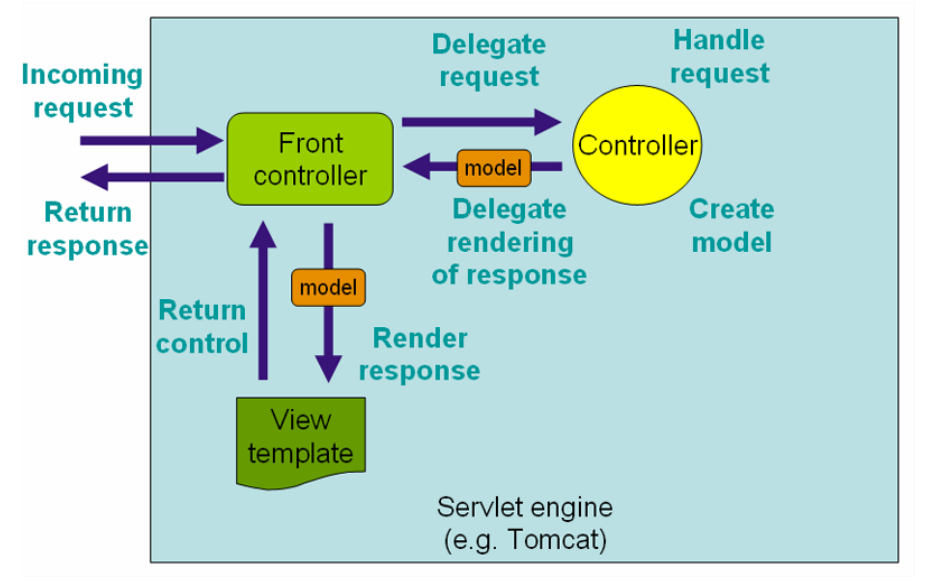
1. **Vẽ và trình bày cách Spring MVC xử lý request**

-M: Model

-V: Views

-C: Controller



* Có thêm front controller để xử lý
* **Luồng thực thi:** Khi có 1 requet thì dispatcher servlet sẽ nhận sau đó dispatcher sẽ sử dụng handler mapping sẽ xác định gửi dữ liệu đến controller con tương ứng sau đó controller sẽ làm việc với model để kiểm tra và lấy dữ liệu 🡪 controller sẽ gửi trả kết quả về front controller và dữ liêu được render đến view-templates tương ứng sau đó sẽ gửi về lại front controller để hiện thị ra cho người dùng

1. **Trình bày cơ chế Dependency Injection trong Spring**

* Là cơ chê giảm sự phụ thuộc giữa 2 module, không phụ thuộc trực tiếp vào nhau (module cấp cao không phụ thuộc module cấp thấp mà thông qua abtract).

1. **Có bao nhiêu cách để thực hiện Dependency Injection? Trình bày?**

* **3 cách:**
* **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

1. **Framework là gì ? Framework khác Library chỗ nào ?**

* **Framework** là bộ khung có các quy tắc và tiêu chuẩn để lập trình viên bắt buộc phải tuần theo để phát triển ứng dụng

|  |  |
| --- | --- |
| Framework | Library |
| * Có quy tắc * Dựa theo mà làm * Thay đổi cấu trúc code dự án đáp ứng các nguyên tắc đề ra | * Không tuân theo quy tắc * Gọi ra mà chạy * Cung cấp các tính năng tiện ích |

1. **Spring Framework là gì ?**

* Là bộ khung mvc phát triển của Spring.
* Spring Framework là một framework ứng dụng và **bộ chứa đảo ngược điều khiển**(sử dụng cơ chế IoC) cho nền tảng Java.

1. **Lợi ích của Spring Framework?**

* Giúp lập trình viên tập trung vào nghiệp vụ
* Theo các quy tắc để xây dựng hệ thống 1 cách nhanh chóng
* Dễ mở rộng dễ bảo trì

