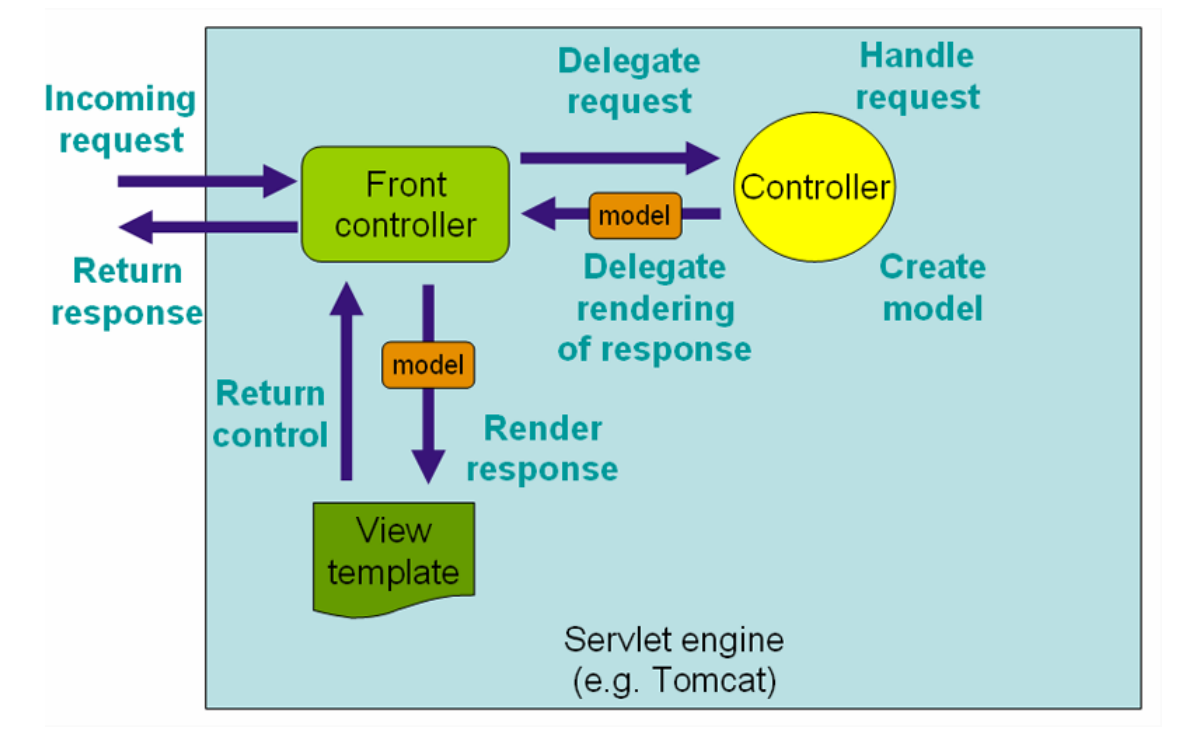
1. Vẽ và trình bày cách Spring MVC xử lý request

- Đầu tiên khi có request từ người dùng tới, front controller (ở đây là class dispatcher servlet) sẽ tiếp nhận toàn bộ các request đó

- Thông qua URL handle mapping mà front controller sẽ điều hướng request tới controller tương ứng

- Tầng model sẽ xử lý các nghiệp vụ tính toán logic và thông qua controller để trả về model and view cho front controller

- Lúc này thông qua view resolver để ánh xạ tên view đến view tương ứng

- Cuối cùng front controller sẽ chuyển view đó đến cho người dùng

2 Trình bày cơ chế Dependency Injection trong Spring

- DI – dependency injection là 1 cơ chế để triển khai IOC, được sử dụng để loại bỏ sự phụ thuộc giữa các mã của chương trình, giúp cho việc bảo trì, mở rộng chương trình dễ dàng hơn

- Các depency được cung cấp và điều khiển bởi framework hoặc container

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

- Có 3 cách để thực hiện DI: khác nhau ở chỗ là vị trí tiêm phụ thuộc. spring chỉ có 2 loại đầu, không có loại thứ 3

+ Constructor injection

+ Setter injection, có biến thể là field injection – là loại injection hay sử dụng phổ biến nhất trong spring boot

+ interface injection: loại này không có trong spring

4 Framework là gì ? Framework khác Library chỗ nào ?

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

- Mỗi framework cung cấp 1 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 chứng năng và công cụ để hỗ trợ quá trình phát triển ứng dụng

\* Khác biệt giữa Framework và Library

Điểm khác biệt lớn nhất giữa Framework và Library đó chính là cơ chế “Inversion of Control”

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

• Với Framework: Framework nắm quyền điều khiển

Ngoài ra:

+Library: được tự do sd, k bị ràng buộc bởi cấu trúc hoặc quy tắc nào.

+FrameWork: phải tuân thủ theo luật chơi của Framework. Framework định hình cấu trúc của ứng dụng 🡪 Lập trình viên phải điền mã vào ô trống mà framework cung cấp.

5 Spring Framework là gì ?

- Là 1 framework dựa trên java

- Được sd để xây dựng các ứng dụng web

- Được cài đặt đầy đủ các đặc tính của MVC Pattern

\* Spring framework được là một [framework](https://nq.com.vn/framework-la-gi) phổ biến của nền tảng java cung cấp đầy đủ tính năng cho việc phát triển và cấu hình các hệ thống java

1. Lợi ích của Spring Framework?

- Spring là một framework giúp các nhà phát triển xây dựng những hệ thống và ứng dụng chạy trên JVM một cách đơn giản, tiện gọn, nhanh chóng và mềm dẻo.

- 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

- Các framework thường có cộng đồng phát triển lớn, các giải pháp đã được đánh giá và thử nghiệm, hệ sinh thái đầy đủ giúp nhanh chóng xây dựng được các giải pháp tuỳ biến

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

- IOC là 1 nguyên lý phát triển phần mềm, trong đó việc điều khiển các đối tượng hoặc các thành phần của hệ thống sẽ được thực hiện bởi framework hoặc các container

- IOC cho phép framework nắm giữ quyền điều khiển luồng thực thi của hệ thống và gọi các mã nguồn khác

8 Bean là gì?

- Bean là 1 đối tượng do Spring IOC container quản lý, tự tạo, tự quản lý vòng đời.

- Việc tạo bean sẽ được quyết định bằng cách đặt các annotation hoặc thông qua config

Trong Spring framework, Bean là một đối tượng được quản lý bởi container của Spring. Bean được định nghĩa bởi một class và có thể được sử dụng bởi các lớp khác thông qua dependency injection.

Có hai cách để triển khai Bean trong Spring:

1. Sử dụng Annotation: Để đánh dấu một class là một Bean, bạn có thể sử dụng các Annotation như @Component, @Service, @Repository, @Controller, @Configuration, @Bean, v.v. Ví dụ, để định nghĩa một Bean sử dụng Annotation, bạn có thể sử dụng @Component như sau:
2. Sử dụng XML configuration: Bạn có thể sử dụng tệp cấu hình XML để định nghĩa các Bean trong Spring. Ví dụ, để định nghĩa một Bean sử dụng XML configuration, bạn có thể sử dụng cấu trúc như sau:

9 Trong Spring có bao nhiêu Bean Scope?

* 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.

10 @Autowired là gì?

- @Autowired là 1 annotation của Spring, mục đích đánh dấu vị trí sẽ tiêm phụ thuộc (DI)

11 @Component có ý nghĩa gì?

- @Component được sử dụng để định nghĩa một khuôn mẫu chung cho toàn bộ dự án.

