1. Vẽ và trình bày cách Spring MVC xử lý request

Request từ client gửi đến server được Front controller tiếp nhận (đại diện Front Controller là Dispatcher Servlet). Sau đó Dispatcher Servlet sẽ chuyển đến Controller và Handling Mapping sẽ quyết định method nào của Controller sẽ thực hiện yêu cầu (@Controller và @RequestMapping, @GetMapping, @PostMapping). Sau đó Controller xử lý nghiệp vụ và sẽ trả lại 1 view thông qua ViewResolver (Model/ModelMap/ModelAndView) cho Front Controller (Dispatcher Servlet). Dispatcher Servlet sẽ trả lại client view template nhờ ViewResolver (ánh xạ tên view đến view tương ứng).

1. Trình bày cơ chế Dependency Injection trong Spring
2. DI là cơ chế tiêm phụ thuộc, tức là tiêm các bean vào trong chương trình viết bằng mô hình Spring MVC, để hạn chế (nới lỏng) sự phụ thuộc giữa các module (các tầng), chỉ phụ thuộc vào abstraction (interface). DI là một Design Pattern triển khai nguyên lý thiết kế phần mềm IOC (Inversion of Control, trong khi IOC cũng là một trong những nguyên lý phát xuất từ nguyên lý cốt lõi trong SOLID đó là DIP – Dependency Inversion Principle – Nguyên lý đảo ngược phụ thuộc). IOC là nguyên lý 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 có trước 🡪 Framework).
3. 3 Có bao nhiêu cách để thực hiện Dependency Injection? Trình bày?

Có 3 cách thực hiện DI, dó là contructor injection setter injection, interface injection

1. Framework là gì ? Framework khác Library chỗ nào ?
2. – **Framework** có thể hiểu **là** một khung chương trình, người dùng bổ sung code **và** tuân theo quy tắc để tạo ra ứng dụng. Còn **Library** chỉ cung cấp các chức năng tiện ích hay các class để sử dụng trong quá trình xây dựng ứng dụng
3. Framework là ứng dụng phần mềm có tính trừu tượng cao, nó cung cấp những tính năng chung thông dụng, có sẵn và có thể tùy biến được.
4. Sự khác nhau cơ bản giữa Framework và Library là cơ chế IOC: Framework nắm quyền điều khiển chương trình/ứng dụng (Code Framework sẽ gọi code của bạn) trong khi Library thì được điều khiển bởi ứng dụng/chương trình (Code của bạn sẽ gọi code Library).

5 Spring Framework là gì ?

Là nền tảng mã nguồn mở để phát triển những ứng dụng web Java EE theo mô hình MVC.

Spring gồm có:

1. Front Controller (Đại diện là Dispatcher Servlet) : lắng nghe và chuyển hướng request.
2. Controller: xử lý nghiệp vụ, thao tác ở các tầng Repository (Data) và Service (nghiệp vụ). Trả lại view kèm với data model.
3. ViewResolver: ánh xạ tên view tương ứng với đối tượng view cụ thể.
4. Lợi ích của Spring Framework?
5. + Sử dụng được và có thể tùy biến các công cụ/chức năng có sẵn.
6. + Tạo sự đồng nhất và đồng bộ code khi phát triển những dự án lớn.
7. + Hạn chế sự phụ thuộc lẫn nhau giữa các tầng, từ đó dễ phân chia công việc và thực hiện công việc song song nhau.
8. + Dễ bảo trì, dễ mở rộng.
9. + Ít quan tâm tới Database nên hạn chế lỗi SQL.

7 Nguyên lý đảo ngược quyền điều khiển (Inversion of Control) là gì ?

Là nguyên lý thiết kế dựa trên nguyên lý thứ 5 trong 5 nguyên tắc thiết kế phần mềm SOLID – Dependency Inversion Principle – Nguyên lý đảo ngược phụ thuộc, tức là để cho Framework thực hiện quyền điều khiển ứng dụng (các đối tượng, các thành phần của hệ thống). Framework sẽ gọi code của lập trình viên khi cần thiết. Trong đó Tiêm phụ thuộc (Dependency Injection) là một Design Pattern triển khai cơ chế IOC này.

