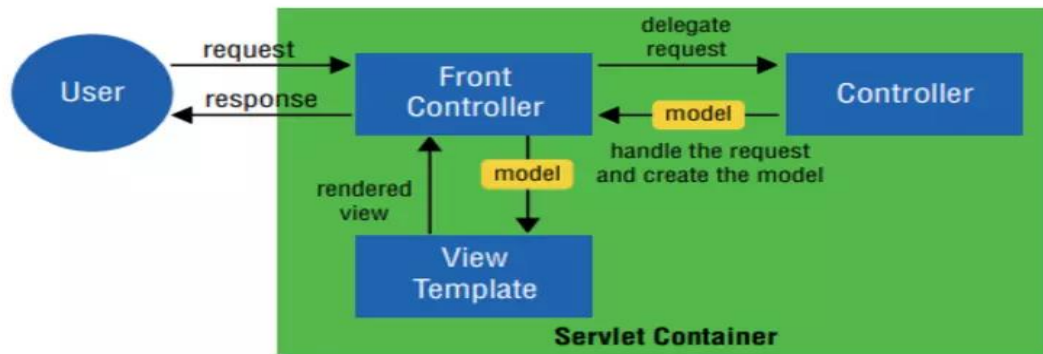


## 1. Vẽ và trình bày cách Spring MVC xử lý request



Luồng thực thi :

- Khi có 1 request được gửi tới Front Controller (Dispatcher Servlet) sẽ tiếp nhận
  - Sau đó Front Controller sử dụng cơ chế Handler Mapping để ánh xạ đến controller tương ứng xử lý request
  - Controller xử lý request xong thì trả kết quả lại cho Front Controller
  - Front Controller (Dispatcher Servlet) render dữ liệu lên view thông qua view resolver kết quả được trả lại cho Front Controller
  - Front Controller trả kết quả về cho người dùng.
- ## 2. Trình bày cơ chế Dependency Injection trong Spring
- Dependency Injection (tiêm phụ thuộc) là cơ chế giảm sự phụ thuộc giữa 2 class. Tức là class cấp cao không phụ thuộc trực tiếp vào class cấp thấp mà thông qua 1 abstraction
- ## 3. Có bao nhiêu cách để thực hiện Dependency Injection? Trình bày?
- Có 3 cách @Autowired :
    - Constructor : các biến phụ thuộc(dependency) được cung cấp thông qua hàm tạo lớp ( constructor
    - Setter : các biến phụ thuộc sẽ được truyền vào 1 class thông qua các hàm setter
    - Interface : client triển khai 1 interface có hàm setter dành cho việc nhận biến phụ thuộc
- ## 4. Framework là gì ? Framework khác Library chỗ nào ?
- Framework là 1 ứng dụng trừu tượng có sẵn bộ khung, luồng code để tạo ra ứng dụng 1 cách nhanh chóng
  - Phân biệt
    - Giống : đều cung cấp tính năng để tái sử dụng code
    - Khác : điểm khác biệt giữa framework và library là cơ chế loc

Library	Framework
- Bao gồm các class, method hỗ trợ code dễ dàng	- Bộ khung có sẵn, để xây dựng ứng dụng 1 cách nhanh chóng, gồm tập hợp các library
- Ứng dụng nắm quyền điều khiển ( code người viết gọi code library )	- Framework nắm quyền điều khiển ứng dụng ( code framework gọi code người viết)
- Hoạt động bị động	- Hoạt động chủ động
- Không có cơ chế loc	- Có cơ chế IoC