- Class được đánh dấu với annotation @Component thì sẽ được tạo thành 1 bean và sẽ được spring container quản lý, đồng thời các class này sẽ phải không thuộc tầng controller, service hay repository thì mới dùng @component được

12 Trình bày ý nghĩa của Front Controller và Controller

- Mục đích của Front controller là để handle tất cả request từ client tới server sau đó dựa vào URL handle mapping để điều hướng tới controller tương ứng và là nơi xử lý tất cả các yêu cầu 🡪 có thể dễ dàng xác thực, phân quyền, ghi log,…

- Ý nghĩa của Controller là: 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 thông tin về view cùng model lại cho front controller.

13 Trình bày ý nghĩa của ModelAndView Interface, ModelAndView Class

- ModelAndView Interface : Mục đích là để vừa trả về dữ liệu vừa trả về trang view, nếu 1 tham số thì chỉ trả về view, nếu 3 tham số sẽ vừa trả về view vừa trả về dữ liệu

- ModelAndView Class: có nhiệm vụ quan trọng trong việc tạo ra các hồi đáp (response). Các thể hiện của ModelAndView giúp dễ dàng tạo liên kết giữa data model và view. Trong đó data model là đối tượng chứa dữ liệu để hiển thị lên view

14 Trình bày ý nghĩa của ModelMap

- Class ModelMap là lớp con của LinkedHashMap được sử dụng trong việc xây dựng data model bằng sử dụng kiểu dữ liệu cặp gồm key và value. ModelMap sử dụng phương thức addAttribute() để tạo liên kết giữa model và tên logic trong model. Thuộc tính đặt trong ModelMap có thể sử dụng để hiển thị dữ liệu dạng form trong sự kiện submit form.

15 Trình bày ý nghĩa của ViewResolver Interface

- View resolver là cơ chế xử lý tầng view của spring mvc, dùng để ánh xạ tên view đến view tương ứng

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

- Trong web service, PUT dùng để update dữ liệu, trong khi POST dùng để thêm mới dữ liệu,

\* GET dùng để truy xuất dl, DELETE dùng để xóa dl

17 @RequestMapping làm gì?

Annotation RequestMapping dùng để ánh xạ một request tới một phương thức. Chúng ta có thể sử dụng @RequestMapping chú thích cho một phương thức hoặc một lớp. Một phương thức chú thích bởi annotation @RequestMapping sẽ trở thành phương thức xử lý request và sẽ được gọi khi dispatcher servlet nhận được một request với URL phù hợp.

20 Trình bày cơ chế Data Binding trong Spring

- Data binding là cơ chế liên kết dữ liệu đầu vào và đầu ra với model, giúp cho việc tương tác dl dễ dàng hơn vì được liên kết tới các đối tượng biểu diễn dữ liệu

- Data Binding trong Spring hoạt động dựa trên DataBinder

Cơ chế Data Binding trong Spring: dữ liệu do người dùng nhập vào thông qua DataBinder trả về kết quả là Binding Result. DataBinder bao gồm các bộ xử lý Property editor( biến đổi các input thành các thuộc tính), bộ định dạng Formatter ( chuyển dối các input (string) thành các kiểu dữ liệu mong muốn), bộ lọc dữ liệu Validator( dùng để validate dữ liệu) và bộ chuyển đổi ConversionService

Data Binding trong Spring là quá trình liên kết dữ liệu giữa các đối tượng Java và các thành phần giao diện người dùng như HTML form, JSP hoặc các trình xử lý Thymeleaf. Khi sử dụng Data Binding, Spring sẽ tự động chuyển đổi dữ liệu giữa các đối tượng Java và các thành phần giao diện người dùng, giúp giảm thiểu mã lặp và tăng tính hiệu quả của ứng dụng.

Spring hỗ trợ Data Binding thông qua các lớp DataBinder và các Annotation như @ModelAttribute, @RequestParam, @PathVariable. Các lớp này cho phép Spring tự động chuyển đổi đối tượng Java sang các giá trị của các trường trong form HTML và ngược lại.

Spring cũng hỗ trợ các tính năng nâng cao của Data Binding như Validation. Validation cho phép bạn kiểm tra tính hợp lệ của dữ liệu được nhập vào từ form HTML trước khi nó được liên kết với đối tượng Java. Spring cung cấp các Annotation như @Valid và @Validated để hỗ trợ tính năng Validation này.

21 So sánh RequestParam và PathVariable ?

- Giống nhau: đều thực hiện chung 1 nhiệm vụ là lấy tham số từ người dùng gửi lên server

- Khác nhau: RequestParam có thể lấy dl từ form submit còn PathVariable thì chỉ lấy dl trên URL ở dạng value

Các ưu điểm của path variable là:

1. Tính tường minh: Khi sử dụng path variable, các tham số được truyền rõ ràng và dễ đọc trong URL, giúp cho các yêu cầu HTTP trở nên dễ hiểu và dễ bảo trì hơn.
2. Khả năng mở rộng: Path variable cho phép bạn dễ dàng mở rộng ứng dụng bằng cách thêm các đoạn đường dẫn mới vào URL mà không cần thay đổi các tham số hiện có. Điều này giúp cho việc thêm mới các chức năng hoặc tính năng cho ứng dụng trở nên dễ dàng hơn.
3. Độ linh hoạt: Path variable cho phép bạn tùy chỉnh các giá trị đầu vào cho mỗi yêu cầu HTTP, giúp cho ứng dụng trở nên linh hoạt hơn trong việc xử lý các yêu cầu khác nhau.
4. Khả năng cache: Path variable cho phép trình duyệt lưu trữ các yêu cầu HTTP trong bộ nhớ cache, giúp giảm tải cho máy chủ và cải thiện hiệu suất của ứng dụng.

22 Thymeleaf là gì?

- Thymeleaf là 1 bộ xử lý view được sử dụng cho các ứng dụng web và các ứng dụng độc lập

- Thymeleaf được xây dựng phù hợp với các tiêu chuẩn web, đặc biệt là HTML5

- Thymeleaf cho phép xử lý 6 loại templeate: HTML, XML, Javascript, TEXT, CSS, RAW

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

- Vòng lặp trong Thymeleaf, ta sử dụng cú pháp => th:each = “object, iter :${list}”

- Đối tượng iter trong vòng lặp th: each cung cấp các thuộc tính sau:

+ Index: trả về chỉ số lần lặp hiện tại, bắt đầu từ 0

+ Count: trả về chỉ số lần lặp hiện tại, bắt đầu từ 1

+ Size: trả về tổng số phần tử trong vòng lặp

+ Current: tương ứng với object của lần lặp hiện tại

+ Even/Odd: vòng lặp chẵn hoặc lẽ (boolean)

+ first/last: trả về true nếu vòng lặp đầu tiên/ cuối cùng

- Sử dụng điều kiện trong thymeleaf: cú pháp th:if “${conditions}”. Th if trả về true trong các tường hợp sau:

- Một giá trị Boolean là true

- Một số có giá trị # 0

- Một ký từ có giá trị # ‘0’

- Một chuỗi có giá trị # “false”. “no”, “off”

- Một giá trị không phải là Boolean, số, ký tự hoặc chuỗi

24 Formatter là gì ? Converter là gì ?

- Formatter: Chỉ hỗ trợ chuyển dữ liệu từ client sang Server (tầng web), giúp chuyển dữ liệu từ String (Client) sang kiểu dữ liệu của mình chọn (Server)

- Converter: Hỗ trợ chuyển đổi 2 kiểu dữ liệu bất kỳ, dùng ở đâu cũng được (dùng ở bất cứ tầng nào của ứng dụng) và hay dùng ở back-end

25 ORM là gì ?

- ORM (object-relational mapping): Là 1 kĩ thuật liên kết các đối tượng trong lập trình với các đối tượng trong database

- ORM giúp cho phép truy xuất dễ dàng đến dữ liệu thông qua đối tượng trong lập trình

