reactor-examples (0.0.1)

Maksim Kostromin

Table of Contents

Chapter 1. Introduction

Chapter 2. intro

combine streams with: `Flux.merge`, `Flux.zip`, `Flux.first`, …

create streams with: `Flux.just`, `Flux.fromIterable`, `Flux.range`, `Flux.interval`, …

transform streams with: `Flux.skip`, `Flux.map`, `Flux.flatMap`, `Flux.filter`

Read project reference documentation

docker / docker-compose

```
./mvnw
./mvnw package docker-compose:up -P docker
./mvnw docker-compose:down -P docker

./gradlew
./gradlew composeUp
./gradlew composeDown
```

Chapter 3. Implementations

links:

- Helpful presentation
- Reactor intro

build

```java
package daggerok;

import lombok.extern.slf4j.Slf4j;
import reactor.core.publisher.Flux;

@Slf4j
public class App {
  public static void main(String[] args) {
    Flux.range(0, 10)
        .map(integer -> integer * integer)
        .filter(integer -> integer % 3 != 0)
        .buffer(3)
        .subscribe(integers -> log.info("result: {}", integers));
  }
}
```

build

```
./mvnw -p ./map-filter-buffer/pom.xml
java -jar ./map-filter-buffer/target/*-all.jar

# gradle jar
./gradlew -b ./map-filter-buffer/build.gradle
java -jar ./map-filter-buffer/build/libs/*-all.jar
```

build

```java
package daggerok;

import lombok.extern.slf4j.Slf4j;
import reactor.core.publisher.Flux;

@Slf4j
public class App {
  public static void main(String[] args) {
    Flux.range(0, 10)
        .map(integer -> integer * integer)
        .filter(integer -> integer % 3 != 0)
        .retry()
        .subscribe(integers -> log.info("result: {}", integers));
  }
}
```

build

```
./mvnw -p ./map-filter-retry/pom.xml
java -jar ./map-filter-retry/target/*-all.jar

# gradle jar
./gradlew -b ./map-filter-retry/build.gradle
java -jar ./map-filter-retry/build/libs/*-all.jar
```

Chapter 4. Links

- GitHub repo
- GitHub pages