

中山大学移动信息工程学院本科生实验报告

(2017 年秋季学期)

课程名称：移动应用开发

任课教师：郑贵锋

年级	大三	专业 (方向)	移动互联网
学号	15352408	姓名	张镓伟
电话	13531810182	Email	709075442@qq.com
开始日期	2017.10.27	完成日期	2017.10.30

一、 实验题目

Broadcast 使用

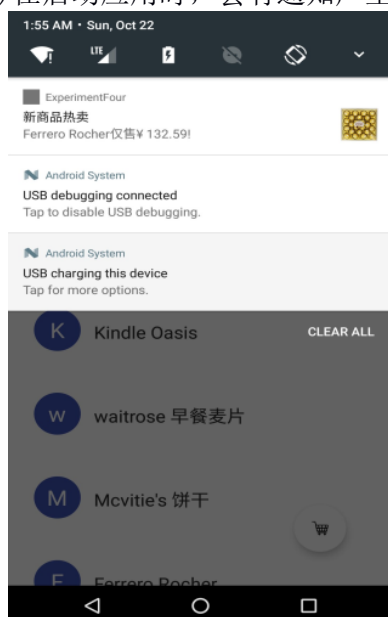
二、 实验目的

1. 掌握 Broadcast 编程基础
2. 掌握动态注册 Broadcast 和静态注册 Broadcast
3. 掌握 Notification 编程基础
4. 掌握 EventBus 编程基础

三、 实验内容

在实验三的基础上，实现静态广播、动态广播两种改变 Notification 内容的方法。
具体要求：

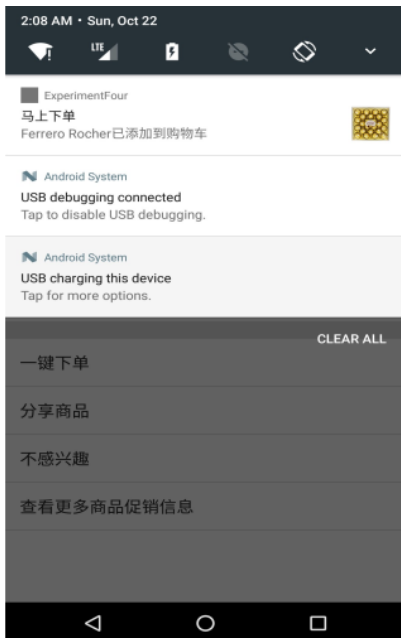
- (1) 在启动应用时，会有通知产生，随机推荐一个商品：



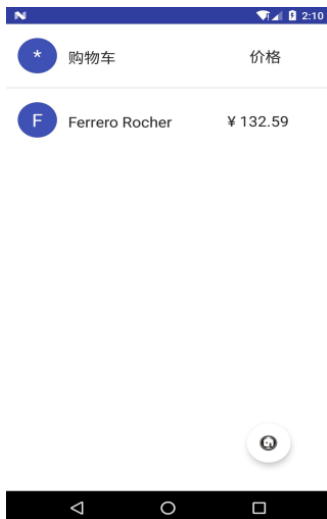
- (2) 点击通知跳转到该商品详情界面：



(3)点击购物车图标，会有对应通知产生，并通过 Eventbus 在购物车列表更新数据:



(4)点击通知返回购物车列表:

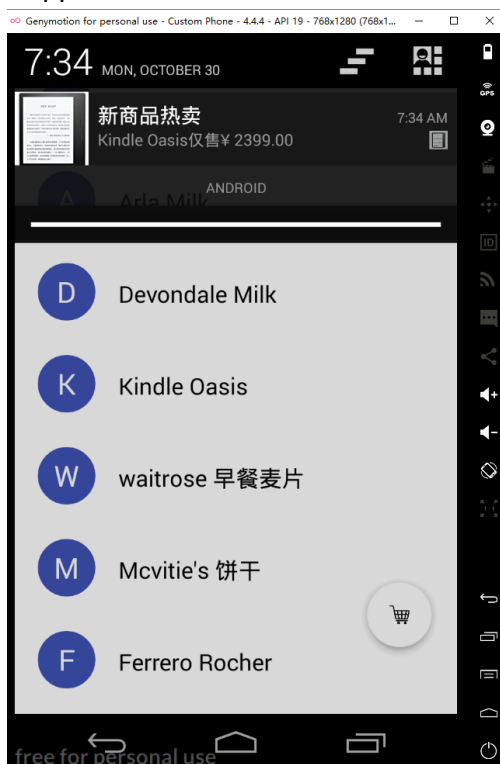


(5)实现方式要求:启动页面的通知由静态广播产生，点击购物车图标的通知由动态广播产生。

四、 课堂实验结果

(1) 实验截图

1. app 启动时随机推荐



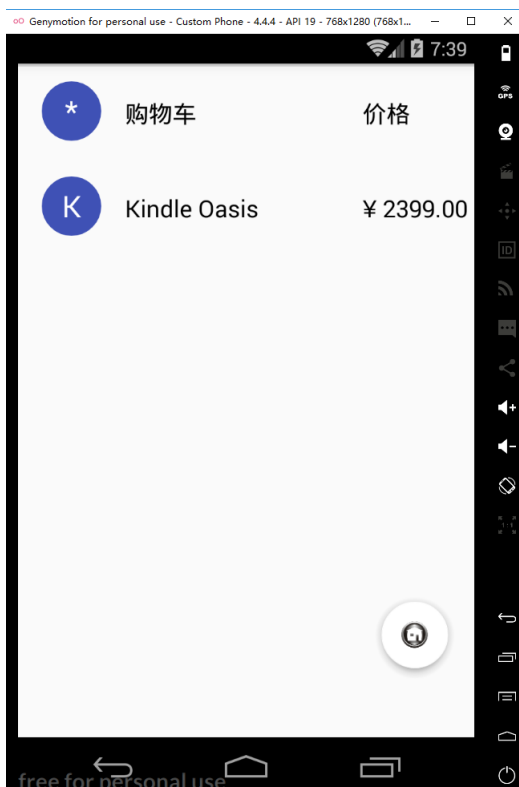
2. 点击通知进入详情页



3. 点击加入购物车，出现动态广播通知



4. 点击通知返回购物车



(2) 实验步骤以及关键代码

静态广播部分：

1.静态广播类，重写其中的 `onReceive` 方法，当接收到对应广播是进行数据处理，产生通知。这里除了使用 `Bundle` 和 `Notification` 产生通知之外，还需要设置点击通知产生跳转 `activ` 的事件，这一步使用 `pendingIntent` 存好跳转需要储存的信息，以及要打开哪个 `activity`，然后再 `Notification.Builder` 中调用 `setContentIntent(pendingIntent)`使之生效。

```
public class StaticReceiver extends BroadcastReceiver {
    private String Name;
    private String Price;
    private String Info;
    private NotificationManager mNotifyMgr;
    final String STATICACTION = "StaticFilter";

    @Override
    public void onReceive(Context context, Intent intent) {
        if (intent.getAction().equals(STATICACTION)) {
            Bundle bundle = intent.getExtras();
            Name = bundle.getString("Name");
            Price = bundle.getString("Price");
            Info = bundle.getString("Info");
            Bitmap bm = BitmapFactory.decodeResource(context.getResources(), ItemImage.getImg(Name));
            Notification.Builder builder = new Notification.Builder(context);

            //设置点击跳转事件
            Intent mIntent = new Intent(context, ItemsDetails.class);
            mIntent.putExtra("Name", Name);
            mIntent.putExtra("Price", Price);
            mIntent.putExtra("Info", Info);
            //PendingIntent 4个参数，第一个是上下文，第二个是requestCode，第三个Intent，
            //第4个是对参数的操作表示，常用的是FLAG_CANCEL_CURRENT和FLAG_UPDATE_CURRENT
            PendingIntent pendingIntent = PendingIntent.getActivity(context, 1, mIntent,
                PendingIntent.FLAG_CANCEL_CURRENT);

            builder.setContentTitle("新商品热卖") //设置通知栏标题：发件人
                .setContentText(Name + "仅售" + Price) //设置通知栏显示内容：短信内容
                .setTicker("新商品热卖，走过路过不要错过！") //通知首次出现在通知栏，带上升动画效果
                .setLargeIcon(bm) //设置通知大ICON
                .setSmallIcon(ItemImage.getImg(Name)) //设置通知小ICON（通知栏）
                .setAutoCancel(true) //设置这个标志当用户单击面板就可以让通知自动取消
                .setContentIntent(pendingIntent);

            //获取状态通知栏管理
            NotificationManager manager = (NotificationManager) context.getSystemService(context.NOTIFICATION_SERVICE);
            //绑定Notification，发送通知请求
            Notification notify = builder.build();
            manager.notify(1, notify);
        }
    }
}
```

2. 在 `MainActivity` 里，使用随机数选择一个商品，利用 `bundle` 和 `intent` 将图片和文字内容发送出去。

```
//静态广播，在启动应用时会有通知产生随机推荐一个商品
Random random = new Random();
int rand = random.nextInt(items.size());
Intent intentBroadcast = new Intent(STATICACTION);
intentBroadcast.putExtra("Name", items.get(rand).getName());
intentBroadcast.putExtra("Price", items.get(rand).getPrice());
intentBroadcast.putExtra("Info", items.get(rand).getInfo());
sendBroadcast(intentBroadcast);
```

