**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: |
| 2 | * - Client Request -> Front Controller -> Controller -> Front Controller -> View -> Front Controller -> Client Spring Font Controller (DispatcherServlet) nhận yêu cầu nó sẽ tìm đến Handler Mapping thích hợp. * - Handler Mapping sẽ ánh xạ yêu cầu của client đến controller thích hợp. * - Sau đó DispatcherServlet sẽ gửi yêu cầu đến Controller thích hợp.   - Sau khi thực hiện tiến trình từ yêu cầu của client, nó thực thi các logic được xác định trong controller và cuối cùng trả về đối tượng ModelAndView.  - Dựa trên các giá trị trong ModelAndView Controller mà nó sẽ view, có thể là JSP, FreeMaker,tiles do file cấu hình của chúng ta config.  - Controller sẽ gửi đến cho Servlet container và cuối cùng Servlet container sẽ gửi kết quả cho người dùng.  Trình bày cơ chế Dependency Injection: |
| 3 | - DI là một cách thực hiện của IoC (Inversion of Control: đảo ngược quyền điều khiển), Design Pattern cho phép xóa bỏ sự phụ thuộc giữa các module.  - Vậy có thể hiểu Injection là mình giao cho một ai đó (ví dụ spring framework chẳng hạn) khởi tạo object và nhúng object đó và các Dependency (module) ( hay được gọi là bean) đang dùng.  - Khi 1 ứng dụng Spring chạy thì IOC Container sẽ thực hiện quét các class được đánh dấu là bean và tạo ra 1 objecr singleton, bỏ vào IOC Container. Khi có 1 bean phụ thuộc vào 1 bean khác thì IOC sẽ tìm trong Container, nếu chưa có thì tạo, nếu có rồi thì lấy ra và inject vào bean cần nó.  Có bao nhiêu cách để thực hiện Dependency Injection? Trình bày? |
| 4 | + Triển khai DI có 3 cách: Interface, Constructor, Setter.  - Constructor: Container sẽ gọi 1 constructor với các tham số, mỗi tham số đại diện cho 1 dependency. Spring nhận biết các tham số thông qua kiểu dữ liệu, tên của thuộc tính và chỉ số của tham số.  Ví dụ: Class Teacher {  private Student student;  public Teacher(Student) { this.student = student };}  - Setter: Container sẽ gọi các phương thức setter để khởi tạo 1 bean. (inject từng thuộc tính)  Inject từng phần: bạn có thể inject từng thuộc tính bằng setter injection nhưng không thể làm với constructor.  Overriding: Setter injection ghi đè lại constructor injection, nếu ta dùng cả constructor và setter injection, IoC container sẽ sử dụng setter injection.  \* Chuyển đổi: Ta có thể dễ dàng thay đổi giá trị bằng setter injection mà ko cần tạo một thể hiện mới của bean.  (Lưu ý: nếu sử dụng setter injection thì class của bạn phải có hàm khởi tạo mặc định không tham số, nếu dùng constructor injection thì phải có hàm khởi tạo với các tham số tương ứng với injection).  Framework là gì ? Framework khác Library chỗ nào ? |
| 5 | - Framework là một phần mềm có tính trừu tượng cao, cung cấp các chức năng chung - thông dụng nhất và có khả năng tùy biến.  - Điểm khác biệt so với Library:  + Library: lập trình viên sẽ nắm quyền điều khiển (Code của mình sẽ gọi code thư viện)  + Framework: framework sẽ nắm quyền ứng dụng (Code của Framework sẽ gọi code của mình)  => Điểm khác biệt chính giữa Framework và Library là cơ chế IoC.  Spring Framework là gì ? |
| 6 | - Spring Framework là một framework dùng để phát triển các ứng dụng Java, mã nguồn mở, sử dụng PoJO, Dependency Injection.  + Gồm các thành phần: Test, Spring Core (Beans, Core, Context, SpEL), AOP (Aspect Oriented Programming), Data Access(JDBC, ORM), Web(Web Socket, Servlet, Web, Portler)  Lợi ích của Spring Framework? |