- ORM giúp lập trình viên tập trung thao tác với các đối tượng mà không cần quan tâm đến database thực tế đang dùng

26 JPA là gì? Spring Data JPA là gì ?

- JPA: java persistence API cung cấp các đặc tả để duy trì, đọc, quản lý dữ liệu từ đối tượng java đến các quan hệ trong cơ sở dữ liệu. JPA cung cấp một mô hình POJO persistence cho phép ánh xạ các table sang các class hoặc ánh xạ các mối quan hệ giữa các table trong database sang các mối quan hệ giữa các đối tượng

- Spring data JPA: giúp cho việc sử dụng JPA trong ứng dụng Spring trở nên dễ dàng hơn, giúp cải tiến JPA tiêu chuẩn, đơn giản hóa tầng truy xuất, tự tạo ra các repository, tự tạo ra các câu lệnh truy vấn JPA thông qua tên method, ghi log, phân trang. Spring data JPA là 1 phần của dự án Spring data, có thể sd Spring data jpa với các framwork khác nhau như: Hibernate, OpenJPA,…

27 Hibernate là gì ? Hibernate làm gì với database

- Hibernate là 1 framework ORM

- JPA là tập hợp các interface được định nghĩa theo quy chuẩn thì hibernate là người triển khai ( implement) các interface này

- Hibernate cung cấp một cơ chế bộ nhớ đệm (caching) để giảm số lần truy cập vào cơ sở dữ liệu, giúp tăng hiệu suất cho ứng dụng của bạn. Khi một giao dịch được kích hoạt, Hibernate sẽ lưu trữ các đối tượng trong session. Khi một truy vấn được thực hiện liên tục, giá trị được lưu trữ trong session sẽ được sử dụng lại. Khi một giao dịch mới bắt đầu, dữ liệu sẽ được lấy lại từ cơ sở dữ liệu và được lưu trữ trong session. Điều này giúp giảm số lần truy cập vào cơ sở dữ liệu và tăng hiệu suất cho ứng dụng của bạn.

Hibernate là một framework ORM (Object-Relational Mapping) cho Java, cho phép các lập trình viên làm việc với cơ sở dữ liệu một cách đơn giản, linh hoạt và hiệu quả. Hibernate giúp tạo ra một lớp trừu tượng giữa cơ sở dữ liệu và ứng dụng, giúp cho việc truy cập và thao tác với dữ liệu trở nên dễ dàng hơn.

Hibernate sử dụng các Annotation và XML để ánh xạ các đối tượng Java với các bảng trong cơ sở dữ liệu. Khi sử dụng Hibernate, các lập trình viên không cần phải viết các câu lệnh SQL phức tạp để truy cập dữ liệu, thay vào đó, Hibernate tự động tạo ra các câu lệnh SQL tương ứng và thực thi chúng.

Hibernate cũng cung cấp các tính năng nâng cao như caching, lazy loading và transaction management. Caching giúp giảm thiểu thời gian truy cập cơ sở dữ liệu bằng cách lưu trữ dữ liệu trong bộ nhớ tạm thời. Lazy loading giúp tối ưu hóa việc tải dữ liệu từ cơ sở dữ liệu bằng cách chỉ tải dữ liệu khi nó thực sự cần thiết. Transaction management giúp quản lý các giao dịch với cơ sở dữ liệu, đảm bảo tính toàn vẹn và độ tin cậy của dữ liệu.

Hibernate là một trong những framework ORM phổ biến nhất trong cộng đồng Java, được sử dụng rộng rãi trong các dự án phát triển ứng dụng web và các ứng dụng Java khác.

28 Liệt kê một số annotation của hibernate?

- @Entity: mô tả 1 class là Entity hoặc là 1 table

- @Table: Mô tả class này sẽ mapping với table trong database

- @Id: mô tả đây là primary key

- @Transient: Dung để mapping 1 kiểu dữ liệu không tạo cột trong database, mô tả column này không tồn tại trong database

- @Column: mô tả đây là 1 column, mô tả tên cột trong database, kiểu dữ liệu trong database

- @OnetoOne: quan hệ 1-1

- @Onetomany: mô ta quan hệ 1-N với 1 bảng khác

- @ManytoOne: mô tả quan hệ N-1 với 1 bảng khác

- @ManytoMany: mô tả quan hệ N-N với bảng khác

- @joinColumn: thường dùng trong mối quan hệ 1-N, N-1 với ý nghĩa foreign key của table có mối quan hệ 1

29 Entity là gì?

- Entity là đối tượng đại diện cho dữ liệu ở trong ứng dụng

- Entity thường là POJO (plain old java object)

- Entity sẽ được ánh xạ tới 1 bảng trong database

30 Persistence Context & Entity Manager là gì?

- Persistence context là tập các thể hiện của entity được quản lý, tồn tại trong 1 kho dữ liệu

- Entity manager là 1 interface trong JPA, cung cấp API để quản lý đối tượng và thực hiện các thao tác trên database như lưu, cập nhật, xóa dl

1. persist(entity): Thêm một đối tượng mới vào cơ sở dữ liệu.
2. find(entityClass, primaryKey): Tìm kiếm một đối tượng trong cơ sở dữ liệu dựa trên khóa chính.
3. merge(entity): Cập nhật một đối tượng đã tồn tại trong cơ sở dữ liệu.
4. remove(entity): Xóa một đối tượng khỏi cơ sở dữ liệu.
5. refresh(entity): Lấy lại trạng thái mới nhất của một đối tượng từ cơ sở dữ liệu.
6. createQuery(queryString): Tạo một câu truy vấn JPA để truy vấn cơ sở dữ liệu.
7. createNamedQuery(queryName): Tạo một câu truy vấn đã được định nghĩa trước trong các tệp tin XML.
8. flush(): Đẩy các thay đổi của đối tượng đang được quản lý bởi Entity Manager xuống cơ sở dữ liệu.
9. clear(): Xóa toàn bộ bộ nhớ đệm của Entity Manager.
10. getReference(entityClass, primaryKey): Lấy tham chiếu đến một đối tượng dựa trên khóa chính mà không thực sự tải đối tượng đó từ cơ sở dữ liệu.

Persistence Context là một khối dữ liệu trong JPA, dùng để quản lý tập trung các đối tượng được lưu trữ trong cơ sở dữ liệu. Nó cung cấp một cơ chế để quản lý việc tương tác của các đối tượng trong cùng một phiên làm việc (transaction) với cơ sở dữ liệu. Persistence Context bao gồm tất cả các đối tượng được truy cập và lưu trữ trong phiên làm việc hiện tại, cùng với các thay đổi được thực hiện trên các đối tượng đó.

Entity Manager là một interface trong JPA, cung cấp các phương thức để quản lý và tương tác với các đối tượng được lưu trữ trong cơ sở dữ liệu. Nó đại diện cho một phiên làm việc với cơ sở dữ liệu và là cầu nối giữa Persistence Context và cơ sở dữ liệu thực tế. Entity Manager cung cấp các phương thức để tạo, cập nhật, xóa và truy vấn các đối tượng, cũng như quản lý các phiên làm việc (transaction) với cơ sở dữ liệu.

