**BỘ CÂU HỎI PHỎNG VẤN MODULE 4**

**SPRING MVC**

|  |  |
| --- | --- |
| **STT** | **Câu hỏi** |
| **Tổng quan Spring MVC** | |
| **1** | **Vẽ và trình bày cách Spring MVC xử lý request**    Spring Web MVC Cung cấp một Front Controller để xử lý hoặc lắng nghe mỗi khi có request tới ứng dụng  Front controller được coi là trung tâm của một ứng dụng web, cung cấp một điểm truy cập duy nhất cho tất cả các request tới.  Front controller gửi các request tới các bộ xử lí, cho phép các nhà phát triển sử dụng nhiều tính năng khác của framework.  Front controller sẽ chọn lựa các controller phù hợp để chuyển yêu cầu đến controller đó xử lí. Controller này có nhiệm vụ lấy dữ liệu từ request, thực thi các bước nghiệp vụ, và sau đó trả về response cho Front controller tiếp tục xử lí. Response ở đây có thể là tên logic của một view, tên logic của một data model và dẽ liệu thực tế để liên kết.  Thông thường, đối tượng của ModelAndView được trả về Front controller. Front controller sẽ xây dựng view, trả về bằng cách sử dụng View Template và các giá trị đã được gán cho dữ liệu ở Model. Kết quả này sau đó sẽ được trả về cho người dùng. |
| **2** | **Trình bày cơ chế Dependency Injection**  Dependency Injection là kỹ thuật tách một class độc lập với các biến phụ thuộc.  Dependency là một loại quan hệ giữa hai class mà trong đó một class hoạt động độc lập và class còn lại phụ thuộc vào class kia. |
| **3** | **Có bao nhiêu cách để thực hiện Dependency Injection? Trình bày?**   * Constructor injection: Các dependency (biến phụ thuộc) được cung cấp thông qua constructor (hàm tạo lớp). * Setter injection: Các dependency (biến phụ thuộc) sẽ được truyền vào 1 class thông qua các setter method (hàm setter). * Interface injection: Dependency sẽ cung cấp một Interface, trong đó có chứa hàm có tên là Inject. Các client phải triển khai một Interface mà có một setter method dành cho việc nhận dependency và truyền nó vào class thông qua việc gọi hàm Inject của Interface đó. |
| **4** | **Framework là gì ? Framework khác Library chỗ nào ?**      **Giống nhau:** Đều cung cấp các chức năng được viết sẵn để có thể tái sử dụng nhiều lần.  **Khác nhau:**  Mỗi Framework chứa một tập hợp các Library.  Framework mang lại bộ khung, cung cấp những hướng dẫn để chúng ta tuân theo nhưng phải trong phạm vi các tiêu chuẩn của nó. Còn Library chỉ đơn thuần là công cụ tiện ích, có thể thoải mái sử dụng và tuỳ biến. |
| **5** | **Spring Framework là gì ?**  Spring là một framework giúp các nhà phát triển xây dựng những hệ thống và ứng dụng chạy trên JVM một cách đơn giản, tiện gọn, nhanh chóng và mềm dẻo. |
| **6** | **Lợi ích của Spring Framework?**   * Spring cho phép lập trình viên sử dụng POJOs. Việc sử dụng POJOs giúp bạn không phải làm việc với EJB, ứng dụng, các luồng chạy, cấu hình… đơn giản hơn rất nhiều. * Spring được tổ chức theo kiểu mô đun. Số lượng các gói và các lớp khá nhiều, nhưng bạn chỉ cần quan tâm đến những gì bạn cần và không cần quan tâm đến phần còn lại. * Spring hỗ trợ sử dụng khá nhiều công nghệ như ORM Framework, các logging framework, JEE, các thư viện tạo lịch trình (Quartz và JDK timer)… * Module Web của Spring được thiết kế theo mô hình MVC nên nó cung cấp đầy đủ các tính năng giúp thay thế các web framework khác như Struts. |
| **7** | **Nguyên lý đảo ngược quyền điều khiển (Inversion of Control) là gì ?**  Inversion of Control (IoC) dịch là đảo ngược điều khiển. Ý của nó là làm thay đổi luồng điều khiển của ứng dụng, giúp tăng tính mở rộng của một hệ thống. |