| 7 | - Nhẹ, giúp các ứng dụng có hiệu năng cao, dễ kiểm thử, tái sử dụng code,…  - Spring cho phép lập trình viên sử dụng POJOs nên ứng dụng các luồng chạy, cấu hình,… đơn giản hơn so với EJB.  - Hỗ trợ nhiều công nghệ như ORM FrameWork, J2EE  Nguyên lý đảo ngược quyền điều khiển (Inversion of Control) là gì ? |
| 8 | - IoC là nguyên lý trong phát triển phần mềm. Trong đó, việc điều khiển các đối tượng hoặc thành phần của hệ thống sẽ được thực hiện bởi Framework hoặc các Container Giải thích IoC? Lợi ích của IoC ? |
| 9 | Bean là gì? |
| 10 | - Bean là đối tượng được Framework và Container quản lý.  - Trong Spring, Bean là một đối tượng được khởi tạo, lắp ráp và quản lý bởi Spring IoC Container.  - Các cách tạo bean: dùng xml, dùng @Bean Trong Spring có bao nhiêu Bean Scope? |
| 11 | **+ Singleton Scope**: Khi một Bean được khai báo là Singleton thì Bean đó là duy nhất trong Spring IoC và được share cho tất cả các Beans khác nếu cần sử dụng nó.  **+ Prototype Scope**: Khác với Singleton Scope, Bean (đối tượng) sẽ được tạo ra mới mỗi khi có một yêu cầu tạo Bean. Như vậy mỗi lần gọi tới Bean mà có Scope là Prototype thì nó sẽ tạo ra một đối tượng (Bean) trong Spring IoC container.  **+ Request Scope:** Spring Container sẽ tạo bean (đối tượng) mới khi có một request (yêu cầu) từ người dùng. Sau khi Request (yêu cầu) xử lý xong thì Bean sẽ bị xoá đi.  **+ Session Scope:** Scope Session sẽ tồn tại chừng nào Sesion ở HTTP. Nó sẽ bị xoá đi khỏi Spring IoC khi Session ở Web bị xoá hoặc hết hiệu lực.  **+ Application scope:** Application Scope được tạo một lần cho toàn bộ ứng dụng Web Application. Application Scope được chứa đựng như một ServletContext, nó cũng gần tương tự như Singleton Scope nhưng nó là Singleton cho từng ServeletContext. @Autowire là gì? |
| 12 | - @Autowire đánh dấu cho Spring biết rằng sẽ tự động inject bean tương ứng vào vị trí được đánh dấu  @Component có ý nghĩa gì? - @Component dùng để đánh dấu một class Java là một Bean |
| **Spring Controller** | |
| **13** | Trình bày ý nghĩa của Controller |
| 14 | - Đón nhận request, trả về respone (data model + view (tên vật lý -> view resolver) cho Front controller. Trình bày ý nghĩa của ModelAndView Interface |
| 15 | - ModelAndView Interface là sự kết hợp của ModelMap và View. Nó cho phép Controller trả về một giá trị bao gồm Model và View. Trình bày ý nghĩa của ModelMap Interface |
| 16 | - ModelMap cho phép truyền dữ liệu từ Controller sang View để hiển thị. ModelMap có các tính chất của Map Trình bày ý nghĩa của ViewResolver Interface |
| 17 | - ViewResolver Interface là trình phân giải view cho phép ánh xạ "tên" của View sang đối tượng của View tương ứng. Định nghĩa URI với các phương thức khác nhau như GET, POST, PUT, PATH, DELETE |
| 18 | Phân biệt POST với GET |
| 19 | |  |  | | --- | --- | | Get | Post | | Kém bảo mật (vì hiển thị thông tin trên URL). | Bảo mật (không hiển thị thông tin trên URL). | | Giới hạn số lượng ký tự hiển thị <2048 | Không giới hạn ký tự | | Chỉ gữi được dữ liệu dạng text | Có thể gữi giá trị dạng binary (file, images, …) | | Có thể truy cập được từ thanh address của trình duyệt. | Không thể truy cập được từ thanh address của trình duyệt. |   Phân biệt POST với PUT |
| 20 | + POST: thêm mới dữ liệu  + PUT: chỉnh sửa dữ liệu Thao tác với form trong ứng dụng Spring MVC |
| 21 | @RequestMapping làm gì? |
