1. **Spring Boot khác gì với Spring Framework?**

Spring boot là 1 nhánh, 1 module của Spring FrameWork.  
 - Về việc cấu hình thì Spring boot được hỗ trợ sẵn nên không phải cấu hình nhiều, phức tạp như Spring FrameWork

- Spring Boot sử dụng Anonation để cấu hình còn Spring FrameWork thì sử dụng XML để cấu hình  
 - Spring Boot có gói sẵn Server như TomCat,… - Spring FrameWork thì không

1. **Những cải tiến Spring Boot so với Servlet JSP trước đây**

1. **JAR khác WAR ở những điểm nào?**

* Java Archive ( JAR) là tệp nén mã nguồn, class - Web Application Archive(WAR) là tệp nén chứa các tệp liên quan đến ứng dụng Web
* JAR chạy file từ command - WAR cần có web Server để chạy
* JAR đóng gói cả web server như TomCat.
* JAR thích hợp để triển khai trong Docker image
* WAR thích hợp để chạy nhiều tệp WAR trên cùng 1 web Server

1. **Maven khác Gradle ở những điểm nào?**

* Maven sử dụng mô hình POM để quản lý thông tin dự án - Gradle sử dụng Groovy, Kotlin DSL ưu tiên tốc độ và sự linh hoạt
* Maven khai báo thông tin quản lý dự án ở file pom.xml - Gradle thì khai báo ở file build.gradle và settings.gradle
* Maven cũ hơn, phổ biến và nhiều bài hướng dẫn hơn - Gradle mới hơn, chạy nhanh hơn
* Maven sử dụng XML cấu hình dài dòng - Gradle sử dụng Script cấu hình ngắn gọn dễ học.

1. **annotation @ResponseBody có ý nghĩa gì?**

Trả lời: Để đánh dấu 1 phương thức sẽ trả về dữ liệu ngay mà không thông qua view Engine

1. **annotation @GetMapping có tác dụng gì đối với phương thức trong Controller?**

Trả lời: Để đánh dấu 1 phương thức sẽ hứng , xử lý request kiểu “GET”.

1. **annotation @Controller thuộc package nào?**

Trả lời : @controller thuộc package `org.springframework.stereotype.Controller`

1. **annotation @SpringBootApplication bao gồm những annotation nào bên trong?**

Trả lời: Bao gồm : @SpringBootConfiguration , @EnableAutoConfiguration , @ComponentScan

1. **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 thuộc tính nào?**

Trả lời : Sửa thuộc tính Java.Version

1. **Khi muốn trả về dữ liệu kiểu JSON cần phải thêm tham số nào vào @GetMapping?**

Trả lời : produces=MediaType.APPLICATION\_JSON\_VALUE

1. **Nếu có hai controller khác nhau nhưng có 2 phương thức mà @GetMapping cùng trỏ vào một đường dẫn khi chạy ứng dụng, điều gì sẽ xảy ra?**

Trả lời: Có 2 trường hợp :

- Trường hợp 1 : Nếu 2 phương thức @GetMapping đó có 1 phương thức có @ResponseBody thì khi chạy chương trình sẽ hứng và xử lý theo phương thức có @ResponseBody

- Trường hợp 2: Cả 2 phương thức đó cùng có hoặc không có @ResponseBody thì khi chạy chương trình sẽ báo lỗi “Cannot map…….”

1. **Trong file pom.xml có nhiều element , trong đó lại có element , thường có 2 giá trị trong scope là runtime và test**

<dependency>

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

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<dependency>

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

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

<scope>test</scope>

</dependency>

Trả lời: thẻ Scope runtime giới hạn việc dependency đó sẽ không được sử dụng trong quá trình compile mà chỉ sử dụng trong runtime

- Thẻ scope test giới hạn việc dependency đó sẽ không được sử dụng trong quá trình runtime mà chỉ sử dụng khi compile và unit test

1. **Khi ta tạo một class Book như dưới đây**

public class Book {

private int id;

private String title;

private String author;

}

Hãy liệt kê các cách trong Visual Studio Code hay IntelliJ để tự động sinh ra các phương thức getter, setters, equalsTo, hash, toString

Trả lời: Dùng Source Action

1. Cách để thay đổi port mặc định của ứng dụng Spring Boot sang port 9000 trong file application.yml như thế nào?

Trả lời : Trong file application.yml gõ cú pháp ``server:

port : 9000``

1. Có thể định nghĩa một key/value mới trong application.properties được không?

Trả lời: Có thể tự định nghĩa bằng cách sử dụng annotation @Value

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

Trả lời: Có thể cùng tồn tại song song.