| **8** | **Giải thích IoC? Lợi ích của IoC ?**   * IoC là một nguyên lý trong phát triển phần mềm, trong đó việc điều khiển các đối tượng hoặc các thành phần của hệ thống được thực hiện bởi framework hoặc các container * IoC cho phép framework nắm giữ quyền điều khiển luồng thực thi của hệ thống và gọi các mã nguồn khác * Lợi ích:   + Tách rời việc thực thi (execution) và việc triển khai (implementation)   + Dễ chuyển đổi giữa các implementation   + Dễ phân tách module hơn   + Dễ kiểm thử hơn, bằng cách tách rời các thành phần riêng lẻ |
| **9** | **Bean là gì?**  Một bean là một đối tượng được khởi tạo, lắp ráp, và được quản lý bởi một Spring IoC container. Các bean này được tạo ra bằng siêu dữ liệu cấu hình mà bạn cung cấp cho container, ví dụ dưới dạng định nghĩa XML <bean/>. |
| **10** | **Trong Spring có bao nhiêu Bean Scope?**  Có 5 scope được định nghĩa cho Spring Bean:   * Singleton: Chỉ duy nhất một thể hiện của bean sẽ được tạo cho mỗi container. Đây là scope mặc định cho spring bean. Khi sử dụng scope này cần chắc chắn rằng các bean không có các biến/thuộc tính được share. * Prototype: Một thể hiện của bean sẽ được tạo cho mỗi lần được yêu cầu(request) * Request: giống với prototype scope, tuy nhiên nó dùng cho ứng dụng web, một thể hiện của bean sẽ được tạo cho mỗi HTTP request. * Session: Mỗi thể hiện của bean sẽ được tạo cho mỗi HTTP Session * Global-Session: Được sử dụng để tạo global sesion bean cho các ứng dụng Portlet. |
| **11** | **@Autowire là gì?**  @Autowired đánh dấu cho Spring biết rằng sẽ tự động inject bean tương ứng vào vị trí được đánh dấu |
| **12** | **@Component có ý nghĩa gì?**  @Component là một Annotation (chú thích) đánh dấu trên các Class để giúp Spring biết nó là một Bean. |
| **Spring Controller** | |
| **13** | **Trình bày ý nghĩa của Controller**  Đọc câu 1.  Spring controller nhận ủy quyền xử lý các request từ phía front controller. Công việc cơ bản của nó là nhận dữ liệu cần thiết từ front controller, xác định view cần sử dụng, triệu gọi lớp xử lý nghiệp vụ cần thiết, sắp xếp dữ liệu mà view cần đến vào model, gửi thông tin về view cùng model lại cho front controller. |
| **14** | **Trình bày ý nghĩa của ModelAndView Interface**  Class ModelAndView có nhiệm vụ quan trọng trong việc tạo ra các hồi đáp (response). Các thể hiện của ModelAndView giúp dễ dàng tạo liên kết giữa data model và view. Trong đó data model là đối tượng chứa dữ liệu để hiển thị lên view. |
| **15** | **Trình bày ý nghĩa của ModelMap Interface**  Class ModelMap là lớp con của LinkedHashMap được sử dụng trong việc xây dựng data model bằng sử dụng kiểu dữ liệu cặp gồm key và value. ModelMap sử dụng phương thức addAttribute() để tạo liên kết giữa model và tên logic trong model. Thuộc tính đặt trong ModelMap có thể sử dụng để hiển thị dữ liệu dạng form trong sự kiện submit form. |
| **16** | **Trình bày ý nghĩa của ViewResolver Interface**  Các ViewResolver (trình phân giải view) là các đối tượng chịu trách nhiệm lấy các đối tượng View cho một hành động và vị trí cụ thể. Thông thường, các controller yêu cầu ViewResolvers chuyển tiếp đến một view với một tên cụ thể (một chuỗi được trả về bởi phương thức). Sau đó, tất cả các view resolver trong ứng dụng thực thi theo thứ tự cho đến khi một trong số đó có thể phân giải view trong trường hợp một đối tượng View được trả về và điều khiển được truyền tới nó cho việc render HTML. |
| **17** | **Định nghĩa URI với các phương thức khác nhau như GET, POST, PUT, PATH, DELETE** |
| **18** | **Phân biệt POST với GET** |
| **19** | **Phân biệt POST với PUT** |
| **20** | **Thao tác với form trong ứng dụng Spring MVC** |
| **21** | **@RequestMapping làm gì?**  Đánh dấu URI |
| **22** | **Trình bày cơ chế Data Binding** |
| **23** | **Thuộc tính consumes trong các Request Mapping là gì ?** |
| **Thymeleaf** | |
