Applying Key Operators in the Flowable Class: Case Study ex2

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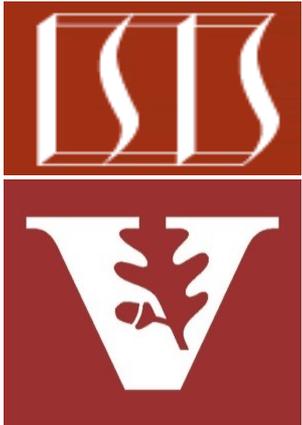d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

Professor of Computer Science

Institute for Software
Integrated Systems

Vanderbilt University
Nashville, Tennessee, USA

- Case study ex2 applies RxJava Flowable features to demonstrate how a Subscriber running in one Scheduler context can exert flow-control on a backpressure-aware Publisher that runs in a different Scheduler context
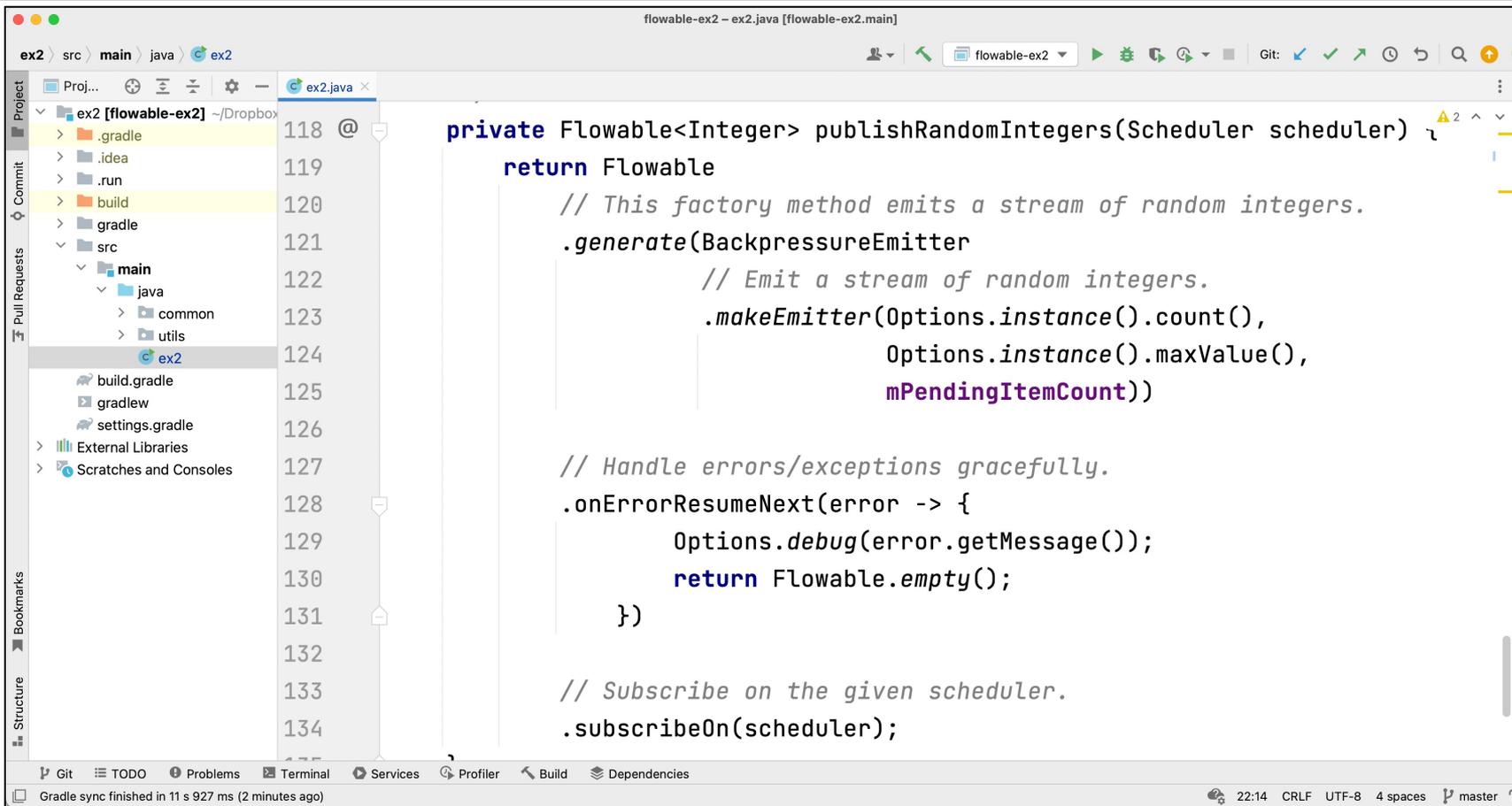
```
return Flowable
    .generate(BackpressureEmitter
        .makeEmitter(count,
                     maxValue,
                 mPendingItemCount)),

    .onErrorResumeNext(error -> {
        debug(error.getMessage());
        return Flowable.empty();
    })

    .subscribeOn(scheduler);
```

Applying Key Operators in the Flowable Class to ex2

Applying Key Operators in the Flowable Class to ex2


```java
private Flowable<Integer> publishRandomIntegers(Scheduler scheduler)
    return Flowable
        // This factory method emits a stream of random integers.
        .generate(BackpressureEmitter
            // Emit a stream of random integers.
            .makeEmitter(Options.instance().count(),
                Options.instance().maxValue(),
                mPendingItemCount))

        // Handle errors/exceptions gracefully.
        .onErrorResumeNext(error -> {
            Options.debug(error.getMessage());
            return Flowable.empty();
        })

        // Subscribe on the given scheduler.
        .subscribeOn(scheduler);
```

See github.com/douglascraigschmidt/LiveLessons/tree/master/Reactive/Flowable/ex2

End of Applying Key Operators in the Flowable Class: Case Study ex2