1. **Nguyên lý đảo ngược quyền điều khiển (Inversion of Control) là gì ?**

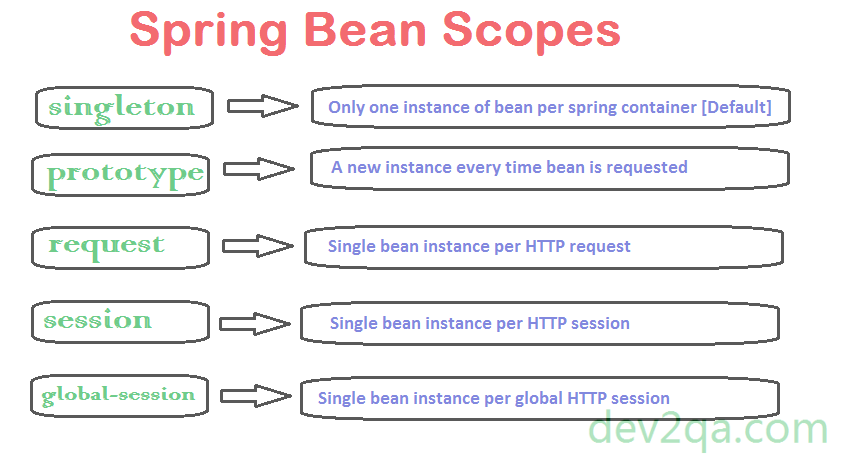
* IoC đảo ngược dòng điều khiển so với luồng điều khiển truyền thống.
* Trong IoC, các phần được viết tùy chỉnh của chương trình máy tính nhận luồng điều khiển từ khung chung.

1. **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/>.

1. **Trong Spring có bao nhiêu Bean Scope?**

* **Singleton** (mặc định): IoC container chỉ tạo đúng duy nhất 1 object từ class bean này
* **Prototype**: return một bean object riêng biệt cho mỗi lần sử dụng.
* **Request**: tạo mỗi bean cho mỗi request
* **Session**: tạo mỗi bean cho mỗi session
* **Global** **session**: tạo mỗi bean cho mỗi global session



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

1. **@Autowire là gì?**

* **@Autowired** là cơ chế tiêm đối tượng vào trong class.
* khi yêu cầu @Autowired thì nó sẽ lấy đối tượng có sẵn trong ApplicationContext để tiêm vào class

1. **@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 instance và đưa vào ApplicationContext để quản lý.

1. **Trình bày ý nghĩa của Front Controller**

* Một class được đánh dấu là **@Controller** thì để khai báo Class đó là một **controller** và có nhiệm vụ **mapping request** trên url vào các method tương ứng trong controller.
* Ví dụ:

<https://levunguyen.com/laptrinhspring/2020/04/28/tong-hop-cac-annotation-trong-spring-web/#13-controller>

Nếu em thấy đoạn mã này trong web.xml thì xin chúc mừng đó chính là font controller ( hay còn được gọi là dispatcherServlet). Còn spring boot thì nó tự thêm cho ta rồi. Ko cần config này nữa.

1. **Trình bày ý nghĩa của ModelAndView class**

* ModelAndView là sự kết hợp của 2 mục đích ModelMap and View . Nó cho phép controller trả về 1 giá trị bao gồm Model và View .Nó đại diện cho một view cùng với các dữ liệu sử dụng trong view đó

1. **Trình bày ý nghĩa của ModelMap class**

* Lớp con của Linked Hash Map
* Giống Model và ModelAndView đều dùng để gửi dữ liệu từ controller lên views nhưng có thêm cái các phương thức của Map

1. **Trình bày ý nghĩa của ViewResolver Interface**

* Là thành phẩn để ánh xạ đến các view templates tương ứng. Được xác định dựa vào prefix(tiền tố) suffix(hậu tố)

1. **Phân biệt POST với PUT thường sử dụng để làm gì?**

* Post đăng ký 1 thông tin
* Put là kiểm tra dữ liệu nhập vào nếu có thì cập nhật không thì thêm mới dữ liệu

1. **@RequestMapping làm gì?**

* **@RequestMapping** để ánh xạ các URL tới các handler method của controller
* **@RequestMapping** có các biến thể khác như**@GetMapping, @PostMapping, @PutMapping…**

1. **Trình bày cơ chế Data Binding trong Spring (thể hiện thông qua DataBinder)**

* Giúp rút gọn việc truyền dữ liệu từ views xuống controller (dùng @ModelAttribute để nhận đối tượng thay vì dùng nhiều @Requestparam để nhận từng thuộc tính).
* Giúp validate dữ liệu
* Formatters và Conversion

1. **Thuộc tính consumes trong các Request Mapping là gì ?**

* **consumes** định dạng trước phương thức đầu vào **produces** định dạng đầu ra

### 20. Thymeleaf là gì?

Thymeleaf là một Java template engine (bộ công cụ)dùng để xử lý View hoạt động trên HTML, XML, Javascript, CSS ,text và RAW.

