Overview of Spring WebFlux

Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

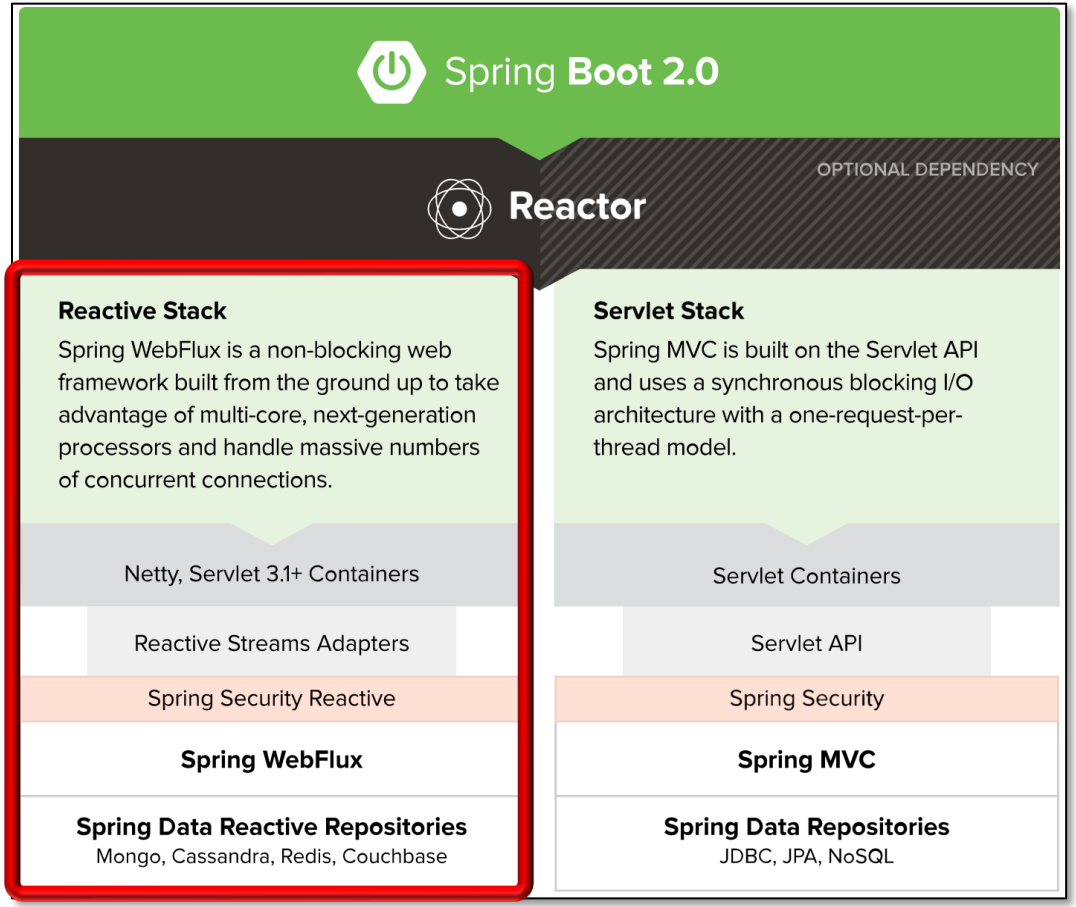
Professor of Computer Science

Institute for Software
Integrated Systems

Vanderbilt University
Nashville, Tennessee, USA

Learning Objectives in this Lesson

- Understand the structure & functionality of the Spring WebFlux framework supported by Spring Boot 2.0

