**Exercise 7**

## CallRecord（通话记录）

### 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

* 广播的静态注册和使用
* 使用广播处理处理事件

1. 需求分析

手机最重要的功能就是通话功能，同样储存通话记录也是必不可少的。该案例使用广播接收者自己实现通话记录的功能。包括呼出电话、已接来电、未接来电以及通话产生的时间。

1. 设计思路（实现原理）
2. 注册监听电话状态的广播；
3. 在广播的onReceive()方法中使用SQLite记录通话记录；
4. 将通话记录展示在ListView上。
5. 预备知识点：

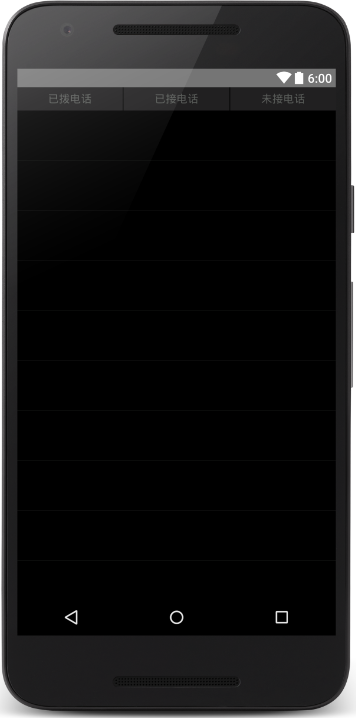
https://cloud.tencent.com/developer/article/1394213

<http://www.cnblogs.com/jerehedu/p/4877863.html>

### 二、案例实现

**（1）创建第一个界面**

创建一个名为“CallRecord”的程序并创建第一个界面MainActivity，该Activity用于展示已拨、已接和未接来电，程序主界面如图6-1所示。



1. 主界面

MainActivity界面对应的布局文件（activity\_main.xml）如下所示：

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

<RelativeLayoutxmlns: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"

android:background="#000000"

tools:context=".MainActivity" >

<LinearLayout

android:id="@+id/ll\_title"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="#212021"

android:orientation="horizontal"

android:weightSum="3" >

<RelativeLayout

android:id="@+id/rl\_dialedCalls"

android:layout\_width="wrap\_content"

android:layout\_height="match\_parent"

android:layout\_weight="1" >

<TextView

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

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:text="已拨电话"

android:textColor="#5a5d5a" />

</RelativeLayout>

<View

android:layout\_width="1dp"

android:layout\_height="30dp"

android:layout\_gravity="center\_vertical"

android:background="#000000" />

<RelativeLayout

android:id="@+id/rl\_receivedCall"

android:layout\_width="wrap\_content"

android:layout\_height="match\_parent"

android:layout\_weight="1" >

<TextView

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

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:text="已接电话"

android:textColor="#5a5d5a" />

//https://encycolorpedia.cn/search?q=000000

</RelativeLayout>

<View

android:layout\_width="1dp"

android:layout\_height="30dp"

android:layout\_gravity="center\_vertical"

android:background="#000000" />

<RelativeLayout

android:id="@+id/rl\_missedCall"

android:layout\_width="wrap\_content"

android:layout\_height="match\_parent"

android:layout\_weight="1" >

<TextView

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

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:gravity="center\_horizontal"

android:text="未接电话"

android:textColor="#5a5d5a" />

</RelativeLayout>

</LinearLayout>

<ListView

android:id="@+id/listview"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_below="@+id/ll\_title"

android:divider="#313431"

android:dividerHeight="1px" />

</RelativeLayout>

**（2）创建数据库**

在项目中创建一个包，命名为db，并创建一个数据库类，用于储存通话记录，具体代码如下：

1. package cn.itcast.callrecord.db;
2. import android.content.Context;
3. import android.database.sqlite.SQLiteDatabase;
4. import android.database.sqlite.SQLiteOpenHelper;
5. public class SQLiteHelper extends SQLiteOpenHelper {
6. public SQLiteHelper(Context context) {
7. super(context, "CallRecords.db", null, 1);
8. }
9. @Override
10. public void onCreate(SQLiteDatabasedb) {
11. db.execSQL("create table phone (id integer primary key autoincrement," +
12. "phone varchar(20)," +
13. "classify varchar(20)," +
14. "date varchar(20))");
15. }
16. @Override
17. public void onUpgrade(SQLiteDatabasedb, intoldVersion, intnewVersion) {
18. }
19. }