### 21.Sử dụng lặp trong Thymeleaf như thế nào? Sử dụng điều kiện trong Thymeleaf như thế nào?

* vòng lặp : th:each  cú pháp <HtmlTag th:each="item, iState : ${items}">
* điều kiện :
* th:if, th:unless .Trong một vài tình huống bạn muốn một đoạn nào đó của Thymeleaf Template xuất hiện trong kết quả nếu một điều kiện nhất định nào đó được đánh giá là true (Đúng). Để làm được điều này bạn có thể sử dụng thuộc tính (attribute) th:if,phủ định lại ta có th:unless .

## th:switch, th:case .Chương trình sẽ đánh giá lần lượt các case (trường hợp) từ trên xuống dưới, nếu tìm thấy 1 case được đánh giá là true (đúng) nó sẽ "render" mã trong case này, tất các case khác sẽ bị bỏ qua.

## th:case = "\*" là case mặc định của cấu trúc th:switch/ th:case. Nếu tất cả các case ở phía trên bị đánh giá là false thì mã của case mặc định sẽ được "render".

**22**.**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. Formatter phù hợp hơn với tầng web (web-tier), còn converter thì có thể sử dụng ở bất cứ tầng nào.

### 23.ORM là gì ?

- ORM (**O**bject **R**elational **M**apping), là một kỹ thuật/cơ chế lập trình thực hiện ánh xạ CSDL sang các đối tượng trong các ngôn ngữ lập trình hướng đối tượng.

### 24. JPA là gì? Spring Data JPA là gì ?

- JPA là viết tắt của Java Persistence API, là một đặc các tiêu chuẩn của Java để làm việc với cơ sở dữ liệu quan hệ.

- Spring Data là một module của Spring Framework. Mục đích của Spring Data là giảm thiểu việc thực hiện quá nhiều bước để có thể implement được JPA

**25.Hibernate là gì ? Hibernate làm gì với database.**

**- Hibernate là gì**

+ Là một thư viện ORM (Object Relational Mapping)

+ Giúp lập trình viên viết ứng dụng Java có thể map các **object** với các **table** trong database

**- Hibernate làm gì với database**

**+ Truy vấn cơ sở dữ liệu độc lập:** Nó tạo ra các truy vấn cơ sở dữ liệu độc lập. Vì vậy, bạn không cần phải viết các truy vấn cơ sở dữ liệu cụ thể

**+ Tạo bảng tự động:** Hibernate framework cung cấp phương tiện để tạo ra các bảng cơ sở dữ liệu tự động.

**+ Đơn giản lệnh join phức tạp:** Có thể lấy dữ liệu từ nhiều bảng một cách dễ dàng với Hibernate framework.

**+ Cung cấp thống kê truy vấn và trạng thái cơ sở dữ liệu:** Hibernate hỗ trợ bộ nhớ cache truy vấn và cung cấp số liệu thống kê về truy vấn và trạng thái cơ sở dữ liệu

**26.Liệt kê một số annotation của hibernate?**

**-** @Entity

- @Table

- @Column

- @Id

- @GeneratedValue

- @ManyToOne

- @OneToMany

- @OneToOne

- @ManyToMany

**27.Entity là gì?**

**-** Entity là một thực thể

- Là các lớp đơn giản, chỉ dùng để mô tả dữ liệu chứ không xử lý các logic nghiệp vụ.

- Entity sẽ được ánh xạ (mapping) tới một bảng trong CSDL.

- Trong JPA, một entity cần tuân thủ:

+ Được gắn với annotation javax.persistence.Entity(@Entity)

+ Có một constructor là public và không có tham số

+ Không được khai báo final

+ Các biến đối tượng cần được khai báo là private, protected hoặc ở mức package-private

**28.Persistence Context & Entity Manager là gì ?**

- Entity Manager cung cấp các phương thức thao tác với các entity(CRUD)

