Overview of Spring WebMVC

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 WebMVC framework supported by Spring Boot 2.0

See docs.spring.io/spring-framework/docs/3.2.x/spring-framework-reference/html/mvc.html
Overview of Spring WebMVC
Overview of Spring WebMVC

- Spring WebMVC

See docs.spring.io/spring-framework/docs/3.2.x/spring-framework-reference/html/mvc.html
Overview of Spring WebMVC

- Spring WebMVC
  - Built on the Servlet API & uses a synchronous I/O architecture w/one-thread-per-request model
Overview of Spring WebMVC

- **Spring WebMVC**
- Built on the Servlet API & uses a synchronous I/O architecture w/one-thread-per-request model
- Each request is handled by a thread that blocks until it is able to fully process the request

A request to a list of flights from a database over the network might take a few seconds, which blocks threads from servicing other requests & responses

See [en.wikipedia.org/wiki/Blocking_(computing)](en.wikipedia.org/wiki/Blocking_(computing))
Overview of Spring WebMVC

• Spring WebMVC
  • Built on the Servlet API & uses a synchronous I/O architecture w/one-thread-per-request model
  • Each request is handled by a thread that blocks until it is able to fully process the request
  • Blocking calls are a natural form of back pressure that forces the caller to wait

See medium.com/@jayphelps/backpressure-explained-the-flow-of-data-through-software-2350b3e77ce7
Overview of Spring WebMVC

- Spring WebMVC
  - Built on the Servlet API & uses a synchronous I/O architecture w/one-thread-per-request model
  - Each request is handled by a thread that blocks until it is able to fully process the request
  - Blocking calls are a natural form of back pressure that forces the caller to wait
  - Thereby eliminating the need for end-to-end rate control

See en.wikipedia.org/wiki/Rate_limiting
Overview of Spring WebMVC

- Spring WebMVC
- Built on the Servlet API & uses a synchronous I/O architecture w/one-thread-per-request model
  - Each request is handled by a thread that blocks until it is able to fully process the request
    - Blocking calls are a natural form of back pressure that forces the caller to wait
  - However, many threads may be needed to handle bursty clients

See [www.baeldung.com/java-web-thread-pool-config](http://www.baeldung.com/java-web-thread-pool-config)
Overview of Spring WebMVC

- **Spring WebMVC**
  - Built on the Servlet API & uses a synchronous I/O architecture w/one-thread-per-request model
  - Network communication uses common Java collection types

See [docs.oracle.com/javase/8/docs/technotes/guides/collections/overview.html](http://docs.oracle.com/javase/8/docs/technotes/guides/collections/overview.html)
Overview of Spring WebMVC

• Spring WebMVC
  • Built on the Servlet API & uses a synchronous I/O architecture w/one-thread-per-request model
  • Network communication uses common Java collection types
    • e.g., Java List & Map collections

```java
public class FlightController {
    ...
    @GetMapping(FLIGHT_DATES)
    List<LocalDate>
        findDepartureDates
            (@RequestParam String departureAirport,
             @RequestParam String arrivalAirport)
            { ... }
    ...
```

See [flights-microservices/-/blob/master/src/main/java/server/flight/FlightController.java](flights-microservices/-/blob/master/src/main/java/server/flight/FlightController.java)
Overview of Spring WebMVC

• **Spring WebMVC**
  • Built on the Servlet API & uses a synchronous I/O architecture w/one-thread-per-request model
  • Network communication uses common Java collection types
  • WebMVC endpoints send & return Java collections in one fell swoop
Overview of Spring WebMVC

- **Spring WebMVC**
  - Built on the Servlet API & uses a synchronous I/O architecture w/one-thread-per-request model
  - Network communication uses common Java collection types
- **WebMVC endpoints** send & return Java collections in one fell swoop
  - Client latency may suffer & thus not be as responsive as possible

See [en.wikipedia.org/wiki/Spinning_pinwheel](en.wikipedia.org/wiki/Spinning_pinwheel)
End of Overview of Spring WebMVC