| 22 | - @RequestMapping được dùng để ánh xạ các request tới Controller tương ứng. Trình bày cơ chế Data Binding |
| 23 | - Data Binding là cơ chế liên kết dữ liệu đầu vào và đầu ra với một đối tượng Model. Thuộc tính consumes trong các Request Mapping là gì ? - Thuộc tính consumes được dùng để định dạng dữ liệu khi client request lên server. |
| **Thymeleaf** | |
| 24 | Template là gì? Template Engine là gì? Thymeleaf là gì? |
| 25 | - Thymeleaf là Java template engine phía máy chủ, một bộ xử lý view, được sử dụng cho các ứng dụng web và các ứng dụng độc lập. - Template là tệp đựng layout được thiết kế sẵn.  - Được xây dựng phù hợp với tiêu chuẩn web, hỗ trợ HTML5  - Thymeleaf cho phép xử lý các loại templates HTML, TEXT, XML, JAVASCRIPT, CSS,... Sử dụng lặp trong Thymeleaf như thế nào? Sử dụng điều kiện trong Thymeleaf như thế nào?  th:each="loop : ${loopList}";  th:if="${condition}"; |
| **Data Binding** | |
| 26 | Data Binding là gì? |
| 27 | -Là cơ chế liên kết dữ liệu đầu vào và đầu ra của đối tượng model. Các form đều được liên kết với một đối tượng biểu diễn dữ liệu phía sau -> dữ liệu trên form sẽ được tự động chuyển đổi thành các thuộc tính của đối tượng liên kết với form.  -Data binding hoạt động dựa trên đối tượng data binder  + Formatters xử lý các format như dateTime, …  + Validator  + Property Editors: chuyển data từ text input -> kiểu dữ liệu, thuộc tính mong muốn.  Formatter là gì ? Converter là gì ? - Formatter cũng hoạt động giống như converter, tức là chuyển đổi một kiểu dữ liệu sang kiểu dữ liệu khác. Tuy nhiên, kiểu dữ liệu nguồn của Formatter là String, trong khi đó converter có thể làm việc với bất cứ kiểu dữ liệu nguồn nào. |
| **ORM và JPA** | |
| 28 | ORM là gì ? |
| 29 | - ORM (Object Relational Mapping) là kỹ thuật liên kết đối tượng trong lập trình và đối tượng trong CSDL.  => Là quá trình biến đổi giữa một record trong Database và một Object trong ngôn ngữ lập trìnhJPA là gì? Spring Data JPA là gì ? |
| 30 | - JPA (Java Persistence API) là một bộ đặc tả tiêu chuẩn của Java, trong đó mô tả các thao tác quản lý dữ liệu quan hệ trong ứng dụng Java. Cung cấp mô hình cho phép ánh xạ các bảng trong CSDL với các lớp.  - Spring Data JPA là một module của Spring Framework giúp cải tiến JPA tiêu chuẩn, đơn giản hóa tầng truy xuất dữ liệu, tự tạo ra Repository, tạo các truy vấn JPA thông qua tên phương thức. Hibernate là gì ? Hibernate làm gì với database |
| 31 | - Hibernate là một ORM Framework mã nguồn mở giúp lập trình viên viết ứng dụng Java có thể map các Object với hệ CSDL, và hỗ trợ thực hiện các khái niệm OOP với CSDL.  - Hibernate có thể hỗ trợ tạo bảng và các dữ liệu thông qua ngôn ngữ Java. Liệt kê một số annoutation của hibernate? |
| 32 | - @Entity, @Table, @Id, @GenerateValue, @Column, @OneToOne, @OneToMany, @ManyToOne, @ManyToMany, … Entity là gì? |
| 33 | - Entity là đối tượng đại diện cho dữ liệu trong ứng dụng.  - Mỗi Entity sẽ được mapping với một table trong CSDL.  Mỗi Entity cần tuân thủ:  + Được gắn annotation @Entity  + Có constructor public rỗng  + Không được khai báo final  + Thuộc tính của đối tượng có access modifier là private, protected hoặc ở mức package private Persistence Context & Entity Manager là gì? |
| 34 | - Persistence Context là một tập các thể hiện của Entity, được dùng quản lý Entity trong một kho dữ liệu.  - Entity Manager là một interface cung cấp các phương thức cho việc tương tác với các Entity.  Các phương thức thông dụng:  + persit  + merger  + remove  + find Câu lệnh truy vấn động là gì? Câu lệnh truy vấn tĩnh là gì? |
