1. Phân biệt ORM, JPA, Hiberneta, Spring Data JPA?

* ORM: là kỹ thuật liên kết giữa các đối tượng lập trình trong CSDL, cho phép truy xuất dễ dàng đến dữ liệu thông qua các đối tượng lập trình
* JPA: là interface mô tả cách quản lý các mối qhe trong dl trong ứng dụng và cung cấp 1 mô hình cho phép ánh xạ các bảng trong DB và class trong java
* Hibernate là 1 ORM framework (JPA là tập hợp các interface được định nghĩa theo quy chuẩn thì hibernate sẽ là người triển khai các interface này )
* Spring data JPA giúp cho việc sd JPA trong Spring trở nên dễ dàng hơn. Nó giúp cải tiến JPA tiêu chuẩn đơn giản hóa tầng truy xuất dl, tự tạo ra repository, tạo các câu truy vấn JPA thông qua tên phương thức

1. Phân biệt session cookie ?

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

|  |  |
| --- | --- |
| Cookie | Session |
| **Cookie** là các thông tin tạm thời được lưu tại máy tính client, và được tạo ra khi client truy cập vào trang web | **Session** là một phiên làm việc của client với website có kết nối với database, các thông tin tạm thời sẽ được lưu tại máy chủ |
| **Cookie** được lưu trữ trên trình duyệt của người dùng. | **Session** không được lưu trữ trên trình duyệt. |
| Dữ liệu **cookie** được lưu trữ ở phía client. | Dữ liệu **session** được lưu trữ ở phía server. |
| Dữ liệu **cookie** dễ dàng sửa đổi hoặc đánh cắp khi chúng được lưu trữ ở phía client. | Dữ liệu **session** không dễ dàng sửa đổi vì chúng được lưu trữ ở phía máy chủ. |
| Dữ liệu **cookie** có sẵn trong trình duyệt đến khi hết hạn. | Sau khi đóng trình duyệt sẽ hết phiên làm việc (session) |

1. Phân biệt web service và web truyền thống

|  |  |
| --- | --- |
| **Website** | **Web service** |
| - Có giao diện | - Không có giao diện |
| - Được sử dụng bởi con người | - Được sử dụng bởi các ứng dụng khác |
| - Website hoạt động đa nền tảng vì chúng cần tinh chỉnh để phù hợp các trình duyệt hay hệ điều hanh khác nhau | - Web service độc lập về nền tảng, sử dụng giao thức chuẩn để giao tiếp |
| - Website được truy cập trong các thành phần của giao diện như button, form | - Truy cập thông qua các phương thức HTTP: get, post, put, delete, patch |
| - Website là ứng dụng đầu cuối, người dùng truy cập website qua URL, qua đó nhận được những dữ liệu như text, hình ảnh, âm thanh có thể dễ dàng hiểu được | - Là 1 khái niệm rộng hơn, dữ liệu trả về từ web service, người dùng khó có thể hiểu được. Dữ liệu đó được các ứng dụng khác sử dụng và chế biến thành các dữ liệu mà người dùng có thể đọc được |

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

|  |  |
| --- | --- |
| **Spring Boot** | **Spring MVC** |
| Spring Boot là một mô-đun của Spring để đóng gói ứng dụng dựa trên Spring với các giá trị mặc định hợp lý. | Spring MVC là model-view-controller web framework trong khuôn khổ Spring framework. |
| Cung cấp các cấu hình mặc định để xây dựng khung công tác được cung cấp bởi Spring. | Cung cấp các tính năng sẵn sàng sử dụng để xây dựng một ứng dụng web. |
| Không cần phải xây dựng cấu hình theo cách thủ công. | Nó yêu cầu cấu hình xây dựng theo cách thủ công. |
| **Không yêu cầu đối với bộ mô tả triển khai.** | **Cần có bộ mô tả Triển khai.** |
| Nó tránh mã viết sẵn và kết hợp các phần phụ thuộc lại với nhau trong một đơn vị duy nhất. | Nó chỉ định từng phụ thuộc riêng biệt. |
| Nó làm giảm thời gian phát triển và tăng năng suất. | Cần nhiều thời gian hơn để đạt được điều tương tự. |

1. Trình bày sự khác nhau của mô hình MVC và Spring MVC ?
2. @Component có ý nghĩa gì? Khác gì với @Service và @Repository

@Component là một annotation được sử dụng để đánh dấu một class được tạo ra với mục đích là để định nghĩa các bean được quản lý bởi Spring IoC container. Spring chỉ tìm kiếm và chọn ra những class được đánh dấu với @Component annotation để khởi tạo và đăng ký với ApplicationContext.

@Service và @Repository cũng là các annotation được sử dụng để đánh dấu các bean là thành phần được quản lý của Spring, nhưng chúng có mục đích sử dụng khác nhau@Service được dùng để đánh dấu các class thuộc tầng service, còn @Repository được dùng để đánh dấu các class thuộc tầng giao tiếp với database1.

