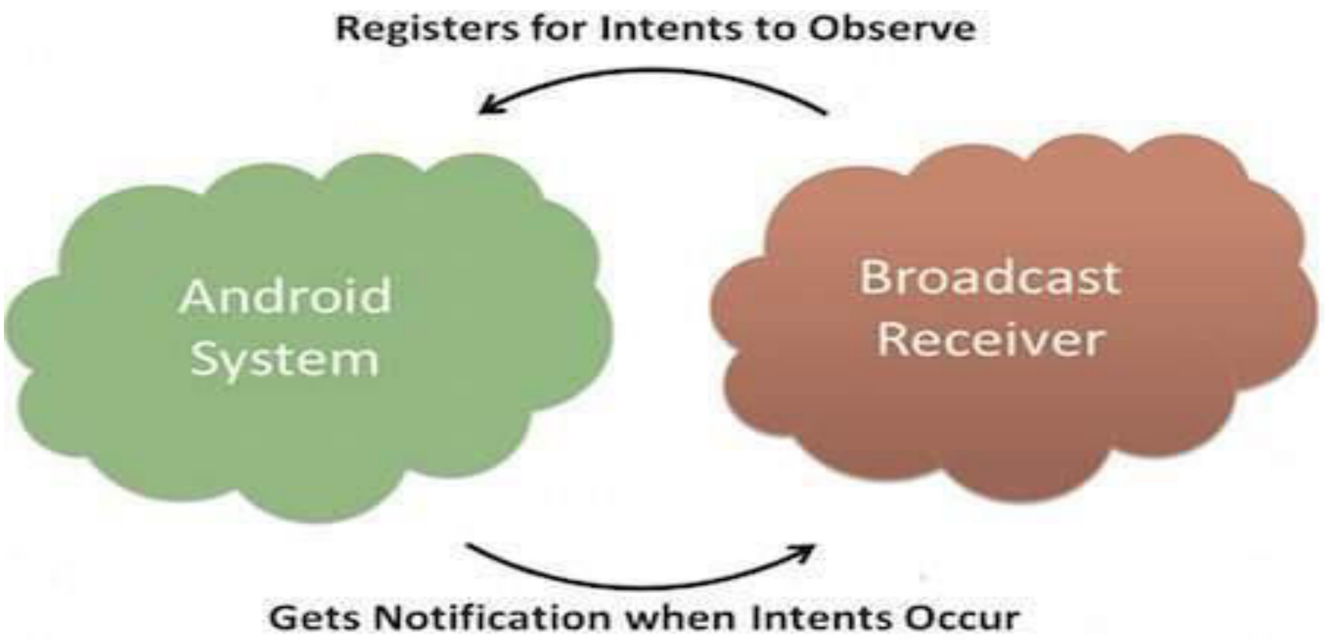

Registers for certain Intents
Android System
Broadcast Receiver
Gets notified when Intent happens

Broadcast Receiver Implementation

- To set up a Broadcast Receiver in android application we need to do the following **two** things.

  1. **Creating** a **BroadcastReceiver**

  2. **Registering** a **BroadcastReceiver**


Registers for Intents to Observe
Android System
Broadcast Receiver
Gets Notification when Intents Occur

Creating a BroadcastReceiver

```
public class MyReceiver extends BroadcastReceiver {
 @Override
public void onReceive(Context context, Intent intent) {
     Toast.makeText(context, "Intent
     Detected.",Toast.LENGTH_LONG).show(); } }
```

- **BroadcastReceiver** is an abstract class with the **onReceiver()** method being abstract.
- **The onReceiver()** method is first called on the registered Broadcast Receivers when **any event occurs**.
- **The intent object** is passed with all the additional data.
- **A Context object** is used to start an **activity or service** using **context.startActivity(myIntent);** or **context.startService(myService**);

Registering a BroadcastReceiver

- **A BroadcastReceiver can be registered in two ways.**

1. By defining it in the *AndroidManifest.xml* file.

```
<receiver android:name=".ConnectionReceiver" >
    <intent-filter>
      <action android:name="android.net.conn.CONNECTIVITY_CHANGE" />
     </intent-filter>
</receiver>
```

2. By defining it programmatically.

```
IntentFilter filter = new IntentFilter();

intentFilter.addAction(getPackageName() + "android.net.conn.CONNECTIVITY_CHANGE");

MyReceiver myReceiver = new MyReceiver();

registerReceiver(myReceiver, filter);
```

Unregistering a BroadcastReceiver

- To unregister a broadcast receiver in **onStop()** or **onPause()** of the activity the following snippet can be used.

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

System-wide events that you can register

- Following are some of the important system wide generated intents.

