

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**



Learning Objectives in this Part of the Lesson

- 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

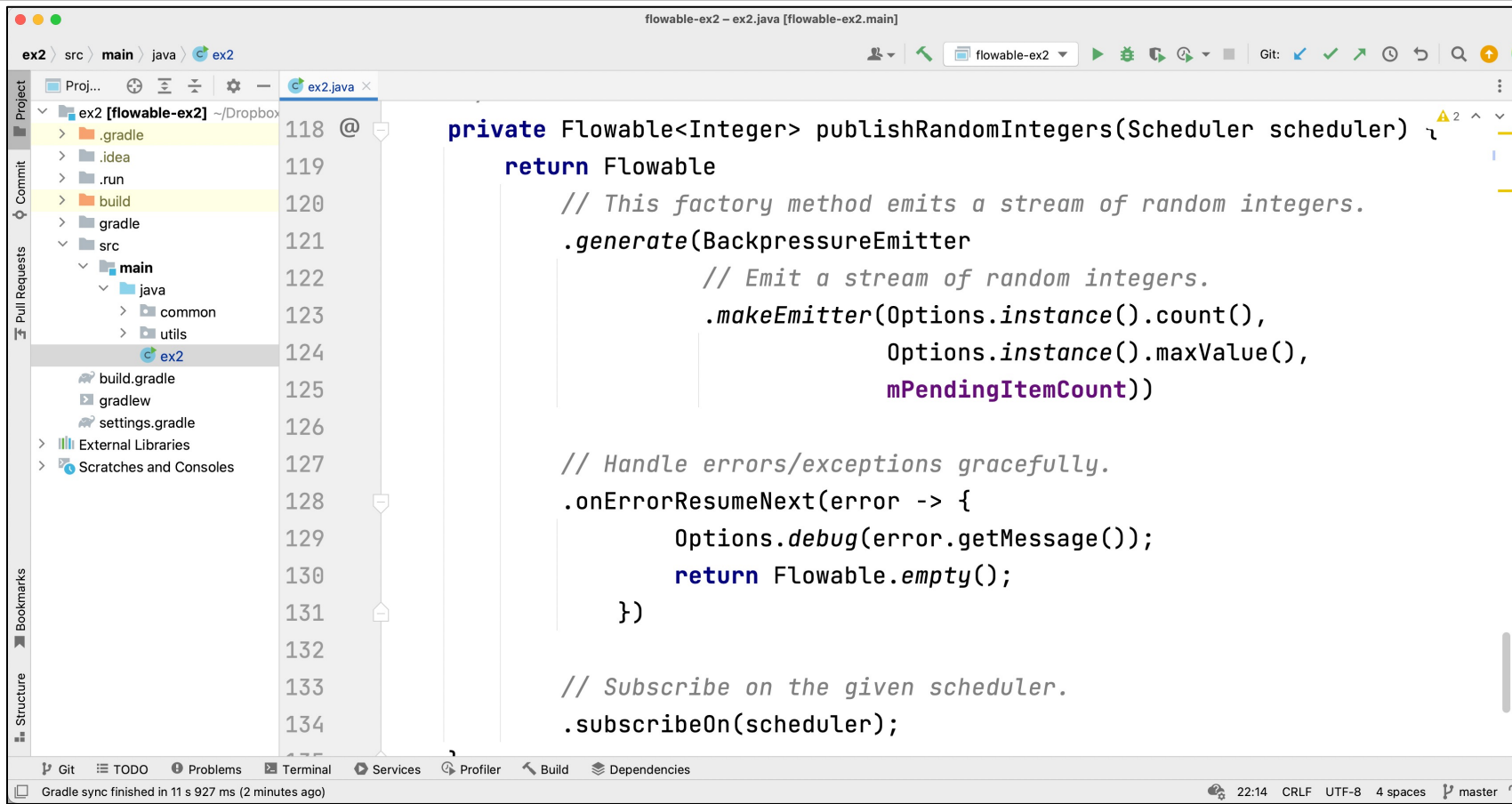
```
return Flowable
    .generate(BackpressureEmitter
        .makeEmitter(count,
            maxValue,
            mPendingItemCount)) ,

    .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



The screenshot shows an IDE window titled "flowable-ex2 - ex2.java [flowable-ex2.main]". The left sidebar displays a project structure for "ex2 [flowable-ex2]" with folders like ".gradle", ".idea", ".run", "build", "gradle", "src", and "main". The "main" folder is expanded, showing "java" and "utils" subfolders. The "ex2" file is selected under "java". The main editor area shows the following Java code:

```
118 @  
119  
120 private Flowable<Integer> publishRandomIntegers(Scheduler scheduler) {  
121     return Flowable  
122         // This factory method emits a stream of random integers.  
123         .generate(BackpressureEmitter  
124             // Emit a stream of random integers.  
125             .makeEmitter(Options.instance().count(),  
126                 Options.instance().maxValue(),  
127                 mPendingItemCount))  
128  
129         // Handle errors/exceptions gracefully.  
130         .onErrorResumeNext(error -> {  
131             Options.debug(error.getMessage());  
132             return Flowable.empty();  
133         })  
134  
135         // Subscribe on the given scheduler.  
136         .subscribeOn(scheduler);  
137 }
```

The bottom status bar indicates "Gradle sync finished in 11 s 927 ms (2 minutes ago)" and "22:14 CRLF UTF-8 4 spaces master".

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

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