

Programação reativa em Java

Construa um software mais escalável com programação reativa

Luiz Costa - Outubro 2021

Sumário

- O que é programação reativa ?
- Escalabilidade
- Reactive vs Servlet
- Benefícios
- Desvantagens
- Hands-on
- Referências
- Perguntas

O que é programação reativa?

A programação reativa é um paradigma de programação assíncrona preocupado com o processamento de fluxos de dados e propagação de mudanças



Escalabilidade

Qualidade do sistema que consegue suportar um **aumento relativamente elevado de carga** sem que isso afete negativamente o seu desempenho.



Reactive stack

- Netty, Servlet 3.1 (SSE)
- Reactive Streams Adapters
- Spring Security Reactive
- Spring WebFlux
- Spring Data Reactive
Repositories
 - Mongo, Cassandra, Redis,
Couchbase, R2DBC

Servlet stack

- Servlet Container
- Servlet API
- Spring Security
- Spring MVC
- Spring Data Repositories
 - JDBC, JPA, NoSQL

Benefícios

ESCALABILIDADE

Dado um aumento de demanda, o sistema reativo por natureza é elástico e escalável e deve responder ao aumento dessa demanda.

USO EFICIENTE DA CPU

Não há a mudança de trabalho entre CPU's

CÓDIGO CLARO

Preocupação com o fluxo de dados, e uma API bem extensa para tratamento de erros, retries, filtros e transformações.

Desvantagens

MÃO DE OBRA

Esse item está relacionado com o próximo item, visto que esse modelo de desenvolvimento ainda não está tão difundido, e com isso carece de mão de obra.

MUDANÇA DE PENSAMENTO

É uma mudança no modelo de desenvolvimento, caso o time/software esteja acostuma com outro modelo de programação.

CONTROLE DE TRANSAÇÃO

Não tão simple quanto o uso de ThreadLocal, com SpringMVC.



Hands-on \o/

O código fonte está disponível em:
<https://github.com/luizugustavocosta/progamação-reactiva-em-java-com-spring>

Passo a passo para executar o projeto

1

Instale e Configure MongoDB

Crie 2 collections, com os nomes:
`listingsAndReviews`, `movies`

2

Download o código

```
git clone  
https://github.com/luizgustavocosta/progamacao-reactiva-em-java-com-spring.git
```

5

Execute o projeto `movie-api` e `review-api`

Esses dois projetos usam MongoDB como database reativo.

- `movie-api`
`java -jar movie-api/target/movie-api-0.0.1-SNAPSHOT.jar`
- `review-api`
`java -jar review-api/target/review-api-0.0.1-SNAPSHOT.jar`

3

Construa o projeto

- `cd progamacao-reactiva-em-java-com-spring`
- `mvn clean verify`

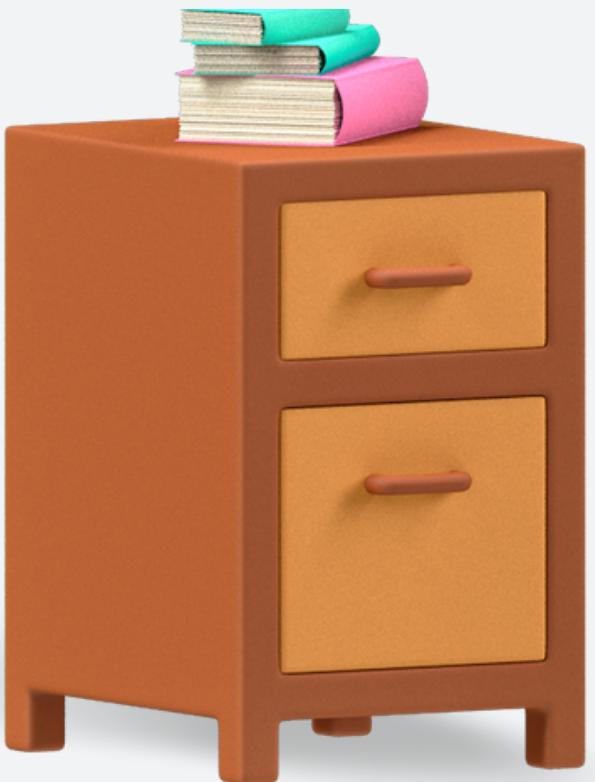
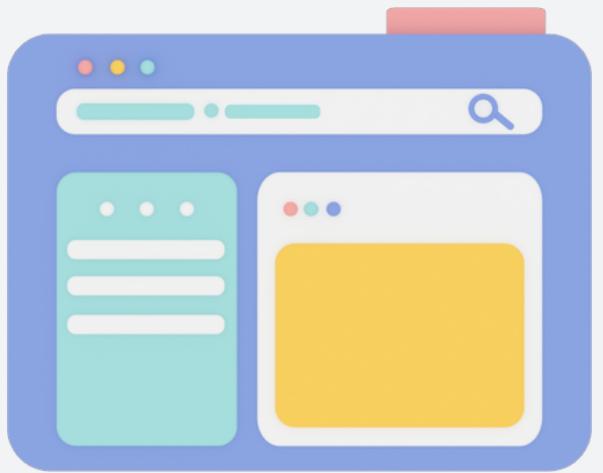
4

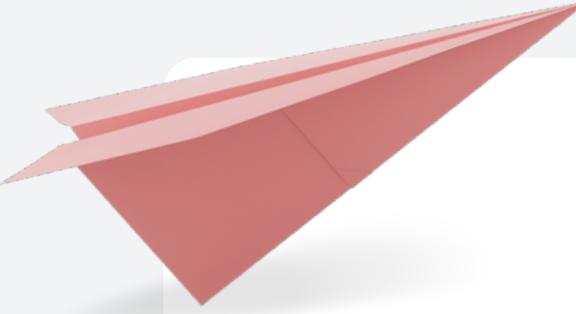
Execute o projeto `customer-api`

- Esse projeto usa H2 como banco de dados reativo
- `java -jar customer-api/target/customer-api-0.0.1-SNAPSHOT.jar`

Referências

- <https://github.com/luizgustavocosta/progamacao-reactiva-em-java-com-spring>
- Vernon, Vaughn. Reactive, Domain-Drive Design. Kalele 2021
- <https://www.baeldung.com/rxjava-vs-java-flow-api>
- <https://www.baeldung.com/java-9-reactive-streams>
- https://en.wikipedia.org/wiki/Reactive_programming#Programming_models_and_semantics
- <https://www.youtube.com/watch?v=z0a0N9OgaAA> [Tomasz Nurkiewicz]
- <https://www.infoq.com/articles/Designing-Implementing-Using-Reactive-APIs/>
- <https://www.oreilly.com/radar/reactive-programming-vs-reactive-systems/>
- <https://www.youtube.com/watch?v=ZNXCmG2MTzw> [Venkat]
- <https://www.reactivemanifesto.org/pt>
- <https://quarkus.io/blog/mutiny-back-pressure/>
- <https://rxmarbles.com>
- <https://projectreactor.io/docs/core/release/api/reactor/core/publisher/Flux.html>
- <https://github.com/ReactiveX/RxJava>
- <https://www.thoughtworks.com/radar/languages-and-frameworks/webflux>
- <https://github.com/rstoyanchev/demo-reactive-spring>
- <https://www.youtube.com/watch?v=M3jNn3HMeWg> [Defog]
- <https://hazelcast.com/blog/migrating-from-imperative-to-reactive/>





Perguntas?

