中山大学数据科学与计算机学院软件工程（移动信息工程）本科生实验报告

移动应用开发实验报告

姓名	学号	班级	电话	邮箱	日期
蔡烨	15352010	1501班	15989010669	caiy29@mail2.sysu.edu.cn	2017/11/5

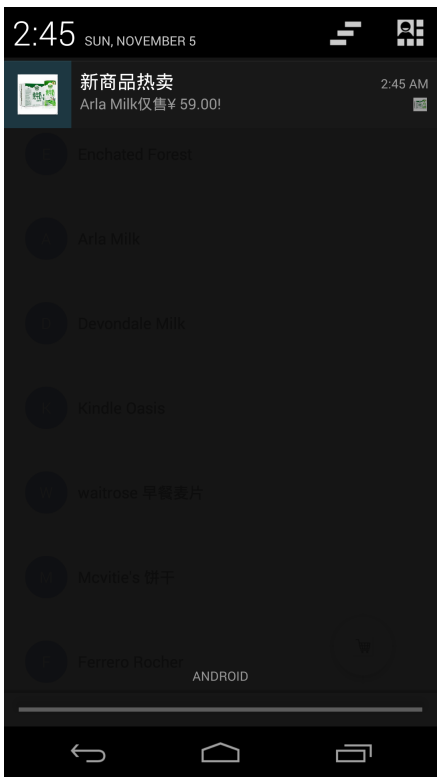
1. 实验题目

Broadcast的使用

2. 实现内容

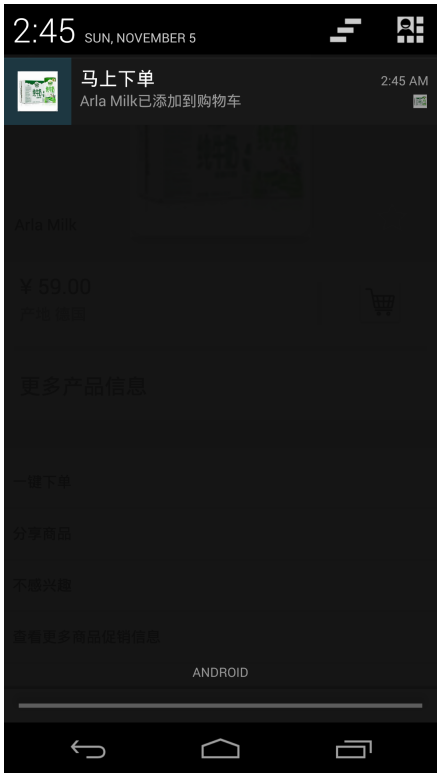
在实验三的基础上，实现静态广播、动态广播两张改变Notification内容的方法。

(1)在应用启动时，会有通知产生，随机推荐一个商品：

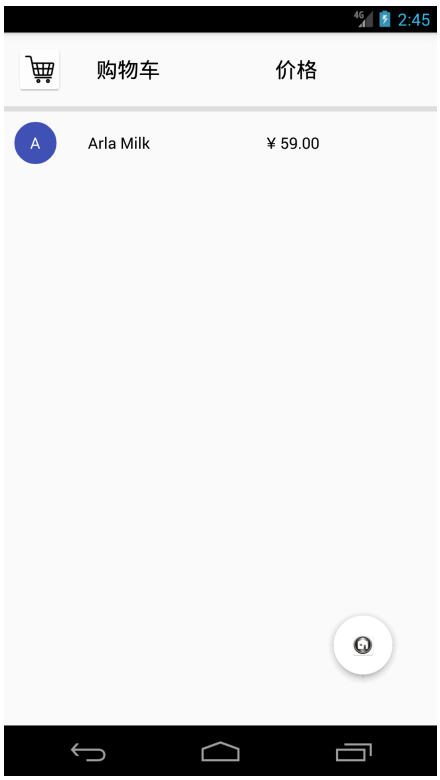


(2)点击通知跳转到该商品详情界面：

(3)点击购物车图标，会有对应的通知产生，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：



3. 实验过程

1. 修改注册文件，注册静态广播

```
<receiver android:name=".myBroadcastReceiver">
    <intent-filter>
        <action android:name="com.example.caiye.lab3.start"/>
    </intent-filter>
</receiver>
```

2. 在MainActivity.java中，发送静态广播，此时设置的intent的action需要和注册时的name保持一致。

```
//产生随机数
Random random = new Random();
int i = random.nextInt(name.length);
//发送静态广播
Bundle bundle = new Bundle();
bundle.putString("name",name[i]);
bundle.putString("price",price[i]);
bundle.putInt("pic",pic[i]);
bundle.putString("info",info[i]);

Intent intentBroadcast = new Intent();
intentBroadcast.setAction("com.example.caiye.lab3.start");
intentBroadcast.putExtras(bundle);
sendBroadcast(intentBroadcast);
```

3. 新建一个类BroadcastReceiver的继承类myBroadcastReceiver，重写onReceive函数，接收静态广播。接收的判断标准是intent的action是不是之前发送时候的那个，也就是注册文件中的name。

```
public void onReceive(Context context, Intent intent) {
    //静态广播，应用启动时
    if(intent.getAction().equals("com.example.caiye.lab3.start")){
        Bundle bundle = intent.getExtras();
        String showName = bundle.getString("name");
        String showPrice = bundle.getString("price");
        Bitmap bm = BitmapFactory.decodeResource(context.getResources(),bundle.getInt("pic"));

        Notification.Builder builder = new Notification.Builder(context);
        builder.setContentTitle("新商品热卖")
                .setContentText(showName+"仅售"+showPrice+"!")
                .setLargeIcon(bm)
                .setSmallIcon(bundle.getInt("pic"))
                .setAutoCancel(true);
        //获取状态通知栏管理
        NotificationManager notificationManager = (NotificationManager)context.getSystemService(Context.NOTIFICATION_SERVICE);
        //点击notification，跳转到商品详情
        Intent intent1 = new Intent(context,DetailsActivity.class);
        intent1.putExtras(bundle);
        PendingIntent pendingIntent = PendingIntent.getActivity(context,0,intent1,PendingIntent.FLAG_UPDATE_CURRENT);
        builder.setContentIntent(pendingIntent);
        //绑定notification，发送通知请求
        Notification notify = builder.build();
        notificationManager.notify(0,notify);
    }
```

