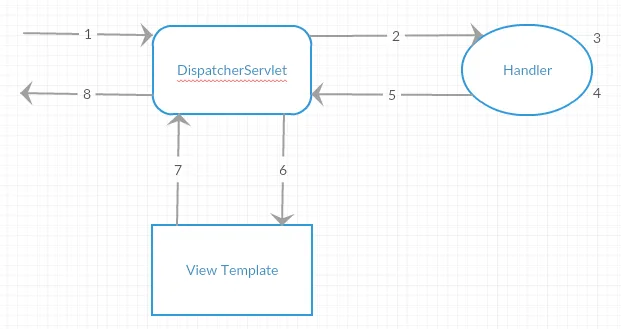
1 Vẽ và trình bày cách Spring MVC xử lý request



1. Request từ người dùng tới ứng dụng web của chúng ta, **DispatcherServlet**sẽ là đối tượng nhận request.
2. Tìm và điều hướng request tới handler phù hợp, ở đây là các Controllers trong ứng dụng web của chúng ta.
3. Xử lý request
4. Chuẩn bị model và chọn view hiển thị.
5. Trả về kết quả xử lý request cho **DispatcherServlet.**
6. **DispatcherServlet**sẽ gọi View Template phù hợp để xử lý việc hiển thị trên giao diện bằng cách sử dụng model.
7. View template trả kết quả về cho **DispatcherServlet.**
8. Trả response về cho người dùng.

2Trình bày cơ chế Dependency Injection trong Spring

\*Dependency Injection là 1 design pattern ( nguyên tắc thiết kế ) trong lập trình- nguyên tắc này sẽ giúp khởi tạo các phụ thuộc áp dụng để giảm sự phụ thuộc để dễ dàng cho viết code , thực hiện test , bảo trì và thay đổi source code về sau Các phụ thuộc nên là interface để dễ dàng cho viết bảo trì và thay đổi

\*Dependency Injection trong Spring Dependency Injection

trong Spring đang được thực hiện dựa vào Spring IOC container Spring IOC container sẽ giúp implement DI bằng cách khởi tạo các dependency , sau đó tiêm vào các phụ thuộc

3 Có bao nhiêu cách để thực hiện Dependency Injection? Trình bày?

\* Có 3 cách để thực hiện DI

+ 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 ?

+ Framework là các ứng dụng phần mềm có tính trừu tượng(abstraction), cung cấp các tính năng chung và thông dụng, có thể tùy biến để tạo nên những ứng dụng cụ thể khác nhau

+ Mỗi framework cung cấp một phương pháp riêng biệt để xây dựng và triển khai ứng dụng

+ Mỗi framework bao gồm một môi trường tổng thể, tái sử dụng được nhằm cung cấp các chức năng và công cụ để hỗ trợ quá trình phát triển ứng dụng

+ Điểm khác biệt lớn nhất giữa Framework và Library đó chính là cơ chế “Inversion of Control”(đảo ngược kiểm soát)

5 Spring Framework là gì ?

Spring framework là một Framework phát triển các ứng dụng Java được sử dụng bởi hàng triệu lập trình viên. Nó giúp tạo các ứng dụng có hiệu năng cao, dễ kiểm thử, sử dụng lại code…

6 Lợi ích của Spring Framework?

+Spring nhẹ và trong suốt (nhẹ: kích thước nhỏ, version cơ bản chỉ khoảng 2MB; trong suốt: hoạt động một cách trong suốt với lập trình viên)

+Spring là một mã nguồn mở, được phát triển, chia sẻ và có cộng đồng người dùng rất lơn.

+Spring Framework được xây dựng dựa trên 2 nguyên tắc design chính là: Dependency Injection và Aspect Oriented Programming.

+Những tính năng core (cốt lõi) của Spring có thể được sử dụng để phát triển Java Desktop, ứng dụng mobile, Java Web. Mục tiêu chính của Spring là giúp phát triển các ứng dụng J2EE một cách dễ dàng hơn dựa trên mô hình sử dụng POJO (Plain Old Java Object)

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

**Inversion of Control (IoC)** là một nguyên lý thiết kế trong công nghệ phần mềm trong đó các thành phần nó dựa vào để làm việc bị đảo ngược quyền điều khiển khi so sánh với lập trình hướng thủ thục truyền thống.

8 Bean là gì?

bean là những module chính của chương trình, được tạo ra và quản lý bởi Spring IoC container

9 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.

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

10 @Autowired là gì?

@Autowired đánh dấu cho Spring biết rằng sẽ tự động inject bean tương ứng vào vị trí được đánh dấu.

11 @Component có ý nghĩa gì?

@Component: là một annotation của class. Nó dùng để đánh dấu class Java là một bean.

