I. Spring Core

@SpringBootApplication : bao gồm các annotation như Cấu hình tự động , tự động ComponentScan, tự động cấu hình và tự đăng kí bean. 

- Khi lệnh run được thực thi sẽ tạo ra 1 container gọi là ApplicationContext để chứa 1 không gian lưu trữ và sẽ tìm kiếm các bean để lưu trữ (còn gọi là các Object java ví dụ là các dev tạo ra các new để quản lí ).

-Component Scan: để quét các bean cần quét. Nếu không đánh dấu @ComponentScan thì nó sẽ quét hết tất cả.

Các bean đó được đặt chú thích trên mỗi class như : @Component, @Service, @Repository, @Controller.

* Autowired: Sử dụng để injection các dependency vào các thành phần mình đã đánh dấu bằng @Autowired

20/12/2023

1. Ôn lại Contructor Injection.
2. Setter Injection
3. Field Injection(không khuyến khích dùng): sử dụng nó thì không cần khởi tạo Contructor Injection hay Setter Injection để tiêm

* Sử dụng nó sẽ khiến Test Junit trở nên khó khăn.

1. Autowiring(Contructor, Setter, Field) and Qualifiers:

21/12/2023

1. Học lại Qualifiers: được sử dụng khi có nhiều beans cùng kiểu và để biết được dependency injection sẽ tiêm vào hàm nào

24/12/2023

1. Coding: Qualifiers

* Nếu trong Component có:
* @Component("BaseballCoach")

Thì bên Dependency Injection kia cũng phải Đặt Qualifiers tên trùng với điểm cần tiêm vào. Còn khi @Component không đặt tên quy ước gì thì tên sẽ được viết thường chữ đầu khi tiêm vào

1. Primany annotion:

* Không nên sử dụng chung giữa Primary và Qualifier
* @Qualifier có mức độ ưu tiên cao hơn so với Primary
* Và chỉ một beans được đặt là @Primary. Nếu có 2 primary component thì sẽ xảy ra lỗi.
* Học primany cho biết nó tồn tại thôi chứ không nên sử dụng vì nó không tường minh và rõ ràng.

1. Lazy Initiazation:

* Tức là khi mình đặt @Lazy thì bean đó sẽ được khởi tạo khi được dung đến.
* Nếu không có @Lazy thì các beans sẽ được khởi tạo khi chạy application
* Khi muốn tạo lazy cho tất cả các beans thì thêm dòng này vào property 
* Nó cũng có một số nhược điểm:

26/12/2023

9.Bean Scopes:

* Là phiên bản được tạo và bean được chia sẻ như thế nào.
* Scope mặc định trong spring là: singleton
* Singleton: được chia sẻ và cùng trỏ tới 1 bean
* Prototype scope: là sẽ tạo ra 1 bean mới khác với singleton không thích dung chung bean

27/12/2023

10.Bean Lifecycle(Vòng đời của bean):

* Xác định init(được gọi sau khi bean được khởi tạo) và detroy(sẽ được gọi trước khi bean được huỷ)
* Thêm annotation @PostConstruct và @PrteDestroy
* Spring không quản lí vòng đời có phạm vi là prototype để đảm bảo tính tự do.
* Bean Lifecycle: Mục đích quản lí bean được chi tiết hơn.

29/12/2023

11. Spring IOC Container(DI):

* Spring famework hỗ trợ 2 loại BeanFactory và ApplicationContext giúp khởi tạo và quản lí các bean

12. Java Config Bean:

* Nó khác với sử dụng @Component

II.Hibernate-JPA/Crud

1. JPA Annotations- Overview

30/12/2023

1. @Overide

* Dùng trong interface
* Dùng để ghi đè phương thức cha
* Phương thức cha thường là 1 cái gì đó chung chung
* Ví dụ Cha là: Tiếng kêu. Con sẽ là cụ thể của tiếng kêu đó ví dụ : MeoMeo

1. Lambda:

* FunctionalInterface: chỉ được có 1 phương thức trừu tượng.
* Nếu interface chỉ có 1 phương thức trừu tượng thì đó được gọi là FunctionalInterface.
* Lambda là cách làm đơn giản hoá của cách FunctionalInterface
* Với biểu thức có 1 dòng thì mình không cần dấu ngoặc nhọn

1. Biến toàn cục:

* Nằm ngoài mọi hàm được sử dụng cho tất cả mọi hàm.

