

Android消息传递之EventBus 3.0使用详解

2022-03-06 11:16:43 By 小峰

EventBus示例:

之前做图片社交App的时候，需要处理一个点赞数据的同步，比如在作品的详情页点赞需要同时更新列表页该作品的点赞数量，这里还是以此为例。

1.) build.gradle添加引用

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

2.) 定义一个事件类型

```
public class DataSynEvent {  
    private int count;  
  
    public int getCount() {  
        return count;  
    }  
  
    public void setCount(int count) {  
        this.count = count;  
    }  
}
```

3.) 订阅/解除订阅

订阅

```
EventBus.getDefault().register(this);//订阅
```

解除订阅

```
EventBus.getDefault().unregister(this);//解除订阅
```

4.) 发布事件

```
EventBus.getDefault().post(new DataSynEvent());
```

5.) 订阅事件处理

```
@Subscribe(threadMode = ThreadMode.MAIN) //在ui线程执行
public void onDataSynEvent(DataSynEvent event) {
    Log.e(TAG, "event---->" + event.getCount());
}
```

ThreadMode总共四个：

- MAIN UI主线程
- BACKGROUND 后台线程
- POSTING 和发布者处在同一个线程
- ASYNC 异步线程

6.) 订阅事件的优先级

事件的优先级类似广播的优先级，优先级越高优先获得消息

```
@Subscribe(threadMode = ThreadMode.MAIN,priority = 100) //在ui线程执行 优先级100
public void onDataSynEvent(DataSynEvent event) {
    Log.e(TAG, "event---->" + event.getCount());
}
```

7.) 终止事件往下传递

发送有序广播可以终止广播的继续往下传递，EventBus也实现了此功能

```
EventBus.getDefault().cancelEventDelivery(event) ;//优先级高的订阅者可以终止事件往下传递
```

8.) 处理代码混淆

```
-keepattributes *Annotation*
-keepclassmembers class ** {
    @org.greenrobot.eventbus.Subscribe <methods>;
}
```

```
-keep enum org.greenrobot.eventbus.ThreadMode { *; }
```

```
# Only required if you use AsyncExecutor
```

```
-keepclassmembers class * extends org.greenrobot.eventbus.util.ThrowableFailureEvent {  
    <init>(java.lang.Throwable);  
}
```

版权声明：

本站遵循 署名-非商业性使用-相同方式共享 2.5 共享协议。

转载请注明转自**疯人峰语**并标明URL。

本文链接：<https://www.iiii.name/?p=2428>