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

- Câu lệnh truy vấn động là câu lệnh được tạo ra trong repository lúc lấy dl từ DB. Câu lệnh truy vấn động dễ sửa nhưng khi dùng sẽ chạy chậm hơn truy vấn tĩnh. Câu lệnh truy vấn động dùng phương thức createQuery() của lớp Entity Manger

- Câu lệnh truy vấn tĩnh là câu truy vấn trong ứng dụng, không thể thay đổi trong quá trình thực thi. Câu lệnh truy vấn tĩnh sẽ chạy nhanh hơn câu lệnh truy vấn động. Câu lệnh truy vấn tĩnh dùng phương thức createNameQuery() của lớp Entity Manger

Câu lệnh truy vấn động cho phép tạo câu lệnh truy vấn tại thời điểm chạy của ứng dụng, dựa trên các điều kiện đầu vào được cung cấp bởi người dùng hoặc các tác nhân khác. Các tham số truyền vào cho câu lệnh truy vấn động có thể thay đổi tùy thuộc vào tình huống, điều này giúp tạo ra các câu lệnh truy vấn động có tính linh hoạt cao. Các ngôn ngữ lập trình như Java cung cấp các thư viện để tạo câu lệnh truy vấn động, ví dụ như JDBC trong Java.

Câu lệnh truy vấn tĩnh là một câu lệnh truy vấn được xác định và cố định trước, và được sử dụng để truy vấn cơ sở dữ liệu một cách thường xuyên. Câu lệnh truy vấn tĩnh được định nghĩa và kiểm tra tại thời điểm biên dịch hoặc triển khai ứng dụng, và không thể thay đổi trong quá trình chạy. Các ngôn ngữ lập trình như SQL thường sử dụng câu lệnh truy vấn tĩnh để truy vấn cơ sở dữ liệu.

32 Trạng thái của Entity bao gồm những gì?

Có 4 trạng thái gồm:

- New: object mới được tạo ra, chưa kết nối với database (hay còn gọi là transient – tạm thời)

- Persistence: Object đã được ánh xạ tới database để tạo bảng, lúc này object được quản lý bởi persistence context, lúc này mọi thay đổi của entity sẽ được hibernate ghi lại và đồng bộ xuống database

- Detached (tạm thời tách ra khỏi persistence context): Một khi persistence context tạm thời đóng lại, mọi entity liên kết trước đó sẽ bị tách ra, lúc này các thay đổi của entity sẽ không được đồng bộ với database

-Remove: Entity bị xóa khỏi cơ sở dữ liệu

33 Có bao nhiêu loại mapping trong Hibernate?

Có 4 loại Mapping trong Hibernate : One-to-One, One-to-Many, Many-to-One và Many-to-Many để ánh xạ các mối quan hệ giữa các bảng trong cơ sở dữ liệu quan hệ.

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

Để sử dụng interface JpaRepository, bạn cần cung cấp hai thông tin: kiểu của entity và kiểu của khóa chính (primary key) của entity đó. Sau đó, Spring Data JPA sẽ tự động tạo ra một implementation cho interface này và bạn có thể sử dụng các phương thức được cung cấp bởi JpaRepository

VD: public interface BookRepository extends JpaRepository<Book, Long> {…}

35 Validation dữ liệu là gì?

- Là thực hiện đánh giá xác minh tính hợp lệ của dữ liệu đầu vào

36 Trình bày cách triển khai validate dữ liệu trong Spring

- Trong spring có 2 cách triển khai validate dữ liệu:

+ Cách 1: Dùng annotation, có 2 loại là annotation có sẵn và annotation tự định nghĩa. Cách này chỉ validate được dữ liệu đơn (là trường dữ liệu có annotation trên nó)

+ Cách 2: Dùng custom validator do mình tự xây dựng. Cách này có thể validate liên item hoặc những trường hợp đơn item mà mình muốn dùng validator

37 Binding Result là gì?

- Binding result là 1 interface của Spring MVC, extend Erros interface. Mục đich chính của nó là cho phép thực hiện validator, chứa kết quả các lỗi có thể có. Biding result phải được đặt sau model object cần validate

Binding Result là một đối tượng trong Spring Framework được sử dụng để lưu trữ kết quả của quá trình binding (ràng buộc) giữa dữ liệu được gửi lên từ form trên giao diện người dùng và đối tượng Model trong ứng dụng web.

Khi một form được gửi lên, Spring sẽ thực hiện quá trình binding để ánh xạ các trường dữ liệu trên form với các thuộc tính của đối tượng Model tương ứng. Trong quá trình này, nếu có lỗi xảy ra, ví dụ như dữ liệu không hợp lệ hoặc không thể ánh xạ được với thuộc tính của đối tượng Model, Binding Result sẽ lưu trữ các thông tin lỗi này. Binding Result cung cấp các phương thức để kiểm tra xem liệu có lỗi xảy ra trong quá trình binding hay không, và lấy ra thông tin chi tiết về các lỗi đó.

Các phương thức chính của BindingResult bao gồm:

1. hasErrors(): Kiểm tra xem có lỗi nào trong BindingResult hay không.
2. getAllErrors(): Lấy tất cả các lỗi trong BindingResult dưới dạng danh sách các đối tượng ObjectError.
3. getFieldError(): Lấy lỗi liên quan đến một trường cụ thể trong BindingResult dưới dạng đối tượng FieldError.
4. getGlobalError(): Lấy lỗi toàn cục trong BindingResult dưới dạng đối tượng ObjectError.
5. rejectValue(): Thêm một lỗi vào BindingResult với các thông tin cụ thể về trường và giá trị không hợp lệ.
6. reject(): Thêm một lỗi toàn cục vào BindingResult với các thông tin cụ thể.

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

- AOP (Aspect oriented programing) Lập trình hướng khía cạnh là 1 kỹ thuật lập trình nhằm phân tách chương trình thành các module riêng lẻ, phân biệt và không phụ thuộc vào nhau

- Khi hoạt động chương trình sẽ kết hợp các module để thực hiện các chức năng nhưng khi sửa đổi 1 chức năng thì chỉ cần sửa đổi 1 module

- AOP gồm các yếu tố:

+ core concern: logic xử lý chính của chương trình

+ cross cutting concern: là những xử lý phụ, cần được thực hiện của chương trình khi logic chính được xử lý

AOP (Aspect-Oriented Programming) là một phương pháp lập trình hướng đối tượng cho phép tách rời các khía cạnh (aspects) không liên quan đến logic chính của ứng dụng, như ghi log, xử lý ngoại lệ, kiểm tra quyền truy cập, và tái sử dụng chúng ở nhiều nơi khác nhau trong ứng dụng. AOP được sử dụng trong các framework như Spring để giúp tách rời các khía cạnh như logging, transaction management, security và caching.

AOP gồm những yếu tố sau:

* Aspect: một khía cạnh của hệ thống, chẳng hạn như logging, caching, hay security.
* Join point: một điểm trong mã của ứng dụng, chẳng hạn như một phương thức được gọi hoặc một biến được truy cập.
* Advice: một hành động được thực hiện tại một join point cụ thể, chẳng hạn như trước hoặc sau khi một phương thức được gọi.
* Pointcut: một tập hợp các join point, cho phép bạn chỉ định các join point nào sẽ được áp dụng cho một advice cụ thể.
* Weaving: quá trình nối các khía cạnh (aspects) vào mã của ứng dụng để tạo ra ứng dụng hoàn chỉnh.

39 Các loại Advice?

Có 3 loại advice:

+ Before advice: sẽ chạy trước khi xử lý 1 method

+ After advice: chạy sau khi xử lý 1 method, trong after sẽ chia thành 3 loại là after returning, after throwing, after finally

+ Around advice:chạy cả trước và sau khi xử lý 1 method, phải gọi phương thức, tùy mình chọn việc aspect sẽ chạy như thế nào

40 JoinPoint là gì?

- JointPoint là 1 điểm trong chương trình có thể chèn xử lý phụ. Joinpoint có thể là 1 phương thức, 1 ngoại lệ, hoặc 1 trường dữ liệu bị thay đổi

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

1. Tạo một class Java đại diện cho Aspect, và chú thích nó bằng @Aspect. Trong class này, có thể định nghĩa các phương thức được chú thích bằng các annotation như @Before, @After, @Around, để chỉ định các hành động cần thực hiện trước hoặc sau các phương thức của các đối tượng khác.

