

# 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	2015	专业（方向）	移动信息工程（互联网方向）
学号	15352151	姓名	柯博
电话	13671412922	Email	<a href="mailto:Kebo@mail2.sysu.edu.cn">Kebo@mail2.sysu.edu.cn</a>
开始日期		完成日期	

### 一、 实验题目

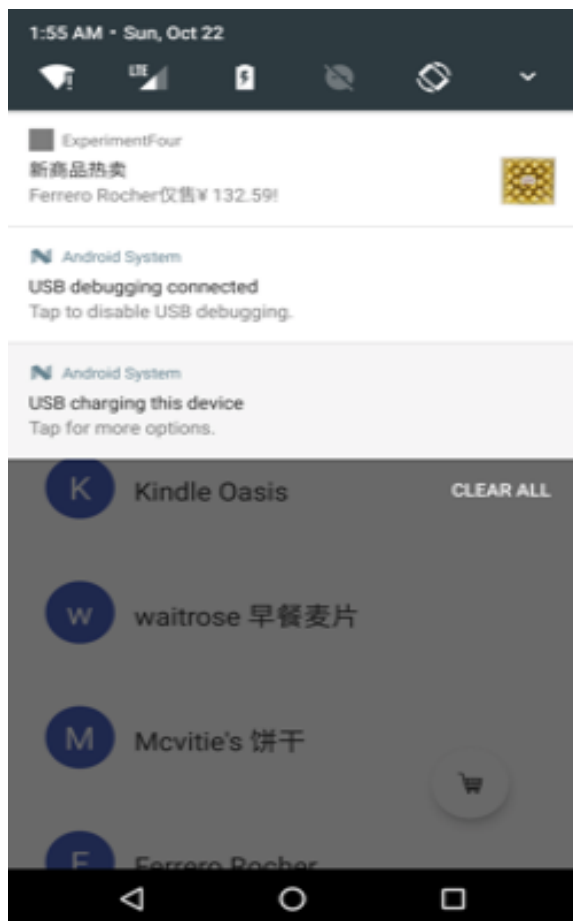
实验四、Broadcast 的使用

### 二、 实现内容

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：



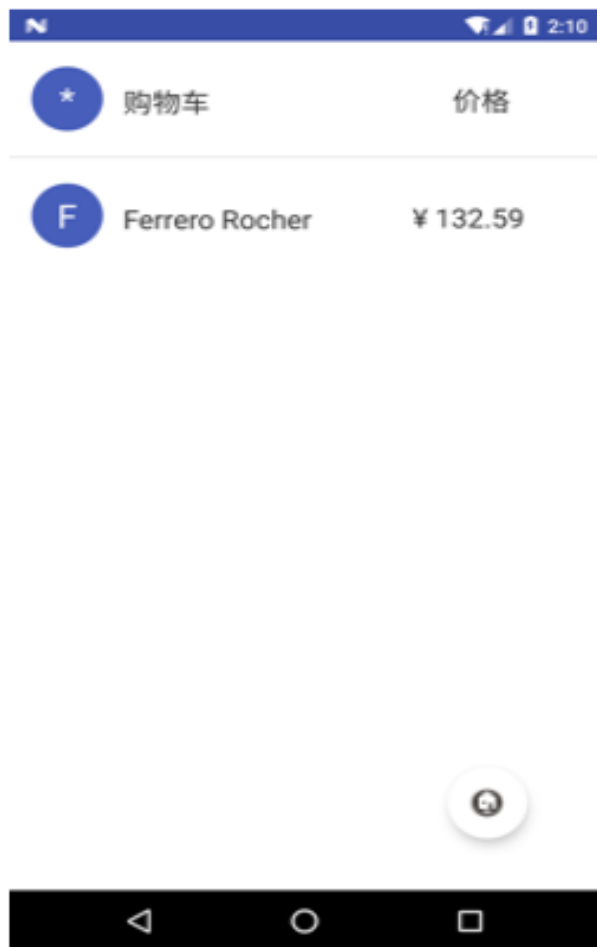
(2)点击通知跳转到该商品详情界面:



(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:



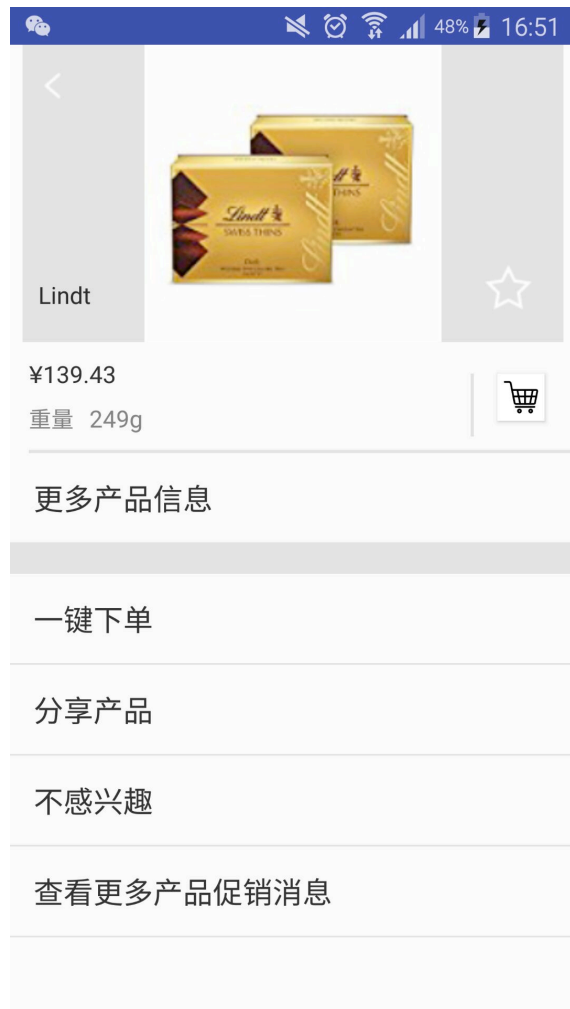
(5)实现方式要求:启动页面的通知由静态广播产生，点击购物车图标的通知由动态广播产生。

### 三、 课堂实验结果

(1) 实验截图



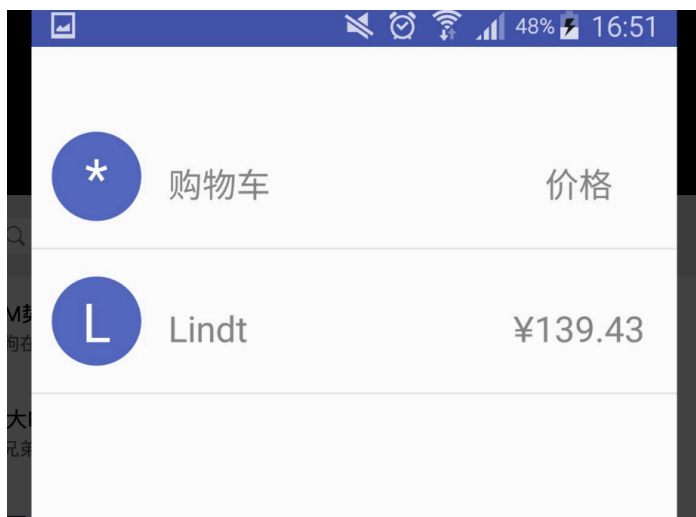
启动应用产生通知，点击通知进入商品详情：



点击购物车按钮，产生通知，商品加入购物车；



点击通知进入购物车列表；



再次启动应用，随机推荐另一个商品；



## (2) 实验步骤以及关键代码

根据实验内容，先实现应用启动的静态广播：

在 MainActivity 启动时发送静态广播；

```
//静态广播
Intent intentbroadcast =new Intent(STATICACTION);
sendBroadcast(intentbroadcast);
```

在 AndroidManifest.xml 中注册静态广播；

```
<receiver android:name=".Receiver">
    <intent-filter>
        <action android:name="com.example.kebo.lab4.staticaction"></action>
    </intent-filter>
</receiver>
```

新建 Receiver.java 文件，新建类 Receiver 继承自 BroadcastReceiver，重写 onCreate 函数，实现当收到静态广播时，发送一个通知，随机推荐商品：

```
public void onReceive(Context context, Intent intent) {
    my_app = (ShoppingApp) context.getApplicationContext();
    if (intent.getAction().equals(STATICACTION)) { //静态广播
        //随机数生成推荐商品
        Random random = new Random();
        idx = random.nextInt( bound: 10);
        String name = my_app.item_data.data.get(idx).get("name").toString();
        String price = my_app.item_data.data.get(idx).get("price").toString();
        int image = my_app.item_data.imgid.get(idx);
        Bitmap bm = BitmapFactory.decodeResource(context.getResources(), image);
        //通知栏状态管理
        NotificationManager manager = (NotificationManager) context.getSystemService(
        //实例化通知栏构造器
        Notification.Builder builder = new Notification.Builder(context);
        //对Builder进行配置
        builder.setContentTitle("每日商品推荐") //设置通知栏标题：发件人
            .setContentText(name+"仅售"+price+"! ") //通知内容
            .setTicker("您有一条新消息") //通知首次出现在通知栏，带上动画效果
            .setLargeIcon(bm) //设置通知大ICON
            .setSmallIcon(image) //设置通知小ICON
            .setAutoCancel(true); //当用户点击，通知自动取消
        //绑定Intent，单击通知时进入DetailActivity
        Intent mintent = new Intent(context, DetailActivity.class);
        mintent.putExtra( name: "itemid", idx);
        PendingIntent mPendingIntent = PendingIntent.getActivity(context, requestCode
        builder.setContentIntent(mPendingIntent);
        //绑定Notification，发送通知请求
        Notification notify = builder.build();
        manager.notify( id: 0, notify);
    }
}
```

