## 一、概述

EventBus是一款针对**[Android](http://lib.csdn.net/base/android" \o "Android知识库" \t "http://blog.csdn.net/harvic880925/article/details/_blank)**优化的发布/订阅事件总线。主要功能是替代Intent,Handler,BroadCast在Fragment，Activity，Service，线程之间传递消息.优点是开销小，代码更优雅。以及将发送者和接收者解耦。  
1、下载EventBus的类库  
源码：[https://github.com/greenrobot/EventBus](https://github.com/greenrobot/EventBus" \t "http://blog.csdn.net/harvic880925/article/details/_blank)

2、基本使用

（1）自定义一个类，可以是空类，比如：

**[java]** [view plain](http://blog.csdn.net/harvic880925/article/details/40660137" \o "view plain" \t "http://blog.csdn.net/harvic880925/article/details/_blank) [copy](http://blog.csdn.net/harvic880925/article/details/40660137" \o "copy" \t "http://blog.csdn.net/harvic880925/article/details/_blank)

1. **public** **class** AnyEventType {
2. **public** AnyEventType(){}
3. }

（2）在要接收消息的页面注册：

**[java]** [view plain](http://blog.csdn.net/harvic880925/article/details/40660137" \o "view plain" \t "http://blog.csdn.net/harvic880925/article/details/_blank) [copy](http://blog.csdn.net/harvic880925/article/details/40660137" \o "copy" \t "http://blog.csdn.net/harvic880925/article/details/_blank)

1. eventBus.register(**this**);

（3）发送消息

**[java]** [view plain](http://blog.csdn.net/harvic880925/article/details/40660137" \o "view plain" \t "http://blog.csdn.net/harvic880925/article/details/_blank) [copy](http://blog.csdn.net/harvic880925/article/details/40660137" \o "copy" \t "http://blog.csdn.net/harvic880925/article/details/_blank)

1. eventBus.post(**new** AnyEventType event);

（4）接受消息的页面实现(共有四个函数，各功能不同，这是其中之一，可以选择性的实现，这里先实现一个)：

**[java]** [view plain](http://blog.csdn.net/harvic880925/article/details/40660137" \o "view plain" \t "http://blog.csdn.net/harvic880925/article/details/_blank) [copy](http://blog.csdn.net/harvic880925/article/details/40660137" \o "copy" \t "http://blog.csdn.net/harvic880925/article/details/_blank)

1. **public** **void** onEvent(AnyEventType event) {}

（5）解除注册

**[java]** [view plain](http://blog.csdn.net/harvic880925/article/details/40660137" \o "view plain" \t "http://blog.csdn.net/harvic880925/article/details/_blank) [copy](http://blog.csdn.net/harvic880925/article/details/40660137" \o "copy" \t "http://blog.csdn.net/harvic880925/article/details/_blank)

1. eventBus.unregister(**this**);

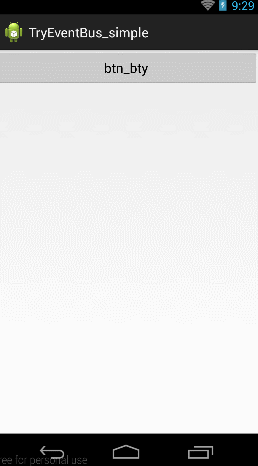
顺序就是这么个顺序，可真正让自己写，估计还是云里雾里的，下面举个例子来说明下。

首先，在EventBus中，获取实例的方法一般是采用EventBus.getInstance()来获取默认的EventBus实例，当然你也可以new一个又一个，个人感觉还是用默认的比较好，以防出错。

## 二、实战

先给大家看个例子：

当击btn\_try按钮的时候，跳到第二个Activity，当点击第二个activity上面的First Event按钮的时候向第一个Activity发送消息，当第一个Activity收到消息后，一方面将消息Toast显示，一方面放入textView中显示。



按照下面的步骤，下面来建这个工程：

### 1、基本框架搭建

想必大家从一个Activity跳转到第二个Activity的程序应该都会写，这里先稍稍把两个Activity跳转的代码建起来。后面再添加EventBus相关的玩意。

**MainActivity布局（activity\_main.xml）**

**[html]** [view plain](http://blog.csdn.net/harvic880925/article/details/40660137" \o "view plain" \t "http://blog.csdn.net/harvic880925/article/details/_blank) [copy](http://blog.csdn.net/harvic880925/article/details/40660137" \o "copy" \t "http://blog.csdn.net/harvic880925/article/details/_blank)

1. **<LinearLayout** xmlns:android="http://schemas.android.com/apk/res/android"
2. xmlns:tools="http://schemas.android.com/tools"
3. android:layout\_width="match\_parent"
4. android:layout\_height="match\_parent"
5. android:orientation="vertical"**>**
7. **<Button**
8. android:id="@+id/btn\_try"
9. android:layout\_width="match\_parent"
10. android:layout\_height="wrap\_content"
11. android:text="btn\_bty"**/>**
12. **<TextView**
13. android:id="@+id/tv"
14. android:layout\_width="wrap\_content"
15. android:layout\_height="match\_parent"**/>**
17. **</LinearLayout>**