5. Spring Framework là gì ?
  - Là 1 framework của java được xây dựng dựa trên mô hình MVC
6. Lợi ích của Spring Framework?
  - Tính đơn giản và kiểm thử tốt
  - Đảo ngược quyền điều khiển (Ioc)
  - Dung lượng nhẹ
  - Hỗ trợ lập trình hướng khía cạnh AOP
  - Xây dựng ứng dụng web theo mô hình MVC
  - Cấu hình và vòng đời được quản lý bởi Spring Framework
  - Xử lý ngoại lệ tốt
7. Nguyên lý đảo ngược quyền điều khiển (Inversion of Control) là gì ?
  - Inversion of Control là nguyên lý đảo ngược quyền điều khiển, trong đó việc điều khiển các đối tượng và thành phần của hệ thống được thực hiện bởi framework
8. Bean là gì?
  - Bean là những module chính của chương trình, được tạo ra và quản lý bởi Spring IoC Container (framework tự khởi tạo và quản lý vòng đời)
9. Trong Spring có bao nhiêu Bean Scope?
  - Có 5 bean scope
    - Singleton : chỉ duy nhất một thể hiện của bean sẽ được tạo ra cho mỗi container. Đây là scope mặc định của bean
    - Prototype : một thể hiện của bean sẽ được tạo ra cho mỗi lần được yêu cầu (request)
    - Request : giống với prototype, tuy nhiên nó dùng cho ứng dụng web, một thể hiện của bean sẽ tạo ra mỗi HTTP Request
    - Session : mỗi thể hiện của bean sẽ được tạo cho mỗi HTTP Session
    - Global-session : được sử dụng để tạo global session bean cho các ứng dụng Portlet
10. @Autowired là gì?
  - Autowired là 1 annotation dùng để tìm các đối tượng phụ thuộc cần sử dụng vào class
11. @Component có ý nghĩa gì?
  - Component là một annotation dùng để đánh dấu class java là một bean. Khi spring quét qua các annotation có đánh dấu là @Component thì nó sẽ tạo bean cho class đó
12. Trình bày ý nghĩa của Controller
  - Controller là nơi nhận request từ người dùng, xử lý request, xây dựng dữ liệu cho view (model) và chọn view để trả lại kết quả cho người dùng
13. Trình bày ý nghĩa của ModelAndView Interface
  - ModelAndView là 1 interface, là sự kết hợp giữa 2 khía cạnh truyền dữ liệu và view.
  - Nó cho phép controller trả về 1 giá trị gồm model (truyền dữ liệu) và view.
  - Khai báo 1 đối tượng khi sử dụng cần phải khai báo thêm tên view
14. Trình bày ý nghĩa của ModelMap Interface
  - ModelMap là 1 class được khai báo như là tham số ở method trong Controller
  - Do triển khai từ Map nên được tích hợp thêm các phương thức của Map như : put, allput, ....
  -
15. Trình bày ý nghĩa của ViewResolver Interface
  - ViewResolver là cơ chế xử lý tầng view trong Spring MVC
  - ViewResolver ánh xạ tên của view sang đối tượng view tương ứng

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

- Giống : đều có sự thay đổi về dữ liệu

POST	PUT
- Thêm mới (create)	- Cập nhật(edit)

17. @RequestMapping làm gì?

- Là một anotation dùng để ánh xạ các request đến các method trong controller tương ứng

Request Param	Path Variable
+ Dùng trích xuất dữ liệu từ đường dẫn theo dạng (key, value)`	+ Dùng trích xuất dữ liệu từ đường dẫn theo dạng (value)
+ Kém bảo mật	+ Bảo mật hơn -> có thể sử dụng regex trên đường dẫn

18. Trình bày cơ chế Data Binding trong Spring

- Data Binding là cơ chế liên kết dữ liệu đầu vào hoặc đầu ra với các object model, thông qua cơ chế data binder. Với 3 nhiệm vụ sau :
  - PropertyEditors : liên kết với các thuộc tính
  - Validators : kiểm tra dữ liệu hợp lệ
  - Formater/Converter : chuyển đổi dữ liệu

19. Thuộc tính consumes trong các Request Mapping là gì ?

- Consumes là chỉ chấp nhận các request có content-type giống với trị khai báo bên trong consumes

20. Thymeleaf là gì?

- Thymeleaf là 1 java template engine dùng để xử lý và tạo html, css, js, xml, text
- Mục đích là tạo ra các template thân thiện đơn giản
- Ưu điểm :
  - Hạn chế các thư viện thẻ
  - Có thể mở trực tiếp file html để xem giao diện
  - Hỗ trợ html 5 ở tất cả version

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

- Sử dụng th:each

22. Formatter là gì ? Converter là gì ?

- Formatter và converter giống nhau là đều để chuyển đổi kiểu dữ liệu
- Formatter kiểu nguồn là String, còn Converter kiểu nguồn có thể là bất cứ kiểu dữ liệu nào
- Formatter phù hợp với tầng web còn Converter thì có thể sử dụng ở bất cứ tầng nào
- Trong trường hợp chuyển đổi dữ liệu nhập vào từ một trường input trong spring mvc nên chọn formatter

