Reaktive Programmierung

mit Spring Boot und Project Reactor

Torsten Kohn

Software Engineer bei der
Comsysto Reply GmbH in München

- Java, JavaScript, TypeScript
- Spring Boot, Angular, React
- Git, Maven, Gradle, Jenkins, Nexus, Sonar
- Amazon Web Services

Inhalt

Einführung: Warum das Thema?

- Imperative Programmierung
- blocking I/O
- skalieren
- Threads

Was ist Reaktive Programmierung?

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

- Programmierparadigma
- Kombination von Observer (push) & Iterator (pull)
- sparsamer Umgang mit Ressourcen
- Pionierarbeit: Reactive Extentions (Rx) für .NET
- Standardisierung für JVM durch Reactive Streams

Reactive Streams - Spezifikation

- The purpose of Reactive Streams is to provide a standard for asynchronous stream processing with non-blocking back pressure.

  Quelle: Reactive Streams

- Spezifikation
  Java-API
  Technology-Compatibility-Kit
  Beispiel-Implementierungen

- Collaboration von verschiedenen Entwicklern von folgenden Unternehmen:

Kaazing	Netflix	Pivotal
Lightbend	Red Hat	Twitter

Publisher

```
package org.reactivestreams;

public interface Publisher<T> {

    public void subscribe(Subscriber<? super T> s);

}
```

Subscriber

```
package org.reactivestreams;

public interface Subscriber<T> {

    public void onSubscribe(Subscription s);

    public void onNext(T t);

    public void onError(Throwable t);

    public void onComplete();

}
```

Subscription

```java
package org.reactivestreams;

public interface Subscription {

    public void request(long n);

    public void cancel();

}
```

Processor

```java
package org.reactivestreams;

public interface Processor<T, R>
    extends Subscriber<T>, Publisher<R> {

}
```

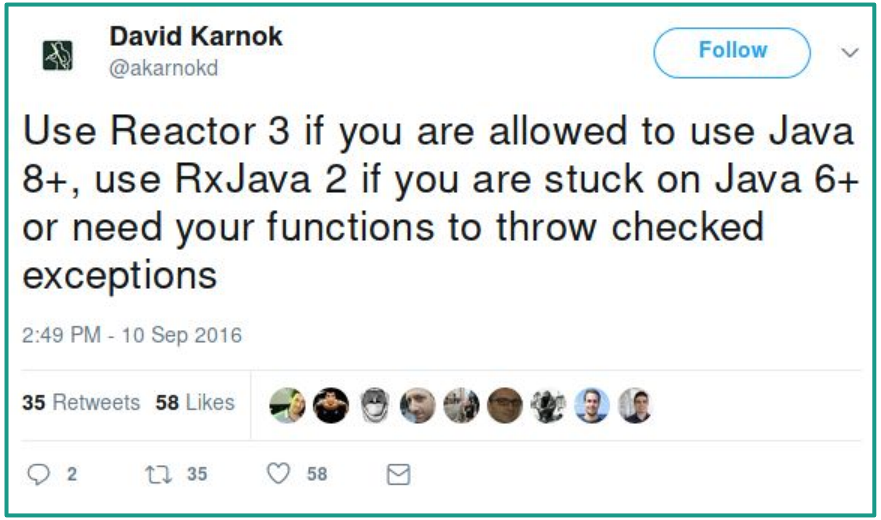
Implementierungen

RxJava

akka

Project Reactor

RxJava vs Reactor



David Karnok
@akarnokd

Follow

Use Reactor 3 if you are allowed to use Java 8+, use RxJava 2 if you are stuck on Java 6+ or need your functions to throw checked exceptions