3.在 AndroidManifest 中注册静态广播，一种比较方便的方式是，创建静态广播类时，在 AS 左边目录，右键->New->Other->Broadcast Reciver，这样 AS 会帮我们自动在 AndroidManifest 中注册，接下来我们只要加一句广播的名称即可。

```
<receiver
    android:name=".StaticReceiver"
    android:enabled="true"
    android:exported="true">
    <intent-filter>
        <action android:name="StaticFilter" />
    </intent-filter>
</receiver>
```

自定广播名称
即Intent的那个
STATICACTION

动态广播部分：

1.动态广播类，基本与静态那个差不多。

```
public class DynamicReceiver extends BroadcastReceiver {
    private String Name;
    private String Price;
    private String Info;
    private NotificationManager mNotifyMgr;
    final String DYNAMICACTION = "com.example.freedom.MyDynamicFilter";

    @Override
    public void onReceive(Context context, Intent intent) {
        if (intent.getAction().equals(DYNAMICACTION)) {
            Bundle bundle = intent.getExtras();
            Name = bundle.getString("Name");
            Bitmap bm = BitmapFactory.decodeResource(context.getResources(), ItemImage.getImg(Name));
            Notification.Builder builder = new Notification.Builder(context);
            //设置点击跳转事件
            Intent mIntent = new Intent(context, MainActivity.class);
            //PendingIntent 4个参数，第一个是上下文，第二个是requestCode，第三个Intent，
            //第四个是对参数的操作表示，常用的是FLAG_CANCEL_CURRENT和FLAG_UPDATE_CURRENT
            PendingIntent pendingIntent = PendingIntent.getActivity(context, 2, mIntent,
                PendingIntent.FLAG_UPDATE_CURRENT);

            builder.setContentTitle("马上下单") //设置通知栏标题: 发件人
                .setContentText(Name + "已添加到购物车") //设置通知栏显示内容: 短信内容
                .setTicker("您已成功下单") //通知首次出现在通知栏，带上升动画效果
                .setLargeIcon(bm) //设置通知大ICON
                .setSmallIcon(ItemImage.getImg(Name)) //设置通知小ICON (通知栏)
                .setAutoCancel(true) //设置这个标志当用户单击面板就可以让通知自动取消
                .setContentIntent(pendingIntent);

            //获取状态通知栏管理
            NotificationManager manager = (NotificationManager) context.getSystemService(context.NOTIFICATION_SERVICE);
            //绑定Notification, 发送通知请求
            Notification notify = builder.build();
            manager.notify(2, notify);
        }
    }
}
```

2.动态广播不能在 AndroidManifest 注册，只能在代码中动态注册和销毁，这里我选择在详情页的 activity 的 onCreate 时注册，并在销毁这个 activity 时注销广播，即在 onDestroy 中注销。

```
//动态广播注册
IntentFilter dynamic_filter = new IntentFilter();
dynamic_filter.addAction(DYNAMICACTION); //添加动态广播的Action
registerReceiver(dynamicReceiver, dynamic_filter); //注册自定义动态广播消息

@Override
protected void onDestroy() {
    super.onDestroy();
    //广播的注销
    unregisterReceiver(dynamicReceiver);
}
```

3.动态广播的触发设置在详情页点击购物车图标时

```
//点击购物按钮
shopcarbutton= (ImageButton) findViewById(R.id.itemdetail_r2_shopcar);
shopcarbutton.setOnClickListener((view) -> {
    Intent intentBroadcast = new Intent(DYNAMICACTION);
    intentBroadcast.putExtra("Name", name);
    intentBroadcast.setFlags(Intent.FLAG_ACTIVITY_NO_HISTORY);
    sendBroadcast(intentBroadcast);
    Toast.makeText(getApplicationContext(), "商品已添加到购物车", Toast.LENGTH_SHORT).show();
    EventBus.getDefault().post(new MessageEvent(name, price, info));
});
```

Eventbus 的使用:

1.这次 eventbus 是为了点击加入购物车时将商品信息传给购物车列表，先声明一个事件类用来存储要传递的信息。

```
public class MessageEvent {
    private String Name;
    private String Price;
    private String Info;
    public MessageEvent(String name, String price, String info) {
        this.Name=name;
        this.Price=price;
        this.Info=info;
    }
    public String getName() { return Name; }
    public String getPrice() { return Price; }
    public String getInfo() { return Info; }
}
```