**（3）创建业务bean**

在项目创建一个包，命名为bean并在其中创建一个业务类Phone，其中包含phone、date、classify三种属性，具体代码如下所示：

1. package cn.itcast.callrecord.bean;
2. public class Phone {
3. private int id;
4. private String phone;
5. private String classify;
6. private String date;
7. public String getDate() {
8. return date;
9. }
10. public void setDate(String date) {
11. this.date = date;
12. }
13. public String getClassify() {
14. return classify;
15. }
16. public void setClassify(String classify) {
17. this.classify = classify;
18. }
19. public intgetId() {
20. return id;
21. }
22. public void setId(int id) {
23. this.id = id;
24. }
25. public String getPhone() {
26. return phone;
27. }
28. public void setPhone(String phone) {
29. this.phone = phone;
30. }
31. public Phone(String phone, String classify, String date) {
32. super();
33. this.phone = phone;
34. this.classify = classify;
35. this.date = date;
36. }
37. public Phone(String phone, String date) {
38. super();
39. this.phone = phone;
40. this.date = date;
41. }
42. public Phone() {
43. }
44. }

**（4）创建数据库操作类PhoneDao**

同样在db包中创建一个操作数据库的类，本案例只需要查询所有数据以及插入数据两种方法，具体代码如下：

1. package cn.itcast.callrecord.db;
2. import android.content.ContentValues;
3. import android.content.Context;
4. import android.database.Cursor;
5. import android.database.sqlite.SQLiteDatabase;
6. import java.util.ArrayList;
7. import java.util.List;
8. import cn.itcast.callrecord.bean.Phone;
9. public class PhoneDao {
10. private SQLiteHelper helper;
11. public PhoneDao(Context context) {
12. helper = new SQLiteHelper(context);
13. }
14. public void insert(Phone psw) {
15. SQLiteDatabasedb = helper.getWritableDatabase();
16. ContentValues values = new ContentValues();
17. values.put("phone", psw.getPhone());
18. values.put("classify", psw.getClassify());
19. values.put("date", psw.getDate());
20. db.insert("phone", null, values);
21. db.close();
22. }
23. public List<Phone>queryAll() {
24. SQLiteDatabasedb = helper.getReadableDatabase();
25. Cursor c = db.query("phone",
26. new String[]{"phone", "classify", "date"}, null, null, null,
27. null, null);
28. List<Phone> list = new ArrayList<Phone>();
29. while (c.moveToNext()) {
30. String phone = c.getString(c.getColumnIndex("phone"));
31. String classify = c.getString(c.getColumnIndex("classify"));
32. String date = c.getString(c.getColumnIndex("date"));
33. Phone number = new Phone(phone, classify, date);
34. list.add(number);
35. }
36. c.close();
37. db.close();
38. return list;
39. }
40. }

