Applying Key Operators in the Flux Class: Case Study ex1

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

- Case study ex1 shows how to use Flux operators just(), fromIterable(), fromArray(), from(), doOnNext(), map(), mapNotNull(), mergeWith(), repeat(), & subscribe() to create, reduce, multiply, & display BigFraction objects synchronously

```
Flux
   .just(BigFraction.valueOf(100,3),
         BigFraction.valueOf(100,4),
         BigFraction.valueOf(100,2),
         BigFraction.valueOf(100,1))

   .map(fraction -> fraction
      .multiply(sBigReducedFraction))

   .subscribe
      (fraction -> sb.append(" = "
         + fraction.toMixedString()
         + "\n"),
       error -> sb.append("error"),
       () -> BigFractionUtils
        .display(sb.toString()));
```

See github.com/douglascraigschmidt/LiveLessons/tree/master/Reactive/flux/ex1

Learning Objectives in this Part of the Lesson

- Case study ex1 shows how to use Flux operators just(), fromIterable(), fromArray(), from(), doOnNext(), map(), mapNotNull(), mergeWith(), repeat(), & subscribe() to create, reduce, multiply, & display BigFraction objects synchronously
  - It also shows how to use the Mono fromCallable() operator

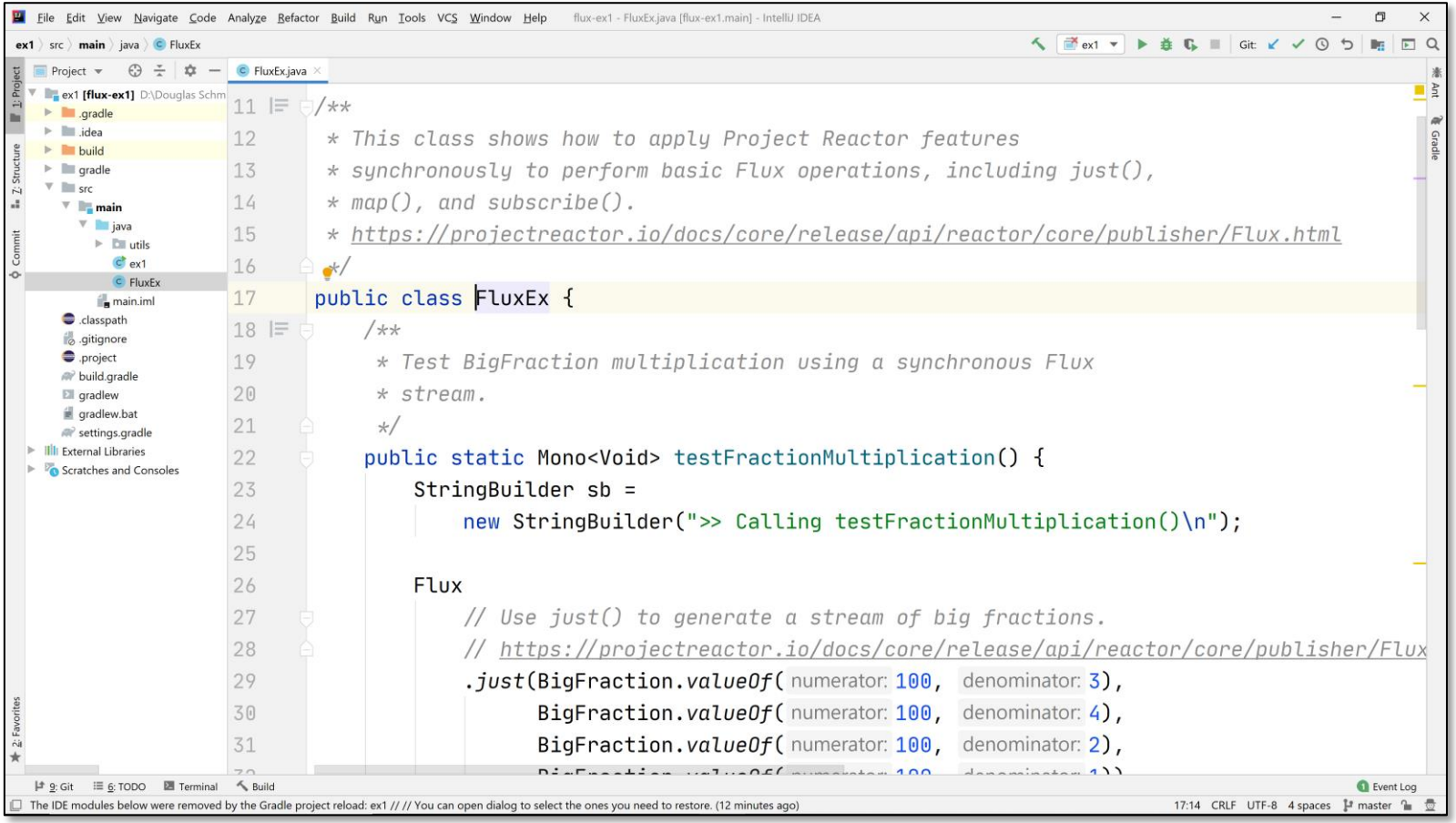
```
Flux<BigFraction> f2 = Flux
    .from(Mono
          .fromCallable(() ->
             makeBigFraction
              (random, true)))

    .repeat(4);
...
```

See github.com/douglascraigschmidt/LiveLessons/tree/master/Reactive/flux/ex1

Applying Key Operators in the Flux Class to ex1

Applying Key Operators in the Flux Class to ex1



See github.com/douglascraigschmidt/LiveLessons/tree/master/Reactive/flux/ex1

End of Applying Key Operators in the Flux Class: Case Study ex1