

RxJava

RxJava:

1, Observables: 被观察者, 事件源

2, Subscribers: 观察者

3, 操作符

map: 把一个事件转换为另一个事件
类型转换

flatMap: 接受一个Observable的输出作为输入, 同时输出另一个Observable, 新的Observable是Subscriber所接收到的。

filter: 对数据进行过滤

take: 固定返回数量

创建Observable:

1, 可通过Create () 创建。

2, 已有的数据结构创建。

你可以使用 `just()` 和 `from()` 方法将对象, 列表, 对象属性转换为发射那些对象的Observable:

```
Observable<String> o = Observable.from("a", "b", "c");

def list = [5, 6, 7, 8]
Observable<Integer> o = Observable.from(list);

Observable<String> o = Observable.just("one object");
```

同步和异步的Observable ();

操作符:

1, skip () 跳过开始几个。

2, take () 固定显示数量。

3, Merge () 组合两个Observable。