**（5）创建广播接收者**

接下来创建用于监听电话状态的广播接收者CallReceiver类，具体代码如下所示：

1. package cn.itcast.callrecord;
2. import java.sql.Date;
3. import java.text.SimpleDateFormat;
4. import android.annotation.SuppressLint;
5. import android.content.BroadcastReceiver;
6. import android.content.Context;
7. import android.content.Intent;
8. import android.telephony.PhoneStateListener;
9. import android.telephony.TelephonyManager;
10. import android.widget.Toast;
11. import cn.itcast.callrecord.bean.Phone;
12. import cn.itcast.callrecord.db.PhoneDao;
13. public class CallReceiver extends BroadcastReceiver {
14. private static intlastState = TelephonyManager.CALL\_STATE\_IDLE;
15. http://www.cnblogs.com/jerehedu/p/4877863.html
16. private static boolean flag = false;
17. /\*\* 区分呼入呼出标记 \*/
18. private static booleanisOuting = false;
19. /\*\* 当前操作电话号码 \*/
20. private static String currentNumber = null;
21. private String listenerNumber;
22. private PhoneDaodao;
23. @Override
24. public void onReceive(final Context context, Intent intent) {
25. dao = new PhoneDao(context);
26. if (intent.getAction().equals(Intent.ACTION\_NEW\_OUTGOING\_CALL)) {
27. // 呼出
28. isOuting = true;
29. String phonenumber = getResultData();// 被呼叫的电话号码
30. // 更新电话号码
31. if (null != phonenumber&& !"".equals(phonenumber)) {
32. currentNumber = phonenumber;
33. }
34. } else {
35. if (!flag) {
36. TelephonyManager manager = (TelephonyManager) context
37. .getSystemService(Context.TELEPHONY\_SERVICE);
38. manager.listen(new PhoneStateListener() {
39. @SuppressLint({ "NewApi", "ShowToast" })
40. @Override
41. public void onCallStateChanged(int state,
42. String incomingNumber) {
43. super.onCallStateChanged(state, incomingNumber);
44. flag = true;
45. if (state == TelephonyManager.CALL\_STATE\_RINGING) {
46. // 来电状态
47. isOuting = false;
48. listenerNumber = incomingNumber;
49. Toast.makeText(context, "来电提醒: " + listenerNumber,
50. Toast.LENGTH\_SHORT).show();
51. } else if (state == TelephonyManager.CALL\_STATE\_OFFHOOK) {
52. if (isOuting) {
53. setData(context, currentNumber,"dialed",
54. "呼出电话： ");
55. } else {
56. setData(context, listenerNumber,"received",
57. "已接来电： ");
58. }
59. }
60. if (lastState == TelephonyManager.CALL\_STATE\_RINGING
61. && state == TelephonyManager.CALL\_STATE\_IDLE) {
62. setData(context, incomingNumber,"missed","未接来电： ");
63. }
64. lastState = state;
65. }
66. private void setData(final Context context,
67. String incomingNumber,Stringclassicy,
68. String hint) {
69. Toast.makeText(context, hint + incomingNumber,
70. Toast.LENGTH\_SHORT).show();
71. SimpleDateFormat formatter = new SimpleDateFormat(
72. "yyyy年MM月dd日HH:mm:ss ");
73. Date curDate = new Date(System.currentTimeMillis());
74. // 获取当前时间
75. String date = formatter.format(curDate);
76. dao.insert(new Phone(incomingNumber, classicy, date));
77. }
78. }, PhoneStateListener.LISTEN\_CALL\_STATE);
79. }
80. }
81. }
82. }

**（6）添加颜色属性**

在res/values/colors.xml文件中添加2个颜色属性，示例代码如下所示：

<color name="white">#ffffff</color>

<color name="gray">#5a5d5a</color>

接下来在layout文件夹中创建listview\_item文件，具体代码如下所示：

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

<RelativeLayout

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<TextView

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

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

android:layout\_alignParentTop="true"

android:layout\_marginLeft="26dp"

android:layout\_marginTop="20dp"

android:text="电话号码"

android:textColor="#ffffff"

android:textSize="16dp" />

<TextView

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

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignLeft="@+id/tv\_number"

android:layout\_below="@+id/tv\_number"

android:layout\_marginBottom="10dp"

android:layout\_marginLeft="28dp"

android:layout\_marginTop="5dp"

android:text="时间"

android:textColor="#ffffff"