**新建一个Activity，SecondActivity布局（activity\_second.xml）**

**[html]** [view plain](http://blog.csdn.net/harvic880925/article/details/40660137" \o "view plain" \t "http://blog.csdn.net/harvic880925/article/details/_blank) [copy](http://blog.csdn.net/harvic880925/article/details/40660137" \o "copy" \t "http://blog.csdn.net/harvic880925/article/details/_blank)

1. **<LinearLayout** xmlns:android="http://schemas.android.com/apk/res/android"
2. xmlns:tools="http://schemas.android.com/tools"
3. android:layout\_width="match\_parent"
4. android:layout\_height="match\_parent"
5. android:orientation="vertical"
6. tools:context="com.harvic.try\_eventbus\_1.SecondActivity" **>**
8. **<Button**
9. android:id="@+id/btn\_first\_event"
10. android:layout\_width="match\_parent"
11. android:layout\_height="wrap\_content"
12. android:text="First Event"**/>**
14. **</LinearLayout>**

**MainActivity.java （点击btn跳转到第二个Activity）**

**[java]** [view plain](http://blog.csdn.net/harvic880925/article/details/40660137" \o "view plain" \t "http://blog.csdn.net/harvic880925/article/details/_blank) [copy](http://blog.csdn.net/harvic880925/article/details/40660137" \o "copy" \t "http://blog.csdn.net/harvic880925/article/details/_blank)

1. **public** **class** MainActivity **extends** Activity {
3. Button btn;
5. @Override
6. **protected** **void** onCreate(Bundle savedInstanceState) {
7. **super**.onCreate(savedInstanceState);
8. setContentView(R.layout.activity\_main);
10. btn = (Button) findViewById(R.id.btn\_try);
12. btn.setOnClickListener(**new** View.OnClickListener() {
14. @Override
15. **public** **void** onClick(View v) {
16. // TODO Auto-generated method stub
17. Intent intent = **new** Intent(getApplicationContext(),
18. SecondActivity.**class**);
19. startActivity(intent);
20. }
21. });
22. }
24. }

到这，基本框架就搭完了，下面开始按步骤使用EventBus了。

### 2、新建一个类FirstEvent

**[java]** [view plain](http://blog.csdn.net/harvic880925/article/details/40660137" \o "view plain" \t "http://blog.csdn.net/harvic880925/article/details/_blank) [copy](http://blog.csdn.net/harvic880925/article/details/40660137" \o "copy" \t "http://blog.csdn.net/harvic880925/article/details/_blank)

1. **package** com.harvic.other;
3. **public** **class** FirstEvent {
5. **private** String mMsg;
6. **public** FirstEvent(String msg) {
7. // TODO Auto-generated constructor stub
8. mMsg = msg;
9. }
10. **public** String getMsg(){
11. **return** mMsg;
12. }
13. }

这个类很简单，构造时传进去一个字符串，然后可以通过getMsg()获取出来。

### 3、在要接收消息的页面注册EventBus：

在上面的GIF图片的演示中，大家也可以看到，我们是要在MainActivity中接收发过来的消息的，所以我们在MainActivity中注册消息。

通过我们会在OnCreate()函数中注册EventBus，在OnDestroy（）函数中反注册。所以整体的注册与反注册的代码如下：

**[java]** [view plain](http://blog.csdn.net/harvic880925/article/details/40660137" \o "view plain" \t "http://blog.csdn.net/harvic880925/article/details/_blank) [copy](http://blog.csdn.net/harvic880925/article/details/40660137" \o "copy" \t "http://blog.csdn.net/harvic880925/article/details/_blank)

1. **package** com.example.tryeventbus\_simple;
3. **import** com.harvic.other.FirstEvent;
5. **import** de.greenrobot.event.EventBus;
6. **import** android.app.Activity;
7. **import** android.content.Intent;
8. **import** android.os.Bundle;
9. **import** android.util.Log;
10. **import** android.view.View;
11. **import** android.widget.Button;
12. **import** android.widget.TextView;
13. **import** android.widget.Toast;
15. **public** **class** MainActivity **extends** Activity {
17. Button btn;
18. TextView tv;
20. @Override
21. **protected** **void** onCreate(Bundle savedInstanceState) {
22. **super**.onCreate(savedInstanceState);
23. setContentView(R.layout.activity\_main);
24. //注册EventBus
25. EventBus.getDefault().register(**this**);
27. btn = (Button) findViewById(R.id.btn\_try);
28. tv = (TextView)findViewById(R.id.tv);
30. btn.setOnClickListener(**new** View.OnClickListener() {
32. @Override
33. **public** **void** onClick(View v) {
34. // TODO Auto-generated method stub
35. Intent intent = **new** Intent(getApplicationContext(),
36. SecondActivity.**class**);
37. startActivity(intent);
38. }
39. });
40. }
41. @Override
42. **protected** **void** onDestroy(){
43. **super**.onDestroy();
44. EventBus.getDefault().unregister(**this**);//反注册EventBus
45. }
46. }

