

Data Interaction of Fragment Based on the Open-source Framework EventBus3.1

Zhiwei Xia

Shandong Transportation Vocational College, Weifang, Shandong, 261206, China

Abstract

When Android is developed, it uses ViewPager + Fragment to realize multiple page switches to realize data interaction and use the advantages of flexible EventBus scheduling to complete data transmission and update the UI.

Keywords

EventBus; data interaction; Fragment

基于开源框架 EventBus3.1 的 Fragment 的数据交互

夏智伟

山东交通职业学院, 中国·山东 潍坊 261206

摘要

Android开发时用到ViewPager+Fragment实现多个页面切换时实现数据交互时使用EventBus调度灵活的优势,完成数据传递并更新UI。

关键词

EventBus; 数据交互; Fragment

1 引言

在进行 Android 开发时经常会用到 ViewPager+Fragment 实现多个页面切换时实现数据交互的情况,常用方式使用 Handler 和 BroadCast 来实现数据。为什么越来越多的开发者会选择使用 EventBus,它给项目带来了什么?它与 Android 原有的消息处理机制有什么区别和优缺点?项目在什么场景下采用 EventBus 会比较合适?

2 Handler+Message、BroadCast 和 EventBus 之比较

2.1 Handler+Message

Handler 常用于线程之间的通信,在主线程中使用 Runnable 和 Message 对象实现数据的传递,每个 Handler 实例都会被绑定到主线程中使用。消息在进行传递时,一般 Handler 在调用线程时创建 Handler 的内部类重写 HandlerMessage 方法,通过 SendMessage 方法进行消息的发

布处理消息时通过 switch 多分支选择语句进行消息分发和相应的处理,实现了事件发布者和接受者之间的高耦合。而使用 Handler 时最大的好处是发生问题时,可以非常明确、快速地进行定位,通过 msg.what 理清每一条消息流的逻辑。

2.2 BroadCast

广播是相对消耗时间、空间最多的一种方式,但是广播作为四大组件非常重要的组件,许多系统级的事件都是通过广播来完成的,在 BroadcastReceiver 的 onReceive 方法中,可以获得 Context、intent 参数,这两个参数可以调用许多的 sdk 中的方法,而 eventbus 获得这两个参数相对比较困难;他的优势体现在与 sdk 连接紧密,如果需要同 android 交互的时候,广播的便捷性会抵消掉它过多的资源消耗,广播的不可替代性在于它可以跨进程进行通信,也就是不同 APP 之间可以通过广播进行传递数据,但同一个 APP 内部的通信,使用广播是较为消耗资源和笨重的。

2.3 EventBus

EventBus 的优势是调度灵活,使用时无需像广播一样关注 Context 的注入与传递,也解除了 Handler 所带来的耦合,是 EventBus 比之于广播、观察者等方式最大的优点,

【作者简介】夏智伟(1981-),女,回族,中国山东潍坊人,副教授,从事智能终端应用开发研究。

但 EventBus 也有很明显的缺陷,在 EventBus 中事件的分发是通过注解函数的参数类型确定的,因此在事件发布有多个相同参数时,很难从事件发布者开始理清消息流,当程序代码适量时,这是一个合理的要求,而当程序太大时,这将成为一种负担。

3 基于开源框架 EventBus3.1 的 Fragment 的数据交互的案例分析

本案例是使用 ViewPager 和 Fragment,展示使用开源框架 EventBus3.0 实现 Activity 和 Fragment 之间数据的传递。

3.1 使用 ViewPager2 完成主页面的搭建

```
<androidx.viewpager2.widget.ViewPager2
    android:id="@+id/viewpager2"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
/>
```

此页面使用 Google 的 androidx 组件包最新升级的 ViewPager2, ViewPager 2 底层是用 RecyclerView 实现的,首先需要在 build.gradle 中添加 ViewPager2 和 ViewPager 2。需要使用 RecyclerView 的 adapter 的依赖包。

```
implementation 'androidx.viewpager2:viewpager2:1.0.0';
implementation 'androidx.recyclerview:recyclerview:1.1.0'
```

创建 Fragment 的布局。

3.2 创建 ViewPager2 的适配器类

ViewPager2 使用的是 FragmentStateAdapter 适配器类。在编辑 ViewPager2Adapter 适配器类时首先要添加完成实例化操作的构造函数,完成多个 Fragment 列表的初始化。createFragment 方法中使用 list.get(position) 方法获取 Fragment 列表中的数据项,getItemCount 方法中使用 list.size() 获取 Fragment 列表中数据的个数。自定义 addFragment 方法,判断如果 fragment 不为空的情况下使用 list.add(fragment) 将 fragment 添加到列表中。