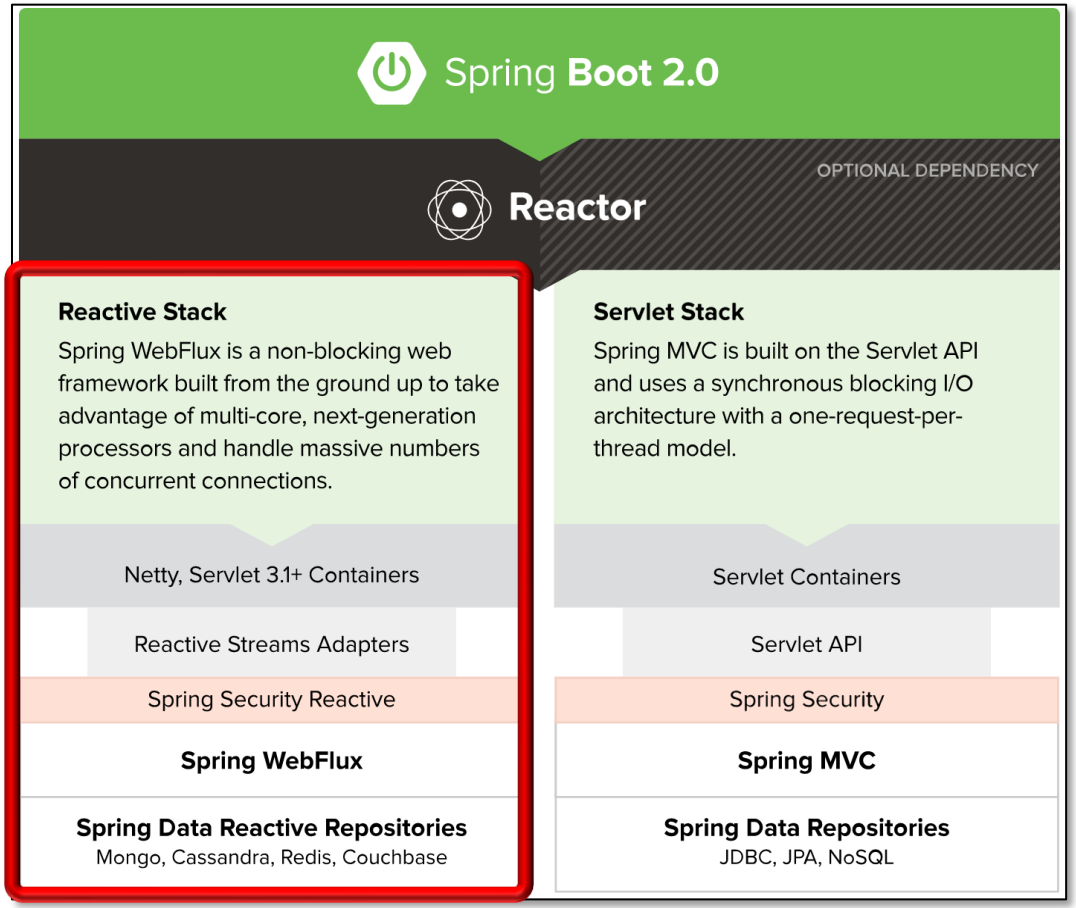


See docs.spring.io/spring-framework/docs/current/reference/html/web-reactive.html#webflux

