6. Rx Java

개요

- 동시성을 적극적으로 끌어안을 필요가 있다. (Embrace Concurrency.)
- 자바 Future를 조합하기 어렵다는 점을 해결해야 한다. (Java Futures are Expensive to Compose.)
- 콜백 방식의 문제점을 개선해야 한다. (Callbacks Have Their Own Problems.)

참고 자료

- 2013/2 netflix 기술 블로그에서 시작함.
  - https://medium.com/netflix-techblog/reactive-programming-in-the-netflix-api-with-rxjava-7811c3a1496a
- 커니의 코틀린 (책 & 블로그) - Rx Java 2.x 버전에 대한 설명이므로 현재 3.x 버전이 나와서 자세한 내용은 github 내용을 참조해야 함.
  - https://www.androidhuman.com/tag/kotlin/
  - 1.x와 2.x는 구조 자체가 달라졌으므로 2.x 버전 이상으로 학습해야함.
- http://reactivex.io/
- https://github.com/ReactiveX/RxJava
- Ap2 기완선임님 : ReactiveX
- Rx Java Scheduler Thread Worker 적용
  - https://blog.hansoolabs.com/676

정의

- Reactive Extensions란.
  - ReactiveX라고 부르며, 이벤트 기반 비동기 프로그래밍.
- 기존 callback 방식과 다른점?
  - 기존 callback 방식
    - 매 이벤트마다 그에 대응하는 동작 정의.
  - Reactive Extensions
    - 이벤트를 이벤트 스트림에 전달하고, 이벤트 스트림을 관찰하다가 원하는 이벤트 감지하면 이에 따른 동작 수행.

장점

- 비동기 이벤트 처리를 쉽게 할 수 있다.
  - 기존 안드로이드에서 비동기 작업은 AsyncTask를 사용해야한다.
    - 사용법 : https://developer.android.com/reference/android/os/AsyncTask
  - 작성하는 코드량을 줄일 수 있다.
- 이벤트나 데이터를 쉽게 가공 및 분배할 수 있다.
  - 데이터 형태변환이나 이벤트 무시에 대해서 원본 이벤트 스트림에서 특정 이벤트만 받는 스트림 방식으로 다양한 조건에 대응하게 구현할 수 있다.

용어

- Observable
  - 이벤트를 만드는 주체로 이벤트 스트림을 통해 만든 이벤트를 내보낸다. (emit)
  - 0 ~ N개까지의 이벤트를 만들 수 있다.
  - 특정 조건으로 종료할 수 있고, 계속 유지하여 이벤트를 만들 수 있다.
- Observer
  - Observable에서만든 이벤트에 반응한다. (react)
  - 이벤트를 받았을 때 수행할 작업을 정의한다.
  - Observable에서 발행한 이벤트를 Observer가 구독해야 이벤트를 받을 수 있다. (subscribe)
- Operators
  - 이벤트 스트림을 통해 전달되는 이벤트를 변환한다.

    - 특정 조건을 만족하는 이벤트만 이벤트 스트림에 적용하거나, 이벤트 개수를 바꿔줄 수 있는 작업들을 할 수 있다.
    - 중첩하여 사용가능하다.
- Scheduler
    - 작업을 수행할 Thread를 지정한다.
    - I/O 작업에 수행할 Thread, 계산에 수행할 Thread, UI 업데이트를 위한 Main Thread등을 지정할 수 있다.
    - 예제) main Thread 지정

▪ subscribe

```
.subscribeOn(Schedulers.io())
.observeOn(AndroidSchedulers.mainThread())
```

main Thread에 해당하는 스케줄러는 RxAndroid에 포함되어 있으므로 RxAndroid를 build.gradle에 의존성 추가 해야함.

```
implementation 'io.reactivex.rxjava2:rxandroid:2.1.0'
```

```
github : https://github.com/ReactiveX/RxAndroid
```

- Disposable
    - Observer가 Observable을 구독할 때 생성되는 객체
    - Observable에서 만드는 이벤트 스트림과 필요한 리소스를 관리한다.
    - 구독해제를 수행할 수 있다. (unsubscribe)
    - 여러 개의 Disposable를 관리할 수 있는 객체가 있다. (CompositeDisposable)
    - 생명주기에 따라 리소스를 관리해야하는 Activity, Fragment에서 CompositeDisposable을 자주 사용한다.

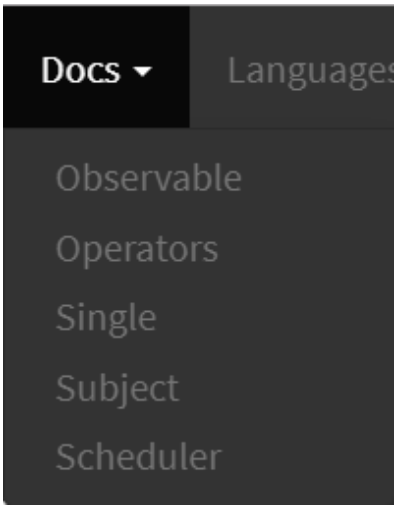
- Thread Worker
- 예제 소스

▪ 예제

```
class SampleRepositoryActivity : AppCompatActivity(){

    ...

    //       CompositeDisposable   .
    interanl val disposables = CompositeDisposable()
    ...
    override fun onStop(){
        super.onStop()
        //      .
        dislosables.clear()
    }
    private fun showSampleRepositoryInfo(data: String){

        //REST API    .
        disposables.add(api.getSampleRepository(data)

            // REST API      .
            .map { ... }

            // Observable     flatMap .
            .flatMap {
                if ( 0 == it.cnt) {
                    //      .
                    // (   .)
                    Observable.error(IllegalStateException
("No search result"))
                }else {
                    //      .
                    Observable.just(it.items)
                }
            }
```

```
                //       AndroidSchedulers.mainThread() .
                .observeOn(AndroidSchedulers.mainThread())

                //     .
                .doOnSubscribe {
                    ...
                }

                //       .
                .doOnTerminate {
                    ...
                }

                //  .
                .subscribe({
                        //        .
                        ...
                }) {
                        //
                        ...
                }
        }
}


: https://faith-developer.tistory.com/22 [ ]
```

Rx Docs



- Observable
- Operators
- Single
- Subject
- Scheduler

Single

- http://reactivex.io/documentation/ko/single.html

예제 소스

- Retrofit Library & RxJava 적용 (HTTP 통신 예제)
  - https://github.com/kunny/kunny-kotlin-book/tree/rxjava/simple-github/src/main
  - Retrofit Library 응답을 observable 형태로 변환해주는 adpater-rxjava2 Libary 추가해줘야함.