In spring boot webflux based microservice, who is the subscriber?

Asked 4 years, 1 month ago Active 6 months ago Viewed 2k times

6

3

Note: Here the terms Subscriber and Subscription are being used from the reactive streams specification.

Consider the following @RestController methods in a spring boot webflux based microservice.

```
@GetMapping(path = "/users", produces = MediaType.APPLICATION_JSON_VALUE)
public Flux<TradingUser> listUsers() {
    return this.tradingUserRepository.findAll();
}

@GetMapping(path = "/users/{username}", produces = MediaType.APPLICATION_JSON_VALUE)
public Mono<TradingUser> showUsers(@PathVariable String username) {
    return this.tradingUserRepository.findByUserName(username);
}
```

1. Here "who/what" will act as the "Subscriber"? I assume the spring boot framework provides a Subscriber(?) Can someone please provide details or any links around this?
2. Say I am invoking the restful endpoint above using client like postman/curl/browser, then in that case how can the client signal demand to the reactive server? (Only the Subscriber has a handle on the Subscription object with the request(n) method to signal demand. However, since Subscriber is probably also on the server side implemented by the spring boot framework, how can the actual client signal the demand?) I am obviously missing something.

spring-boot spring-webflux reactive-streams

Share Edit Follow

edited Jan 10, 2018 at 8:48

asked Jan 10, 2018 at 6:51

Jatin
587 6 16

You controller code above is the `Publisher` . And your postman/browser is `Consumer` – pvpkiran Jan 10, 2018 at 8:50

1 @pvpkiran - sorry mate, your comment does not add any value. Question clearly asks who/what the Subscriber is, not consumer. – Jatin Jan 10, 2018 at 22:59

2 Answers

Active | Oldest | Score

8

Currently with HTTP, the *exact backpressure information* is not transmitted over the network, since the HTTP protocol doesn't support this. This can change if we use a different wire protocol.

So the reactive streams demand is translated to/from the actual read/writes at the HTTP level.

If you looks at Spring Framework's `org.springframework.http.server.reactive.ServletHttpHandlerAdapter` , you'll see that this class does the adaptation between Servlet 3.1 Async I/O and Reactive Streams. It does implement a specific `Subscriber` class.

There are other specific adapter implementations for this as well: Undertow, Jetty, Tomcat, Reactor Netty. If the underlying server supports reactive streams, we'll simply let the server handle the demand. If not, a `Subscriber` implementation is used.

Share Edit Follow

answered Jan 11, 2018 at 8:11

Brian Clozel
50.2k 15 140 162

So the reactive streams demand is translated to/from the actual read/writes at the HTTP level. -- can you please elaborate this? What do you mean by read/writes? Do you mean to say the back-pressure relies on TCP flow control? – Jatin Jan 12, 2018 at 3:10

1 It relies on reading and writing in the OS TCP buffers. This in turn translates into TCP control flow (e.g. if a buffer is full, the OS can use TCP control flow to stop incoming data) – Brian Clozel Jan 12, 2018 at 5:20

2

Inside the dependency `org.springframework.spring-web` there is a function named `public void service(...` which calls `.subscribe` in `ServletHttpHandlerAdapter` . I think it is confusing sometimes to understand that the framework is handling this subscription under the hood when many tutorials on WebFlux show the subscriber of a Mono or Flux explicitly to demonstrate how reactive streams work but here it is done by the framework for us.

Share Edit Follow

answered Aug 8, 2021 at 22:57

caladeve
406 4 12