2. Đăng ký Aspect với Spring Framework bằng cách sử dụng @Component hoặc @Bean. Aspect được đăng ký sẽ được Spring Framework quản lý và áp dụng cho các phương thức của các đối tượng khác.

3. Sử dụng các annotation như @Autowired hoặc @AspectJProxyFactory để áp dụng Aspect cho các đối tượng khác trong ứng dụng.

42 Mục đích và cách xử lý ngoại lệ trong Spring Web MVC như thế nào?

Mục đích của việc xử lý ngoại lệ trong Spring Web MVC là để đảm bảo rằng khi có lỗi xảy ra trong ứng dụng, người dùng sẽ nhận được một thông báo lỗi thân thiện và rõ ràng thay vì một trang lỗi mặc định do máy chủ tạo ra.

Spring Web MVC cung cấp nhiều cách để xử lý ngoại lệ, bao gồm:

- Sử dụng annotation @ExceptionHandler để định nghĩa một phương thức xử lý ngoại lệ trong controller hoặc trong một lớp được đánh dấu với annotation @ControllerAdvice.

- Sử dụng các implementation của interface HandlerExceptionResolver để định nghĩa cách xử lý ngoại lệ cho toàn bộ ứng dụng.

Việc xử lý ngoại lệ một cách hợp lý sẽ giúp cho người dùng có trải nghiệm tốt hơn và giúp cho nhà phát triển dễ dàng xác định và khắc phục các vấn đề trong ứng dụng.

1. Sử dụng @ExceptionHandler: @ExceptionHandler là một annotation được sử dụng để xử lý ngoại lệ trong một Controller cụ thể. Bằng cách đánh dấu một phương thức trong Controller bằng @ExceptionHandler với kiểu ngoại lệ cụ thể, phương thức đó sẽ được gọi khi có ngoại lệ xảy ra trong phương thức xử lý yêu cầu.

2. Sử dụng @ControllerAdvice: @ControllerAdvice là một annotation được sử dụng để xử lý ngoại lệ trong toàn bộ ứng dụng. Bằng cách đánh dấu một class với @ControllerAdvice và định nghĩa các phương thức được chú thích bằng @ExceptionHandler với các kiểu ngoại lệ cụ thể, các phương thức này sẽ được gọi khi có ngoại lệ xảy ra trong bất kỳ Controller nào trong ứng dụng.

3. Sử dụng SimpleMappingExceptionResolver: SimpleMappingExceptionResolver là một class được sử dụng để ánh xạ các ngoại lệ cụ thể với các trang lỗi cụ thể. Bằng cách cấu hình SimpleMappingExceptionResolver trong file cấu hình Spring, các ngoại lệ xảy ra trong ứng dụng sẽ được ánh xạ với các trang lỗi tương ứng.

44 Cookie là gì?

Cookie là các tệp được trang web người dùng truy cập tạo ra. Nó giúp trải nghiệm người dùng tốt hơn bằng cách lưu thông tin duyệt web. Với cookie, các trang web có thể duy trì trạng thái đăng nhập, ghi nhớ tùy chọn trang web và cung cấp nội dung phù hợp với người dùng 🡪 Cookie thường được sd để luu thông tin của người dùng trên trình duyện web

Cookie là một đoạn dữ liệu được lưu trữ trên máy tính của người dùng thông qua trình duyệt web khi họ truy cập vào một trang web cụ thể. Cookie thường được sử dụng để lưu trữ các thông tin liên quan đến người dùng và phiên làm việc của họ, giúp cho việc duyệt web trở nên thuận tiện hơn.

Cookie được tạo ra bởi server và gửi đến trình duyệt của người dùng thông qua HTTP header của trang web. Mỗi cookie bao gồm một tên, một giá trị, và một số thuộc tính khác như thời gian sống (lifetime), đường dẫn (path), và tên miền (domain). Sau khi được lưu trữ trên máy tính của người dùng, cookie sẽ được gửi lại kèm theo mỗi yêu cầu đến server của trang web tương ứng.

45 Phân biệt Session và Cookie

- Giống nhau: Đều được dùng để lưu trữ thông tin người dùng

- Khác nhau

+ Session là các tệp được lưu trữ ở server chứa thông tin người dùng còn Cookie cũng là các tệp lưu trữ nhưng được lưu trữ ở client

+ Một session kết thúc khi đóng trình duyệt, cookie kết thúc tùy thuộc vào mình cài đặt thời gian sống cho nó

+ Một session phụ thuộc vào cookie tức là nó chỉ hiện cookie trùng với session id, còn cookie không phụ thuộc vào session, nó sẽ truyền toàn bộ cookie có lưu trữ tại máy liên quan đến trang web

+ Session bảo mật hơn vì lưu ở server, cookie kém bảo mật do lưu ở client nên dễ sửa đổi

1. Định nghĩa: Cookie là một đoạn dữ liệu được lưu trữ trên máy tính của người dùng thông qua trình duyệt web khi họ truy cập vào một trang web cụ thể, trong khi Session là một khái niệm trong lập trình web để lưu trữ thông tin phiên làm việc của người dùng trên máy chủ.

2. Điểm mạnh: Cookie thường được sử dụng để lưu trữ các thông tin liên quan đến người dùng và phiên làm việc của họ, giúp cho việc duyệt web trở nên thuận tiện hơn và tiết kiệm thời gian. Session thường được sử dụng để lưu trữ các thông tin phiên làm việc của người dùng trên máy chủ, giúp cho việc xử lý dữ liệu trên máy chủ trở nên dễ dàng hơn và giảm tải cho máy khách.

3. Thời gian sống: Cookie có thời gian sống có thể được thiết lập bởi server hoặc bởi người dùng, trong khi Session có thời gian sống được quy định bởi server và được duy trì trên máy chủ trong suốt phiên làm việc của người dùng.

4. An toàn: Cookie có thể bị đánh cắp hoặc sử dụng để giám sát hoạt động của người dùng trên trang web, trong khi Session được giữ an toàn trên máy chủ và không thể bị đánh cắp từ máy khách.

5. Sử dụng: Cookie thường được sử dụng để lưu trữ các thông tin như tên đăng nhập, ngôn ngữ sử dụng, cài đặt giao diện, lịch sử truy cập, và các thông tin khác liên quan đến người dùng và phiên làm việc của họ. Session thường được sử dụng để lưu trữ các thông tin phiên làm việc của người dùng trên máy chủ, giúp cho việc xử lý dữ liệu trên máy chủ trở nên dễ dàng hơn và giảm tải cho máy khách.

46 Session là gì? cho ví dụ về session?

- Là 1 phiên làm việc giũa client và server

- Session bắt đầu khi thực hiện request đầu tiên tới server và kết thúc khi cilent dừng làm việc vs server

- Session được lưu trữ ở trên server

- Ta có thể truy cập giá trị của biến session khi

+ User không đóng brower

+ User không log out

+ Session chưa bị hết hạn (Thường server sẽ có quy định thời gian hiệu lực của session)

Session là một khái niệm trong lập trình web để lưu trữ thông tin phiên làm việc của người dùng trên máy chủ. Khi một người dùng truy cập vào trang web, một phiên làm việc (session) sẽ được tạo ra và duy trì trên máy chủ để lưu trữ các thông tin liên quan đến phiên làm việc của người dùng đó

Session được tạo ra bằng cách sử dụng một số thông tin như session ID và thời gian sống của session. Session ID là một chuỗi ngẫu nhiên được tạo ra để xác định phiên làm việc của người dùng trên máy chủ, trong khi thời gian sống của session quy định thời lượng phiên làm việc được duy trì trên máy chủ trước khi nó bị xóa bỏ.

