EventBus是一款针对Android优化的发布/订阅（publish/subscribe）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。

简化了应用程序内各组件间、组件与后台线程间的通信。

优点是开销小，代码更优雅。以及将发送者和接收者解耦。比如请求网络，等网络返回时通过Handler或Broadcast通知UI，两个Fragment之间需要通过Listener通信，这些需求都可以通过EventBus实现。

使用

 onEvent：事件的处理在和事件的发送在相同的线程，所以事件处理时间不应太长，不然影响事件的发送线程。

 onEventMainThread: 事件的处理会在UI线程中执行。事件处理时间不能太长，长了会出现臭名远之的ANR。

 onEventBackgroundThread：事件的处理会在一个后台线程中执行。虽然名字是BackgroundThread，事件处理是在后台线程，但事件处理时间还是不应该太长，因为如果发送事件的线程是后台线程，会直接在当前后台线程执行事件；如果当前线程是UI线程，事件会被加到一个队列中，由一个线程依次处理这些事件，如果某个事件处理时间太长，会阻塞后面的事件的派发或处理。

 onEventAsync：事件处理会在单独的线程中执行，主要用于在后台线程中执行耗时操作，每个事件会开启一个线程，但最好限制线程的数目



注册,解绑,处理事件,发送EventBus.getDefault().post("test!");

源码级

EventBus是一个观察者模式的实现，三要素

* Event：事件
* Subscriber：事件订阅者，接收特定的事件。
* Publisher:事件发布者，用于通知Subscriber有事件发生。

<http://p.codekk.com/blogs/detail/54cfab086c4761e5001b2538>

基本类的概念和重要类的作用,记住几个重要的就OK

<http://www.jcodecraeer.com/a/anzhuokaifa/androidkaifa/2016/0417/4152.html>

详细用法,写的搅乱