Sự khác biệt chính giữa 3 annotation này là dùng để phân loại giữa các layer với nhau.

1. Phân biệt post,put và patch?

POST: dùng để tạo dữ liệu còn Put dùng cập nhật dữ liệu.

PUT là phương thức bảo toàn dữ liệu nên khi chúng ta ấn submit bao nhiêu lần đi chăng nữa thì dữ liệu cũng chỉ tính như chúng ta submit 1 lần. Còn đối với POST thì chúng ta submit quá nhiều lần thì có thể nó sẽ cho ra những kết quả khác nhau.

PUT: để update, ta phải gửi 1 bản ghi đầy đủ các field để yêu cầu cập nhật. Nếu chỉ gửi 1 số field nhất định thì những field còn lại sẽ bị xóa (tức bị null)

PATCH: giống PUT về ý nghĩa nhưng khác nhau về cách làm việc, nó chỉ thay đổi những field được yêu cầu thay vì thay đổi toàn bộ record.

1. So sánh JPA với Spring Data JPA

- JPA: java persistence API cung cấp 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: 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,…và nó 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.

1. So sánh RequestParam và PathVariable ?

RequestParam và RequestVariable là hai thuộc tính được sử dụng trong các framework web để truyền giá trị từ client (browser) đến server

- RequestParam : gửi dữ liệu theo key- value,

- ParthVariable : gửi dữ liệu trên url và chỉ có value.

1. Giải thích join point và point cut

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

Point cutlà tập hợp các điểm trong chương trình để chèn xử lý phụ. Point cut được xác định bằng cách sd các biểu thức chính quy trên tên phương thức để xác định vị trí trong source code

1. Phân biệt giữa IoC & DI?

- 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

- 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. Các framework hỗ trợ IOC: Spring cho java hoặc Angular cho javascript

1. Authentication là gì ? Authorization là gì ?

|  |  |
| --- | --- |
| Authentication | Authorization |
| - Authentication là hoạt động nhằm thiết lập hoặc chứng thực một thông điệp hoặc một đối tượng nào đó đáng tin cậy | - Authorization / Access Control là tiến trình quyết định xem 1 principal có được phép thực hiện một hành động nào đó trong ứng dụng hay không |

1. Framework khác Library chỗ nào ?

|  |  |
| --- | --- |
| Framework | Library |
| - Framework bao gồm nhiều library, được hiểu là một khung chương trình, người dùng bổ sung code và tuân theo quy tắc của framwork để tạo ra ứng dụng. | - Library cung cấp các chức năng tiện ích hay các class để sử dụng trong quá trình xây dựng ứng dụng. |

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.

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

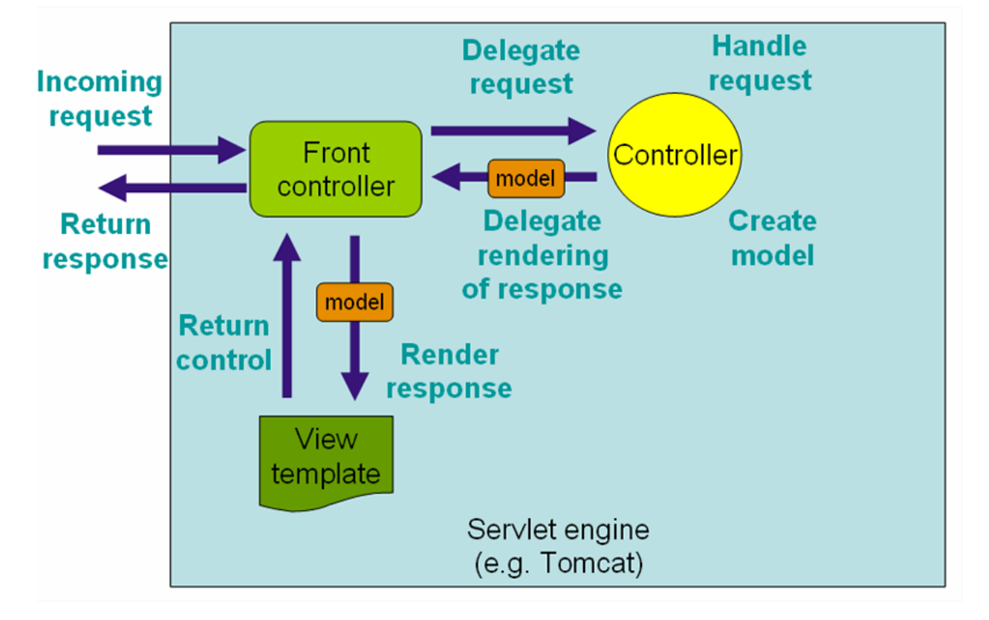
- Câu lệnh truy vấn động là câu lệnh được tạo ra trong repository lúc lấy dl từ DB. Câu lệnh truy vấn động dễ sửa nhưng khi dùng sẽ chạy chậm hơn truy vấn tĩnh. Câu lệnh truy vấn động dùng phương thức createQuery() của lớp Entity Manger

- 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

1. Trình bày sự khác nhau của Model, ModelMap và ModelAndView

* Model là một interface trong khi đó ModelMap là một Class.
* Model là một interface nó chứa đựng 4 phương thức addAttribute và một phương thức mergeAttribute .
* ModelMap cài đặt lớp Map interface. Nên nó thêm các phương thức của Map.
* 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.