android:textSize="16dp" />

</RelativeLayout>

**（7）主界面逻辑**

在MainActivity的onResume()方法中进行数据库的查询以及数据的展示，界面中有三个按钮，点击按钮时展示按钮所对应的数据，具体代码如下所示：

1. package cn.itcast.callrecord;
2. import android.annotation.SuppressLint;
3. import android.app.Activity;
4. import android.os.Bundle;
5. import android.view.View;
6. import android.view.View.OnClickListener;
7. import android.view.ViewGroup;
8. import android.widget.BaseAdapter;
9. import android.widget.ListView;
10. import android.widget.RelativeLayout;
11. import android.widget.TextView;
12. import java.util.ArrayList;
13. import java.util.List;
14. import cn.itcast.callrecord.bean.Phone;
15. import cn.itcast.callrecord.db.PhoneDao;
16. @SuppressLint({"NewApi", "ResourceAsColor"})
17. public class MainActivity extends Activity implements OnClickListener {
18. private ListViewlistView;
19. privatePhoneDaodao;
20. private MyAdapter adapter;
21. private Phone phoneNumber;
22. private List<Phone> dialed, received, missed, dataList;
23. private TextViewtv\_receivedCall, tv\_missedCall, tv\_dialedCalls;
24. private RelativeLayoutrl\_receivedCall, rl\_missedCall, rl\_dialedCalls;
25. @Override
26. protected void onCreate(Bundle savedInstanceState) {
27. super.onCreate(savedInstanceState);
28. setContentView(R.layout.activity\_main);
29. rl\_receivedCall = (RelativeLayout) findViewById(R.id.rl\_receivedCall);
30. rl\_missedCall = (RelativeLayout) findViewById(R.id.rl\_missedCall);
31. rl\_dialedCalls = (RelativeLayout) findViewById(R.id.rl\_dialedCalls);
32. tv\_receivedCall = (TextView) findViewById(R.id.tv\_receivedCall);
33. tv\_missedCall = (TextView) findViewById(R.id.tv\_missedCall);
34. tv\_dialedCalls = (TextView) findViewById(R.id.tv\_dialedCalls);
35. rl\_receivedCall.setOnClickListener(this);
36. rl\_missedCall.setOnClickListener(this);
37. rl\_dialedCalls.setOnClickListener(this);
38. listView = (ListView) findViewById(R.id.listview);
39. dao = new PhoneDao(this);
40. }
41. @Override
42. protected void onResume() {
43. super.onResume();
44. dataList = dao.queryAll();
45. dialed = new ArrayList<Phone>();
46. received = new ArrayList<Phone>();
47. missed = new ArrayList<Phone>();
48. if (dataList != null) {
49. for (Phone p : dataList) {
50. String classify = p.getClassify();
51. if ("received".equals(classify)) {
52. received.add(new Phone(p.getPhone(), p.getDate()));
53. } else if ("dialed".equals(classify)) {
54. dialed.add(new Phone(p.getPhone(), p.getDate()));
55. } else if ("missed".equals(classify)) {
56. missed.add(new Phone(p.getPhone(), p.getDate()));
57. }
58. }
59. }
60. adapter = new MyAdapter(dialed);
61. listView.setAdapter(adapter);
62. setTextColor(R.color.gray, R.color.gray, R.color.white);
63. }
64. private class MyAdapter extends BaseAdapter {
65. private List<Phone> list;
66. public MyAdapter(List<Phone> list) {
67. this.list = list;
68. }
69. @Override
70. public intgetCount() {
71. return list.size();
72. }
73. @Override
74. public Object getItem(int position) {
75. return list.get(position);
76. }
77. @Override
78. public long getItemId(int position) {
79. return position;
80. }
81. @Override
82. public View getView(int position, View convertView, ViewGroup parent) {
83. phoneNumber = list.get(position);
84. View item = convertView != null ? convertView : View.inflate(
85. MainActivity.this, R.layout.listview\_item, null);
86. TextViewtv\_number = (TextView) item.findViewById(R.id.tv\_number);
87. TextViewtv\_date = (TextView) item.findViewById(R.id.tv\_date);
88. tv\_number.setText("电话号码: " + phoneNumber.getPhone());
89. tv\_date.setText(phoneNumber.getDate());
90. return item;
91. }
92. }
93. @Override
94. public void onClick(View v) {
95. switch (v.getId()) {
96. case R.id.rl\_dialedCalls:
97. adapter = new MyAdapter(dialed);
98. listView.setAdapter(adapter);
99. setTextColor(R.color.gray, R.color.gray, R.color.white);
100. break;
101. case R.id.rl\_receivedCall:
102. adapter = new MyAdapter(received);
103. listView.setAdapter(adapter);
104. setTextColor(R.color.white, R.color.gray, R.color.gray);
105. break;
106. case R.id.rl\_missedCall:
107. adapter = new MyAdapter(missed);
108. listView.setAdapter(adapter);
109. setTextColor(R.color.gray, R.color.white, R.color.gray);
110. break;
111. }
112. }
113. private void setTextColor(int one, int two, int three) {
114. tv\_receivedCall.setTextColor(getResources().getColor(one));
115. tv\_missedCall.setTextColor(getResources().getColor(two));
116. tv\_dialedCalls.setTextColor(getResources().getColor(three));
117. }
118. }