1. Bean là gì?
2. **Spring Beans** chính **là** những Java Object mà từ đó tạo nên khung sườn của một **Spring** application. Chúng được cài đặt, lắp ráp và quản lý bởi **Spring** IoC container. Những **bean** này được tạo ra bởi configuration metadata được cung cấp từ container, ví dụ, **trong** tag nằm **trong** file XML

9 Trong Spring có bao nhiêu Bean Scope?

Các **Bean Scope trong Spring**. **Có** 5 **scope** được định nghĩa cho **Spring Bean**: Singleton: Chỉ duy nhất một thể hiện của **bean** sẽ được tạo cho mỗi containe

* 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
* Trong 5 scope trên thì 3 scope cuối chỉ dùng trong ứng dụng web.

1. @Autowire là gì?
2. @Autowired là đánh đấu 1 Bean được tạo và quản lý bởi IOC container (vị trí tiêm Bean).
3. 11 @Component có ý nghĩa gì?

@Component là một [Annotation](https://loda.me/java-huong-dan-tu-tao-mot-annotations-loda1554297400922) (chú thích) đánh dấu trên các Class để giúp **Spring** biết nó là một Bean

1. Trình bày ý nghĩa của Controller
2. Nơi tiếp nhận và xử lý các resquets của client đồng thời trả về các reponse tương ứng

13 Trình bày ý nghĩa của ModelAndView Interface

**Là sự kết hợp của 2 khía cạnh truyền dữ liệu và view.** Nó là một container chứa cả view và cả model data ở trong 1 kết quả view trả về

1. Trình bày ý nghĩa của ModelMap Interface
2. **ModelMap cài đặt lớp Map interface. Nên nó thêm các phương thức của Map.**

15 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)

17 Phân biệt POST với PUT thường sử dụng để làm gì?

**POST** : Gửi dữ liệu đến máy chủ **để** tạo một thực thể mới. **Thường** được **sử dụng** khi tải lên tệp hoặc gửi biểu mẫu web. **PUT** : Tương tự **POST**, nhưng được **sử dụng để** thay thế một thực thể hiện có. VÒI : Tương tự như **PUT**, nhưng được **sử dụng để** chỉ cập nhật các trường nhất định trong một thực thể hiện có.

1. @RequestMapping làm gì?
2. @**RequestMapping là** một trong những annoation sử dụng nhiều nhất trong Spring MVC. Annotation @**RequestMapping** được sử dụng để map request với class hoặc method xử lý request đó. @**RequestMapping** có thể được áp dụng với controller class hoặc method trong controller class.

19 Trình bày cơ chế Data Binding trong Spring

Data Binding là cơ chế liên kết dữ liệu đầu vào hoặc đầu ra với các đối tượng model. Hay nói cách khác đó là sự kết nối dữ liệu của bean đặt trong model đến các điều khiển trên form.

Data Binding giúp cho việc tương tác với dữ liệu trở nên dễ dàng.

Các Form đều được liên kết với một đối tượng dữ liệu được biểu diễn ở phía sau nó.

Khi tương tác với Form, dữ liệu nhập vào Form sẽ được tự động chuyển đổi thành thuộc tính của đối tượng mà nó liên kết.

Data Binding còn hỗ trợ Data Conversion dữ liệu và Validate dữ liệu.

1. Thuộc tính consumes trong các Request Mapping là gì ?
2. Thuộc tính ‘consumes’ trong @RequestMapping quy định Content-Type mà Request có thể gửi lên Server (để Controller tiếp nhận và xử lý).

21 Thymeleaf là gì?

Là 1 thư viện engine của java dc sử dụng đẻ xử lý và tạo ra các file html, xml,javascript,raw, có thể làm việc ở môi trường web và non web giúp tạo ra các teamplate đơn giản dể sửa chửa

1. Sử dụng lặp trong Thymeleaf như thế nào? Sử dụng điều kiện trong Thymeleaf như thế nào?
2. Sử dụng vòng lặp trong thymeleaf bằng th:each , điều kiện là th:if
3. Formatter là gì ? Converter là gì ?
4. Chuyển đổi kiểu dữ liệu này sang kiểu dữ liệu khác ,dữ liệu của formate là string , còn converter có thể chuyển hầu hết các kiểu dữ liệu
5. ORM là gì ?
6. Là kỹ thuật ánh xạ các CSDL trong database sang các đối tượng trong lập trình

