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

* Sau khi user gửi request lên server, dispatcherServlet sẽ tiếp nhận request và điều hướng tới controller thích hợp, sau đó controller sẽ gọi service, service gọi repo để thao tác với DB, sau đó dữ liệu được chuyển hóa thành model và gửi lại cho controller. Controller sẽ trả lại tên view và model cho dispatcherServlet, dispathcherServl sẽ dung viewresolver để ánh xạ tên view đến view tương ứng và hiển thị cho người dung.

1. Trình bày cơ chế Dependency Injection

* Khởi tạo đối tượng

|  |  |
| --- | --- |
|  | 1. Có bao nhiêu cách để thực hiện Dependency Injection? Trình bày |
|  | * Có 3 cách để thực hiện DI   + Setter Injection  + Constructor Injection  + Interface injection  Spring chỉ hổ trợ setter và constructor |

* Biết được object nào mà class đang cần để nhúng vào

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

• 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

• Diểm khác biệt lớn nhất giữa Framework và Library đó chính là cơ chế “InversionofControl”

• Với Library: Ứng dụng nắm quyền điều khiển (control)

• Với Framework: Framework nắm quyền điều khiển

Lợi ích của Framework

• Framework giải quyết các vấn đề thông dụng, giúp lập trình viên tập trung vào xử lý nghiệp vụ

• Giúp tăng tốc độ phát triển

• Cung cấp môi trường làm việc tiêu chuẩn, giúp dễ giao tiếp giữa các bên khi cùng tham gia phát triển

1. Spring Framework

• Spring Framework cung cấp một mô hình đầy đủ cho việc phát triển và cấu hình các hệ thống Java lớn

• Các tính năng lõi:

• Core: IoC container, Events, Resources, i18n, Validation, Data Binding, Type Conversion, SpEL, AOP.

• Testing: Mock objects, TestContext framework, Spring MVC Test, WebTestClient.

• Data Access: Transactions, DAO support, JDBC, ORM, Marshalling XML.

• Web Servlet: Spring MVC, WebSocket, SockJS, STOMP messaging.

• Web Reactive: Spring WebFlux, WebClient, WebSocket.

• Integration: Remoting, JMS, JCA, JMX, Email, Tasks, Scheduling, Cache.

• Languages: Kotlin, Groovy, Dynamic languages.

1. Lợi ích Spring Framework

+Spring Framework cho phép lập trình viên sử dụng POJOs mà ko cần làm việc với EJB

+ Spring hổ trợ nhìu loại công nghệ ORM Framework, JEE, các thư viện có khả năng tạo ra lập trình

+ Module web của Spring được thiết kế theo mô hình MVC nên nó cung cấp đầy đủ tính năng giúp thay thế các web framework khác như Struts.

1. Nguyên lý đảo ngược quyền điều khiển (Inversion of Control)

* Khi không có IOC thì nếu muốn khởi tạo và quản lý đối tượng thì bản thân ta phải là người làm. Còn khi sử dụng IOC thì việc quản lý đối tượng sẽ do bên thứ 3 đảm nhận (Framework)

1. Lợi ích IOC

* Tách rời việc thực thi và triển khai
* Dễ dàng phân tách module hơn
* Dễ kiểm thử hơn, bằng cách tách rời các thành phần riêng lẻ

1. Bean là gì

* Bean là những module chính của chương trình được tạo ra và quản lý bởi spring ioc container. Bất kỳ class Java POJO nào cũng có thể là Spring Bean nếu nó được cấu hình và khởi tạo thông qua container bằng việc cung cấp các thông tin cấu hình (các file config .xml, .properties..)

1. Trong spring có bao nhiên bean Scope

A screenshot of a computer

Description automatically generated

1. @Autowired là gì?

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

1. @component là gì?

Là 1 annotation đánh dấu trên các class để cho chúng biết là được quản lý bởi Spring boot. Khi chạy chương trình thì spring sẽ quét qua và tạo bean cho chúng

1. Ý nghĩa controller

Spring controller nhận ủy quyền xử lý các request từ phía front controller. Công việc cơ bản của nó là nhận dữ liệu cần thiết từ front controller, xác định view cần sử dụng, triệu gọi lớp xử lý nghiệp vụ cần thiết, sắp xếp dữ liệu mà view cần đến vào model, gửi tên view cùng model lại cho front controller.

1. Ý nghĩa ModelAndView

* ModelAndView đai diện cho 1 view cùng với các dữ liệu sử dụng trong view đó.
* ModelAndView có thể kèm theo status của response

1. ViewResolver ?

* ViewResolver có nhiệm vụ ánh xạ tên view đến view tương ứng

1. Khác nhau POST, GET, PUT, DEL, PATH

A close up of text

Description automatically generated

1. Modelmap ?

* Nó giống như model (model dung để truyền dữ liệu từ controller sang view để hiển thị ). Ưu điểm của Modelmap là cung cấp khả năng chuyển tiếp 1 tập giá trị và xử lý chúng như thể chúng nằm trong Map.

