

1、依赖

compile 'org.greenrobot:eventbus:3.0.0'

2、在接收消息的类里对eventbus注册，接收消息，eventbus反注册注册：

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

反注册：界面销毁时反注册 防止发生内存泄漏

```
@Override  
protected void onDestroy() {  
    EventBus.getDefault().unregister(this);  
    super.onDestroy();  
}
```

接收消息：在需要用到消息的类的任何地方来接收

//通过自定义类来接收消息（发送消息也是通过自定义类来实现）

```
@Subscribe(threadMode = ThreadMode.MAIN)  
public void onMonEvent(MessEvent event){  
    String string = event.getString();  
    list3.add(string);  
}
```

//通过EventBus直接发送的消息的接收方法（这种方法不需要自定义类）

```
@Subscribe(threadMode = ThreadMode.MAIN)  
public void onMonEvent(String event){  
    list3.add(event);  
}
```

3、发送消息 在需要发送消息的任何地方来发送

通过自定义类发送消息

```
String shopLineName = "1234567889";
```

```
EventBus.getDefault().post(new MessEvent(shopLineName));
```

直接通过EventBus的POST方法直接发送消息（这种方法不需要自定义类）

```
EventBus.getDefault().post("12121222");
```

4、创建一个自定义类：用来发送 接收消息 定义相应的成员变量

```
public class MessEvent {
```

```
    private String msg;
```

```
    public MessEvent(String msg){
```

```
        this.msg = msg;
```

```
    }
```

```
    public String getString(){
```

```
        return msg;
```

```
    }
```

```
}
```

从Activity A类 发送消息到 Activity B类（接收信息类）

注意：B类是由A类跳转进入的，意味着A类创建之后B类还没有创建且B类的注册接收信息也不生效，

这种情况下要采用StickyEvent(粘性事件)

具体操作：

A类：先创建用于发送信息

直接发送

```
EventBus.getDefault().postSticky("你大爷的大爷!~~~~~我来了，草草");
```

通过事件对象发送

```
EventBus.getDefault().postSticky(new MessageEvent("你大爷的大爷!"));
```

B类：由A类跳转后进入 用于接收信息

注册：

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

返注册：

```
EventBus.getDefault().removeStickyEvent(MessageEvent.class); //手动移除消息
```

```
EventBus.getDefault().unregister(this);
```

接收信息：

直接接收信息

```
/**
 * sticky = true 重要的一个参数必须要有，否则无法收到信息
 */
@Subscribe(threadMode = ThreadMode.MAIN,sticky = true)
public void onEvent(String event) {
    if (event!= null) {

        /**将接收到的信息添加到集合或者数组或者StringBuffer等容器，不能直接在这里为Viwe设置值或其它赋值操作*/

        // mBuffer.append(event+"我真的收到信息了666666");
        // brr[0] = event+"我真的收到信息了999999";
        arr.add(event+"我真的收到信息了");

    }else {
        Toast.makeText(this,"没有收到",Toast.LENGTH_SHORT).show();
    }
}
```

通过事件对象接收信息

```
/**
 * sticky = true 重要的一个参数必须要有，否则无法收到信息
 */
@Subscribe(threadMode = ThreadMode.MAIN,sticky = true)
public void onEvent(MessageEvent event) {
    if (event.getMsg() != null) {
```

/**将接收到的信息添加到集合或者数组或者StringBuffer等容器，不能直接在这里为Viwe设置值或其它赋值操作*/

```
// mBuffer.append(event.getMsg()+"我真的收到信息了666666");
// brr[0] = event.getMsg()+"我真的收到信息了999999";
arr.add(event.getMsg()+"我真的收到信息了");
}else {
    Toast.makeText(this,"没有收到",Toast.LENGTH_SHORT).show();
}
}
```

B类中的设置数据到View控件的方法：

onCreate(Bundle savedInstanceState)的方法中找到控件后，通过集合对象来获取数据为控件设置接收到的数据

```
mtextviews = (TextView) findViewById(R.id.main_textviews_t1);
mtextviews.setText(arr.get(0).toString().trim()); 设置数据
```