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

```java
.justin(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.toString() + "\n"),
    error -> sb.append("error"),
    () -> BigFractionUtils.display(sb.toString()));
```

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

\[
\text{Flux<BigFraction> f2 = Flux}
  \text{.from(Mono}
  \text{.fromCallable(() ->
      makeBigFraction
        (random, true)))}
  \text{.repeat(4);}
\]

- It also shows how to use the Mono fromCallable() operator

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

```java
public class FluxEx {

    public static Mono<Void> testFractionMultiplication() {
        StringBuilder sb = new StringBuilder("\n\n        // Use just() to generate a stream of big fractions.
        // https://projectreactor.io/docs/core/release/api/reactor/core/publisher/Flux.html
        Flux.
        
        * This class shows how to apply Project Reactor features
        * synchronously to perform basic Flux operations, including just(),
        * map(), and subscribe().
        * https://projectreactor.io/docs/core/release/api/reactor/core/publisher/Flux.html
        */

        return Flux.just(BigFraction.valueOf(numerator: 100, denominator: 3),
                         BigFraction.valueOf(numerator: 100, denominator: 4),
                         BigFraction.valueOf(numerator: 100, denominator: 2),
                         BigFraction.valueOf(numerator: 100, denominator: 1));
    }

    public static void main(String[] args) {
        testFractionMultiplication().subscribe(System.out::println);
    }
}
```

See github.com/douglascraigschmidt/LiveLessons/tree/master/Reactive/flux/ex1
End of Applying Key Operators in the Flux Class: Case Study ex1