25 JPA là gì? Spring Data JPA là gì ?

Jpa là 1 bản thiết kế dựa trên kỹ thuật ORM, spring data jpa là 1 triển khai của sprng data giúp việc triển khai các kho dữ liệu JPA đc dễ dàng hơn

1. Hibernate là gì ? Hibernate làm gì với database
2. Hibernate là triển khai của bản thiết kế Jpa , hibernate có thế kết nối và truy vấn dữ liệu trong database dc dễ dàng , thông qua việc tạo các câu truy vấn

27 Liệt kê một số annotation của hibernate?

@entity, @ id, @ genear,@table,

1. Entity là gì?
2. Entity là 1 class tương ứng với 1 table trong database

29 Persistence Context & Entity Manager là gì?

**Persistence context** có thể coi là một “môi trường” chứa toàn bộ các đối tượng mà ta tạo ra và lưu vào csdl trong mỗi session

EntityManager là 1 interface cung cấp các API cho việc quản lý và tương tác với Entity.

1. Câu lệnh truy vấn động là gì? Câu lệnh truy vấn tĩnh là gì?
2. Câu lệnh truy vấn động là câu lệnh truy vấn cần truyền tham số vào để có thể truy vấn và tái sử dụng được. Câu lệnh truy vấn động được tạo nên bởi phương thức **em.createQuery**()
3. Còn câu lệnh truy vấn tĩnh thì không cần phải truyền tham số vào, câu lệnh truy vấn tĩnh được tạo bằng cách sử dụng phương thức **createNamedQuery**() của EntityManager:

31 Trạng thái của Entity bao gồm những gì?

Managed 🡪*sessionFactory.openSession()*

Detatched 🡪Session.evict(entity) or Session.clear()

Transient

Deleted/Removed 🡪Session.delete(entity).

32 Có bao nhiêu loại mapping trong Hibernate?

1. Collection Mappings: Các collection phát sinh từ các mối quan hệ 1-n, n-n
2. Association Mappings: 1-n, 1-1, n-n, n-1
3. Component Mappings: 1 class (không phải là Entity) là thành viên của class khác.

33 SessionFactory là gì?

Là một interface giúp tạo ra session kết nối đến database bằng cách đọc các cấu hình trong Hibernate configuration.

34 Validation dữ liệu là gì?

Validation dữ liệu là xác minh dữ liệu đầu vào có hợp lệ hay là không.

1. Trình bày cách triển khai validate dữ liệu trong Spring

Có 3 cách để validate dữ liệu trong Spring:

1. **Cách 1**: Dùng các annotation có sẵn trong Spring: @Size, @NotNull, @NotEmpty, @NotBlank, @Email, @Pattern(regex = ”^[A-Za-z]\*$”), @Min, @Max, @NumberFormat., @DateTimeFormat, @Past, @Future.
2. **Cách 2**: Dùng custom validate: Class cần validate đối tượng sẽ implements interface **Validator** và ghi đè 2 method supports và validate.

Trong cả 2 cách trên thì trong Controller phải có @Valid/@Validated ở trước @ModelAttribute và tiếp đến là Class BindingResult.

Method validate hoặc annotation trên các trường sẽ gọi Spring check xem có lỗi hay không.

Đối tượng bindingResult sẽ chứa các error xuất hiện (nếu có) và ném trả về view cho người dùng.

1. Sử dụng custom annotation: tạo 1 annotation và 1 class cho Annotation đó. Class này sẽ implements interface **ConstraintValidator<**Tên\_annotation dataType> và ghi đè 2 method**: initialize + isValid(**String value, ConstraintValidatorContext context**)**

36 Binding Result là gì?

Là class chứa các thông tin về việc đối tượng được tạo nên có vi phạm các ràng buộc (constraint) hay không.

1. AOP là gì? AOP gồm những yếu tố nào?
2. AOP là Aspect Oriented Programming – lập trình hướng khía cạnh, tức là một kỹ thuật lập trình nhằm phân tách chương trình thành các module riêng rẽ, phân biệt và không phụ thuộc lẫn nhau.