Overview of Spring WebFlux

Overview of Spring WebFlux

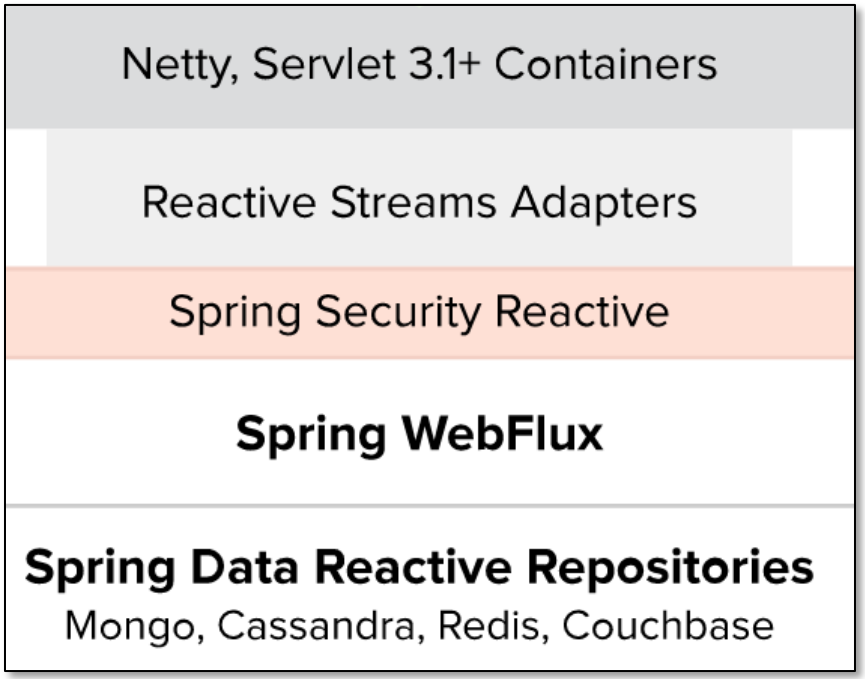
- Spring WebFlux



See docs.spring.io/spring-framework/docs/current/reference/html/web-reactive.html#webflux

Overview of Spring WebFlux

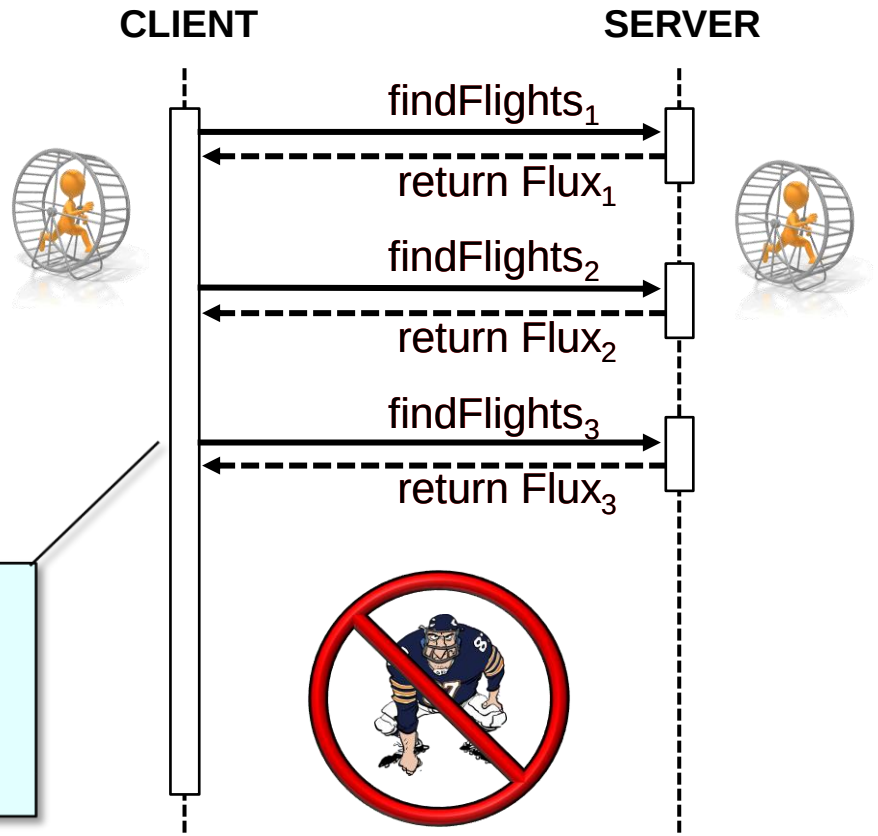
- Spring WebFlux
  - A non-blocking web framework that leverages multiple cores & handles large # of concurrent connections



See en.wikipedia.org/wiki/Non-blocking_algorithm

Overview of Spring WebFlux

- Spring WebFlux
  - A non-blocking web framework that leverages multiple cores & handles large # of concurrent connections
    - Requests are handled in an entirely asynchronous (& "lazy") manner



A request to a list of flights from a database over the network might take a few seconds, but the threads servicing requests & responses don't block

See en.wikipedia.org/wiki/Asynchrony_(computer_programming)

Overview of Spring WebFlux

- Spring WebFlux
  - A non-blocking web framework that leverages multiple cores & handles large # of concurrent connections
    - Requests are handled in an entirely asynchronous (& "lazy") manner
    - A small # of threads are thus required

