Applying Key Operators in the Observable Class: Case Study ex4 (Part 1)

Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

Professor of Computer Science

Institute for Software Integrated Systems

Vanderbilt University
Nashville, Tennessee, USA
Learning Objectives in this Part of the Lesson

• Part 1 of case study ex4 applies Observable operators fromArray(), toFlowable(), map(), subscribeOn(), flatMap(), subscribe(), & a thread pool to create, multiply, & display BigFraction objects asynchronously

```
Single
   .fromCallable(() ->
       .makeBigFraction(sRANDOM, true))
   .flatMapObservable(bf1 ->
       Observable
       .fromArray(bigFractionArray)
   .flatMap(bf2 -> Observable
       .fromCallable(() -> bf2)
       .subscribeOn(scheduler)
       .map(__ -> bf2
           .multiply(bf1)))
   .toFlowable(LATEST)
   .subscribe(blockingSubscriber);
```

Learning Objectives in this Part of the Lesson

- Part 1 of case study ex4 explores Observable operators fromArray(), toFlowable(), map(), subscribeOn(), flatMap(), the Flowable operator subscribe(), & the computation() thread pool to asynchronously create, multiply, & display BigFraction objects.
- It also shows Single operators subscribeOn(), fromCallable(), flatMapObservable(), & flatMapCompletable().

```java
Single
  .fromCallable(() ->
    .makeBigFraction(sRANDOM, true))
  .flatMapObservable(bf1 ->
    Observable
      .fromArray(bigFractionArray)
      .flatMap(bf2 -> Observable
        .fromCallable(() -> bf2)
        .subscribeOn(scheduler)
        .map(__ -> bf2.multiply(bf1))))
  .toFlowable(LATEST)
  .subscribe(blockingSubscriber);
```

See [github.com/douglas craig schmidt/LiveLessons/tree/master/Reactive/Observable/ex4](github.com/douglas craig schmidt/LiveLessons/tree/master/Reactive/Observable/ex4)
Learning Objectives in this Part of the Lesson

- Part 1 of case study ex4 explores Observable operators fromArray(), toFlowable(), map(), subscribeOn(), flatMap(), the Flowable operator subscribe(), & the computation() thread pool to asynchronously create, multiply, & display BigFraction objects.

- It also shows Single operators subscribeOn(), fromCallable(), flatMapObservable(), & flatMapCompletable().

- In addition, it shows how to create & use a generic blocking Subscriber.

```java
class BlockingSubscriber<T>
    implements Subscriber<T> {
    ...
    final CountDownLatch mLatch;
    ...
    @Override
    public void onComplete() {
        ...
        mLatch.countDown();
    }
    ...
}
```

See github.com/douglascraigschmidt/LiveLessons/tree/master/Reactive/Observable/ex4
Applying Key Operators in the Observable Class to ex4
Applying Key Operators in the Observable Class to ex4

/*
 * This class shows how to reduce and/or multiply big fractions
 * asynchronously and concurrently using many advanced RxJava
 * Observable operations, including fromIterable(), map(), generate(),
 * take(), flatMap(), flatMapCompletable(), fromCallable(), filter(),
 * reduce(), collectInto(), subscribeOn(), onErrorReturn(), and
 * Schedulers.computation(). It also shows advanced RxJava Single and
 * Maybe operations, such as ambArray(), subscribeOn(), and
 * doOnSuccess().
 */

public class ObservableEx {
  /**
   * Create a random number generator.
   */
  private static final Random sRANDOM = new Random();

  /**
   * Use an asynchronous Observable stream and a pool of threads to
   * showcase exception handling of BigFraction objects.
   */

  public static Completable testFractionExceptions() {
    StringBuffer sb =
End of Applying Key Methods in the Observable Class: Case Study ex4 (Part 1)