38 Các loại Advice?

Advice: là những code để thực hiện Aspect, nó là 1 hành động weave advice vào chương trình chính.

Các loại Advice: @After, @Before, @Around, @AfterThrowing, @AfterReturning

39 JoinPoint là gì?

JoinPoint: điểm gia nhập, là một vị trí nào đó trong luồng thực thi chương trình mà Aspect (cụ thể là Advice) được weave (đan, dệt) vào chương trình chính 🡪JoinPoint = Core Concern x Cross-cutting Concerns.

PointCut: chỉ ra tập hợp các joint point để công cụ AOP (Aspect) đan các advice vào **core-concern**.

1. Nêu cách triển khai AOP mà bạn dùng trong dự án

Cách triển khai AOP dùng trong dự án: Ghi log chương trình mượn sách, trả sách.

Khi người dùng đăng nhập và xem danh mục sách hoặc khi trả sách thì ghi lại log thông tin người mượn, thời gian…

41 Xử lý ngoại lệ trong Spring Web MVC như thế nào?

Mục tiêu của xử lý ngoại lệ (Exception Handling) trong Spring là không gửi thông tin ngoại lệ tới client (người dùng). Mấu chốt của Exception Handling là bắt ngoại lệ đúng lúc và có hướng xử lý (handle) phù hợp.

Các cách xử lý ngoại lệ trong Spring Web MVC:

1. Sử dụng try-catch: dùng để xử lý ngoại lệ nghiệp vụ, xử lý tại Controller (hoặc ở Service).

Sử dụng annotation @ExceptionHandler: Tạo 1 class ExceptionHandler có annotation là @ControllerAdvice. Class này sẽ có 1 method trả về 1 view với annotation ở đầu method là @ExceptionHandler(BookNotFoundException.class) – ví dụ

42 Cookie là gì?

**Cookie:** là các tệp được trang web (website) tạo ra khi người dùng nhập vào. Cookie giúp tăng trải nghiệm trực tuyến người dùng bằng cách lưu thông tin người dùng ở browser. (lấy giá trị của Cookie: cookie.getValue())

@CookieValu sử dụng để truy cập dữ liệu được đặt ở trong bất kỳ http cookie nào.

43 Phân biệt Session và Cookie

1. Session: lưu trữ thông tin ở Server, trong khi Cookie lưu trữ thông tin ở browser của client.
2. Session kết thúc khi đóng trình duyệt, trong khi Cookie chỉ kết thúc khi hết hạn thời gian sống (setMaxAge()).
3. Session bảo mật hơn vì lưu ở server, trong khi Cookie thì lưu ở browser nên có thể bị đánh cắp thông tin, sửa hay xóa thông tin.
4. Session phụ thuộc vào Cookie (Value “JSESSIONID” là một ID của Session được tạo ra từ Server và gửi về máy client. Khi người dùng đăng nhập browser để gửi request lên server thì server sẽ kiểm tra và so sánh JSESSIONID của Cookie (ở Browser) và JSESSIONID ở server xem thông tin có giống nhau không, nếu có thì server sẽ trả lại thông tin.). Vì thế nên Session phụ thuộc vào Cookie, trong khi Cookie không phụ thuộc vào Session.

44 Nêu cách triển khai Session mà bạn dùng trong dự án

Cách triển khai Session dùng trong dự án: lấy thông tin đăng nhập của User hiển thị lên tất cả các page trong ứng dụng. Hoặc là ứng dụng tạo giỏ hàng thì đối tượng giỏ hàng được lưu ở trong session để có thể chuyển đổi từ trang product list/add sang trang giỏ hàng mà không bị mất dữ liệu.