| **24** | **Template là gì? Template Engine là gì? Thymeleaf là gì?**  Template định nghĩa một khuôn mẫu sẵn, sau đó ứng dụng sẽ đổ dữ liệu vào template để tạo ra một response thực tế và trả về cho người dùng.  Thymeleaf là thư viện, cung cấp các cách thức để xây dựng khuôn, khung mẫu. |
| **25** | **Sử dụng lặp trong Thymeleaf như thế nào? Sử dụng điều kiện trong Thymeleaf như thế nào?**  Thêm thư viện của Thymeleaf và dùng theo ý muốn |
| **Data Binding** | |
| **26** | **Data Binding là gì?** |
| **27** | **Formatter là gì? Converter là gì?** |
| **ORM và JPA** | |
| **28** | **ORM là gì?**  ORM (Object-Relational Mapping) là kỹ thuật liên kết giữa các đối tượng trong lập trình với các đối tượng trong CSDL |
| **29** | **JPA là gì? Spring Data JPA là gì ?**   * Java Persistence API, là một giao diện cung cấp các interface. Có các method hỗ trợ tương tác với CSDL. * Spring Data JPA Là 1 thư viện phát triển các method JPA hoàn chỉnh nhất * Hỗ trợ cho làm việc với Spring * Điều kiện của Spring Data JPA có thể hoạt động là phải hoạt động trên 1 framework đang triển khai JPA |
| **30** | **Hibernate là gì? Hibernate làm gì với database**  Hibernate là một thư viện ORM (Object Relational Mapping) mã nguồn mở giúp lập trình viên viết ứng dụng Java có thể map các objects (pojo) với hệ quản trị cơ sở dữ liệu quan hệ, và hỗ trợ thực hiện các khái niệm lập trình hướng đối tượng với cớ dữ liệu quan hệ. |
| **31** | **Liệt kê một số annotation của hibernate?** |
| **32** | **Entity là gì?**  Là thực thể trong CSDL, tương ứng với 1 Class |
| **33** | **Persistence Context & Entity Manager là gì?**  Persistence Context là tập các thể hiện của entity được quản lý, tồn tại trong một kho dữ liệu.  Persistence context có thể được coi như một container hoặc là cache cấp đầu tiên cho tất cả các đối tượng mà chúng ta đã load hoặc lưu vào cơ sở dữ liệu trong một session. |
| **34** | **Câu lệnh truy vấn động là gì? Câu lệnh truy vấn tĩnh là gì?**  Câu lệnh truy vấn tĩnh là câu lệnh truy cấn không có tham số truyền vào và ngược lại |
| **35** | **Trạng thái của Entity bao gồm những gì?**  New (Mới tạo ra, chưa được lưu xuống CSDL)  Persistent (được ghi xuống CSDL và quản lý trong Persistence Context)  Detached (tạm thời tách ra khỏi Persistence Context)  Removed: |
| **36** | **Có bao nhiêu loại mapping trong Hibernate?**  1-n, n-1, n-n, 1-1 |
| **37** | **SessionFactory là gì?**  Quản lý việc tạo ra session khi cần thao tác với CSDL  Là một interface giúp tạo ra session kết nối đến database bằng cách đọc các cấu hình trong Hibernate configuration. |
| **38** | **Liệt kê ba trạng thái của đối tượng trong hibernate?**   * Transient: Đối tượng không có quan hệ với Session hiện tại của Hibernate. * Persistent: Đối tượng có quan hệ với Session hiện tại của Hibernate và trạng thái của nó được đồng bộ với cơ sở dữ liêu * Detache: Đối tượng đã từng có trạng thái persistent nhưng hiện tại đã không còn giữ quan hệ với Session… |
| **Validation** | |
| **39** | **Validation dữ liệu là gì?**  Là kiểm tra tính hợp lệ |
| **40** | **Trình bày cách triển khai validate dữ liệu trong Spring.**  Thêm thư viện, dùng annotation |
| **41** | **Binding Result là gì?**  Là nơi để chứa kết quả của việc kiểm tra tính hợp lệ (có thể lỗi hay không lỗi) |
| **AOP** | |
| **42** | **AOP là gì? AOP gồm những yếu tố nào?**  Aspect Oriented Programming (AOP) – Lập trình hướng khía cạnh: là một kỹ thuật lập trình (kiểu như lập trình hướng đối tượng) nhằm phân tách chương trình thành cách moudule riêng rẽ, phân biệt, không phụ thuộc nhau. |
| **43** | **Các loại Advice?**  @Before : chạy trược method  @After: Chạy trong 2 trường hợp method chạy thành công hay có exception  @AfterReturning: Chạy khi method chạy thành công  @AfterThrowing: Chạy khi method có exception |
