Applying Key Methods in the Mono 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

- Recognize key operators in the Mono class & know how they are applied in a detailed case study.

See projectreactor.io/docs/core/release/api/reactor/core/publisher/Mono.html
Learning Objectives in this Part of the Lesson

- Recognize key operators in the Mono class & know how they are applied in a detailed case study
- Case study ex1 shows how to apply the just(), fromCallable(), map(), doOnSuccess(), & then() operators to create, reduce, transform, and display a Big Fraction synchronously

```java
return Mono
    .fromCallable(() -> BigFraction
        .reduce(sUnreducedFrac))
    .doOnSuccess(bf ->
        logBigFraction
        (sUnreducedFrac, bf, sb))
    .map(BigFraction::toMixedString)
    .doOnSuccess(bf ->
        displayMixedBigFraction(bf, sb))
    .then();
```

See [github.com/douglascraigschmidt/LiveLessons/tree/master/Reactive/mono/ex1](https://github.com/douglascraigschmidt/LiveLessons/tree/master/Reactive/mono/ex1)
Applying Key Methods in the Mono Class to ex1
Applying Key Methods in the Mono Class to ex1

```java
public class MonoEx {

    /**
     * Test synchronous BigFraction reduction using a mono and a pipeline of operations that run on the calling thread.
     */

    public static Mono<Void> testFractionReductionSync1() {
        StringBuilder sb =
            new StringBuilder(">> Calling testFractionReductionSync()\n");

        // Create a new unreduced big fraction.
        BigFraction unreducedFraction =
            BigFraction.valueOf(new BigInteger(sBI1),
                               new BigInteger(sBI2),
                               reduce: false);

        return Mono
```
End of Applying Key Methods in the Mono Class: Case Study ex1