1. ModelMap ?

* Giống như model ( dung để chuyển dữ liệu từ controller sang view để hiển thị ). Nó có khả năng chuyển 1 tập dữ liệu và xử lý chúng như thể trong map .

1. RequestMapping?

* Là cơ chế ánh xạ các request tới các action tương ứng trong controller

1. Databinding là gì ? cơ chế ?

* Là cơ chế liên kết dữ liệu đầu vào ( hoặc đầu ra) với các đối tượng model
* Databinding giúp cho việc tương tác với dữ liệu trở dễ dàng hơn
* Khi tương tác với form, dữ liệu trên form sẽ được chuyển đổi thành các thuộc tính của đối tượng liên kết với nó
* Databinding hoạt động dựa trên databider
* khi người dung gửi dữ liệu lên server, spring MVC sử dụng databider để ánh xạ các tham số của request tới các thuộc tính của dối tượng được chỉ định trong controller
* có 3 thành phần liên quan đến việc ánh xạ dữ liệu

+ formatter : chuyển đổi dữ liệu này sang kiểu dữ liệu khác

+ propertyEditor: chuyển đổi string sang kiểu dữ liệu khác

+ validator : kiểm tra tính hợp lễ của dữ liệu

1. consume trong requestMapping?

* dùng để ánh xạ tới thuộc tính content-type của request

1. template ? template Engine? Thymeleaf?

* Template là một cách thức thông dụng để hiển thị kết quả của một request về cho người dùng. Template định nghĩa một khuôn mẫu sẵn, sau đó ứng dụng sẽ đổ dữ liệu vào template để tạo ra một response thực tế và trả về cho người dùng.
* Template giúp cho việc xây dựng giao diện trở nên dễ dàng hơn thông qua việc tách các mã nguồn nghiệp vụ (chứa trong controller) với mã nguồn giao diện (chứa trong các file template).
* Template đó chính là 1 mẫu bố cục chung cho tất cả các trang có sử dụng lại những thành phần giống nhau mà không phải viết lại toàn bộ, từ đó trên mỗi trang, chỉ cần thay đổi ở một số nơi được chỉ định trên trang từ template.
* Thymeleaf là 1 bộ xử lý view được sử dụng cho các ứng dụng web và các ứng dụng độc lập
* Thymeleaf có thể xử lý HTML, XML, JS, CSS và cả text
* Thymeleaf cung cấp 1 cách thức xử xây dựng template hiện đại dễ bảo trì, phù hợp với các tiêu chuẩn của web đặc biệt là HTML5

1. Formatter và converter?

* Formatter chuyển đổi từ string sang 1 kiểu dữ liệu khác và ngược lại
* Converter chuyển đổi từ kiểu dữ liệu này sang kiểu dữ liệu khác

1. ORM

* Là kỹ thuật lập trình liên kết giữa các đối tượng trong lập trình với các đối tượng trong CSDL
* ORM cho phép truy xuất dễ dàng đến dữ liệu thông qua các đối tượng lập trình
* ORM giúp lập trình viên tập trung thao tác các đói tượng, ko cần quá quan tâm đến CSDL

1. JPA ? Spring data jpa?

* JPA là 1 bộ đặc tả tiêu chuẩn của java, trong đó mô tả các thao tác quản lý dữ liệu quan hệ trong các ứng dụng java
* Spring data JPA giúp cho việc sử dụng JPA trong ứng dụng spring trở nên dễ dàng hơn. Có thể sử dụng spring data JPA với các framework như hibernate, OpenJPA,….

1. Hibernate? Hibernate làm gì với DB?

* Hibernate là 1 thư viện ORM mã nguồn mở giúp lập trình viên viết các ưng dụng java có thể map các object với hệ quản trị cơ sở dữ liệu quan hệ, và hổ trọ thực hiện các khái niệm lập trình hướng đối tượng với cơ sở dữ liệu quan hệ.

1. Entity?

* Entity là 1 đói tượng đại diện cho dữ liệu ở trong ứng dụng
* Entity thường là POJO
* Entity sẽ ánh xạ tới 1 bảng trong CSDL
* Entity cần tuân thủ

+ được gắn với annotation Entity

+ có 1 constructor ko tham số

+ các đói tượng cần được khai báo private, protected hoặc package-private

1. Persistence Context & Entity Manager ?

• Persistence Context là tập các thể hiện của entity được quản lý, tồn tại trong một kho dữ liệu

• Interface EntityManager:

• Khai báo các phương thức để tương tác với persistence context

• Tạo hoặc xoá các thể hiện của entity

• Tìm kiếm entity theo khoá chính

• Thực thi các câu lệnh truy vấn lên entity

1. 3 trạng thái của đối tượng trong hibernate

* Transient : ở trạng thái này khi nó mới đc tạo bằng cách sử dụng từ khóa new, nhưng chưa được liên kết với bất kỳ phiên làm việc hibernate nào
* Pesistent: khi 1 đối tượng đã được liên kết với 1 phiên làm việc hibernate
* Detached: khi 1 đối tượng đã liên kết với 1 phiên làm việc hibernate trước đó, nhưng phiên làm việc trước đó đã kết thúc.