其中，Bitmap是位图，是图像处理最重要的类之一，用它可以获取图像文件信息，进行图像剪切、旋转、缩放等操作，并可以指定格式保存图像文件。

4. 添加Eventbus的依赖，声明一个事件类，用来传递商品信息

```
compile 'org.greenrobot:eventbus:3.0.0'
```

```
public class Event {
    String name;
    String price;
    String info;
    int pic;

    Event(String name,String price,String info,int pic){
        this.name = name;
        this.price = price;
        this.info = info;
        this.pic = pic;
    }

    public String getName() { return name; }
    public String getPrice() { return price; }
    public String getInfo() { return info; }
    public int getPic(){ return pic;}
}
```

5. 在MainActivity.java中，准备订阅者：声明并注释订阅方法，可选地指定线程模式；注册订阅者；注销订阅者。

```
@Subscribe(threadMode = ThreadMode.MAIN)
public void onMessageEvent(Event event) {
    carGoods.add(new Good(event.getName(),event.getPrice(),event.getInfo(),event.getPic()));
    listViewAdapter.notifyDataSetChanged();
};
protected void onDestroy(){
    super.onDestroy();
    EventBus.getDefault().unregister(this);//注销eventBus
}

 protected void onCreate(Bundle savedInstanceState) {
     super.onCreate(savedInstanceState);
     setContentView(R.layout.activity_main);
     EventBus.getDefault().register(this);//注册eventBus
```

6. 当购物车按钮被点击时传递Event事件，注册动态广播，发送动态广播，注销动态广播。

```
    //购物车按钮的监听
    car.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Toast.makeText(DetailsActivity.this,"商品已添加到购物车",Toast.LENGTH_SHORT).show();
            //传递event事件
            EventBus.getDefault().post(new Event(showName,showPrice,showInfo,showPic));
            //注册动态广播
            IntentFilter dynamic_filter = new IntentFilter();
            dynamic_filter.addAction("com.example.caiye.lab3.dynamic");
            registerReceiver(dynamicReceiver,dynamic_filter);
            //发送动态广播
            Intent broadcastIntent = new Intent();
            broadcastIntent.setAction("com.example.caiye.lab3.dynamic");
            broadcastIntent.putExtras(bundle);
            sendBroadcast(broadcastIntent);
        }
    });
}
//注销动态广播
protected void onDestroy(){
    super.onDestroy();
    unregisterReceiver(dynamicReceiver);
}
```

7. 在MainActivity.java中，重新onNewIntent函数，使得当activity跳转到MinActivity时，如果intent中传来的布尔类型是true，则是跳转到购物车列表而不是商品列表。

```
@Override
protected void onNewIntent(Intent intent){
    super.onNewIntent(intent);
    if(intent.getBooleanExtra("jump",true)){
        floatingCar.setImageResource(R.mipmap.mainpage);
        carLayout.setVisibility(View.VISIBLE);
        homeLayout.setVisibility(View.INVISIBLE);
    }
}
```

8. 实现BroadcastReceiver子类，重写onReceive方法，这里直接将它写进静态广播的接收器中即可，只需要判断收到的是动态广播，其他操作基本一致。

```
else if(intent.getAction().equals("com.example.caiye.lab3.dynamic")){
    Bundle bundle = intent.getExtras();
    Bitmap bm = BitmapFactory.decodeResource(context.getResources(),bundle.getInt("pic"));

    Notification.Builder builder = new Notification.Builder(context);
    builder.setContentTitle("马上下单")
            .setContentText(bundle.getString("name")+"已添加到购物车")
            .setLargeIcon(bm)
            .setSmallIcon(bundle.getInt("pic"))
            .setAutoCancel(true);
    //获取状态通知栏管理
    NotificationManager notificationManager = (NotificationManager)context.getSystemService(Context.NOTIFICATION_SERVICE);
    //点击notification，跳转到购物车列表
    Intent intent1 = new Intent(context,MainActivity.class);
    intent1.putExtra("jump",true);

    PendingIntent pendingIntent = PendingIntent.getActivity(context,0,intent1,PendingIntent.FLAG_UPDATE_CURRENT);
    builder.setContentIntent(pendingIntent);
    //绑定notification，发生通知请求
    Notification notify = builder.build();
    notificationManager.notify(cnt++,notify);
}
```

9. 将launchMode设置为singleTask，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="singleTask">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

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

4. 实验思考及感想

对我来说本次实验比较难以搞懂的其实是eventbus的准备订阅、注册订阅、注销订阅，一开始都不知道应该写在哪，还是就是位图的使用，都是通过搜索引擎查找，看看网上博客的解释和别人贴上去的其他项目的代码，了解该如何使用，如何写下来。

还有就是广播接收器中的pendingIntent，它和intent的区别在于，intent是及时启动的，随所在的activity消失而消失，而pendingIntent这个类用于处理即将发生的事情，比如在notification中用于跳转页面，但不是马上跳转，可以看成是延迟执行的intent。整个notification的使用TA给的实验指导文档其实说得很清楚，虽然有些看不太懂，但经过搜索之后还是可以基本理解。