3.3 创建 MainActivity 的主要类

①实例化 list。

使用 list = new ArrayList<>() 完成 Fragment 列表的实例化操作。

②实例化并设置适配器。

```
ViewPager2Adapter viewPager2Adapter = new
ViewPager2Adapter(this,list);
```

通过实例化 ViewPager2Adapter 将当前的 FragmentAc-

tivity 和 list 实际数据传入构造函数。

viewPager2.setAdapter(viewPager2Adapter) 完成适配器的设置。

③添加 Fragment 到 ViewPager2 中并设置当前显示的页面。

```
viewPager2Adapter.addFragment(new FragmentOne());
viewPager2Adapter.addFragment(new FragmentTwo());
viewPager2Adapter.addFragment(new FragmentThree());
viewPager2Adapter.addFragment(new FragmentFour());
viewPager2.setCurrentItem(1);
```

使用 viewPager2Adapter.addFragment 方法完成不同 Fragment 的添加并使用 viewPager2.setCurrentItem(1) 设置当前显示的页面为第二个页面。

④使用 viewPager2.registerOnPageChangeCallback 设置页面切换时触发的事件。

⑤添加 EventBus 的依赖库。

在 build.gradle 中添加 EventBus 的依赖库。

```
implementation 'org.greenrobot:eventbus:3.1.1'
```

如图 1 所示。

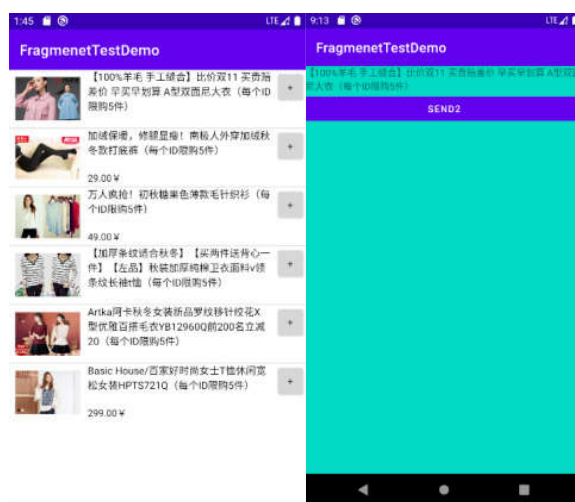


图 1

⑥传递数据。

本案例在 FragmentTwo 中单击按钮跳转到 ListView 视图后单击 ListView 列表视图中的任何一个选项再重新调回到 MainActivity 中的第二个页面 FragmentTwo,同时传递数据并更新页面将数据显示在页面上。

在 ListView 的适配器中完成页面的跳转时并携带数据,代码如下:

```
Intent intent = new Intent(context,MainActivity.class);
```

```
intent.putExtra("id",1);
intent.putExtra("pic",(Integer) list.get(position).
get("pic"));
intent.putExtra("des",list.get(position).get("des").
toString());
intent.setFlags(FLAG_ACTIVITY_NEW_TASK);
context.startActivity(intent);
```

在 MainActivity 的 onResume 方法中完成数据的接收

```
id = getIntent().getIntExtra("id",0);
pic = getIntent().getIntExtra("pic",0);
des = getIntent().getStringExtra("des");
```

⑦发送数据。

在 MainActivity 中添加 EventBus.getDefault().post(new FirstEvent(des)) 完成数据的发送。

⑧接收数据。

在接收页面中添加“EventBus.getDefault().register(-this);”然后复写 onEventMainThread 方法中使用 tvF2.setText(event.getName()) 完成页面数据的更新。

3.4 总结

本案例使用开源框架 EventBus 实现了在 Activity 和另一个 Activity 的 Fragment 之间数据的传递，如果 Fragment 向 Activity 发送数据的方法原理也是相同的，在使用 EventBus 进行数据发送和接收的顺序进行调整即可。

参考文献

- [1] 赵益明.基于智能手机的股票交易系统的研发[D].浙江:浙江工商大学,2017.
- [2] 王恩鹏.某公司智能货柜系统客户端的设计与实现[D].北京:北京邮电大学,2021.
- [3] 基于Android平台的地震灾情采集系统的设计与实现[D].宁夏:宁夏回族自治区地震局,2020.