- Persistence quản lý các entity

**29.Câu lệnh truy vấn động là gì? Câu lệnh truy vấn tĩnh là gì?**

|  |  |
| --- | --- |
| **Câu lệnh truy vấn động** | **Câu lệnh truy vấn tĩnh** |
| * Câu lệnh truy vấn động câu truy vấn **có** tham số * Linh hoạt hơn * Khơi tạo nhiều lần * Truy vấn chậm | * Câu lệnh truy vấn tĩnh câu truy vấn **không** có tham số * Ít linh hoạt * Khởi tạo 1 lần duy nhất * Truy vấn nhanh |

### 31.Có bao nhiêu loại mapping trong Hibernate?

@OneToMany

@ManyToOne

@OneToOne

@ManyToMany

### 32.SessionFactory là gì?

Là một interface giúp tạo ra session kết nối đến database bằng cách đọc các cấu hình trong một file xml

### 33.Validation dữ liệu là gì?

- Là việc kiểm tra dữ liệu trước khi xử lý nhằm đảm bảo dữ liệu đáp ứng các yêu cầu đặt ra.

- Các tầng validate dữ liệu

+ UserInterface(UI): Html, JS.

+ Business layer: Java, PHP

+ Data layer : database

### 34.Trình bày cách triển khai validate dữ liệu trong Spring

- Spring triển khai validate ở tầng Business layer:

- Có 2 cách triển khai validate:

+ Dùng các anotation để validate: @NotNull, @NotBlank ,@NotEmplty, @Email,…

**+** Custom Validate (Tự định nghĩa)

B1: implement Validator vào Dto

B2: Định nghĩa method validate

B3: gọi method validate ở controller để kiểm tra dữ liệu

**Lưu ý: File msg tự định thì phải cấu hình**

### 35.Binding Result là gì?

### - BindingResult là đối tượng của Spring mà giữ kết quả xác thực và ràng buộc và chứa các lỗi có thể xảy ra (đặt bên cạnh đối tượng cần valid)

### 36.AOP là gì? AOP gồm những yếu tố nào?

### - Aspect Oriented Programming (AOP)– lập trình hướng khía cạnh(Bổ trợ cho OOP) – để giải quyết cho công việc được lặp đi lặp lại và không liên quan đến nghiệp vụ của project như: logging, transaction, security

**- AOP gồm những yếu tố nào**

**+** Aspect: Khai báo 1 khối độc lập với OOP

+ Advice: Các hành động mà AOP dùng để quan sát OOP

+ Joinpoint: Các điểm gây ra các hành động ở advice

+ Pointcut: Tập hơn các Joinpoint có khả năng gây ra các hành động ở advice

**37. Các loại advice**

\*Before:

\*After:

\*After returning:

\*After throwing:

\*Around: thực hiển ở đầu và cuối method

**38. JoinPoint là gì**

+ Joinpoint: Các điểm gây ra các hành động ở advice

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

- Đặt @Aspect trên đầu class, trong class sẽ có nhiều phương thức mà mỗi phương thức sẽ là 1 advice từ OOP, trong advice sẽ định nghĩa các pointcut và xử lí theo cách mình mong muốn.

**40. Xử lý ngoại lệ trong spring MVC như thế nào**

3 Cách:

-Try-catch, throw-throws

-Exception Handler

-Afterthrowing

**41. Cookie là gì**

- Cookie là các tệp văn bản được lưu trữ trên Client side và chúng được giữ cho mục đích theo dõi sử dụng.

**42.Phân biệt Session và Cookie**

|  |  |  |
| --- | --- | --- |
|  | Session | Cookie |
| -What-Whent-Where-Security | - Lưu trữ bất kì object nào- Đến khi đóng chương trình- Server và tồn tại JSESSIONID nằm ở Cookie- Không thể chỉnh sửa | - Chỉ lưu được String(lưu ý: sử dụng ‘-’ làm dấu ngăn cách giữa các dữ liệu )-Hết thời gian sống khi được set lúc tạo ra- Client(trình duyệt)-Có thể chỉnh sửa ở Client |