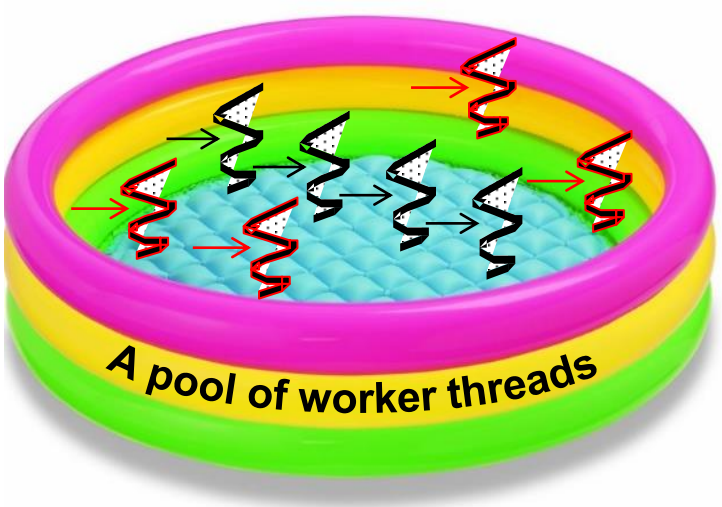
See www.baeldung.com/spring-webflux-concurrency

Overview of Spring WebFlux

- Spring WebFlux
  - A non-blocking web framework that leverages multiple cores & handles large # of concurrent connections
    - Requests are handled in an entirely asynchronous (& "lazy") manner
    - A small # of threads are thus required
      - Typically based on # of processor cores

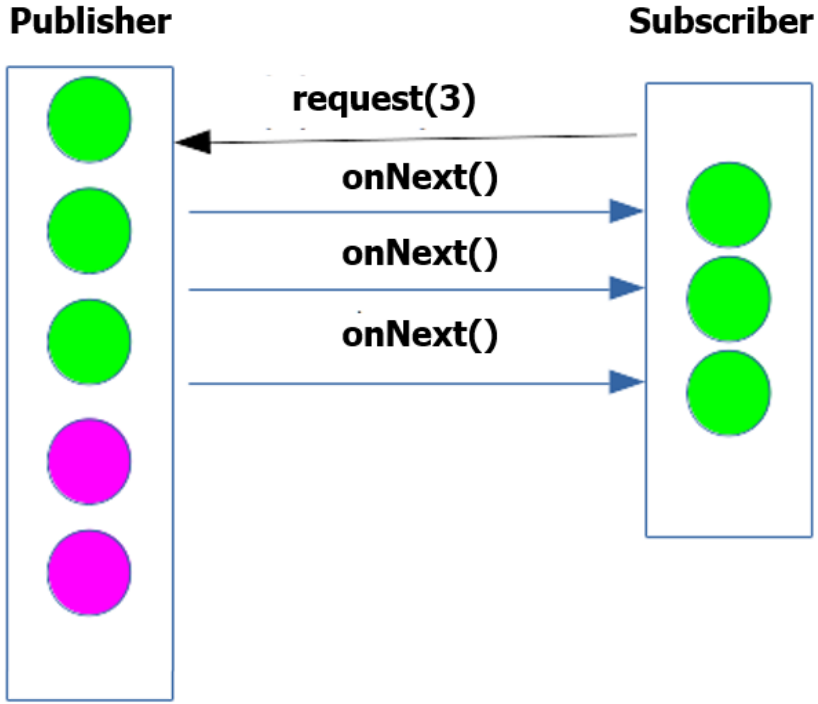
Overview of Spring WebFlux

- Spring WebFlux
  - A non-blocking web framework that leverages multiple cores & handles large # of concurrent connections
    - Requests are handled in an entirely asynchronous (& "lazy") manner
    - A small # of threads are thus required
      - Typically based on # of processor cores
      - I/O-bound operations may require adaptively increasing the # of threads