| 35 | - Câu lệnh truy vấn động là câu truy vấn có các tham số thay đổi.  - Câu lệnh truy vấn tĩnh là câu truy vấn có các tham số cố định. Trạng thái của Entity bao gồm những gì? |
| 36 | + New: tạo mới, 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: xóa những Detached Entity.   Có bao nhiêu loại mapping trong Hibernate? + Có 3 loại mapping trong Hibernate:  - Association Mapping  + OneToOne  + OneToMany  + ManyToOne  + ManyToMany  - Collection Mapping: Set, SortedSet, List, Collection, Map, SortedMap  - Component Mapping |
| 37 | SessionFactory là gì? |
| 38 | - Là một interface giúp tạo ra Session kết nối đến Databate bằng cách đọc các cấu hình trong Hibernate Configuration. Liệt kê ba trạng thái của đối tượng trong hibernate? - detached, persistent và transient. |
| **Validation** | |
| 39 | Validation dữ liệu là gì? |
| 40 | - Validation là thực hiện đánh giá/xác minh tính hợp lệ của dữ liệu đầu vào. Trình bày cách triển khai validate dữ liệu trong Spring |
| 41 | 1. Implement interface Validator vào model cần Validate.  2. Overide các phương thức của Interface theo hình thức regex dữ liệu.  3. Sử dụng các annotation Validate có sẵn: @NotNull, @NotEmpty, @NotBlank,…  4. Sử dụng @Validate, BindingResult ở Controller để thực hiện trả về kết quả khi có lỗi. Binding Result là gì? - Binding Result là một đối tượng chứa các lỗi trong quá trình Validate |
| **AOP** | |
| 42 | AOP là gì? AOP gồm những yếu tố nào? |
| 43 | - AOP (Aspect Oriented Programming) là lập trình hướng khía cạnh, là các đoạn chương trình độc lập với nghiệp vụ dự án (bổ trợ OOP).  AOP gồm các yếu tố:  + Aspect: dùng để khai báo cho các đoạn chương trình độc lập, không liên quan gì đến nghiệp vụ dự án.  + Advice: là các hành động mà Aspect dùng để quan sát OOP.  + Pointcut: là tập hợp các Joinpoint có khả năng xảy ra các hoạt động Advice.  + Joinpoint: những điểm xảy ra Advice. Các loại Advice? |
| 44 | Có 5 loại:  + Before: hoạt động trước khi Joinpoint thực thi.  + AfterReturning: hoạt động sau khi Joinpoint thực thi và trả về giá trị.  + AfterThrowing: hoạt động sau khi Joinpoint thực thi và ném ngoại lệ.  + After: hoạt động khi Joinpoint hoàn thành.  + AroundAdvice: thực thi cả trước và sau Joinpoint. JoinPoint là gì? |
| 45 | - Joinpoint là những điểm xảy ra Advice. Nêu cách triển khai AOP mà bạn dùng trong dự án |
| 46 | 1. Khai báo class AOP và đánh dấu bằng annotation @Aspect, @Component.  2. Khai báo cách Advice với các annotation: @Before, @AfterReturning, @AfterThrowing, @After, @Around  3. Khai báo Pointcut bằng từ khóa "excution".  4. Viết phương thức Advice khi Joinpoint xảy ra. Xử lý ngoại lệ trong Spring Web MVC như thế nào? - Sử dụng annotion @ExceptionHandler hoặc Try/Catch để xử lý các ngoại lệ nghiệp vụ, HandlerExceptionResolver để xử lý ngoại lệ tại request, sử dụng error-page để xử lý lỗi fatal. |
| **Session & Cookie** | |
| 47 | Cookie là gì? |
| 48 | - Cookie là những tệp được trang web người dùng truy cập tạo ra. Cookie giúp trải nghiệm trực tuyến dễ dàng hơn.      Phân biệt Session và Cookie |