### 43.Nêu cách triển khai Session mà bạn dùng trong dự án

### -Tạo: Dùng @SessionAttributes kết hợp với @ModelAttribute khi cả 2 trùng tên thì session sẽ được tạo ra

### -Sử dụng:

### +Dùng @SessionAttribute đặt làm tham số của từng phương thức

### +Dùng @SessionAttributes trên đầu các class controller mà muốn sử dụng session

### 44.Web Service là gì? Lấy ví dụ về web service

### -Web Service là 1 ứng dụng client server giao tiếp với nhau qua giao tiếp HTTP/HTTPS.

### - Web Service cung cấp 1 chuẩn giao tiếp chung giữa các nền tảng, framework, ứng dụng… Các ứng dụng phần mềm được viết bằng các ngôn ngữ lập trình khác nhau và chạy trên các nền tảng khác nhau, có thể sử dụng các dịch vụ web để trao đổi dữ liệu qua mạng máy tính (như Internet) theo cách tương tự như liên lạc giữa các quá trình trên một máy tính

### Ví dụ: google map,

### 45.Phân biệt Web Service và Web truyền thông:

|  |  |  |
| --- | --- | --- |
|  | **Website** | **Webservice** |
| **Input** | Người dùng sẽ truy cập thông qua giao diện, form | Thông qua phương thức GET,POST,PUT,PATCH,DELETE |
| **Output** | Giao diện người dùng (GUI) | Các kiểu dữ liệu JSON ,XML |
| **How** | Hoạt động đa nền tảng nhưng phải thông qua trình duyệt | Hoạt động độc lập nền tảng, không phụ thuộc vào hệ điều hành |
| **Who** | Người dùng thông thường | Các ứng dụng hệ thống |

### 46. SOAP là gì? RESTful là gì?

### -Cả 2 đều là các loại webservice

-SOAP: Gửi dự liệu từ client về Server theo 1 chuẩn nhất định

-RESTful: Gửi dự liệu từ client về Server không theo 1 chuẩn nào cả, không ràng buộc các quy chuẩn.

**47.jQuery là gì? Ajax là gì?**

**- jQuery** là thư viện được viết từ JavaScript, jQuery giúp xây dựng các chức năng bằng Javascript dễ dàng, nhanh và giàu tính năng hơn.

**- AJAX** là viết tắt của JavaScript và XML không đồng bộ (Asynchronous JavaScript and XML).

AJAX là một kỹ thuật mới để tạo ra các ứng dụng web tốt hơn, nhanh hơn và tương tác hơn

### 48.Nêu ví dụ về cách bạn triển khai Ajax trong dự án của bạn

$.ajax({

       headers: {

           'Accept': 'application/json',

           'Content-Type': 'application/json'

       },

       type: "POST",

       data: JSON.stringify(newSmartphone),

       //tên API

       url: "/smartphones",

       //xử lý khi thành công

       success: successHandler(Đây là để gọi lại function nha)

       error: failHandler(Đây là để gọi lại function nha)

   });

headers: Quy định kiểu dữ liệu gửi đi và trả về

type: Phương thức truyền đi

data: Nội dùng truyền đi

url: đường dẫn cần truyền tới

success: Xử lý khi thành công

error: Xử lý khi thất bại

### 50.I18n và L10n là gì?

**Internationalization** (I18N) là quá trình thiết kế một ứng dụng phần mềm để nó có thể được điều chỉnh thích hợp cho nhiều ngôn ngữ và khu vực mà không có sự thay đổi về kỹ thuật.

**Localization** (L10N) là quá trình đáp ứng một phần mềm đã quốc tế hóa cho một vùng hoặc ngôn ngữ cụ thể bằng cách bổ sung các thành phần của ngôn ngữ đang hướng đến, cũng như dịch các đoạn văn bản trong phần mềm qua ngôn ngữ đó.

**51.Nêu cách triển khai I18N và I10N trong dự án Spring**

- Tạo file config -> tạo class AppConfiguration implements WebMvcConfigurer

@Override

-addInterceptors(): tự động phân tích tham số lang đi kèm các request để xác định bản địa hiện tại

@Bean

-MessageSource: setBaseName(tên file.properties)

-LocaleResolver: setDefaultLocale(new Locale( lấy thông tin về bản địa từ session của người dùng))

Tạo file.properties điền các message cần đặt tên

->Qua html gọi message bằng #{message}

**52.Interceptor là gì?**

Chốt chặn, chặn request gửi vào controller, chặn data gửi qua view

*Một trong các ứng dụng****Spring MVC****có sử dụng****Interceptor****là "Ứng dụng web đa ngôn ngữ".*

Bạn có thể sử dụng **Interceptor** để làm một số việc, chẳng hạn như ghi lại **Log**, thêm hoặc cập nhập các cấu hình trước khi request được xử lý bởi **Controller**, ...

