Reactive Spring

Full-stack reactive with Spring 5, Spring Boot 2, & Project Reactor

Mark Heckler
Principal Technologist/Developer Advocate
mark@thehecklers.org
mheckler@pivotal.io
@mkheck

Pivotal

Who am I?

- Author
- Speaker
- Architect & Developer
- Java Champion
- Seeker of a better way

"In a nutshell reactive programming is about
non-blocking, event-driven applications that
scale with a small number of threads with
backpressure as a key ingredient that
aims to ensure producers
do not overwhelm consumers."

–Rossen Stoyanchev, Project Reactor team

Reactive Streams: 4 interfaces

- Publisher<T>
- Subscriber<T>
- Subscription
- Processor<T,R>

Project Reactor: a quick overview

REACTIVE CORE

Reactor is a **fully non-blocking** foundation with efficient demand management. It directly interacts with Java 8 *functional API*, *Completable Future*, *Stream* and *Duration*.

TYPED [0|1|N] SEQUENCES

Reactor offers **2 reactive composable API** Flux [N] and Mono [0|1] extensively implementing Reactive Extensions.

NON BLOCKING IPC

Suited for **Microservices** Architecture, Reactor IPC offers **backpressure-ready network engines** for HTTP (including Websockets), TCP and UDP. Reactive Encoding/Decoding is fully supported.

Let’s code!

Helpful resources

- http://www.reactive-streams.org
- https://projectreactor.io
- https://github.com/mkheck/flux-flix-intro
- https://github.com/joshlong/flux-flix-service

Thanks for coming!

@mkheck
mark@thehecklers.org
mheckler@pivotal.io

Pivotal

@mkheck @springcentral @projectreactor

Pivotal