**Experiment 11**

**1.拦截史迪仔广播，p157**

## 2.BatteryLevel（电池电量）

### 一、案例描述

1. 考核知识点

030006001：广播接收者简介

030006002：广播接收者的创建

1. 练习目标

* 广播的动态注册和销毁

1. 需求分析

广播分为两种注册方式：静态注册和动态注册，上一个案例讲解了静态注册，该案例将对广播进行动态注册与销毁。该案例主要讲解了监听设备的电量和网络状态并展示在界面上。

1. 设计思路（实现原理）
2. 在代码中进行监听电池电量和网络状态广播的注册和注销；
3. 将获取到的状态显示在界面上。

### 二、案例实现

**（1）编写主界面**

创建“BatteryLevel”程序，首先创建MainActivity用于显示信息，该程序的界面如图6-6所示。



1. 主界面

程序主界面对应的布局文件（activity\_main）如下所示：

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

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity" >

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:orientation="vertical" >

<TextView

android:id="@+id/tv\_cell"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="电池电量"

android:textColor="#000000"

android:layout\_margin="10dp"

android:textSize="18dp" />

<TextView

android:id="@+id/tv\_netWork"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="网络状态"

android:textColor="#000000"

android:layout\_margin="10dp"

android:textSize="18dp" />

</LinearLayout>

</RelativeLayout>

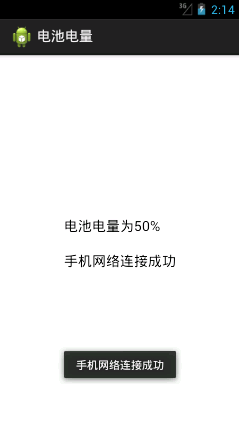
**（2）编写程序代码**

编写好主界面，接下来编写MainActivity类逻辑代码，具体代码如下所示：

1. package cn.itcast.batterylevel;
2. import android.app.Activity;
3. import android.content.BroadcastReceiver;
4. import android.content.Context;
5. import android.content.Intent;
6. import android.content.IntentFilter;
7. import android.net.ConnectivityManager;
8. import android.net.NetworkInfo.State;
9. import android.os.Bundle;
10. import android.widget.TextView;
11. import android.widget.Toast;
12. public class MainActivity extends Activity {
13. private TextView tv\_cell, tv\_netWork;
14. private BatteryReceiver batteryReceiver;
15. private ConnectivityReceiver connectivity;
16. @Override
17. protected void onCreate(Bundle savedInstanceState) {
18. super.onCreate(savedInstanceState);
19. setContentView(R.layout.activity\_main);
20. tv\_cell = (TextView) findViewById(R.id.tv\_cell);
21. tv\_netWork = (TextView) findViewById(R.id.tv\_netWork);
22. // 注册广播接受者java代码
23. IntentFilter intentFilter = new IntentFilter(
24. Intent.ACTION\_BATTERY\_CHANGED);
25. batteryReceiver = new BatteryReceiver();
26. // 注册receiver
27. registerReceiver(batteryReceiver, intentFilter);
28. IntentFilter mFilter = new IntentFilter();
29. mFilter.addAction(ConnectivityManager.CONNECTIVITY\_ACTION);
30. connectivity = new ConnectivityReceiver();
31. registerReceiver(connectivity, mFilter);
32. }
33. /\*\*
34. \* 广播接受者
35. \*/
36. private class BatteryReceiver extends BroadcastReceiver {
37. @Override
38. public void onReceive(Context context, Intent intent) {
39. // TODO Auto-generated method stub
40. // 判断它是否是为电量变化的Broadcast Action
41. if (Intent.ACTION\_BATTERY\_CHANGED.equals(intent.getAction())) {
42. // 获取当前电量
43. int level = intent.getIntExtra("level", 0);
44. // 电量的总刻度
45. int scale = intent.getIntExtra("scale", 100);
46. // 把它转成百分比
47. tv\_cell.setText("电池电量为" + ((level \* 100) / scale) + "%");
48. }
49. }
50. }
51. private class ConnectivityReceiver extends BroadcastReceiver {
52. @Override
53. public void onReceive(Context context, Intent intent) {
54. State wifiState = null;
55. State mobileState = null;
56. ConnectivityManager cm = (ConnectivityManager) context
57. .getSystemService(Context.CONNECTIVITY\_SERVICE);
58. wifiState = cm.getNetworkInfo(ConnectivityManager.TYPE\_WIFI)
59. .getState();
60. mobileState = cm.getNetworkInfo(ConnectivityManager.TYPE\_MOBILE)
61. .getState();
62. if (wifiState != null && mobileState != null
63. && State.CONNECTED != wifiState
64. && State.CONNECTED == mobileState) {
65. Toast.makeText(context, "手机网络连接成功",
66. Toast.LENGTH\_SHORT).show();
67. tv\_netWork.setText("手机网络连接成功");
68. // 手机网络连接成功
69. } else if (wifiState != null && mobileState != null
70. && State.CONNECTED != wifiState
71. && State.CONNECTED != mobileState) {
72. // 手机没有任何的网络
73. Toast.makeText(context, "手机没有网络",
74. Toast.LENGTH\_SHORT).show();
75. tv\_netWork.setText("手机没有网络");
76. } else if (wifiState != null && State.CONNECTED == wifiState) {
77. // 无线网络连接成功
78. Toast.makeText(context, "无线网络连接成功",
79. Toast.LENGTH\_SHORT).show();
80. tv\_netWork.setText("无限网络连接成功");
81. }
82. }
83. }
84. @Override
85. protected void onDestroy() {
86. super.onDestroy();
87. unregisterReceiver(batteryReceiver);
88. unregisterReceiver(connectivity);
89. }
90. }

**（3）运行程序**

运行“BatteryLevel”程序，结果如图6-7所示。



1. 运行界面

如图6-7所示，运行程序之后，当前设备的电量和网络状态都显示在了界面上。

### 三、案例总结

1、广播可以进行静态注册和动态注册。

2、广播在代码中动态注册需要有unregisterReceiver()操作。