| 49 | |  |  | | --- | --- | | Cookie | Session | | Cookie được lưu trữ trên trình duyệt. | Session không được lưu trữ trong trình duyệt của người dùng. | | Dữ liệu cookie được lưu ở phía máy khách | Dữ liệu session chứa thông tin người dùng và được lưu trữ ở phía server. | | Dữ liệu cookie dễ dàng sửa đổi hoặc bị đánh cắp. | Dữ liệu session không dễ dàng sửa đổi vì chúng được lưu trữ ở phía máy chủ. | | Dữ liệu cookie có sẵn trong trình duyệt của chúng ta đến khi hết hạn (setMaxAge()) | Dữ liệu session có sẵn cho trình duyệt chạy. Sau khi đóng trình duyệt sẽ mất thông tin. |   Session phụ thuộc vào cookie. <jsession id>  Nêu cách triển khai Session mà bạn dùng trong dự án 1. Khởi tạo đối tượng Session cùng annotation @ModelAttribute  2. Dùng annotation @SessionAttributes để lưu trữ thông tin của Model Attribute  3. Sử dụng annotation @CookieValue để ràng buộc giá trị của cookie HTTP với tham số phương thức trong Controller.  4. Tạo Cookie và trả về cho Client |
| **Web Service** | |
| 50 | Web Service là gì? Lấy ví dụ về web service |
| 51 | + Web Service là một dịch vụ web cung cấp nền tảng chung cho nhiều ứng dụng. Những ứng dụng này có thể được viết bằng nhiều nền tảng công nghệ khác nhau nhưng có thể giao tiếp với nhau được.  + Hoạt động dựa trên mô hình Client-Server, thông qua giao thức HTTP, HTTPs  + Kiểu dữ liệu dùng để giao tiếp là XML và JSON.  + Web service (Dịch vụ web) là các thành phần ứng dụng được hiển.  thị dưới dạng các dịch vụ trên WWW. Phân biệt Web Service và Web truyền thông |
| 52 | - Web Application:  + Có giao diện.  + Kết quả trả về cho người dùng là view.  + Tương tác giữa User và ứng dụng thông qua form, link, image, button,…  + Thông thường được tương tác giữa User & ứng dụng.  - Web Service:  + Không có giao diện.  + Kết quả trả về chỉ có data (theo format XML hoặc JSON) => User khó đọc.  + Tương tác giữa các ứng dụng thông qua các phương thức HTTP: GET, POST, PUT, DELETE,... và HTTP code.  + Thông thường chỉ có các ứng dụng giao tiếp với nhau trong môi trường Internet. REST là gì? RESTful là gì? |
| 53 | - SOAP (Simple Object Access Portal) là giao thức Internet cho phép các chương trình trong các hệ điều hành khác nhau hoặc trên các máy chủ riêng biệt liên kết với nhau qua Internet. SOAP sử dụng giao thức XML  RESTful (Representational State Transfer) là một tiêu chuẩn trong thiết kế API cho các ứng dụng web để quản lý resource. Sử dụng phương thức HTTP đơn giản để giao tiếp giữa các máy. RESTful sử dụng giao thức JSON hoặc XML jQuery là gì? Ajax là gì? |
| 54 | - jQuery là một thư viện được xây dựng từ JavaScript.  - Ajax (Asynchronous JavaScript & XML) là một kỹ thuật giúp phát triển các trang web động mà không cần tải lại trang. Nêu ví dụ về cách bạn triển khai Ajax trong dự án của bạn |
| 55 | 1. Sử dụng jQuery $(document).ready để thông báo sử dụng jQuery khi HTML load xong.  2. Dùng jQuery để bắt sự kiện.  3. Dùng $.ajax để triển khai Ajax thông qua các thuộc tính: url, dataType, type, success. @RequestHeader và @ResponseHeader có ý nghĩa gì |
| **I18N** | |
| 56 | I18N và I10N là gì? |
| 57 | - I18n (Internationalization): Quốc tế hóa.  - L10n (Localization): Nội địa hóa   Nêu cách triển khai I18N và I10N trong dự án Spring 1. Cấu hình Java: LocalResolver, LocalChangeInterceptor, InterceptorRegistry, MessageSource.  2. Tạo file Resource Bundle để khai báo message bằng các ngôn ngữ khác nhau.  3. Sử dụng Thymeleaf để gán các message vào HTML. |
| 58 | Interceptor là gì? - Interceptor được xem như là một chốt chặn giữa các request gửi đến Controller. |
| **Bảo mật** | |
| 59 | Bảo mật là gì? Cơ chế bảo mật trong Spring như thế nào? |
