

# Rx Java Tutorial

Android

# Introduction

- RxJava – Reactive Extensions for the JVM – a library for composing asynchronous and event-based programs using observable sequences for the Java VM.
- Source Code at  
<https://github.com/ReactiveX/RxJava>  
<https://github.com/ReactiveX/RxAndroid>
- RxJava has become very important because of LiveData which is Android's lifecycle aware version of RxJava.
- RxAndroid is the Android version of RxJava. It provides additional API which is android specific

# Implementation

- Add the gradle dependency:-  
implementation "io.reactivex.rxjava2:rxjava:2.x.y" - x, y vary based on latest version  
implementation "io.reactivex.rxjava2:rxandroid:2.0.2"
- We will be using a recycler view to demonstrate this tutorial.
- In this tutorial, we will demonstrate the use of Observable.fromArray() API
- The subscriber is on Android main/UI thread and the observable is on background thread.

# Code Snippet

```
protected void rxJavaTutorial(){
    //Say this is the data from storage/network
    Observable.fromArray("Lollipop","Marshmallow","Nougat", "Oreo")
        .subscribeOn(Schedulers.newThread())
        .observeOn(AndroidSchedulers.mainThread())
        .subscribe(new Observer<String>() {
            @Override
            public void onSubscribe(Disposable d) {
                Log.d(TAG, msg: "Subscribed");
            }

            @Override
            public void onNext(String s) {
                mDataSet.add(s);
                myAdapter.notifyDataSetChanged();
                Log.d(TAG, msg: "Added: "+ s);
            }

            @Override
            public void onError(Throwable e) {
                Log.d(TAG, msg: "Error");
            }

            @Override
            public void onComplete() {
                myAdapter.notifyDataSetChanged();
                Log.d(TAG, msg: "Completed");
            }
        });
}
```

# Exercise

- What happens if `onStop()` of `MainActivity` is called when the `onNext` was called.
- Try `LiveData`
- Try operators
- Try other types of observables.