在商品详情类中添加动态广播注册函数，并在 onCreate 函数中注册：

```
void registerReceiver() {
    Receiver dynamicReceiver = new Receiver();
    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
}

//注册动态广播
registerReceiver();
```

修改购物车按钮的点击事件处理，将信息交给 EventBus 发送给购物车列表，同时发送动态广播：

```
b_addchart.setOnClickListener((v) -> { //设置添加购物车按钮事件
    my_app.chart_data.add( //将当前商品加入购物车的列表中
        my_app.item_data.data.get(idx),
        my_app.item_data.stared.get(idx),
        my_app.item_data.itemid.get(idx),
        my_app.item_data.imgid.get(idx)
    );
    EventBus.getDefault().post(iid); //用EventBus发送itemid
    Intent intentbroadcast = new Intent(DYNAMICACTION);
    //intentbroadcast.putExtra("itemid", iid);
    sendBroadcast(intentbroadcast); //发送动态广播
    my_app.listview_adpater.notifyDataSetChanged(); //刷新列表
    Toast.makeText( context: DetailActivity.this,
        text: "商品已经添加到购物车", Toast.LENGTH_SHORT).show();
});
```



Receiver.java 中实现接收到动态广播的处理；

```
else if (intent.getAction().equals(DYNAMICACTION)) { //动态广播
    int iid = intent.getIntExtra( name: "itemid", defaultValue: 0);
    idx = my_app.item_data.getIndex(iid);
    String name = my_app.item_data.data.get(idx).get("name").toString();
    String price = my_app.item_data.data.get(idx).get("price").toString();
    int image = my_app.item_data.imgid.get(idx);
    Bitmap bm = BitmapFactory.decodeResource(context.getResources(), image);
    NotificationManager manager = (NotificationManager) context.getSystemService(
    Notification.Builder builder = new Notification.Builder(context);
    builder.setContentTitle("马上下单")
        .setContentText(name+"已经添加到购物车")
        .setTicker("您有一条新消息")
        .setLargeIcon(bm)
        .setSmallIcon(image)
        .setAutoCancel(true);
    Intent mintent = new Intent(context, MainActivity.class);
    //mintent.putExtra("itemid", iid);
    PendingIntent mPendingIntent = PendingIntent.getActivity(context, requestCode
    builder.setContentIntent(mPendingIntent);
    Notification notify = builder.build();
    manager.notify( id: 0, notify);
}
```

在 MainActivity 中完善对 EventBus 数据的处理和动态广播的处理；

```
//注册EventBus
EventBus.getDefault().register( subscriber: this);
}
//当mainActivity启动之后收到intent的响应
@Override
protected void onNewIntent(Intent intent) {
    mode = 2; //切换MainActivity显示的内容为购物车
    lv_items.setVisibility(View.VISIBLE);
    rv_items.setVisibility(View.GONE);
    fab_tochart.setImageResource(R.mipmap.mainpage);
}
@Override
public void onDestroy() {
    super.onDestroy();
}
//EventBus的订阅者方法
@Subscribe(threadMode = ThreadMode.MAIN)
public void onMessageEvent(Integer iid) {
    int idx = my_app.item_data.getIndex(iid);
    my_app.chart_data.add( //将当前商品加入购物车的列表中
        my_app.item_data.data.get(idx),
        my_app.item_data.stared.get(idx),
        my_app.item_data.itemid.get(idx),
        my_app.item_data.imgid.get(idx)
    );
    ((MyListViewAdapter)lv_items.getAdapter()).notifyDataSetChanged();
}
```

### (3) 实验遇到困难以及解决思路

启动应用即闪退问题：静态广播的接收处理 Receiver.java 写了一个空指针，自己还一直没发现；在商品详情页面将该商品添加到购物车之后，弹出的通知内容与商品不符：将商品信息传递给购物车，但是没有传给 Receiver，加一个 Intent 即可；

## 四、 课后实验结果

## 五、 实验思考及感想

相比上个实验，这次的实验可以说是很简单了，但是 Broadcast 作为 Android 的四大核心组件之一，还有很多功能没有在本次实验中体现，正如实验目的中所说的，掌握了 Broadcast 的编程基础。Android 开发之路仍然任重道远。

作业要求：

1. 命名要求：学号\_姓名\_实验编号，例如 15330000\_林 XX\_lab1。
2. 实验报告提交格式为 pdf。
3. 实验内容不允许抄袭，我们要进行代码相似度对比。如发现抄袭，按 0 分处理。