**[从零开始的RxJava2.0教程(四)在安卓中的应用](http://blog.csdn.net/qq_35064774/article/details/53065400)**

# 2. RxAndroid

RxAndroid是RxJava针对**[Android](http://lib.csdn.net/base/android" \o "Android知识库" \t "http://blog.csdn.net/qq_35064774/article/details/_blank)**平台的扩展库。

曾几何时，RxAndroid确实提供了很多的实用的方法，但后来社区上很多人对这库的结构有意见，然后作者便进行了重构，现在只保留了AndroidSchedulers， 现在基本RxAndroid只有一个功能，那就是AndroidSchedulers.mainThread 方法用于指定主线程。

以前那些类被划分到了其他一些Rx库，比如 RxBinding, RxLifecycle。这里我并不打算介绍，因为对于新手来说，为时太早，如果你有兴趣，可以自己查阅相关资料。

# 3. 配合 Retrofit 2

目前来说，Android的网络库基本被 Retrofit + OkHttp 一统天下了。因为它确实很优秀。

当然，RxJava的影响力也不容小觑，Retrofit 也提供了对RxJava 的支持，截止到目前(2016.11.06)为止，官方似乎还没有支持RxJava 2.0。不过你不用担心，已经有大神帮我们做好了。

要使用 Retrofit 需要引入对应的库。

compile 'com.squareup.okhttp3:okhttp:3.4.1'

compile 'com.squareup.okio:okio:1.10.0'

compile 'com.squareup.retrofit2:retrofit:2.1.0'

compile 'com.jakewharton.retrofit:retrofit2-rxjava2-adapter:1.0.0'

retrofit2 依赖于 okhttp3，而 okhttp3 又依赖于okio。最后一项便是大神写好的 retrofit2-rxjava2-adapter 适配器啦。

假设我们现在需要去获取百度页面的html代码。

public interface BaiDuService {

@GET("/")

Flowable<ResponseBody> getText();

}

定义好请求接口。注意，返回值是 Flowable 类型。

Retrofit retrofit = new Retrofit.Builder()

.baseUrl("http://www.baidu.com/")

.addCallAdapterFactory(RxJava2CallAdapterFactory.create())// 添加RxJava2的适配器支持

.build();

BaiDuService service = retrofit.create(BaiDuService.class);

service.getText()

.subscribeOn(Schedulers.newThread())

.observeOn(AndroidSchedulers.mainThread())

.subscribe(new Subscriber<ResponseBody>() {

@Override

public void onSubscribe(Subscription s) {

s.request(Long.MAX\_VALUE);

}

@Override

public void onNext(ResponseBody s) {

Toast.makeText(RxJava2Activity.this, "获取成功", Toast.LENGTH\_SHORT).show();

try {

System.out.println(s.string());

} catch (IOException e) {

e.printStackTrace();

}

}

@Override

public void onError(Throwable t) {

t.printStackTrace();

Toast.makeText(RxJava2Activity.this, "获取失败，请检查网络是否畅通", Toast.LENGTH\_SHORT).show();

}

@Override

public void onComplete() {

System.out.println("任务结束");

}

});

在创建 Retrofit 的时候添加对 RxJava 2 的适配器，这样，请求就可以直接返回 Flowable。 然后就可以进行 RxJava 2 的操作了。

# 4. 从RxJava 2中获取数据

前面我介绍了可以通过 fromX 的方法把数组、列表等数据发射出去。那么有没有办法直接把发射的数据获取出来而不是通过订阅者来输出呢？

List<String> ids = Flowable.range(1, 100)

.map(new Function<Integer, String>() {

@Override

public String apply(Integer integer) throws Exception {

return "id:" + integer;

}

})

.toList().blockingGet();

System.out.println(ids);

比如这里，我发射了1到100总计100个数据，我们可以通过 blockingX 方法来拿到这些数据。