**（8）编写广播接收者**

打开清单文件，receiver标签内添加2个过滤器，示例代码如下所示：

<receiver

android:name=".CallReceiver"

android:enabled="true"

android:exported="true" >

**<intent-filter android:priority="1000" >**

**<action android:name="android.intent.action.PHONE\_STATE" />**

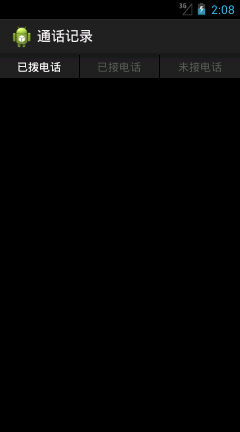
**<action android:name="android.intent.action.NEW\_OUTGOING\_CALL" />**

**</intent-filter>**

</receiver>

**（9）、运行程序**

运行“CallRecord”程序，界面如图6-2所示。



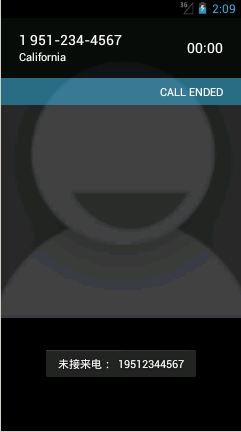
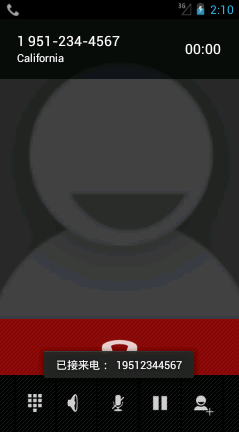
1. 主界面

当前并没有通话记录，把应用关闭或按Home建进入后台，然后呼出一个电话，如图6-3所示。



1. 呼出电话

如图6-3所示，当呼出一个电话时，屏幕上会提示呼出的电话号码。接下来使用DDMS的Emulator Control功能对模拟器进行呼叫，并在模拟器上分别进行多次接听和挂断操作，如图6-4所示。