### 53.Bảo mật là gì? Cơ chế bảo mật trong Spring như thế nào?

Là bảo vệ dữ liệu của bạn từ bất kỳ hình thức truy cập trái phép,bảo vệ dữ liệu an toàn trên môi trường trực tuyến. Trong Spring có cơ chế xác thực (Authentication) và cơ chế phân quyền (Authorization).

### 54.Authentication là gì ? Authorization là gì ?

**- Authentication:** là quá trình xác thực thông tin người dùng có được truy cập vào hệ thống

**- Authorization:** là quá trình phân quyền vói từng account người dùng.

**55.Liệt kê các cơ chế xác thực**

- HTTP Basic( client cung cấp username, password )

- Cookies( khi người dùng đăng nhập lần đầu thì sử dụng cooki để đăng nhập lại )

- Tokens(JWT JSON Web Token)

- Signature(JWT)

- One-Time-Password(OTP)

**56.CSRF là gì ?**

-**CSRF:** Là cách tấn công bằng cách mượn quyền chứng thực từ người dùng để làm thay đổi trang web mà họ có account

**57.CORS là gì ?**

- Cross-Origin Resource Sharing (CORS) là một tiêu chuẩn về bảo mật có ở các web browser thế hệ mới.

- CORS là một cơ chế xác nhận thông qua Header của request. Cụ thể là trên Server sẽ nói với browser về quy định chấp nhận những request từ domain nào và phương thức ra sao (GET, POST, PUT, v.v..)

**58.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.

### 59.Sự khác nhau giữa Spring Boot và Spring MVC là gì?

|  |  |
| --- | --- |
| Spring Boot | Spring MVC |
| * Tích hợp sẵn tomcat * Cung cấp sẵn thư viện * Tự động cấu hình * Không cần file config(xml; java) * Khó fix bug * Có class application và application.properties | * Không có tích hợp tomcat * Không cung cấp * Phải tự cấu hình * Cần phải có file config * Dễ fix bug hơn 1 tí nhưng vẫn khó * Không có |

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

Dùng @ComponentScan đặt trên class

Dùng @Bean đặt trên method để khai báo các Bean

**61.Trình bày Spring Security? @EnableWebSecurity làm gì?**

**Authentication:** Khi nói về authentication là ta nói về chức năng đăng nhập vào hệ thống. Authentication nghĩa là bạn có phải là người dùng của hệ thống hay không.

**Authorization:** Khi nói về authorization ta nói về quyền hạn được phép làm gì?

**@EnableWebSecurity:** là một chú thích đánh dấu. Nó cho phép Spring tìm và tự động áp dụng cho lớp WebSecurity.(Theo Hà đọc như này là quá suôn mồm rồi)

### 62.Trình bày Spring Boot JPA? Cài đặt?

**Spring Boot JPA**

-Spring Boot JPA là một phần trong hệ sinh thái Spring Data, nó tạo ra một layer ở giữa tầng service và database, giúp chúng ta thao tác với database một cách dễ dàng hơn, tự động config và giảm thiểu code thừa thãi.

-Spring Boot JPA đã wrapper Hibernate và tạo ra một interface mạnh mẽ.

**Cài đặt**

-Để thêm Spring JPA vào project, bạn cần thêm dependency spring-boot-starter-data-jpa.

-Để connect tới MySql, chúng ta cần driver tương ứng, vì vậy phải bổ sung thêm cả dependency mysql-connector-java vào pom.xml.

### 63.Để sử dụng interface Jpa Repository cần cung cấp những thông tin nào?

Đối tượng và kiểu dữ liệu id của đối tượng đó.

### 64.Bạn có biết annotation @ConfigurationProperties? Trình bày?

Cho phép truy xuất các thuộc tính được đặt trong các tệp cấu hình 1 cách tự động và mạnh mẽ

**Data Binding là** cơ chế liên kết dữ liệu đầu vào(hoặc đầu ra) với các đối tượng model.

- Khi sử dụng Data binding, các form sẽ được liên kết với 1 đối tượng biểu diễn dữ liệu ở sau(các Class trong model).

- Khi tương tác với form thì các dữ liệu sẽ tự chuyển đổi thành thuộc tính của đối tượng liên kết với form.

- Data binding trong spring hoạt động dựa trên Data Binder.