### 4、发送消息

发送消息是使用EventBus中的Post方法来实现发送的，发送过去的是我们新建的类的实例！

**[java]** [view plain](http://blog.csdn.net/harvic880925/article/details/40660137" \o "view plain" \t "http://blog.csdn.net/harvic880925/article/details/_blank) [copy](http://blog.csdn.net/harvic880925/article/details/40660137" \o "copy" \t "http://blog.csdn.net/harvic880925/article/details/_blank)

1. EventBus.getDefault().post(**new** FirstEvent("FirstEvent btn clicked"));

完整的SecondActivity.**[Java](http://lib.csdn.net/base/java" \o "Java 知识库" \t "http://blog.csdn.net/harvic880925/article/details/_blank)**的代码如下：

**[java]** [view plain](http://blog.csdn.net/harvic880925/article/details/40660137" \o "view plain" \t "http://blog.csdn.net/harvic880925/article/details/_blank) [copy](http://blog.csdn.net/harvic880925/article/details/40660137" \o "copy" \t "http://blog.csdn.net/harvic880925/article/details/_blank)

1. **package** com.example.tryeventbus\_simple;
3. **import** com.harvic.other.FirstEvent;
5. **import** de.greenrobot.event.EventBus;
6. **import** android.app.Activity;
7. **import** android.os.Bundle;
8. **import** android.view.View;
9. **import** android.widget.Button;
11. **public** **class** SecondActivity **extends** Activity {
12. **private** Button btn\_FirstEvent;
14. @Override
15. **protected** **void** onCreate(Bundle savedInstanceState) {
16. **super**.onCreate(savedInstanceState);
17. setContentView(R.layout.activity\_second);
18. btn\_FirstEvent = (Button) findViewById(R.id.btn\_first\_event);
20. btn\_FirstEvent.setOnClickListener(**new** View.OnClickListener() {
22. @Override
23. **public** **void** onClick(View v) {
24. // TODO Auto-generated method stub
25. EventBus.getDefault().post(
26. **new** FirstEvent("FirstEvent btn clicked"));
27. }
28. });
29. }
30. }

### 5、接收消息

接收消息时，我们使用EventBus中最常用的onEventMainThread（）函数来接收消息，具体为什么用这个，我们下篇再讲，这里先给大家一个初步认识，要先能把EventBus用起来先。

在MainActivity中重写onEventMainThread（FirstEvent event），参数就是我们自己定义的类：

在收到Event实例后，我们将其中携带的消息取出，一方面Toast出去，一方面传到TextView中；

**[java]** [view plain](http://blog.csdn.net/harvic880925/article/details/40660137" \o "view plain" \t "http://blog.csdn.net/harvic880925/article/details/_blank) [copy](http://blog.csdn.net/harvic880925/article/details/40660137" \o "copy" \t "http://blog.csdn.net/harvic880925/article/details/_blank)

1. **public** **void** onEventMainThread(FirstEvent event) {
3. String msg = "onEventMainThread收到了消息：" + event.getMsg();
4. Log.d("harvic", msg);
5. tv.setText(msg);
6. Toast.makeText(**this**, msg, Toast.LENGTH\_LONG).show();
7. }

完整的MainActiviy代码如下：

**[java]** [view plain](http://blog.csdn.net/harvic880925/article/details/40660137" \o "view plain" \t "http://blog.csdn.net/harvic880925/article/details/_blank) [copy](http://blog.csdn.net/harvic880925/article/details/40660137" \o "copy" \t "http://blog.csdn.net/harvic880925/article/details/_blank)

1. **package** com.example.tryeventbus\_simple;
3. **import** com.harvic.other.FirstEvent;
5. **import** de.greenrobot.event.EventBus;
6. **import** android.app.Activity;
7. **import** android.content.Intent;
8. **import** android.os.Bundle;
9. **import** android.util.Log;
10. **import** android.view.View;
11. **import** android.widget.Button;
12. **import** android.widget.TextView;
13. **import** android.widget.Toast;
15. **public** **class** MainActivity **extends** Activity {
17. Button btn;
18. TextView tv;
20. @Override
21. **protected** **void** onCreate(Bundle savedInstanceState) {
22. **super**.onCreate(savedInstanceState);
23. setContentView(R.layout.activity\_main);
25. EventBus.getDefault().register(**this**);
27. btn = (Button) findViewById(R.id.btn\_try);
28. tv = (TextView)findViewById(R.id.tv);
30. btn.setOnClickListener(**new** View.OnClickListener() {
32. @Override
33. **public** **void** onClick(View v) {
34. // TODO Auto-generated method stub
35. Intent intent = **new** Intent(getApplicationContext(),
36. SecondActivity.**class**);
37. startActivity(intent);
38. }
39. });
40. }
42. **public** **void** onEventMainThread(FirstEvent event) {
44. String msg = "onEventMainThread收到了消息：" + event.getMsg();
45. Log.d("harvic", msg);
46. tv.setText(msg);
47. Toast.makeText(**this**, msg, Toast.LENGTH\_LONG).show();
48. }
50. @Override
51. **protected** **void** onDestroy(){
52. **super**.onDestroy();
53. EventBus.getDefault().unregister(**this**);
54. }
55. }