| **44** | **JoinPoint là gì?** |
| **45** | **Nêu cách triển khai AOP mà bạn dùng trong dự án** |
| **46** | **Xử lý ngoại lệ trong Spring Web MVC như thế nào?** |
| **Session & Cookie** | |
| **47** | **Cookie là gì?**  Cookie là những tập tin một trang web gửi đến máy người dùng và được lưu lại thông qua trình duyệt khi người dùng truy cập trang web đó. |
| **48** | **Phân biệt Session và Cookie** |
| **49** | **Nêu cách triển khai Session mà bạn dùng trong dự án.**  Dùng annotation hoặc dùng HttpSession |
| **Web Service** | |
| **50** | **Web Service là gì? Lấy ví dụ về web service**   * Web Services là một ứng dụng có thể truy cập thông qua giao thức Web chuẩn (HTTP hoặc HTTPs). Web Services được thiết kế để giao tiếp với các chương tình khác (chứ không phải với User). * Ví dụ: GG map |
| **51** | **Phân biệt Web Service và Web truyền thông**   |  |  | | --- | --- | | **Website** | **Web Service** | | 1. Có UI  2. Người dùng dùng  3. Đa nền tảng (trình duyệt và hệ điều hành khác nhau)  4. Cách thức tương tác : thành phần trong giao diện (button, textbox....)  5. Dữ liệu đầu ra: trang HTML - dễ đọc hiểu  (được tinh chỉnh để cho user đọc) | 1. Không có UI.  2. Ứng dụng dùng  3. Độc lập nền tảng (không quan tâm đến nền tảng nào,  chỉ quan tâm đến giao thức và chuẩn đầu ra)  4. Cách thức tương tác: Truy cập bởi các phương thức HTTP  5. Dữ liệu đầu ra: JSON / XML.  Cần được tinh chỉnh để xuất ra thì user mới đọc được. | |
| **52** | **REST là gì? RESTful là gì?**  Là viết tắt của REpresentational State Transfer (là chuyển trạng thái đại diện)  Là một kiểu kiến trúc lập trình định nghĩa các Quy tắc để thiết kết các web services.  Những Web services đạt chuẩn rest thì được gọi là restful. |
| **53** | **jQuery là gì? Ajax là gì?**  **jQuery** là một thư viện được viết từ Js.  **Ajax** là một nhóm các công nghệ phát triển web được sử dụng để tạo các ứng dụng web giàu tính tương tác, nhanh hơn và mượt mà hơn với sự giúp đỡ của XML, HTML, CSS và JavaScript. |
| **54** | **Nêu ví dụ về cách bạn triển khai Ajax trong dự án của bạn**  Dùng hàm $.ajax({}) để tương tác với CSDL (thêm, sửa, xóa, tìm kiếm) |
| **55** | **@RequestHeader và @ResponseHeader có ý nghĩa gì**  @RequestHeader được sử dụng khi ta muốn lấy dữ liệu được truyền bằng Header của một request (yêu cầu từ client).8  Chúng ta sử dụng @ResponseHeader khi mình muốn trả về thêm dữ liệu cho client ở phần trên cùng của mỗi response. |
| **I18N** | |
| **56** | **I18N và I10N là gì?**  Internationalization (I18N) |
| **57** | **Nêu cách triển khai I18N và I10N trong dự án Spring** |
| **58** | **Interceptor là gì?** |
| **Bảo mật** | |
| **59** | **Bảo mật là gì? Cơ chế bảo mật trong Spring như thế nào?**   * Spring Security là một dự án nổi bật trong hệ sinh thái Spring, cung cấp các dịch vụ bảo mật toàn diện cho các ứng dụng doanh nghiệp có nền tảng Java EE, là một framework tập trung vào việc cung cấp khả năng xác thực (authentication) và phân quyền (authorization) cho ứng dụng Java. * Cơ chế bảo mật trong Spring: * Xác thực (Authentication) * Phân quyền (Authorization) * Phân quyền hệ thống tệp tin * Quản lý tài khoản * Sử dụng SSL để truyền dữ liệu * Mã hoá Firewalls |
| **60** | **Authentication là gì ? Authorization là gì ?**  **Authentication** (xác thực) là một hành động nhằm thiết lập hoặc chứng thực một thông điệp hoặc đối tượng nào đó là đáng tin cậy.  **Authorization** (Cấp quyền/phân quyền) là quá trình xác định xem một người dùng có quyền truy cập một tài nguyên cụ thể để thực hiện một số hành động hay không. |