23. ORM là gì ?

- Object Relational Mapping : là kỹ thuật liên kết giữa các đối tượng trong lập trình với các đối tượng trong cơ sở dữ liệu

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

- Java Persistence API : là bộ đặc tả của java bao gồm các interface được thiết kế từ ý tưởng ORM

- Spring Data JPA là module của spring hỗ trợ việc triển khai Jpa dễ dàng hơn ở tầng repository

25. Hibernate là gì ? Hibernate làm gì với database

- Hibernate là ORM Framework để triển khai Jpa
- Hibernate triển khai và thực thi các interface extends của Jpa

26. Liệt kê một số annotation của hibernate?

@Entity

@Id

@Query

@ManyToOne,....

27. Entity là gì?

- Entity là đối tượng đại diện cho dữ liệu trong ứng dụng
- Entity thường là POJO (Plain Old Java Object), nghĩa là class đơn giản chỉ mô tả dữ liệu, không xử lý nghiệp vụ và có method getter setter
- Entity là đối tượng thể hiện tương ứng của 1 bảng trong cơ sở dữ liệu
  - ➔ Điều kiện để 1 class là entity :
    - + @Entity đầu class
    - + Class hay method không sử dụng final
    - + Phải có constructor mặc định
    - + Thuộc tính có access modifier không được là public

28. Persistence Context & Entity Manager là gì?

- Persistence Context là một tập các thể hiện của entity, được dùng quản lý entity trong 1 kho dữ liệu
- Entity Manager là một interface cung cấp các phương thức để thao tác với entity

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

- Câu lệnh truy vấn tĩnh không thể truyền tham số trong lúc compile-time
- Câu lệnh truy vấn động có thể truyền tham số và thay đổi linh động tùy thuộc giá trị tham số truyền vào

30. Trạng thái của Entity bao gồm những gì?

Entity có 4 trạng thái :

- Transient : mới được tạo chưa lưu vào cơ sở dữ liệu
- Persistent : được ghi xuống cơ sở dữ liệu và quản lý trong persistence context
- Detached : tạm thời tách ra khỏi persistence context
- Removed : bị xóa khỏi cơ sở dữ liệu

31. Có bao nhiêu loại mapping trong Hibernate?

- Association Mapping (Entity Mapping) : là ánh xạ liên kết giữa giữa các lớp entity và các mối quan hệ giữa các bảng chính, gồm 4 kiểu :
  - ManyToOne
  - ManyToMany
  - OneToOne
  - OneToMany

- Collections Mapping : nếu một entity hoặc lớp có 1 tập các giá trị trong 1 trường collections cụ thể, thì chúng ta có thể ánh xạ các giá trị đó bằng các interface collections có sẵn trong java
- Component Mapping : mapping cho 1 lớp có 1 tham chiếu đến 1 lớp khác như là 1 biến thành viên.

### 32. SessionFactory là gì?

- SessionFactory : là một interface giúp tạo ra session kết nối đến database bằng cách đọc các cấu hình trong Hibernate configuration. Mỗi một database phải có 1 sessionfactory

### 33. Validation dữ liệu là gì?

- Validation dữ liệu là ràng buộc dữ liệu đầu vào để đảm bảo tính đúng đắn của dữ liệu

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

Có 3 tầng validation :

UI layer	Business layer	Data layer
<ul style="list-style-type: none"> <li>- Phía client</li> <li>- Dùng ngôn ngữ FE</li> <li>- Validate trực tiếp, phản hồi nhanh, giảm tải cho sever</li> <li>- Bảo mật không cao, nhiều trường hợp không xử lý được</li> </ul>	<ul style="list-style-type: none"> <li>- Phía sever</li> <li>- Dùng ngôn ngữ BE</li> <li>- Ràng buộc chặt chẽ, độ chính xác cao, xử lý đa dạng</li> <li>- Khi có quá nhiều xử lý sẽ gây ảnh hưởng đến sever</li> </ul>	<ul style="list-style-type: none"> <li>- Phía database</li> <li>- Dùng các ràng buộc của sql (not null, check, ..)</li> <li>- Cú pháp đơn giản dễ xử dụng</li> <li>- Khó bắt lỗi nhập liệu, tốc độ chậm, ảnh hưởng nhiều đến database làm web bị chậm</li> </ul>