\* Ví dụ: Session thường được sử dụng để lưu trữ các thông tin như thông tin đăng nhập, thông tin giỏ hàng, các lựa chọn của người dùng, vv.

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

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

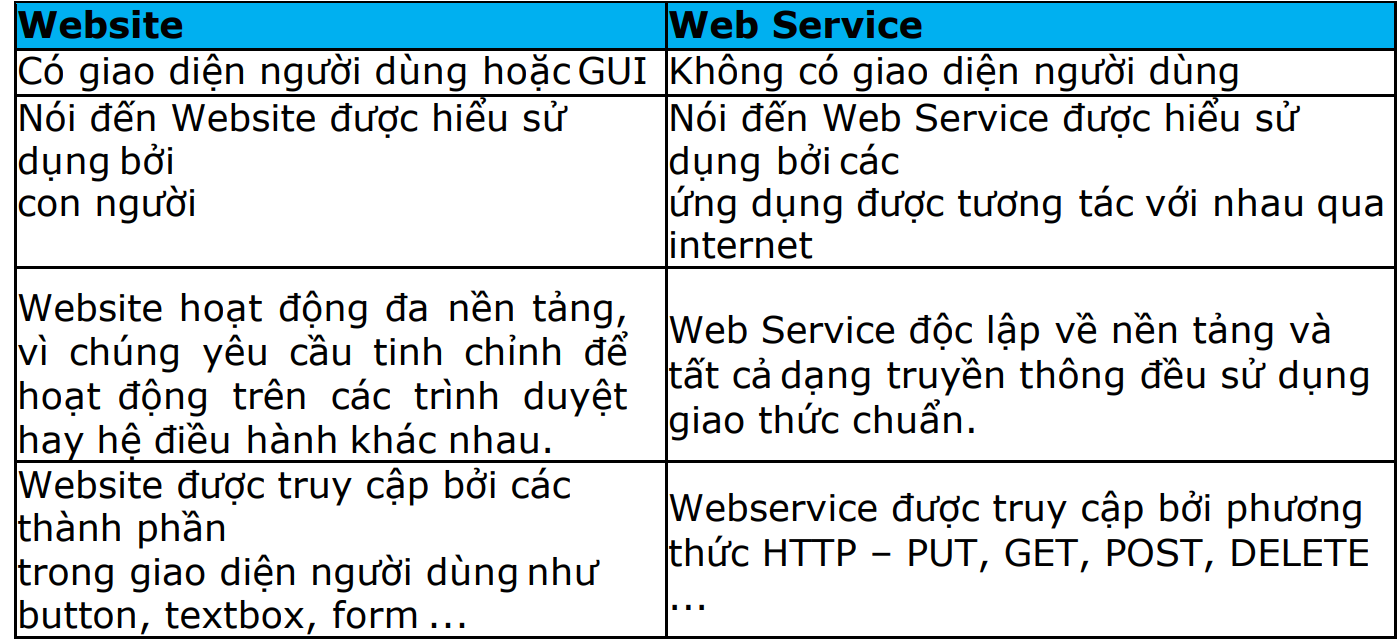
- Web service là các thành phần ứng dựng được hiển thị dưới dạng www. Web service có thể sử dụng để tích hợp với các ứng dụng được của các ngôn ngữ khác nhau và chạy trên các nền tang khác nhau. Web service hoạt động như 1 server trong mô hình ứng dụng client-server, dử dụng giao thức http/https và chỉ thực hiện 1 tác vụ cụ thể. Web service sẽ xử lý và trả ra dữ liệu theo chuẩn, đảm bảo mọi ứng dụng có thể hiểu và sử dụng mà không quan tâm đến loại thiệt bị, hệ điều hành, kiến trúc phần mềm hay ngôn ngữ được sử dụng

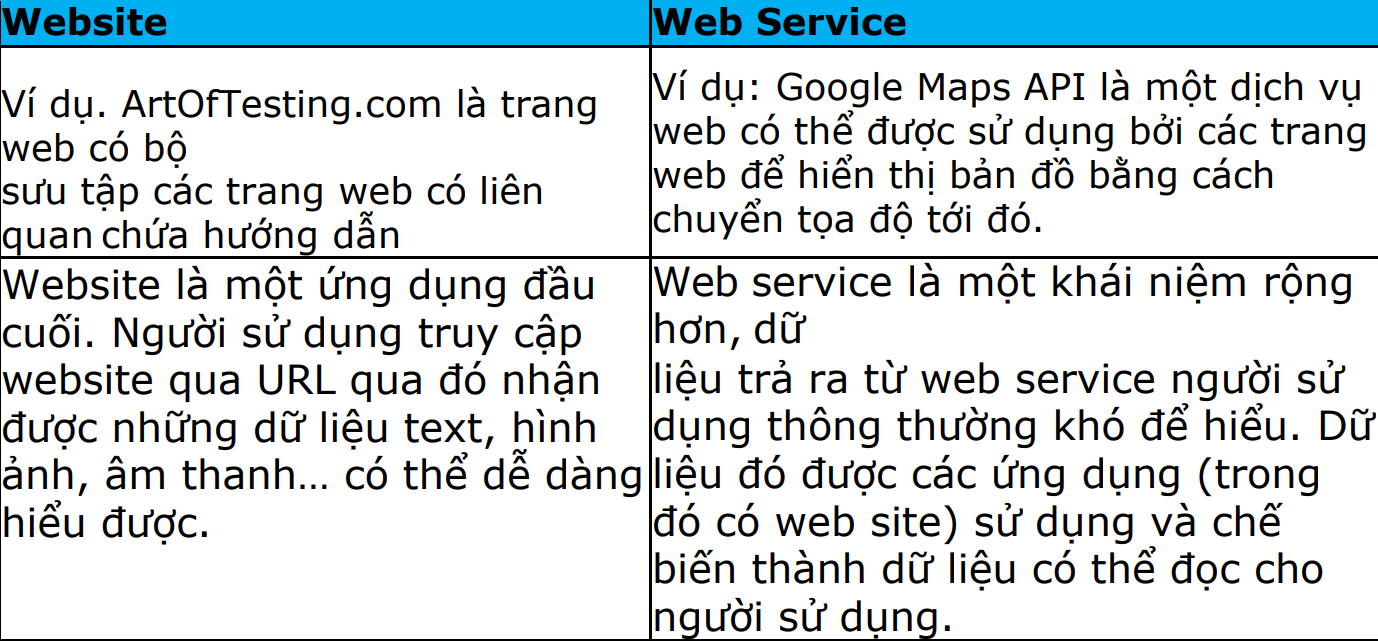
- Kiểu dl đầu ra phô biến của 1 web service thường là XML và JSON

- Ví dụ web service về giá vàng, giá cổ phiếu, giá xăng dầu, thời tiết,…

\* Ví dụ web service về dịch vụ thời tiết là dịch vụ này cung cấp thông tin thời tiết cho các thành phố khác nhau trên thế giới thông qua một API. Các ứng dụng khác có thể gửi yêu cầu đến API của dịch vụ thời tiết để lấy thông tin thời tiết cho một thành phố cụ thể. Dịch vụ thời tiết sẽ trả về thông tin thời tiết dưới dạng XML hoặc JSON và ứng dụng sẽ xử lý thông tin này để hiển thị cho người dùng.

49 Phân biệt Web Service và Web truyền thống





50 SOAP là gì? RESTful là gì?

- SOAP (simple object access protocol) là giao thức sử dụng XML để định nghĩa dữ liệu thuần văn bản và truyền dữ liệu thông qua HTTP

- REST (Representational state transfer) là 1 kiểu cấu trúc cung cấp các quy tắc để xây dựng web service. REST định nghĩa dữ liệu dưới dạng XML hoặc JSON và truyền thông qua HTTP. Các trang web xây dựng dựa trên REST còn được gọi là RESTful, chủ yếu xử lý các hành động CRUD

51 jQuery là gì? Ajax là gì?