| **61** | **Liệt kê các cơ chế xác thực**   * HTTP Basic: là một phương thức để client cung cấp username và password khi thực hiện yêu cầu. * Username và password được nối vào một chuỗi duy nhất: username:password * Chuỗi này được mã hóa với Base64 * Từ khóa Basic được đặt trước giá trị được mã hóa này * Cookies * Tokens * JWT (JSON Web Token) là 1 tiêu chuẩn mở (RFC 7519) định nghĩa cách thức truyền tin an toàn giữa các thành viên bằng 1 đối tượng JSON. Thông tin này có thể được xác thực và đánh dấu tin cậy nhờ vào "chữ ký" của nó. Phần chữ ký của JWT sẽ được mã hóa lại bằng HMAC hoặc RSA. * JWT bao gồm ba phần: * Tiêu đề, chứa loại mã thông báo và thuật toán băm * Tải trọng, chứa các xác nhận quyền sở hữu * Chữ ký, có thể được tính như sau nếu bạn chọn HMAC SHA256:HMACSHA256( base64UrlEncode(header) + "." + base64UrlEncode(payload), secret) * Nhược điểm: Cần phải nỗ lực nhiều hơn để giảm thiểu các cuộc tấn công XSS. * Signature (chữ kí số) * Chữ ký số khóa công khai là mô hình sử dụng các kỹ thuật mật mã để gắn với mỗi người sử dụng một cặp khóa công khai - bí mật và qua đó có thể ký các văn bản điện tử cũng như trao đổi các thông tin mật. Khóa công khai thường được phân phối thông qua chứng thực khóa công khai. * Quá trình sử dụng chữ ký số bao gồm 2 quá trình: tạo chữ ký và kiểm tra chữ ký. * One-time password (OTP – mật khẩu dùng 1 lần) |
| **62** | **CRSF là gì** |
| **63** | **CORS là gì**  CORS là một cơ chế cho phép nhiều tài nguyên khác nhau (fonts, Javascript, v.v…) của một trang web có thể được truy vấn từ domain khác với domain của trang đó. |
| **Spring Boot** | |
| **64** | **Spring Boot là gì?**    Spring Boot là một dự án phát triển bởi JAVA trong hệ sinh thái Spring framework. Nó giúp cho các lập trình viên chúng ta đơn giản hóa quá trình lập trình một ứng dụng với Spring, chỉ tập trung vào việc phát triển business cho ứng dụng. |
| **65** | **Sự khác nhau giữa SpringBoot và SpringMVC là gì?**   |  |  | | --- | --- | | **Spring Boot** | **Spring MVC** | | Spring Boot là một mô-đun của Spring để đóng gói ứng dụng dựa trên Spring với các giá trị mặc định hợp lý. | Spring MVC là model-view-controller web framework trong khuôn khổ Spring framework. | | Cung cấp các cấu hình mặc định để xây dựng khung công tác được cung cấp bởi Spring. | Cung cấp các tính năng sẵn sàng sử dụng để xây dựng một ứng dụng web. | | **Không cần phải xây dựng cấu hình theo cách thủ công.** | **Nó yêu cầu cấu hình xây dựng theo cách thủ công.** | | **Không yêu cầu** đối với bộ mô tả triển khai. | **Cần** có bộ mô tả Triển khai. | | Nó tránh mã viết sẵn và kết hợp các phần phụ thuộc lại với nhau trong một đơn vị duy nhất. | Nó chỉ định từng phụ thuộc riêng biệt. | | Nó làm giảm thời gian phát triển và tăng năng suất. | Cần nhiều thời gian hơn để đạt được điều tương tự. | |
| **66** | **Trong trường hợp bạn muốn tuỳ chỉnh cấu hình cho Spring Boot chỉ tìm kiếm các bean trong một package nhất định.**  @ComponentScan(basePackages = "com.baeldung.componentscan.springapp.animals") |
| **67** | **Cấu hình cho Spring Boot Tìm kiếm các Bean ở nhiều package khác nhau bằng cách nào?**  1. Sử dụng @ComponentScan  2. Sử dụng scanBasePackages tromg @SpringBootApplication. |
| **68** | **Trình bày Spring Security? @EnableWebSecurity làm gì?**  **@EnableWebSecurity** tìm và tự động áp dụng security lên các class trong Spring |
| **69** | **Trình bày Spring Boot JPA? Cài đặt?**  Thêm thư viện, đặt anotation repository, kế thừa JpaRepository |
| **70** | **Để sử dụng interface JpaRepository cần cung cấp những thông tin nào?**  Tên class và kiểu dữ liệu của khóa chính |
| **71** | **Bạn có biết annotation @ConfigurationProperties? Trình bày?**  @ConfigurationProperties annotation cho phép truy xuất các thuộc tính được đặt trong các tệp cấu hình một cách tự động và mạnh mẽ. |