1. Web Service là gì? Lấy ví dụ về web service
2. Web Service là dịch vụ web cung cấp nền tảng chung cho nhiều ứng dụng được viết bằng nhiều nền tảng công nghệ khác nhau để có thể giao tiếp được với nhau.
3. Hoạt động của Web Service được thực hiện dựa trên mô hình client-server, thông qua giao thức HTTPs với **kiểu dữ liệu trả về là JSON/XML.**
4. **Web Service có 3 thành phần chính (theo dạng XML):**
5. **+ SOAP:** Simple Object Access Protocol – giao thức trao đổi thông tin.
6. **+ WSDL:** Web Service Desciption Language – Mô tả thông tin trao đổi.
7. **+UDDI:**  Universal Description, Discovery, and Integration – Mô tả, xuất bản và tìm kiếm Web Services.
8. Ví dụ: sử dụng Web Service cho các ứng dụng tạo ExchangeRate (tỉ giá hối đoái, giá vàng), Dự báo thời tiết (Weather Broadcasting) hay là các thông tin chứng khoán, giá nhà đất, các mẫu banner quảng cáo…

46 Phân biệt Web Service và Web truyền thông

Web Service thì không có giao diện, còn Web truyền thông thì có giao diện.

Web Service thì tương tác giữa các ứng dụng với nhau, trong khi web truyền thông là tương tác giữa người dùng và ứng dụng.

Web Service là độc lập với nền tảng (ko phụ thuộc nền tảng) còn web truyền thông thì đa nền tảng (phụ thuộc vào nhiều ngôn ngữ khác nhau, nhiều hệ điều hành khác nhau…)

Web Service thì truy cập thông tin thông qua các method POST, PUT, GET, DELETE..trong khi web truyền thông thì truy cập thông tin bằng form, link, button…

Web Service trả về response là data thuần JSON/XML nên người dùng khó đọc, trong khi web truyền thông trả về các view là văn bản, hình ảnh, video,…

47 SOAP là gì? RESTful là gì?

**SOAP** = Simple Object Access Protocol – giao thức truy cập đối tượng giản đơn, nó là một loại WebService, làm việc với ngôn ngữ XML nên xử lý chậm.

**REST** = *Representational State Transfer –* Chuyển giao trạng thái đại diện, tức là một loại WebService, nhưng thao tác dựa trên đối tượng JSON/XML nên sử dụng tiện hơn (do JSON xử lý nhanh hơn, code dễ gần với Java hơn).

**RESTful** là ứng dụng được triển khai bằng REST nên tuân thủ các quy tắc của REST:

+ Sử dụng giao thức HTTPs, dữ liệu trả về JSON/XML, độc lập nền tảng, sử dụng các phương thức HTTP rõ ràng

+ Phi trạng thái: không có trạng thái như web truyền thông (không lưu thông tin request header…)

+ Hiển thị cấu trúc thư mục như URIs.

+ Linh động, không yêu cầu chặt chẽ như SOAP (XML yêu cầu chặt chẽ).

48 jQuery là gì? Ajax là gì?

Jquery là thư viện được xây dựng từ JavaScript, với phương châm “write less, do more” – Jquery giúp cho Dev viết code ít hơn, làm đc nhiều việc hơn.

AJAX: Asynchronous JavaScript and XML – là một kỹ thuật giúp phát triển các trang web động mà không cần phải tải lại trang (reload page), nó sử dụng cơ chế bất đồng bộ, giúp cập nhật lại nội dung 1 phần của trang mà không cần reload page, làm cho web mượt mà hơn.

* Giúp người dùng tiếp tục sử dụng trang web trong khi client đang request thông tin từ server.

49 Nêu ví dụ về cách bạn triển khai Ajax trong dự án của bạn

Lưu dữ liệu từ trang create mà ko cần phải chuyển trang hay load lại trang.

Tải nội dung thêm khi click vào nút “xem thêm” mà không cần phải tải lại toàn bộ trang.

50 @RequestHeader và @ResponseHeader có ý nghĩa gì

Header request parameters là các thông số kèm theo khi gửi yêu cầu lên server

Header response parameters là các thông số kèm theo khi server gửi kết quả về cho client.

**The @RequestHeader annotation is used to bind**[**HTTP request header**](https://www.logicbig.com/quick-info/web/http-header.html)**attributes values to controller method parameters.**

51 I18n và L10n là gì?

Internationalization, Localization

Internationlization: quốc tế hóa, là chức năng setting trang web cho việc sử dụng đa ngôn ngữ và tăng tính thương mại, mở rộng nhiều thị trường khác nhau.