1. **android.intent.action.BATTERY_LOW** : Indicates **low battery condition** on the device.

2. **android.intent.action.BOOT_COMPLETED** : This is broadcast once, after the system has **finished booting.**

3. **android.intent.action.CALL** : To perform a **call to someone** specified by the data.

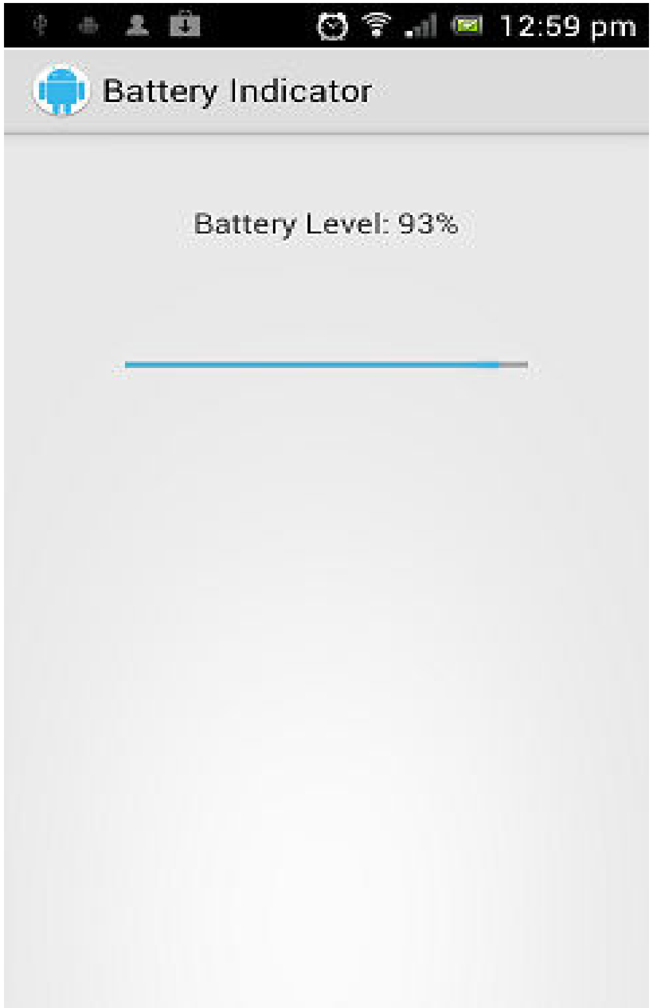
4. **android.intent.action.DATE_CHANGED** : The **date** has changed

5. **android.intent.action.REBOOT** : Have the **device reboot.**

6. **android.net.conn.CONNECTIVITY_CHANGE** : The mobile network or **WiFi** connection is changed (or reset).

Example

```
<?xml version="1.0"encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
 android:layout_width="fill_parent"
 android:layout_height="fill_parent"
 android:orientation="vertical">
<TextView
      android:id="@+id/textfield"
      android:layout_marginTop="40dip"
      android:layout_width="wrap_content"
       android:layout_height="wrap_content"
      android:layout_gravity="center"/>
<ProgressBar
      android:id="@+id/progressbar"
       android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginTop="20dip"
      android:layout_gravity="center"
       android:minWidth="200dip"
       android:minHeight="100dip"
       android:max="100"
      style="?android:attr/progressBarStyleHorizontal"/>
</LinearLayout>
```



12:59 pm

Battery Indicator

Battery Level: 93%

MainActivity.java

```
package com.example.hp1000.myapplicationservice;

Import ..;

Public class BatteryIndicator Activity extends Activity {
    //Create Broadcast Receiver Object along with class definition
   private BroadcastReceiver mBatInfoReceiver = new BroadcastReceiver() {
   @Override
    //When Event is published, onReceive method is called
    public void onReceive(Context c, Intent i) {
           int level = i.getIntExtra("level", 0);  //Get Battery %
           //Find the progressbar creating in main.xml
           ProgressBar pb = (ProgressBar) findViewById(R.id.progressbar);
           //Set progress level with battery % value
          pb.setProgress(level);
          TextView tv = (TextView) findViewById(R.id.textfield);
          tv.setText("Battery Level: "+ Integer.toString(level) + "%"); //Set TextView with text
      }
    };
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.main); //Set layout we created
            //Register the receiver which triggers event
            //when battery charge is changed
            registerReceiver(mBatInfoReceiver, new IntentFilter(
                  Intent.ACTION_BATTERY_CHANGED));
    }
}
```

Reference

- **BroadcastReceiver | Android Developers**
  https://developer.android.com/guide/components/broadcasts.html
  https://developer.android.com/reference/android/content/BroadcastReceiver.html