- Ajax (asynchoronous javascript and xml) là 1 nhóm các công nghệ phát triển web được sử dụng để tạo ra các ứng dụng web giàu tính tương tác, nhanh hơn, mượt hơn với sự giúp đỡ của XML, HTML, CSS, JS,…

- Ajax giúp các trang web được cập nhật mà không cần tải lại trang

- Với Ajax, người dùng có thể tiếp tục sử dụng ứng dụng trong khi các chương trình trên client tạo request để lấy thông tin từ server

- Jquery là một thư việc javascript được thiết kế đơn giản hóa thao tác HTML DOM, xử lý sự kiện, CSS và Ajax. Nó có mã nguồn mở và miễn phí sử dụng

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

F

53 @RequestHeader và @ResponseHeader có ý nghĩa gì

- @Request header được sử dụng khi ta muốn lấy dữ liệu được truyền bằng header của 1 request

- Chúng ta sử dụng @ResponseHeader khi muốn trả về thêm dữ liệu cho client ở phần trên cùng của mỗi response

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

Bảo mật là quá trình đảm bảo tính bảo mật, toàn vẹn và sẵn sàng của dữ liệu và hệ thống thông tin. Bảo mật bao gồm việc ngăn chặn truy cập trái phép, sử dụng, tiết lộ, sửa đổi hoặc phá hủy thông tin.

\* Cơ chế bảo mật trong Spring

- Xác thực(Authentication)

- Phânquyền (Authorization)

- Bộ lọc Servlet (Servlet Filters)

56 Authentication là gì ? Authorization là gì ?

- Authentication – xác thực là 1 hành động nhằm thiết lập hoặc chứng thực 1 thông điệp hoặc đối tượng nào đó đáng tin cậy, xác thực cũng còn có ý nghĩa khác là công nhận nguồn gốc của đối tượng (nói 1 cách khác xác thực là tiến trình thiết lập 1 principal)

- Authorization – phân quyền: là tiến trình quyết định xem 1 principal có được phép thực hiện 1 hành động trong ứng dụng hay không. Principal có thể là 1 người, 1 thiết bị hoặc 1 hệ thống nào đó thực hiện 1 hành động trong ứng dụng. Trước khi phần quyền, principal phải được xác thực

57 Liệt kê các cơ chế xác thực

- Http basic: phương thức để client cung cấp user và password khi thực hiện yêu cầu

- Sử dụng cookies: lưu tại máy client nên dễ bị lộ

- Tokens: JWT (Json web token) là 1 tiêu chuẩn mở định nghĩa cách thức truyền tin an toàn giữa các thành viên 1 đối tượng json. Thông tin này có thể được xác thực và đánh dấu tin cậy dựa vào “chữ ký” của nó. Phần chữ ký của JWT được mã hóa

- Signatures: gồm 2 quá trình là tạo chữ ký số và chứng thực. Là mô hình sử dụng các kĩ thuật mật mã gắn với mỗi người sử dụng 1 cặp khóa công khai – bí mật và qua đó có thể ký các văn bản điện tử cũng như trao đổi các thông tin mật

- One time password: OTP mật khẩu chỉ sử dụng 1 lần

- OAuth2 : xác thực thông qua các ứng dụng

58 CSRF là gì

- CSRF: cross site request forgery là kỹ thuật tấn công bằng cách sử dụng quyền chứng thực của người dùng đối với 1 website nào đó

- CSRF là kỹ thuật tấn công vào người dùng và dựa vào đó hacker có thể thực thi những thao tác phải yêu cầu sự chứng thực

59 CORS là gì

- CORS: cross origin resources sharing: là cơ chế cho phép nhiều tài nguyên khác nhau của 1 trang web có thể được truy vấn từ domain khác với domain của trang web đó

60 Spring Boot là gì?

- Spring Boot là một Dự án nằm trên tầng IO Execution của Spring IO Framework.

- Spring Boot là bước tiếp theo của Spring để giúp Spring dễ dàng hơn trong việc thiết lập và phát triển các ứng dụng. Đối với Spring Boot, cấu hình Spring được giảm thiểu. Spring Boot hỗ trợ các containers nhúng trong việc cho phép các ứng dụng web có thể chạy độc lập mà không cần triển khai trên Máy chủ Web.

61 Sự khác nhau giữa SpringBoot và SpringMVC là gì?

- Spring Boot là một phần mở rộng của [Spring framework](https://hoidapcongnghe.info/spring-framework-la-gi/) cho phép các nhà phát triển rút ngắn quy trình cần thiết để xây dựng một ứng dụng web bằng cách sử dụng cấu hình tự động

- Spring MVC là một thư viện trong Spring framework. Trong khi Spring MVC là một framework (khuôn khổ) thiết yếu tồn tại trong Spring nói chung, thì Spring Boot là một mô-đun tùy chọn được sử dụng để hợp lý hóa quá trình phát triển, có thể bao gồm việc tích hợp Spring MVC trong quá trình xây dựng

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

- Thông qua các annotation @Controller, @Service, @Repository, @Component

63 Trình bày Spring Security? @EnableWebSecurity làm gì?

- Spring security là một dự án bảo mật trong hệ sinh thái spring, cung cấp dịch vụ bảo mật toàn diễn cho các ứng dụng doanh nghiệp có nền tảng Java EE

- Spring security là 1 framewwork tập trung vào việc cung cấp khả năng xác thực và phân quyền cho ứng dụng java

Annotation @EnableWebSecurity được sử dụng để kích hoạt cấu hình bảo mật web trong Spring Security. Bạn có thể thêm annotation này vào một lớp cấu hình @Configuration để định nghĩa cấu hình bảo mật cho ứng dụng của mình.

64. RequestMapping là một Annotation trong Spring Framework được sử dụng để ánh xạ các yêu cầu HTTP đến các phương thức xử lý tương ứng trong ứng dụng web. Với RequestMapping, bạn có thể chỉ định URL, phương thức HTTP, các tham số và các đối tượng phản hồi được trả về cho các yêu cầu HTTP.

Các thuộc tính của RequestMapping bao gồm:

1. value: Chỉ định URL được ánh xạ đến phương thức xử lý. Nếu không có giá trị được chỉ định, URL được ánh xạ sẽ là URL mặc định của phương thức.
2. method: Chỉ định phương thức HTTP được sử dụng cho yêu cầu HTTP. Mặc định là tất cả các phương thức HTTP.
3. params: Chỉ định các tham số yêu cầu HTTP cần phải có để phương thức xử lý được gọi.
4. headers: Chỉ định các tiêu đề yêu cầu HTTP cần phải có để phương thức xử lý được gọi.
5. consumes: Chỉ định các định dạng phương thức yêu cầu HTTP có thể chấp nhận.
6. produces: Chỉ định các định dạng phương thức phản hồi HTTP có thể trả về.

