Spring Boot là mở rộng, cải tiến của Spring framework.

Ưu điểm của Spring Boot:

- Cung cấp sẵn các thư viện dependencies, đơn giản hóa việc xây dựng và cấu hình ứng dụng. Ví dụ Thymeleaf có một thư viện Thymeleaf nhưng Spring Boot sẽ dùng một thư viện Spring Boot Starter Thymeleaf cấu hình dành riêng cho ứng dụng SpringBoot.

- Nhúng sẵn web server bên trong ứng dụng Spring Boot để đơn giản hoá khi đóng gói. Mặc định là TomCat nhưng có thể thay bằng Netty.

- Hỗ trợ lấy thông số, kiểm tra sức khoẻ ứng dụng Spring Boot Actuator

- Có nhiều cách cấu hình ứng dụng sử dụng @Configuration hoặc application.properties hoặc application.yml

- Có sẵn các cấu hình mặc định ví dụ bố trí các thức mục con... để giảm tối đa cấu hình lại.

Cải tiến của Spring Boot so với Servlet JSP:

- Spring Boot cấu hình sẵn gần như mọi thứ. Còn Servlet sử dụng XML để cấu hình

- Spring Boot đóng gói TomCat bên trong. Servlet dùng TomCat ở ngoài

- Servlet gồm các API ở mức thấp xử lý request - response. Trong khi Spring Boot bổ xung thêm pattern Model - View - Controller

3.

- JAR đóng gói tất các file class, tài nguyên vào một file Zip để có thể tái sử dụng như là một thư viện, hay nột ứng dụng độc lập.

- WAR phù hợp để đóng gói ứng dụng web theo cấu trúc để triển khai lên web server.

4.

- Maven tạo ra file pom.xml (dạng XML) để cấu hình build ứng dụng gồm tên ứng dụng, group id, artifactid, thông số Java JDK, các thư viện phụ thuộc dependecies, các plugins. Maven ra đời sớm, phổ biến hơn,nhưng tốc độ build chậm hơn. Maven có hệ thống quản lý các dependencies là https://mvnrepository.com/

- Gradle sử dụng script ngắn gọn hơn, dạng thực thi dòng lệnh hơn là cấu hình. Tốc độ build của Gradle cho dự án lớn tốt hơn Maven.

5.

Annotation @ResponseBody đánh dấu một phương thức trong Controller trả về luôn cho trình duyệt dữ liệu mà không qua xử lý thêm.

6.

Annotation @GetMapping đánh dấu một phương thức trong Controller sẽ hứng / xử lý GET request gửi lên trình duyệt.

7.

Annotation @Controller thuộc package org.springframework.stereotype.Controller

8.

Annotation @SpringBootApplication bao gồm 3 annotation sau đây:

@EnableAutoConfiguration: bật cơ chế cấu hình tự động

@ComponentScan: bật cơ chế quét các @Bean, @Component ở mọi cấp thư mục

@Configuration: cho phép đăng ký phương thức trả về Bean

9. Khi biên dịch một ứng dụng SpringBoot tải trên mạng, IDE báo lỗi không tìm thấy JDK version phù hợp vậy phải vào file pom.xml để sửa thẻ java.version trong file pom.xml

<properties>

<java.version>14</java.version>

</properties>

10.

Khi muốn trả về dữ liệu kiểu JSON cần phải thêm

@GetMapping(value="/json", produces=MediaType.APPLICATION\_JSON\_VALUE)

11.

Nếu có hai controller khác nhau nhưng có 2 phương thức mà @GetMapping cùng trỏ vào một đường dẫn khi chạy ứng dụng, thì không biên dịch được. Báo lỗi có 2 Controller cùng mapping đến một địa chỉ.

12.

Scope runtime chỉ rằng những thư viện này được biên dịch cùng ứng dụng phục vụ lúc chạy. Còn scope test dùng để chạy kiểm thử unit test.

13.

Để tự động sinh ra các phương thức getter, setters, equalsTo, hash, toString trong Visual Studio Code: chuột phải, menu rồi chọn Source Action

14.

Cách để thay đổi port mặc định của ứng dụng Spring Boot sang port 9000 trong file application.yml:

server:

port: 9000

15.

Có thể định nghĩa một key/value mới trong application.properties

1. Trong một dự án SpringBoot có thể có 2 file cấu hình application.yml và application.properties cùng song song tồn tại.

17.

Ưu tiên dùng private để kiểm soát việc đọc (getter) và ghi (setter).

18.

Trong thư mục dự án Spring Boot, chúng ta thường thấy các thư mực controller, model, service, repository. Hoàn toàn có thể để class XXXController.java ở thư mục cùng cấp với Application.java nhưng việc này không khuyến khích. Hãy để class XXXController.java vào trong thư mục controller.

Việc bố trí thư mục giúp phân loại chức năng của các file java tốt hơn, dễ tìm và quản lý hơn.

19.

Đối tượng Application khởi động trước, đọc các cấu hình vào đặc biệt là cấu hình web server rồi mới khởi động TomCat.

20.

Tính năng NIO trong web server TomCat là Non-blocking I/O. Tính năng này đối lập với blocking I/O.

Non Blocking I/O dùng chung 1 thread để xử lý vào ra nhiều kết nối. Còn Blocking I/O với mỗi kết nối phải tạo ra một thread riêng biệt

21.

Dùng thư viện jackson-databind để convert mảng này sang JSON string.

22.

Bổ sung dependency Thymeleaf:

Sửa file pom.xml bổ xung

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-thymeleaf</artifactId>

<version>2.4.1</version>

</dependency>

Cú pháp Thymeleaf để gán giá trị động vào 2 thuộc tính src và alt của thẻ <img>:

<img th:attr="src=@{'/images/' + ${book.id} + '.jpg'}, alt=${book.title}"/>

24. Ta có một mảng 20 phần tử nhưng khách hàng chỉ muốn hiển thị ra tất cả các phần tử ở vị trí lẻ: 1, 3, 5, 7, 9,.. vậy phải làm thế nào?

Cách đơn giản nhất là tạo một ArrayList mới, duyệt qua mảng ban đầu chọn ra phần tử ở các vị trí lẻ 1, 3, 5.

25.

POJO viết tắt của Plain Old Java Object, có nghĩa là đối tượng Java thuần tuý.