How to get started with EventBus in 3 steps

The EventBus API is as easy as 1-2-3.

Before we get started make sure to add EventBus as a dependency to your project.

Contents [hide]

Step 1: Define events

Events are POJO (plain old Java object) without any specific requirements.

Java

```java
public class MessageEvent {

    public final String message;

    public MessageEvent(String message) {
        this.message = message;
    }
}
```

Step 2: Prepare subscribers

Subscribers implement event handling methods (also called “subscriber methods”) that will be called when an event is posted. These are defined with the **@Subscribe** annotation.
Note that with EventBus 3 the method name can be chosen freely (no naming conventions like in EventBus 2).

Java

```java
// This method will be called when a MessageEvent is posted (in the UI thread for Toast)
@Subscribe(threadMode = ThreadMode.MAIN)
public void onMessageEvent(MessageEvent event) {
    Toast.makeText(getActivity(), event.message, Toast.LENGTH_SHORT).show();
}

// This method will be called when a SomeOtherEvent is posted
@Subscribe
public void handleSomethingElse(SomeOtherEvent event) {
    doSomethingWith(event);
}
```

Subscribers also need to **register** themselves to **and unregister** from the bus. Only while subscribers are registered, they will receive events. In Android, in activities and fragments you should usually register according to their life cycle. For most cases onStart/onStop works fine:

```java
@Override
public void onStart() {
    super.onStart();
    EventBus.getDefault().register(this);
}

@Override
public void onStop() {
    EventBus.getDefault().unregister(this);
    super.onStop();
}
```

Step 3: Post events

Post an event from any part of your code. All currently registered subscribers matching the event type will receive it.

```java
EventBus.getDefault().post(new MessageEvent("Hello everyone!"));
```

Learn more

Have a look at the full documentation to learn about all features of EventBus.

ObjectBox is a Techstars'17 company

ObjectBox 0.9.10 – getting closer to 1.0

ObjectBox Documentation Update

Recent Comments

William Ferguson on EventBus 3.1 with plain Java support

DaoMaster on Introducing DaoCompat: greenDAO on ObjectBox

Morten Slott Hansen on Introducing DaoCompat: greenDAO on ObjectBox

Andreas Neumann on Custom Types

DaoMaster on Custom Types

Tags

ActiveAndroid Active Entities Annotations Benchmark beta DaoCompat Data Listeners DbFlow Documentation Encryption EventBus GDG Gradle greenDAO greenrobot-common index JavaDocs JDT main thread mobile database Multithreading NoSQL ObjectBox Open Source ORM OrmLite perfmatters Performance plain Java Presentation Query QueryBuilder Reactive Reactive Observers Reactive Programming Reddit Reflection Release Requery Slides SQLCipher SQLite Techstars thread mode To-Many