65. Spring Boot và Spring MVC là hai framework phát triển ứng dụng web phổ biến trong Java. Tuy nhiên, chúng có sự khác biệt như sau:

1. Mục đích sử dụng: Spring MVC là một framework phát triển ứng dụng web truyền thống, trong khi Spring Boot được tạo ra để giúp người phát triển tạo ứng dụng web nhanh chóng và dễ dàng hơn.
2. Cấu hình: Trong Spring MVC, các cấu hình phải được thực hiện thủ công bằng cách sử dụng tệp cấu hình XML hoặc Java. Trong khi đó, Spring Boot sử dụng cấu hình tự động và tự động cấu hình các thành phần khác nhau của ứng dụng.
3. Tích hợp: Spring Boot cung cấp tích hợp với các công cụ phổ biến như Maven và Gradle, giúp người phát triển quản lý các phụ thuộc và xây dựng ứng dụng một cách dễ dàng hơn. Trong khi đó, Spring MVC không cung cấp tích hợp sẵn với các công cụ này.
4. Đóng gói: Spring Boot cung cấp tính năng đóng gói ứng dụng thành các tệp JAR hoặc WAR độc lập, giúp việc triển khai ứng dụng trở nên đơn giản hơn. Trong khi đó, Spring MVC không cung cấp tính năng này mặc định.
5. Khả năng tự cấu hình: Spring Boot có khả năng tự cấu hình các thành phần của ứng dụng, giúp cho việc triển khai và quản lý ứng dụng trở nên dễ dàng hơn. Trong khi đó, Spring MVC không có tính năng này mặc định.

66. CORS (Cross-Origin Resource Sharing) là một cơ chế được sử dụng để cho phép các trang web từ các nguồn khác nhau tương tác với nhau một cách an toàn trong trình duyệt. CORS cho phép một trang web trong một tên miền hoặc một máy chủ gửi yêu cầu tới một tên miền hoặc máy chủ khác để truy cập các tài nguyên (như tệp, API, v.v.) được lưu trữ trên đó.

Trong trường hợp không có CORS, trình duyệt sẽ chặn các yêu cầu của trang web đến các tài nguyên khác miền để bảo vệ sự riêng tư và an toàn của người dùng. CORS giúp giải quyết vấn đề này bằng cách cho phép các trang web gửi yêu cầu khác miền một cách an toàn, đồng thời đảm bảo rằng các yêu cầu này không gây nguy hiểm cho người dùng.

CORS được triển khai bằng cách sử dụng các tiêu đề HTTP như Access-Control-Allow-Origin, Access-Control-Allow-Methods, Access-Control-Allow-Headers, v.v. để cho phép hoặc từ chối yêu cầu tới các tài nguyên khác miền. Khi một trang web gửi một yêu cầu tới một tài nguyên khác miền, trình duyệt sẽ gửi một yêu cầu tới máy chủ để xác định liệu yêu cầu đó có được phép hoặc từ chối hay không.

67. Data Binder là một tính năng quan trọng trong Spring Framework cho phép liên kết (binding) giữa dữ liệu người dùng và đối tượng Java. Cơ chế Data Binder giúp giải quyết vấn đề chuyển đổi dữ liệu giữa các đối tượng Java và các trường dữ liệu trên giao diện người dùng.

Khi dữ liệu được nhập vào từ giao diện người dùng, Data Binder sử dụng các thuộc tính và phương thức của đối tượng Java để ánh xạ các giá trị vào các trường dữ liệu tương ứng. Sau đó, nó sẽ kiểm tra tính hợp lệ của dữ liệu đầu vào và tạo ra các thông báo lỗi nếu cần thiết.

Cơ chế Data Binder được sử dụng trong Spring Framework được thực hiện bằng cách sử dụng các lớp và giao diện sau:

1. PropertyEditor: Là một giao diện được sử dụng để chuyển đổi giữa các kiểu dữ liệu khác nhau. Nó cho phép các giá trị người dùng được chuyển đổi sang các kiểu dữ liệu phù hợp với các trường dữ liệu trên đối tượng Java.
2. DataBinder: Là một lớp được sử dụng để liên kết (binding) các giá trị người dùng với các thuộc tính của đối tượng Java. Nó sử dụng các PropertyEditor để chuyển đổi các giá trị người dùng sang các kiểu dữ liệu phù hợp trước khi lưu trữ chúng vào đối tượng Java.
3. WebDataBinder: Là một lớp con của DataBinder được sử dụng trong Spring MVC để liên kết các giá trị người dùng với các thuộc tính của một đối tượng Java và kiểm tra tính hợp lệ của chúng.

68. Trong ứng dụng web, việc xác thực và kiểm tra tính hợp lệ của dữ liệu là rất quan trọng để đảm bảo tính toàn vẹn và an toàn của hệ thống. Trong tầng backend (BE) của ứng dụng, bạn có thể sử dụng các cách sau để xác thực và kiểm tra tính hợp lệ của dữ liệu:

1. Sử dụng các annotation của Spring Framework: Spring Framework cung cấp nhiều annotation để xác thực và kiểm tra tính hợp lệ của dữ liệu. Ví dụ, bạn có thể sử dụng @NotNull để đảm bảo rằng một trường không được để trống hoặc @Size để đảm bảo rằng một trường có độ dài nằm trong một phạm vi nhất định.
2. Sử dụng Hibernate Validator: Hibernate Validator là một thư viện xác thực dữ liệu phổ biến trong Java và được sử dụng rộng rãi trong Spring Framework. Thư viện này cung cấp nhiều annotation để kiểm tra tính hợp lệ của dữ liệu, bao gồm @NotNull, @Size, @Email, @Pattern và nhiều hơn nữa.
3. Sử dụng Custom Validator: Bạn có thể tạo ra các validator tùy chỉnh để kiểm tra tính hợp lệ của dữ liệu theo cách của riêng mình. Để làm điều này, bạn cần triển khai interface Validator của Spring Framework và ghi đè phương thức validate để kiểm tra tính hợp lệ của đối tượng.
4. Sử dụng Exception Handler: Nếu dữ liệu không hợp lệ, bạn có thể ném ra một ngoại lệ và bắt nó bằng Exception Handler để xử lý lỗi và trả về thông báo lỗi cho người dùng.
5. Sử dụng API kiểm tra tính hợp lệ của dữ liệu: Ngoài các cách trên, bạn có thể sử dụng các API kiểm tra tính hợp lệ của dữ liệu như Apache Commons Validator hoặc Google Guava Preconditions để kiểm tra tính hợp lệ của dữ liệu.

69. Trong lập trình hướng khía cạnh (Aspect-Oriented Programming - AOP), Join Point và Point Cut là hai khái niệm quan trọng để thực hiện việc tách rời các quan tâm (concerns) khác nhau trong ứng dụng.

* **Join Point**: là một điểm trong mã nguồn của ứng dụng, nơi mà chúng ta có thể chèn các khía cạnh (aspects) để thực hiện các chức năng bổ sung. Các điểm đó được xác định bởi các phương thức (method) hoặc các sự kiện (event) như gọi phương thức, ngoại lệ, xử lý ngoại lệ, thực hiện câu lệnh gọi, hay các hành động khác trong quá trình thực thi chương trình.
* **Point Cut**: là một tập hợp các Join Point, được định nghĩa bởi một biểu thức điều kiện. Point Cut định nghĩa các điểm nào trong mã nguồn sẽ được áp dụng khía cạnh (aspect), nó giúp cho việc quản lý các khía cạnh trong ứng dụng trở nên dễ dàng hơn.

Ví dụ: Giả sử ta có một ứng dụng Spring Boot và muốn thực hiện việc theo dõi thời gian thực thi của tất cả các hàm trong một package nào đó. Để thực hiện điều này, chúng ta sử dụng AOP để chèn một khía cạnh (aspect) vào các Join Point là các phương thức trong package đó.

Trong ví dụ trên, chúng ta sử dụng các annotation của AOP để định nghĩa một khía cạnh (aspect) và một Point Cut. Point Cut được định nghĩa bằng cách sử dụng biểu thức "execution(\* com.example.demo.service.\*.\*(..))" để xác định các phương thức trong package com.example.demo.service. Khi các phương thức đó được gọi, khía cạnh sẽ được chèn vào Join Point tương ứng để thực hiện việc theo dõi thời gian thực thi của chúng.