2.在购物车 activity 中注册订阅者，以及准备订阅者（即写收到订阅后要做什么事情，这里要做的事是将商品添加到购物车，可选地指定线程模式）。注意退出时要注销订阅者。

```
//注册订阅者
EventBus.getDefault().register(this);

//准备订阅者
@Subscribe(threadMode = ThreadMode.MAIN)
public void onMessageEvent(MessageEvent messageEvent) {
    shopcaritems.add(new Items(messageEvent.getName(), messageEvent.getInfo(), messageEvent.getPrice()));
    shopcarAdapter.notifyDataSetChanged();
}

@Override
protected void onDestroy() {
    super.onDestroy();
    //撤销订阅者
    EventBus.getDefault().unregister(this);
}
```

其他部分:

1.在本次实验中，要求点击动态广播通知时要跳转到购物车列表。这里我们首先在 Android 主界面中将 launchMode 设置为了 singleInstance，使得点击 Notification 后不会另外新建一个购物车列表。

```
<activity
    android:name=".MainActivity"
    android:launchMode="singleInstance" <!-- 红色箭头 -->
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

为了在点击打开通知打开的购物车列表而不是商品列表，这两者我是写在了一个 activity 里的，也就是说我需要在点击通知时，打开这个 activity 的时候将购物车列表设置为可见，而将商品列表设置为不可见。但是我们不能新建一个 activity，新建一个 activity 会调用 onCreate 函数，然后

将购物车列表设置为默认不可见。我们需要重写一个叫 `onNewIntent` 的函数，这个函数的作用时，如果之前已经打开过这个 `activity` 了，那么它会将这个 `activity` 从栈里拿出来，而不是新建一个。这时我们在这个函数里设置购物车列表可见，商品列表不可见就可以了。

```
@Override
protected void onNewIntent(Intent intent) {
    super.onNewIntent(intent);
    Intent mIntent=getIntent();
    Shopcar.setVisibility(View.VISIBLE);
    Itemlist.setVisibility(View.INVISIBLE);
    switchbutton.setTag("1");
    switchbutton.setImageResource(R.drawable.mainpage);
}
```

(3) 实验遇到困难以及解决思路

1.这次实验还比较简单，基本功能实现出来都比较容易，按照文档的内容打就可以了。唯一的困难是一开始我不知道怎么在点击动态广播通知时跳转到购物车列表。到购后来询问了同学，知道了有 `onNewIntent` 这个函数，上网查了它的基本使用方法之后解决了问题。

2.另外这次要求给 `Notification` 设置 `LargeIcon`，而这个东西需要使用位图 `Bitmap` 作为参数，上网百度之后知道可以使用 `BitmapFactory.decodeResource` 来获取 `Bitmap` 对象，如下：

```
Bitmap bm = BitmapFactory.decodeResource(context.getResources(),ItemImage.getImg(Name));
```

其中第一个参数是上下文，第二个参数是要加载的位图资源的 `id`。

五、 课后实验结果

这次实验按照基本步骤实现之后有个 bug：点击购买商品 a→出现动态广播→点击动态进入购物车列表→点击悬浮按钮回到商品列表→点击任意非 a 商品→却进入了 a 商品详细信息。产生这个 bug 的原因是点击了通知以后那个商品详情的 `activity` 没有被 finish 掉。查找资料后我找到的解决方式是在调用动态广播类之前，将传给它的 `intent` 设一个 flag，参数我选择的是 `FLAG_ACTIVITY_NO_HISTORY`，这个参数的作用是这个 flag 启动的 `activity` 一旦退出，它不会存在于栈中，也就是相当于被 finish 掉。

```
//点击购物按钮
shopcarbutton= (ImageButton) findViewById(R.id.itemdetail_r2_shopcar);
shopcarbutton.setOnClickListener((view) -> {
    Intent intentBroadcast = new Intent(DYNAMICACTION);
    intentBroadcast.putExtra("Name", name);
    intentBroadcast.setFlags(Intent.FLAG_ACTIVITY_NO_HISTORY);
    sendBroadcast(intentBroadcast);
    Toast.makeText(getApplicationContext(), "商品已添加到购物车", Toast.LENGTH_SHORT).show();
    EventBus.getDefault().post(new MessageEvent(name, price, info));
});
```


六、 实验思考及感想

这次实验学习了 **broadcast** 的使用，以及 **Notification** 和 **EventBus** 的编程基础。通过这次实验我感受到了安卓的 **api** 真的很丰富，功能很全，我所知的也不过冰山一角，还需要继续努力去熟悉。