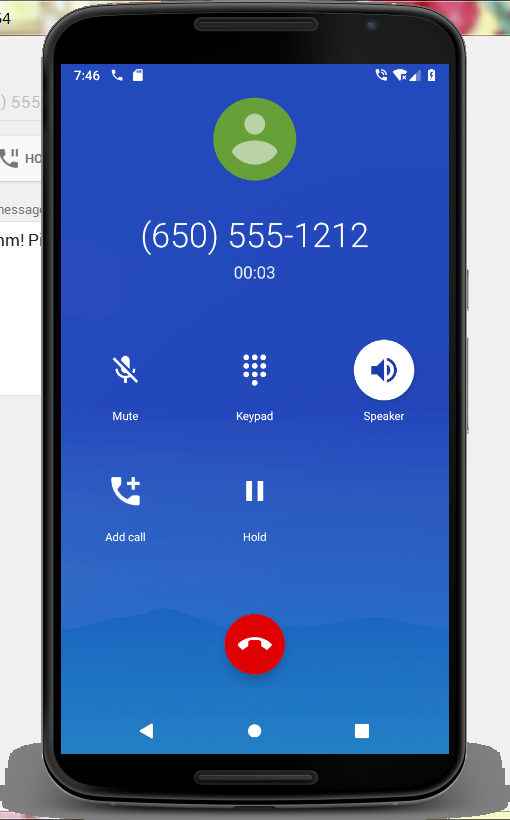
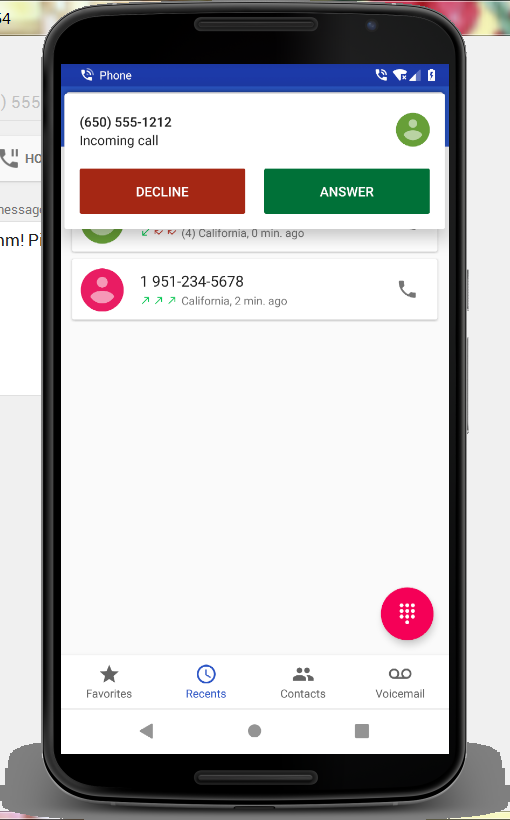
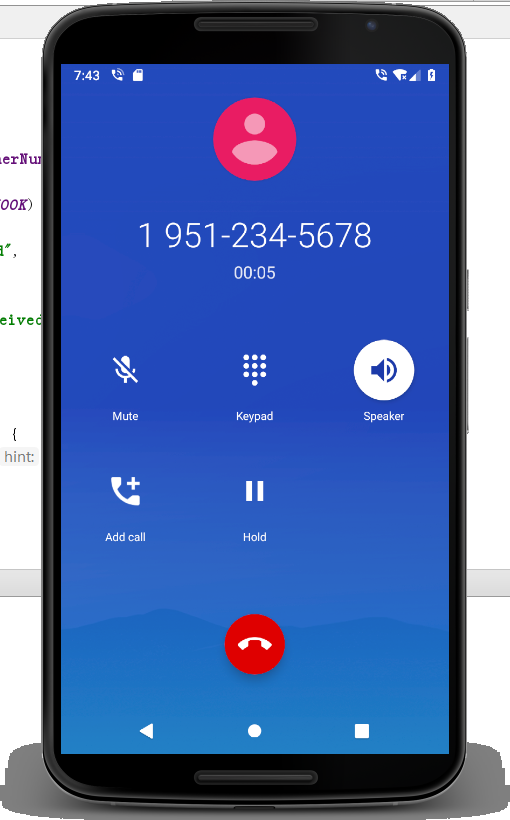
1. 未接来电和已接来电

如图6-4所示，分别进行了挂断电话和接听电话的操作，都有界面提示，接下来再次打开应用，如图6-5所示。



1. 主界面

如图6-5所示，在进行了一系列操作之后主界面中显示出来之前对电话操作的状态，主界面默认显示已拨电话，点击已接电话和未接电话按钮会看到各自对应的记录。



### 三、案例总结

1、需要注意的是，使用广播接收者必须要在清单文件中进行注册。