| 60 | - Bảo mật là sự hạn chế khả năng lạm dụng tài nguyên và tài sản.  - Cơ chế bảo mật trong Spring:  + Authentication (xác thực): là một hành động nhằm thiết lập hoặc chứng thực một thông điệp hoặc đối tượng nào đó đáng tin cậy.  + Authorization (phân quyền): là quá trình xác định xem một đối tượng có quyền truy cập một tài nguyên cụ thể để thực hiện một số hành động hay không. Authentication là gì ? Authorization là gì ? |
| 61 | + Authentication (xác thực): là một hành động nhằm thiết lập hoặc chứng thực một thông điệp hoặc đối tượng nào đó đáng tin cậy.  + Authorization (phân quyền): là quá trình xác định xem một đối tượng có quyền truy cập một tài nguyên cụ thể để thực hiện một số hành động hay không. Liệt kê các cơ chế xác thực |
| 62 | + HTTP Basic: yêu cầu phía Client cung cấp Username & Password để xác thực.  + Cookies.  + Tockens: là một tiêu chuẩn mở định nghĩa cách truyền tin an toàn giữa các thành viên bằng một đối tượng JSON.  + Signature.  + One-time Password (OTP).  + OAuth2: xác thực thông qua ứng dụng thứ 3. CRSF là gì |
| 63 | - CSRF (Cross-site Request Forgery) là một kỹ thuật tấn công giả mạo chính chủ thể của nó. CORS là gì - CORS (Cross-origin Resource Sharing) là một cơ chế cho phép nhiều tài nguyên khác nhau của một trang web có thể truy cấp từ một domain khác với domain của trang. |
| **Spring Boot** | |
| 64 | Spring Boot là gì? |
| 65 | - Spring Boot là một trong số các module của Spring framework, cung cấp tính năng RAD (Rapid Application Development). Spring Boot được sử dụng để tạo ra một ứng dụng độc lập dựa trên Spring mà có thể chạy ngay với rất ít cấu hình cần thêm. Spring Boot không cần đến bất kỳ cấu hình xml nào, mà nó sử dụng quy ước về mô hình thiết kế phần mềm theo cấu hình, có nghĩa là nó giảm tải công việc cho các lập trình viên. Sự khác nhau giữa SpringBoot và SpringMVC là gì? |
| 66 | - Spring Boot:  + Spring Boot là một module của Spring Framework để đóng gói ứng dụng dựa trên Spring với các giá trị mặc định hợp lý.  + Nó cung cấp các cấu hình mặc định để xây dựng Spring-powered Framework.  + Không có yêu cầu với các file mô tả khi triển khai.  + Nó tránh mã viết sẵn và kết hợp các dependencies lại với nhau trong một đơn vị duy nhất.  + Nó làm giảm thời gian phát triển và tăng năng suất.  - Spring MVC:  + Spring MVC dựa trên Spring Framework.  + Nó cung cấp các tính năng có sẵn để xây dựng một ứng dụng web.  + Cần có nhiều file mô tả triển khai.  + Nó chỉ định từng dependency riêng biệt.  + Cần nhiều thời gian để phát triển. Trong trường hợp bạn muốn tuỳ chỉnh cấu hình cho Spring Boot chỉ tìm kiếm các bean trong một package nhất định |
| 67 | Cấu hình cho Spring Boot Tìm kiếm các Bean ở nhiều package khác nhau bằng cách nào? |
| 68 | - Annotation ComponentScan, Sử dụng scanBasePackages tromg @SpringBootApplication   Trình bày Spring Security? @EnableWebSecurity làm gì? |
| 69 | - Spring Security là một Framework của Spring, dành riêng cho việc thiết lập bảo mật ứng dụng bao gồm Authentication và Authorization.  @EnableWebSecurity để kích hoạt SpringSecurity trong dự án. Trình bày Spring Boot JPA? Cài đặt? |
| 70 | Để sử dụng interface JpaRepository cần cung cấp những thông tin nào? |
| 71 | - Interface JpaRepository cần cung cấp thông tin: Đối tượng và Primary Key. Bạn có biết annotation @ConfigurationProperties? Trình bày? |

- @ConfigurationProperties cho phép truy xuất các thuộc tính được đặt trong các tệp cấu hình một các tự động và mạnh mẽ.