Overview of Spring WebFlux

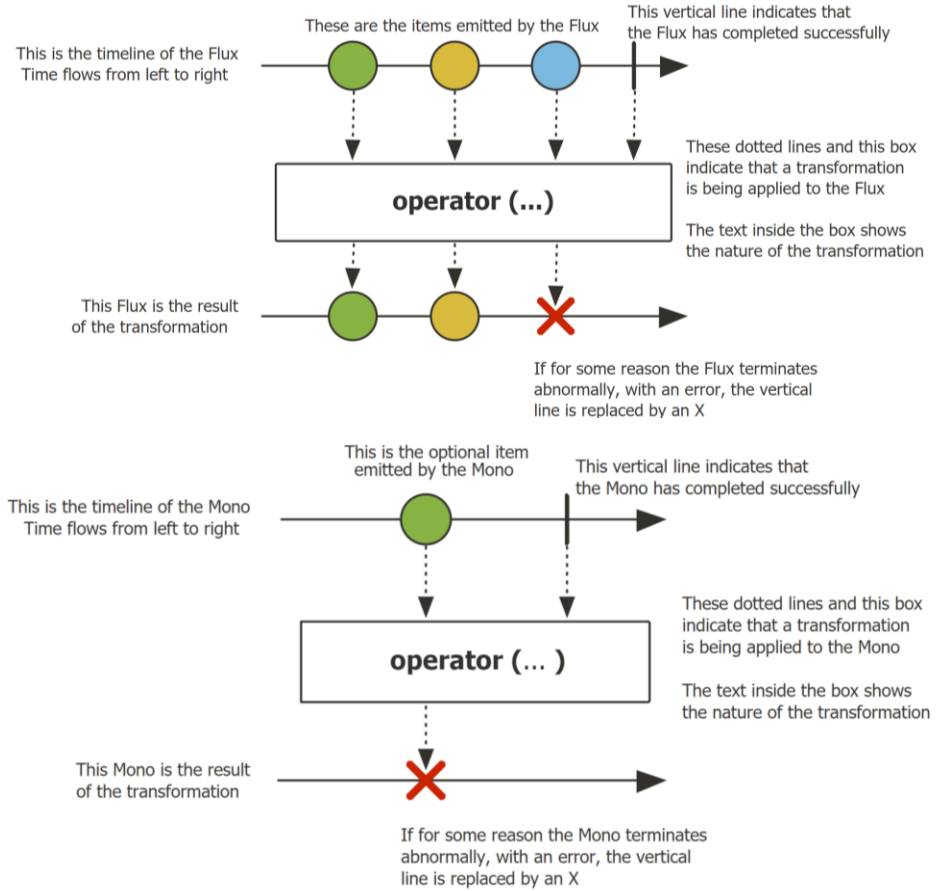
- Spring WebFlux
  - A non-blocking web framework that leverages multiple cores & handles large # of concurrent connections
    - Requests are handled in an entirely asynchronous (& "lazy") manner
    - A small # of threads are thus required
    - However, there's often a need for non-blocking backpressure
      - i.e., control event rate so a fast publisher does not overwhelm a slower subscriber



See www.baeldung.com/spring-webflux-backpressure

Overview of Spring WebFlux

- Spring WebFlux
  - A non-blocking web framework that leverages multiple cores & handles large # of concurrent connections
  - Network communication uses Project Reactor reactive types