12 Trình bày ý nghĩa của Front Controller

* Front Controller sẽ sử dụng Handler Mapping để biết được controller nào sẽ xử lý request đó

13 Trình bày ý nghĩa của ModelAndView Interface

14 Trình bày ý nghĩa của ModelMap

15 Trình bày ý nghĩa của ViewResolver Interface

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

* Để tạo tài nguyên: sử dụng HTTP POST
* Để cập nhật tài nguyên: sử dụng HTTP PUT

18 @RequestMapping làm gì?

**@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

Cơ chế data binding hoạt động dựa trên đối tượng data binding  
- validate dữ liệu

- property editors biến đổi các input thành các thuộc tính  
- formatter biến đổi thuộc tính thành đối tượng

20 Thuộc tính consumes trong các Request Mapping là gì ?

Để quy định định dạng dữ liệu mà một request có thể process khi người dùng request tới  chúng ta có thể sử dụng annotation @RequestMapping với thuộc tính consumes trong Spring MVC định dạng dữ liệu khi user request tới

21 Thymeleaf là gì?

Thymeleaf là một bộ sử lý view được sử dụng cho các ứng dụng web cụ thể ở đây là spring  
Thymeleaf được xây dựng phù hợp với tiêu chuẩn web đặt biệt là htm5

Thymeleaf cho phép xử lý 6 loại temple, html, xml,js,css,text, raw

22 Sử dụng lặp trong Thymeleaf như thế nào? Sử dụng điều kiện trong Thymeleaf như thế nào?

Cú pháp đầy đủ của **th:each** bao gồm 2 biến, biến phần tử (item variable) và biến trạng thái (state variable).

Vòng lặp:

- th:each =” item variable, status  variable: ${list}”

trong đó item variable là phần tử danh sách mỗi lần lặp

Biến trạng thái (status  variable) là một đối tượng hữu ích, nó chứa các thông tin cho bạn biết trạng thái hiện tại của vòng lặp

**-Index** Chỉ số hiện tại của phép lặp , bắt đầu với số 0.

**-count** Chỉ số hiện tại của phép lặp , bắt đầu với số 1

-**size** Tổng số phần tử trong danh sách.

**-even/odd** Kiểm tra xem chỉ số hiện tại của phép lặp là chẵn hay lẻ.

23 Formatter là gì ? Converter là gì ?

24 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

* Cho phép truy xuất dễ dàng đến dữ liệu thông qua các đối tượng lập trình
* Giúp lập trình viên tập trung thao tác với các đối tượng, không cần quá quan tâm đến CSDL thực tế đang dùng
* Tự động hoá rất nhiều thao tác như quản lý giao dịch, validate dữ liệu, cache

25 JPA là gì? Spring Data JPA là gì ?

**JPA – Java Persistence API: cung cấp các dặt tả để duy trì đọc quản lý dữ liệu đối tượng java đến các quan hệ trong csdl**

JPA cung cấp một mô hình POJO persistence cho phép ánh xạ các table/các mối quan hệ giữa các table trong database sang các class/mối quan hệ giữa các object.

26 Hibernate là gì ? Hibernate làm gì với database

27 Liệt kê một số annotation của hibernate?

28 Entity là gì?

29 Persistence Context & Entity Manager là gì?

30 Câu lệnh truy vấn động là gì? Câu lệnh truy vấn tĩnh là gì?

31 Trạng thái của Entity bao gồm những gì?

32 Có bao nhiêu loại mapping trong Hibernate?

33 SessionFactory là gì?

34 Validation dữ liệu là gì?

35 Trình bày cách triển khai validate dữ liệu trong Spring

36 Binding Result là gì?

37 AOP là gì? AOP gồm những yếu tố nào?

38 Các loại Advice?

39 JoinPoint là gì?

40 Nêu cách triển khai AOP mà bạn dùng trong dự án

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

42 Cookie là gì?

43 Phân biệt Session và Cookie

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

45 Web Service là gì? Lấy ví dụ về web service

46 Phân biệt Web Service và Web truyền thông

47 SOAP là gì? RESTful là gì?

48 jQuery là gì? Ajax là gì?

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

50 @RequestHeader và @ResponseHeader có ý nghĩa gì

51 I18n và L10n là gì?

52 Nêu cách triển khai I18N và L10N trong dự án Spring

53 Interceptor là gì?

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

55 Authentication là gì ? Authorization là gì ?

56 Liệt kê các cơ chế xác thực

57 CSRF là gì

58 CORS là gì

59 Spring Boot là gì?

60 Sự khác nhau giữa SpringBoot và SpringMVC là gì?

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

62 Trình bày Spring Security? @EnableWebSecurity làm gì?

63 Trình bày Spring Boot JPA? Cài đặt?

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

65 Bạn có biết annotation @ConfigurationProperties? Trình bày?