1. 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 và tên vật lý 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

1. DI là gì? Các cách triển khai DI?

- 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

\* Các cách triển khai 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: Các dependency (biến phụ thuộc) sẽ được truyền vào 1 class thông qua interface.
* Spring chỉ có constructor injection và setter injection, k có interface injection

1. Framework là gì? Ưu nhược điểm khi sd FW?

- 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.

*Ưu điểm*:

* Tiết kiệm thời gian và công sức khi xây dựng ứng dụng từ đầu bởi vì cấu trúc mã cơ bản hầu hết đã được phát triển và tích hợp sẵn.
* Dễ dàng chỉnh sửa, cải tiến và phát triển từ các tính năng sẵn có để nhanh chóng đưa vào sử dụng khi xây dựng ứng dụng hoặc website.
* Chất lượng được đảm bảo, khả năng bảo mật cao.
* Code an toàn hơn
* Hạn chế code trùng lặp và code thừa
* Giúp code hoạt động nhất quán với ít lỗi hơn
* Giúp làm việc trên các công nghệ phức tạp trở nên dễ dàng hơn
* Chức năng của Framework liên tục được cải tiến
* Một số phân đoạn code và chức năng được xây dựng và thử nghiệm trước. Điều này làm cho các ứng dụng đáng tin cậy hơn
* Kiểm tra và gỡ lỗi code dễ dàng hơn

*Nhược điểm*

* Ngoài những lợi ích thiết thực kể trên, nhược điểm của Framework là gì?
* Bất kỳ lỗi hoặc vấn đề bảo mật nào liên quan đến framework đều có thể ảnh hưởng đến mọi ứng dụng được xây dựng trên framework đó.
* Mỗi Framework đều có giới hạn quy tắc nhất định, do đó không thể sửa đổi cấu trúc cốt lõi của một Framework.
* Tính sẵn có của Framework có thể bị một số cá nhân lợi dụng, sử dụng sai hướng để tìm ra các lỗ hổng và điểm yếu trong Framework.
* Để làm chủ Framework, lập trình viên cần bỏ ra khá nhiều thời gian và công sức.
* Framework thường có kích thước lớn vì vậy nó không thích hợp với việc phát triển các ứng dụng nhỏ.

1. IOC là gì? các cách triển khai IOC?

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. Các framework hỗ trợ IOC: Spring cho java hoặc Angular cho javascript

Các cách triển khai IOC: thông qua 1 số cơ chế: Strategydesign pattern, ServiceLocator pattern,Factory pattern and Dependency Injection(DI)

Lợi ích:

• Tách rời việc thực thi(execution) và việc triển khai (implementation)

• Dễ chuyển đổi giữa các implementation

• Dễ phân tách module hơn

• Dễ kiểm thử hơn, bằng cách tách rời các thành phần riêng lẻ

1. Trình bày ý nghĩa của ViewResolver Interface.

- Controller do người dùng định nghĩa trả về tên của view và các thông tin dữ liệu tới FrontController. Tên của view được trả về có kiểu là String , nên cần xử lý tiếp bởi ViewResolver.

1. @RequestMapping là gì? Các biến thể

- Ánh xạ các HTTP Request tới các phương thức xử lý của MVC và Rest controller.

@GetMapping

@PostMapping

@PutMapping

@DeleteMapping

@PatchMapping

1. Trình bay 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

1. Thymeleaf là gì? Các biểu thức của thymeleaf?

- Thymeleaf là 1 bộ xử lý view được sd 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

\*Các biểu thức

* ${...}: biểu thức thay giá trị của biến
* \*{...}: biểu thức thay thuộc tính của biến
* #{...}: message
* @{...}: biểu thức liên kết URL
* ~{...}: fragement , biểu thức mảnh

1. 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

1. Spring data JPA là gì? Ưu điểm ?

-KN: 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,…

- UD: 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.

1. 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.

1. Trình bày cách triển khai validate dữ liệu trong Spring ?

-b1: sử dụng annotation của spring để quy định rang buộc cho các field.

-b2: sử dụng @Valid hoặc @Validate và @BindingResult trong handle method để kiểm tra tính hợp lệ của dữ liệu.

-b3: sử dụng #Field.hasError() và th:error để hiển thị thông báo lỗi trên trang giao diện.

Các tầng validate dl: BE, FE, Data

1. 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

1. jQuery là gì? Ajax là gì?

- 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

- 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

1. 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

1. 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

1. 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