1. Tại sao các thuộc tính của một class trong Model thường có access modifier là private? Tại không để mặc định hoặc là public cho dễ truy xuất?

public class Book {

private int id; //thường là private

}

Trả lời: Để bảo mật an toàn dữ liệu

1. Trong thư mục dự án Spring Boot, chúng ta thường thấy các thư mực controller, model, service, repository. Hỏi liệu có thể để các file class XXXController.java ở thư mục cùng cấp với Application.java? Việc bố trí thư mục này có ý nghĩa gì?

Trả lời : Có thể .

- Việc bố trí thư mục theo quy tắc chung giúp việc quản lý, bảo trì dễ dàng không gây khó khăn cho các lập trình viên khác. Giúp lập trình viên mới dễ dàng tiếp cận, hiểu dự án, chương trình.

1. Trong tiến trình khởi động ứng dụng Spring Boot, web server TomCat khởi động trước hay là đối tượng Application khởi động trước?

Trả lời: Đối tượng Application khởi động trước

1. Tính năng NIO trong web server TomCat là gì? Câu này cứ Google thôi.

Trình kết nối NIO trong web server TomCat là trình kết nối sử dụng đồng thời 2 nhóm luồng ( threads ) . Một nhóm là poller threads sẽ có một số luồng duy trì giữ kết nối tồn tại với người dùng được kết nối, trong khi đó có bất cứ yêu cầu ( request ) nào được gửi đến thì nhóm Worker Threads sẽ được gọi ra để xử lý request.

1. Ta có một mảng các đối tượng kiểu Book cần trả về kiểu application/json vậy phải dùng thư viện gì để convert mảng này sang JSON string?

Trả lời: Sử dụng thư viện jackson.databind.ObjectMapper

1. Sau khi bạn tạo xong Spring Boot project, bạn phát hiện ra bạn quên không chọn dependency Thymeleaf, vậy bạn phải làm gì để bổ xung dependency này?

Trả lời : Vào file pom.xml điền thêm dependency vào giữa thẻ <dependencies>

<dependencies>

….…

<dependency>

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

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

<version>2.4.0</version>

</dependency>

…...

</dependencies>

1. Hãy mô tả cú pháp Thymeleaf để gán giá trị động vào 2 thuộc tính src và alt của thẻ <img>

<img th:attr=”@{‘đường dẫn’ + car.photo(tên file ảnh) + ‘đuôi file’ }” alt=”${car.name}”>

1. 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? Gợi ý xử lý trong Controller hoặc sử dụng cú pháp Thymeleaf

Trả lời:

- Trong phương thức xử lý của Controller có thể tạo thêm mảng và add nhưng phần tử vị trí lẻ của mảng cũ vào mảng mới .

@GetMapping(value = "/bestsale")

    public String getCar(Model model) {

        Car[] carCollection = { new Car(1, "Triton 4x4 Premium", "Mitsubishi", 869, "triton"),

                new Car(2, "Ranger", "Ford", 1205, "ranger"), new Car(3, "XL7 Premium", "Suzuki", 620, "XL7"),

                new Car(4, "Xe4", "Suzuki", 620, "XL7") };

                Car[] newArr = new Car[carCollection.length/2];

                int j = 0;

                for (int i = 1; i < carCollection.length; i=i+2) {

                    newArr[j] = carCollection[i];

                    j++;

                }

        model.addAttribute(CARS, newArr);

        model.addAttribute(APP\_NAME, appName);

        return "bestsale";

    }

- Sử dụng Thymeleaf each có thêm biến state và đặt điều kiện state.index % 2 ==0 -> sẽ lấy phần tử vị trí lẻ

<ul class="ul-list">

            <th:block th:each="car, state: ${cars}">

            <li th:if="${state.index} % 2 == 0" class="li-list">

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

                <div class="info">

                    <p th:text="${car.name}" class="p-info"></p>

                    <p th:text="${car.manufacturer}" class="p-info"></p>

                    <p th:text="${car.price + ' triệu'}" class="p-info"></p>

                </div>

            </li>

            </th:block>

        </ul>

1. Khái niệm POJO trong java là gì?

POJO (plain old java object) là lớp đối tượng Java thuần túy, không bị hạn chế hay phụ thuộc nghĩa là không extend, implement các lớp, interface chỉ định trước, không chứa các Annotation chỉ định trước. (Thuần túy chỉ dùng Java không dùng các FrameWork)