Các phương pháp validate :

- Annotation Validate : dùng các anotation có sẵn của spring như :
  - @NotNull : không được null, request gửi lên không mất trường
  - @NotEmpty : không được trống/empty
  - @NotBlank : không đc null/khoảng trắng(chuỗi)
  - @Size @Max @Min
  - @Email
  - @Pattern
  - @DateTimeFormat
- Custom Validate :
  - Tại class Dto, implement interface Validator và triển khai method validate()
  - Tại Controller, khởi tạo 1 đối tượng Dto -> truyền đối tượng cần validate và binding result vào method validate
- Custom anotation

### 35. Binding Result là gì?

- Binding Result là 1 đối tượng, nơi spring chứa các kết quả xác thực trong quá trình validate

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

- Aspect Oriented Programming là kỹ thuật lập trình hướng khía cạnh bổ sung cho hướng đối tượng, phân tách chương trình thành các module riêng lẻ, phân biệt, không phụ thuộc lẫn nhau và nó không liên quan đến nghiệp vụ chính như: transaction, security, logging

- Gồm 4 yếu tố :
  - Aspect : là 1 khối độc lập với OOP, dùng để hỗ trợ cho OOP
  - Advice : là các hoạt động mà Aspect dùng để quan sát OOP
  - PointCut : là tập hợp điểm có khả năng xảy ra Advice
  - JoinPoint : là các điểm xảy ra Advice

37. Các loại Advice?

- Before : thực hiện trước khi method hoạt động
- After : thực hiện sau khi method hoạt động
- AfterThrowing : thực hiện sau khi xảy ra exception
- AfterReturning : thực hiện sau khi method trả về giá trị thành công
- Around : thực hiện trước khi method hoạt động

38. JoinPoint là gì?

- JoinPoint là 1 vị trí nào đó trong luồng thực thi của chương trình mà advice sẽ được đan (weave) vào chương trình

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

- Chia nhỏ chương trình thành nhiều module riêng rẽ, hoạt động độc lập và không phụ thuộc nhau
- Khi hoạt động, chương trình sẽ kết hợp các module lại để thực hiện các chức năng nhưng khi sửa đổi 1 chức năng thì chỉ cần sửa 1 module

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

- Dùng try catch
- Dùng throw, throws
- Dùng @ExceptionHandler

41. Cookie là gì?

- Cookie là thành phần lưu trữ dữ liệu được tạo ra ở sever và lưu trữ trên trình duyệt của người dùng. Thông tin này được dùng để nhận biết khi người dùng truy cập 1 trang web

42. Phân biệt Session và Cookie

- Giống : đều được dùng để lưu thông tin tạm thời
- Khác :

	Cookie	Session
Kiểu dữ liệu	Chỉ dạng text(string)	Bất kì kiểu dữ liệu nào
Thời gian tồn tại	<ul style="list-style-type: none"> <li>- Khi hết thời gian sống</li> <li>- Cố tình xóa</li> </ul>	Khi hết phiên làm việc : <ul style="list-style-type: none"> <li>+ Tắt trình duyệt</li> <li>+ Tắt sever</li> <li>+ Hết thời gian TimeOut</li> </ul>
Nơi lưu trữ	Client	Sever (đồng thời tạo cookie JSESSIONID để phân biệt các user)
Bảo mật	Kém bảo mật	Bảo mật hơn

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

- Đặt @SessionAtributes trùng tên với @ModelAttribute
- Sử dụng @SessionAttribute trước tên của tham số trong method của controller -> lấy xuống sử dụng

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

- Webservice là tập hợp bao gồm các tiêu chuẩn và giao thức để trao đổi dữ liệu giữa các ứng dụng, các hệ thống
- Vd : fb, ins sử dụng web service để chạy trên nhiều nền tảng khác nhau như ios, android,...

