SPRING IO

#springio19

Moving From Imperative To Reactive

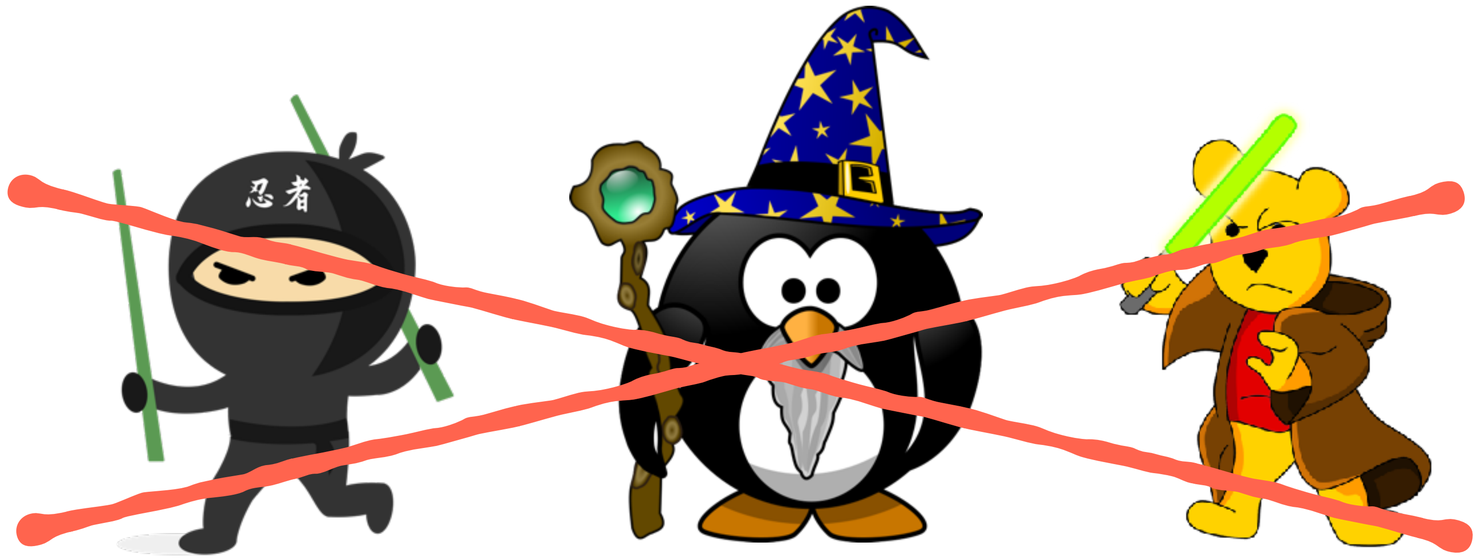
Paul Harris
@twoseat

What is this?

- An Introduction to Reactive Programming
- Some background
- A practical example, coded live!
- For your benefit

Why me?

- Using Project Reactor for over 3 years
- Cloud Foundry Java Client
- Made all the reactive mistakes
- A normal developer



Reactive Manifesto

- Responsive
- Resilient
- Elastic
- Message Driven

Reactive Streams

- Publisher
- Subscriber
- Subscription
- Processor

Project Reactor

- Translates Reactive Streams into a framework you can use
- First commit 10 November 2015
- Forms the basis of reactive support in Spring
- Includes reactor-core, reactor-netty, and reactor-test

Flux & Mono

- Flux - a Publisher of 0 to N elements.
- Mono - a Publisher of 0 or 1 elements

Code!

Summary

- Flux and Mono
- .map() and .flatMap()
- Errors are just values
- Stay reactive!
- Doesn’t replace what you know (e.g. imperative, streams)

Resources

- Reactive Manifesto: https://www.reactivemanifesto.org
- Reactive Streams: https://www.reactive-streams.org
- Project Reactor: https://projectreactor.io
- Choosing an Operator: https://projectreactor.io/docs/core/release/reference/index.html#which-operator
- Spring Web Reactive: https://docs.spring.io/spring/docs/current/spring-framework-reference/web-reactive.html
- Code: https://github.com/twoseat/springio2019
- CF Java Client: https://github.com/cloudfoundry/cf-java-client

Thanks!

Paul Harris
@twoseat