See spring.io/blog/2016/04/19/understanding-reactive-types

Overview of Spring WebFlux

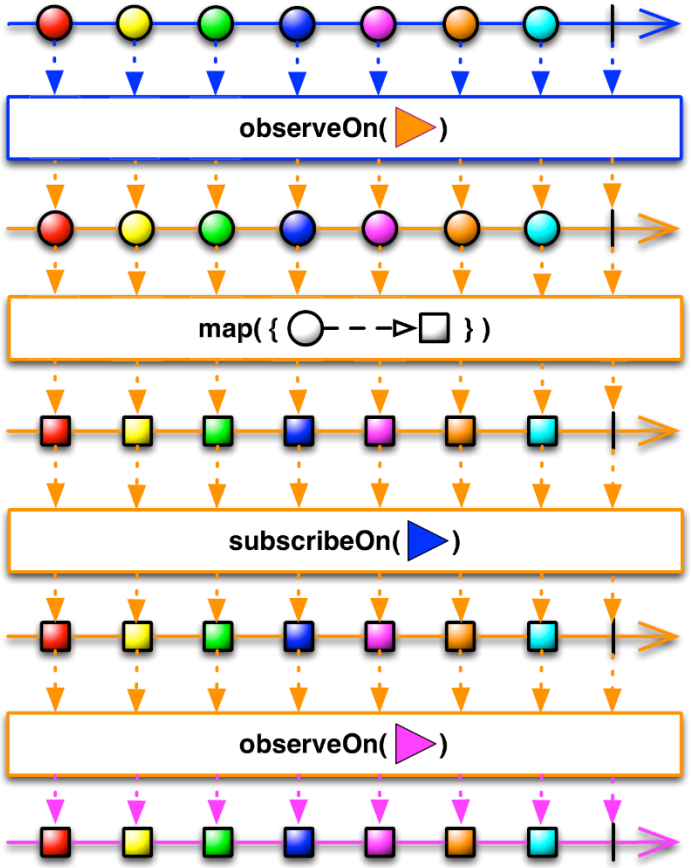
- Spring WebFlux
  - A non-blocking web framework that leverages multiple cores & handles large # of concurrent connections
  - Network communication uses Project Reactor reactive types
    - Work on data sequences of 0..1 (Mono) and 0..N (Flux)

```
public class FlightController {
  ...
  @GetMapping(FLIGHT_DATES)
  Flux<LocalDate>
     findDepartureDates
       (@RequestParam String
         departureAirport,
        @RequestParam String
         arrivalAirport) {...}
  ...
  @GetMapping(EXCHANGE)
  Mono<ExchangeRate> getRate
    (@RequestParam String from,
     @RequestParam String to)
    {...}
```

See flights-reactive-microservices/-/blob/master/src/main/java/server/flight/FlightController.java

Overview of Spring WebFlux

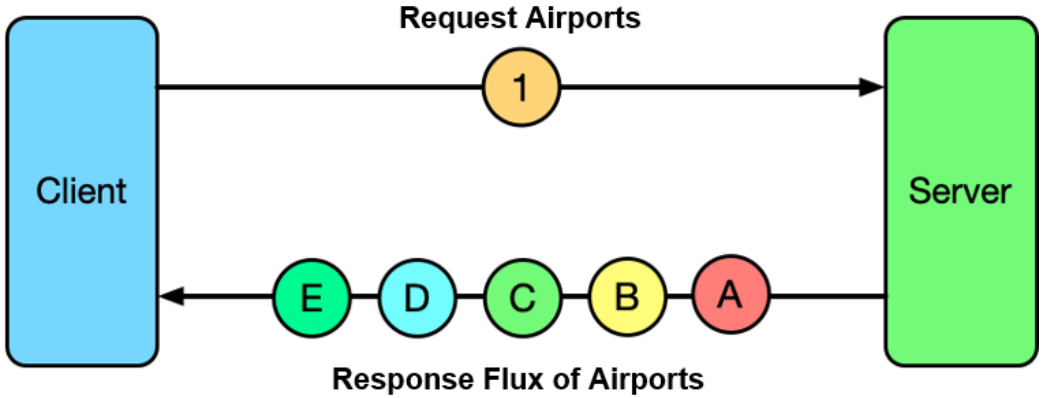
- Spring WebFlux
  - A non-blocking web framework that leverages multiple cores & handles large # of concurrent connections
  - Network communication uses Project Reactor reactive types
    - Work on data sequences of 0..1 (Mono) and 0..N (Flux)
    - Provides a rich set of operators aligned with the ReactiveX vocabulary of operators