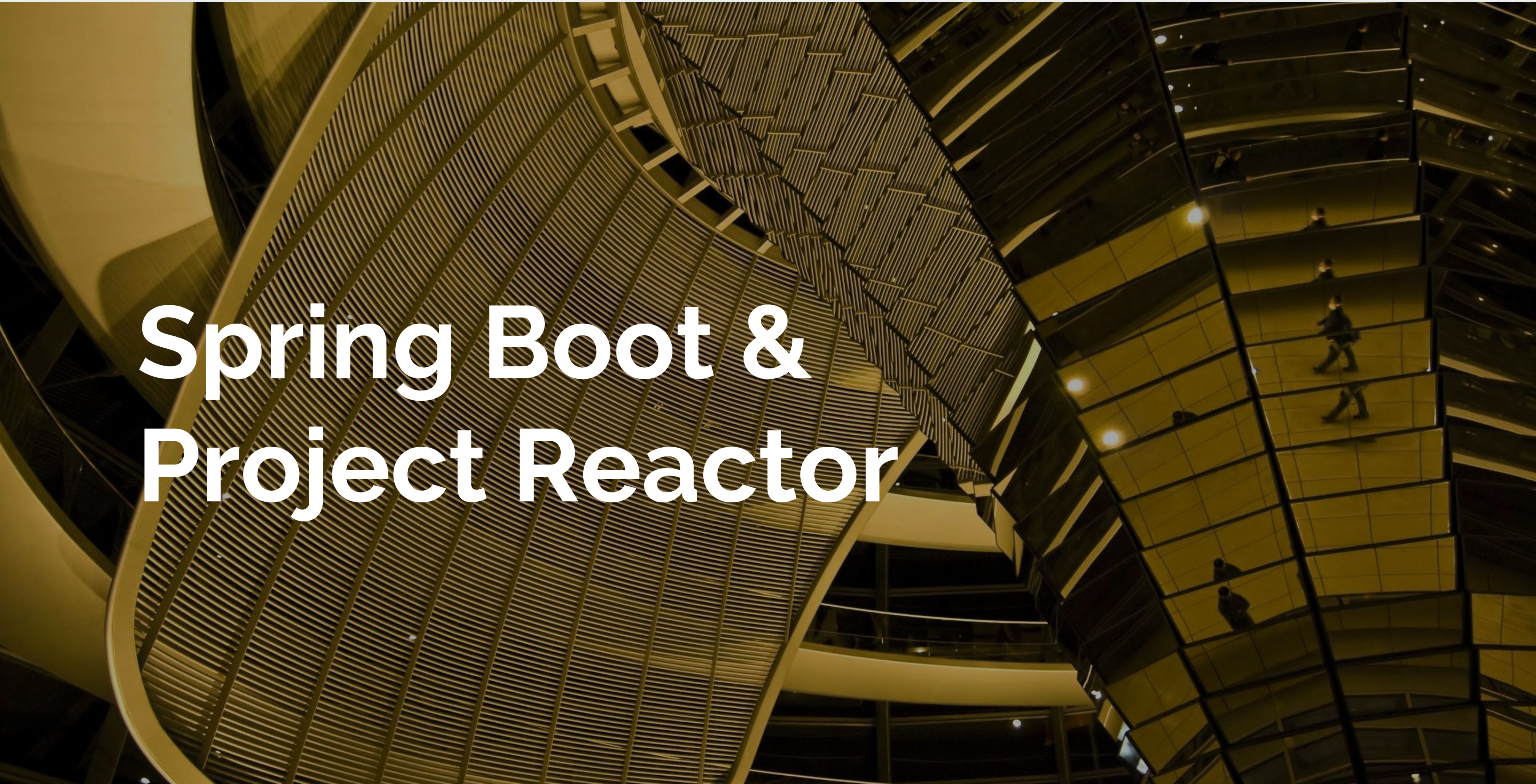
2:49 PM - 10 Sep 2016

35 Retweets 58 Likes

2 35 58

Java 9 Flow API

- keine Implementierung
- keine Spezifikation
- keine Tests / kein Toolkit zum Validieren der eigenen Implementierung



Spring Boot & Project Reactor

Project Reactor

- Java 8 functional API
- Completable Future
- Stream
- Duration

- backpressure-ready network engines
  - HTTP (Websockets)
  - TCP und UDP

Publisher - Mono

0 oder 1 Element

```
Mono.just("Hello World")
        .map(value -> value.split("(?!^)"))
        .flatMapMany(Flux::fromArray)
        .filter(character -> !"L".equalsIgnoreCase(character))
        .collectList()
        .map(list -> String.join("", list))
        .subscribe(System.out::println);

// Heo Word
```

Publisher - Flux

0 oder N Elemente

```
Flux<String> abc = Flux
        .just("A", "B", "C");

Flux<String> cba = Flux
        .fromIterable(
                Arrays.asList("c", "b", "a"));

abc.zipWith(cba).subscribe(System.out::println);
```

```
// [A,c]

// [B,b]

// [C,a]
```

Demo

Unterstützte Technologien

Server und Servlet-Container

Netty

Apache Tomcat

jetty://

undertow

Unterstützte Technologien

NoSQL Datenbanken

mongoDB®

redis

cassandra

Couchbase

- bringt Overhead
- Debugging ist komplizierter
- kapselt Nebenläufigkeit, Threads und Synchronisation

Wie geht es weiter?

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

- Demo Code auf Gitlab
- Project Reactor Dokumentation
- Reactor in Produktion Cloud Foundry Java Client
- Servlet and Reactive Stacks in Spring Framework 5 InfoQ