Localization: phán định ngôn ngữ địa phương để hiển thị phù hợp với từng vùng hay từng tùy chọn người dùng.

52 Nêu cách triển khai I18N và I10N trong dự án Spring

(a). Tạo LocaleChangeInterceptor: Interceptor để gắn biến “language” vào đuôi request để LocaleResolver căn cứ vào biến đó để phán định ngôn ngữ phù hợp.

(b) Setting LocalResolver: Tạo LocalResolver để phán định ngôn ngữ từ request gửi đi theo biến “language” mà Interceptor đã gắn.

(c) Tạo các file message bundle tương ứng với từng ngôn ngữ => với các hậu tố do người dùng quy định

1. Interceptor là gì?
2. Interceptor là bộ lọc đầu vào của request (tương tự như Servlet Filter), định nghĩa các hậu tố cho các request.

54 Bảo mật là gì? Cơ chế bảo mật trong Spring như thế nào?

Bảo mật là cơ chế đảm bảo tính bí mật và an toàn cũng như sự toàn vẹn của thông tin và dữ liệu.

Trong Spring thì bảo mật qua 2 cơ chế: xác thực (authentication) và phân quyền (authorization)

55 Authentication là gì ? Authorization là gì ?

Authen xác thực, author quân quyền

1. Liệt kê các cơ chế xác thực
2. Basic HTTP
3. Cookie
4. Tokens
5. One-Time Password
6. Signatures
7. Certificate
8. Oauth2

57 CSRF là gì

Cross-Site Request Forgery: là một kỹ thuật tạo các request giả mạo xuyên qua các trang.

58 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. **CORS** là viết tắt của từ Cross-origin resource sharing

59 Spring Boot là gì?

**Spring Boot** là một dự án phát triển bởi **JAVA**(ngôn ngữ 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.

60 Sự khác nhau giữa SpringBoot và SpringMVC là gì?

Spring Boot tự động cấu hình và tải các dependency cần thiết cho việc build ứng dụng.

61 Cấu hình cho Spring Boot Tìm kiếm các Bean ở nhiều package khác nhau bằng cách nào?

@Component, @Service, @Repository, @Controller

Sử dụng @ComponentScan, @SpringBootApplication.

62 Trình bày Spring Security? @EnableWebSecurity làm gì?

Spring Security là 1 dự án cung cấp 2 cơ chế bảo mật cơ bản:

+ Authentication: là tiến trình thiết lập một Principal (1 người, thiết bị, hệ thống) có thể thao tác một hành động nào đó lên ứng dụng.

+ Authorization: là tiến trình cấp phép Principal có được phép thực hiện một hành động (truy cập, CRUD) lên tài nguyên của ứng dụng hay không.

Spring Security gồm có 5 thành phần chính:

1. SecurityContext: là interface lưu trữ tất cả chi tiết liên quan đến bảo mật của ứng dụng.
2. SecurityContextHolder: lưu trữ SecurityContext hiện tại của ứng dụng, bao gồm chi tiết cảu Principal đang tương tác với ứng dụng.
3. Đối tượng Authentication: biểu diễn thông tin của Principal.
4. UserDetails: là 1 interface, đại diện cho 1 Principal.
5. GrantedAuthority: là quyền được cấp cho Principal, tiền tố mặc định của quyền là “ROLE\_”. Phương thức getAuthorities() là lấy danh sách các quyền được cấp cho Principal.

Class WebSecurityConfigurerAdapter là một abstract class implement interface WebSecurityConfigurer định nghĩa các cấu hình mặc định cần thiết cho Spring Security. Chúng ta cần sử dụng class này với annotation @EnableWebSecurity để enable hỗ trợ security cho ứng dụng web của chúng ta.

63 Trình bày Spring Boot JPA? Cài đặt?

64 Để sử dụng interface JpaRepository cần cung cấp những thông tin nào?

@Entity, @Id, @GeneratedValue

Validation-api

Hibernate-Validator

1. Bạn có biết annotation @ConfigurationProperties? Trình bày?
2. Cuối cùng là @ConfigurationProperties, annotation này đánh dấu class bên dưới nó là properties, các thuộc tính sẽ được tự động nạp vào khi Spring khởi tạo.