1. sessionFactory?

SessionFactory là một trong những thành phần quan trọng trong Hibernate, mà nhiệm vụ chính của nó là tạo ra các đối tượng Session. SessionFactory là một đối tượng không thay đổi trong quá trình thực thi ứng dụng, được sử dụng để tạo ra các phiên làm việc (Session) mới khi cần thiết.

1. Validation dữ liệu ?

* Thực hiện đánh giá / xác minh tính hợp lệ của dữ liệu đầu vào

1. Binding result?

Binding Result là một đối tượng được sử dụng trong Spring Boot để lưu trữ kết quả của quá trình validation. [Nếu validation fail, method vẫn sẽ được gọi vào, và chúng ta có thể check đối tượng Binding Result kia có chứa lỗi hay không, từ đó xử lý phù hợp](https://viblo.asia/p/dung-validation-bang-tay-trong-spring-boot-phan-1-3P0lPGyoZox" \t "_blank)

1. AOP ? AOP gồm những yếu tố nào?

* AOP là 1 kỹ thuật lập trình nhằm phân tách chương trình thành các module riêng lẽ, phân biệt và ko phụ thuộc
* Khi hoạt động thì chương trình sẽ kết hợp các module lại để thực thi chức năng nhưng khi chỉnh sửa thì chỉ cần sửa 1 module
* AOP chia làm 2 phần concern:
  + Core concern: những xử lý chính của chương trình
  + Cross cutting concern: những xử lý phụ cần được thực thi của chương trình khi core concern gọi

1. Các loại advice?

* Before : thực hiện trước điểm joinpoint
* After : thực hiện sau điểm joinpoint
* Around: thực hiện cả trước và sau
* After Running: thực hiện sau điểm joinpoint và điều kiện là ko xảy ra lỗi
* After throwing : tương tự nhưng điều kiện là xảy ra lỗi

1. JoinPoint là gì ?

* Joinpoint là những điểm trong chương trình có thể chèn xử lý phụ vào, nó có thể là 1 phương thức được gọi, hay là 1 lỗi được throw

1. Xử lý ngoại lệ trong Spring Web MVC ?

* Dung try-catch
* Dùng annotation @ExceptionHandler
* Dung HandlerExceptionResolver : dung để xử lý ngoại lệ tại servlet

1. Cookie là gì? Session ? khác nhau ?

* Cookie được dung để lưu những thông tin tạm thời. Tệp tin coolie được truyền từ server tới browser và được lưu trữ trên máy tính của bạn khi bạn truy cập vào ứng dụng. Khi tắt browser cũng ko mất đi các giá trị vì chúng ta lưu trên máy tính của mình.
* Session là 1 phiên làm việc. là 1 cách giao tiếp giữa client với server. Một session bắt đầu khi client gửi request tới server, nó tồn tại suốt trong quá trình làm việc và mất đi khi tắt browser hoặc hết timeout

A white text on a white background

Description automatically generated

1. Web Service là gì? Lấy ví dụ

+ Webservice là các thành phần ứng dụng được hiển thị dưới dạng các dịch vụ trên WWW.

+ Webservice xây dựng các chuẩn mở và sử dụng các giao thức mở để giao tiếp

+ Webservice hoạt động như một server trong mô hình ứng dụng client server sử dụng giao thức HTTP/HTTPS và chỉ thực hiện một tác vụ cụ thể

+ Với dữ liệu đầu vào xác định, web service sẽ xử lý và trả về dữ liệu đầu ra theo chuẩn đảm bảo mọi ứng dụng có thể hiểu và sử dụng mà không quan tâm đến loại thiết bị, hệ điều hanh hay kiến trúc phần mềm, ngôn ngữ sử dụng

+ Kiểu dữ liệu đầu ra phổ biến của một web service thường là XML hoặc JSON

1. Phân biệt?

A screenshot of a computer

Description automatically generated

1. REST là gì? RESTful là gì?

* 'REST là một kiểu cấu trúc cung cấp các quy tắc để xây dựng webservice.
* REST định nghĩa dữ liệu dưới dạng XML hoặc JSON và truyền thông qua mạng internet sử dụng giao thức HTTP.
* Các trang web service xâyd ựng dựa trên REST được gọi là RESTful, và chủ yếu nhằm xử lý các hoạt động CRUD trên dữ liệu