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

	Webservice	Website
Input	Các giao thức http (get, post, delete, put/patch)	Thông qua các ô input, các sự kiện (onclick, submit,...)
Output	Các dữ liệu dưới dạng xml hoặc json	Có layout (html, css...)
How	Độc lập nền tảng không phụ thuộc bất kỳ ngôn ngữ nào	Đa nền tảng nhưng phải hoạt động dựa trên trình duyệt
Who	Các ứng dụng sử dụng	Người dùng sử dụng

#### 46. SOAP là gì? RESTful là gì?

- Simple Object Access Protocol (SOAP) : là giao thức truy cập webservice dựa trên xml có thể tương tác với ngôn ngữ lập trình khác
- Representation State Tranfer (REST) : là một kiểu kiến trúc chứ không phải giao thức cung cấp các quy tắc để xây dựng webservice (định nghĩa dưới dạng xml hoặc json)

#### 47. jQuery là gì? Ajax là gì?

- jQuery là thư viện của js, dùng để tối ưu (rút gọn) code trong js khi làm việc với các thẻ html
- Ajax là kỹ thuật trao đổi dữ liệu bất đồng bộ với web sever, thay đổi 1 phần trang web mà không cần load lại trang

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

#### 49. @RequestHeader và @ResponseHeader có ý nghĩa gì

- @RequestHeader : được sử dụng khi muốn lấy dữ liệu được truyền bằng Header của 1 request
- @ReponseHeader : được sử dụng khi muốn trả thêm dữ liệu cho client ở phần trên cùng của mỗi reponse

#### 50. I18n và L10n là gì?

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

#### 52. Interceptor là gì?

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

- Bảo mật là sự hạn chế khả năng lạm dụng tài nguyên và tài sản
- Cơ chế bảo mật trong Spring hoạt động dựa trên xác thực (Authentication) và phân quyền (Authorization)

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

- Authetication : là hoạt động nhằm xác thực 1 thông điệp hoặc đối tượng nào đó đáng tin cậy
- Authorization : là quá trình xác định xem 1 đối tượng có quyền truy cập vào 1 tài nguyên cụ thể nào đó để thực hiện 1 số hành động hay không

#### 55. Liệt kê các cơ chế xác thực

- OTP ( one time password ) : mật khẩu sử dụng 1 lần thường được dùng trong lớp bảo mật cuối cùng của các giao dịch trực tuyến
- HTTP basic : mã hoá chuỗi username + password dựa trên các thuật toán như md5, bas64

- Token : thường sử dụng JWT ( json web token) gồm có 3 thành phần header, payload, signature
- Signature : chữ ký số
- Cookie : xác thực dựa trên các thông tin được lưu ở cookie

56. CSRF là gì

- Cross site request forgery (CSRF) : là kỹ thuật tấn công giả mạo chính chủ thể của nó bằng cách sử dụng quyền chứng thực của người dùng đối với 1 website

57. XSS là gì

- Cross Site Scripting là cách tấn công vào trang web để đánh cắp thông tin người dùng thông qua nhúng JS vào trang web

58. Spring Boot là gì?

- Spring boot là 1 framework của java nó giúp đơn giản hoá quá trình lập trình 1 ứng dụng với spring

59. Sự khác nhau giữa SpringBoot và SpringMVC là gì?

Spring boot	Spring mvc
+ Là 1 module của spring để đóng gói ứng dụng dựa trên spring	+ Là 1 framework dùng để xây dựng các ứng dụng web dựa trên spring
+ không cần cấu hình theo cách thủ công	+ cần cấu hình theo cách thủ công
+ giảm thời gian phát triển và tăng năng suất	+ cần nhiều thời gian để phát triển

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

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

- Spring security là 1 loạt các bộ lọc giúp chúng ta xác thực và phân quyền người dùng trước khi họ truy cập vào tài nguyên cần chúng ta
- @EnableWebSecurity là 1 anotation khi gắn ở 1 bean bất kỳ thì sẽ kích hoạt tính năng Spring security của ứng dụng web

62. Trình bày Spring Boot JPA? Cài đặt?

- Spring boot jpa giúp chúng ta sử dụng jpa trong spring 1 cách dễ dàng hơn

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

- Jpa Repository<T,ID> : T: class Entity, Id: kiểu dữ liệu khoá chính

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

- @ConfigurationProperties là anotation cho phép truy xuất các thuộc tính được đặt trong các tệp cấu hình 1 cách tự động và mạnh mẽ