Introduction to Reactive Programming

at Software Craftsmanship New York

by Balint Pato, Andrew Leung, Rafael Huaman

2016/03/08

What is reactive programming?

Background

Reactive

has been around for a while
reactive ~ event driven

- UI events
- Network protocols
- FRP (functional reactive programming) is something else: it works with purely functional programs **composed of functions of time** (can be discrete or continuous)
- Reactive programming works with discrete **events**

Reactive Manifesto

for creating reactive systems
www.reactivemanifesto.org

- Responsive
- Resilient, Elastic
- Message driven

Reactive streams

a new specification

- currently 1.0.0, java only
- low level API, 4 interfaces
    - Subscriber, Producer, Subscription, Processor
- TCK (technology compatibility kit)
- **BACKPRESSURE** - hot vs cold streamtypes

Typesafe

ORACLE

Pivotal

NETFLIX

spray

twitter

redhat

Applied Duality, Inc.

KAAZING

Doug Lea – SUNY Oswego

Composition APIs

solves the problem of organizing reactive code, composing event-driven behaviours

- spring reactor project
- reactive extensions

Reactive Extensions

Reactive Extensions

- Came from Microsoft first:
  - Observer pattern
  - LINQ
  - Schedulers
- has been adopted in many languages: reactivex.io

ReactiveX

http://reactivex.io/intro.html

Reactive programming is programming with asynchronous data streams.
/@andresaltz/

1. **Everything is a stream!**
   You will create an observable - it emits events! Think: mouse, event bus, price ticker, metrics…
2. **You will create new streams via "Queries"** - think LINQ (for .NET folks), or think Stream API for Java folks, think filter, map, reduce, etc. for the functional folks
3. **Schedulers** - concurrency in ReactiveX - schedulers can be blocking, non blocking (threads), delays, etc.

Picture taken from @andrewsaltz gist:
https://gist.github.com/staltz/868e7e9bc2a7b8c1f754

Recommended materials

The introduction to Reactive Programming you've been missing (by @andrestaltz)

Intro to Reactive Programming (SpringOne conference talk by Stephane Maldini, Rossen Stoyanchev)

reactivex.io

Exercise

Simulate the 2016 election!

http://github.com/software-craftsmanship-new-york/rxKata

Rules

- You are excused of writing tests today. Exceptionally. We will prepare a separate session on testing Rx. Enjoy!
- Be prepared to **share your learnings**, that will make the session fun!
- **Form teams of 3**! (not 4, not 2)
- Draw, design, read, learn first! Chop up the work!
- Take babysteps, iterate!
- **Plan:**
  2 x (45 mins session + 10 mins retro)

Recommended iterations

1. [optional] If you know threads well, implement it first without Rx!
2. Implement with a dozen votes first with a blocking Observable (just use a list) and print the individual votes
3. Implement the vote counting query on top of it
4. Make the Observable parallel with Schedulers