See reactivex.io/documentation/operators.html

Overview of Spring WebFlux

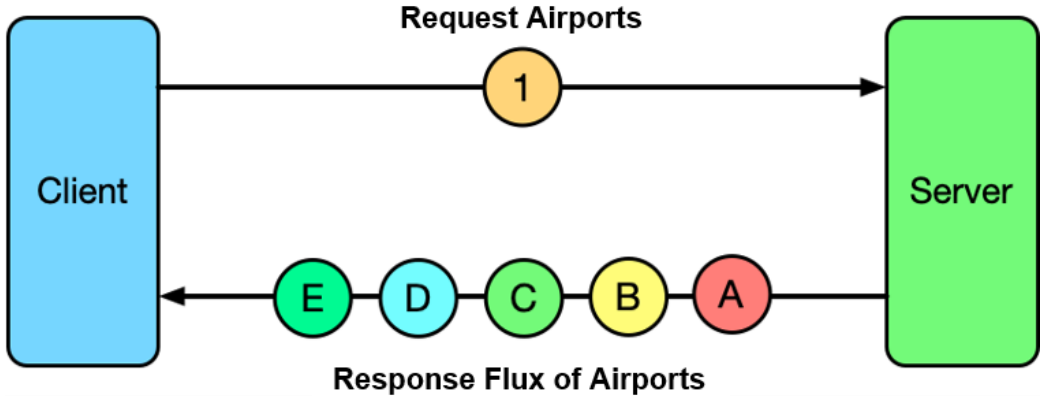
- Spring WebFlux
  - A non-blocking web framework that leverages multiple cores & handles large # of concurrent connections
  - Network communication uses Project Reactor reactive types
  - Reactive Flux types can be streamed element-by-element



See medium.com/@nithinmallya4/processing-streaming-data-with-spring-webflux-ed0fc68a14de

Overview of Spring WebFlux

- Spring WebFlux
  - A non-blocking web framework that leverages multiple cores & handles large # of concurrent connections
  - Network communication uses Project Reactor reactive types
  - Reactive Flux types can be streamed element-by-element
    - Clients can thus be more responsive



See en.wikipedia.org/wiki/Responsiveness

Overview of Spring WebFlux

- Spring WebFlux
  - A non-blocking web framework that leverages multiple cores & handles large # of concurrent connections
  - Network communication uses Project Reactor reactive types
  - Reactive Flux types can be streamed element-by-element
    - Clients can thus be more responsive
  - The stream can be kept "live" via Spring server-sent-events

```
public class FlightController {
  ...
  @GetMapping(RATES,
    produces = MediaType.
      TEXT_EVENT_STREAM_VALUE) {
  Flux<ExchangeRate>
    getRates(@RequestParam
                String toCurrency)
      {...}
  ...
```

See www.baeldung.com/spring-server-sent-events

Overview of Spring WebFlux

- Spring WebFlux
  - A non-blocking web framework that leverages multiple cores & handles large # of concurrent connections
  - Network communication uses Project Reactor reactive types
  - Reactive Flux types can be streamed element-by-element
    - Clients can thus be more responsive
  - The stream can be kept "live" via Spring server-sent-events

STATISTICS Servlet Stack

Executions

	Total	OK	KO
	40000	39994	6
Mean req/s	784.314	784.196	0.118

Response Time (ms)

	Total	OK	KO
Min	202	202	2443
50th percentile	2093	2093	3151
75th percentile	3017	3017	3763
95th percentile	4547	4547	4104
99th percentile	5061	5061	4154
Max	5356	5356	4166
Mean	2102	2101	3216
Std Deviation	1304	1304	660

STATISTICS Reactive Stack

Executions

	Total	OK	KO
	40000	40000	0
Mean req/s	975.61	975.61	-

Response Time (ms)

	Total	OK	KO
Min	201	201	-
50th percentile	315	315	-
75th percentile	422	422	-
95th percentile	865	865	-
99th percentile	1392	1392	-
Max	1883	1883	-
Mean	390	390	-
Std Deviation	232	232	-

The reactive stack can have better response time & the # of request per second

See medium.com/@the.raj.saxena/springboot-2-performance-servlet-stack-vs-webflux-reactive-stack-528ad5e9dadc

End of Overview of Spring WebFlux