III. REST API

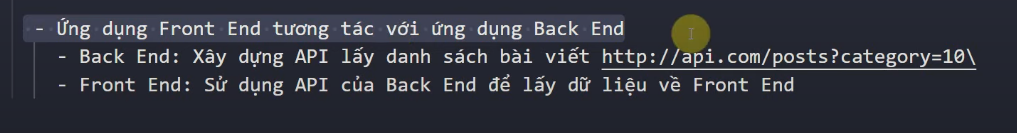
3/1/2024

1. JSON liên kết Data Java POJO: Chỉ là chuyển đổi từ định dạng này sang định dạng khác.

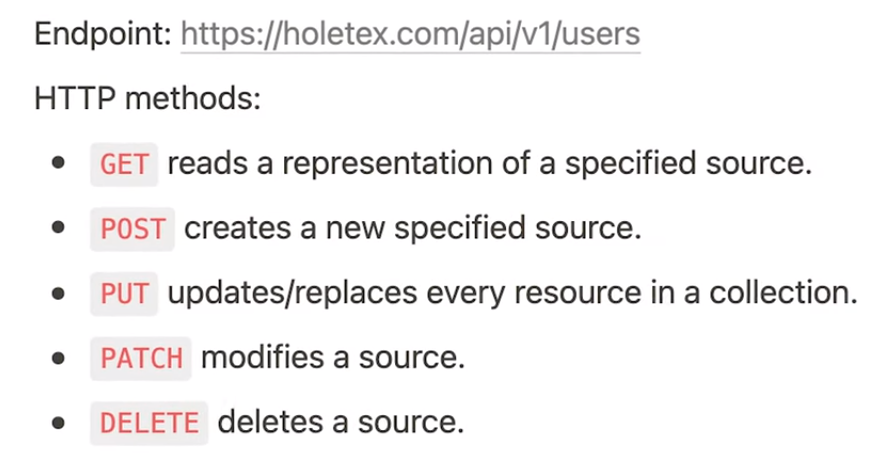
* POJO chỉ là Object java.
* Có thể chuyển từ JSON sang POJO và ngược lại
* Jackson: là thư viện java dung để xử lí JSON
* Spring sẽ tự xử lý tích hợp Jackson.
* Bất cứ dữ liệu JSON nào được chuyển đến REST Controller sẽ được chuyển thành POJO.
* Bất kì dữ đối tượng Java nào được trả về REST Controller đều được đổi thành JSON.

1. REST API

* API : Giúp cho 2 phần mềm có thể tương tác và trao đổi dữ liệu với nhau mà không quan tâm tới code được xây dựng như thế nào.
* Nhưng cần có giao thức chung.
* Và để 2 phần mềm có thể hiểu được nhau thì phải sử dụng chung 1 kiểu giao thức



* REST là ngôn ngữ độc lập.
* Nó không quan tâm bạn sử dụng ngôn ngữ nào.
* Ứng dụng máy chủ và ứng dụng khách có thể sử dụng bất cứ ngôn ngữ lập trình nào



1. Path Variables(Biến đường dẫn)
2. Spring boot REST Global Exception Handling:

* Có thể tái sử dụng code.
* Dùng cho trên nhiều controller được.
* Phải gọi @ControllerAdvice.
* Đây là cách hay nhất dành cho các dự án lớn.

9/1/2024

1. REST API Desgin:

10/1/2024

1. Contructor (Người xây dựng của 1 class nào đó):

* Contructor không tham số: được tạo mặc định nhưng điều kiện là chưa 1 contructor nào được khai báo thì mới có hiệu lực.
* Không có dữ liệu trả về.

1. Extends(Kế thừa): Mở rộng kế thừa mọi thứ từ lớp cha. Và lớp con có thể mở rộng thêm và có thể ghi đè @Override nhưng phải trùng tên phương thức của lớp cha.
2. Tính trừu tượng (Interface, và sử dụng implement) để kết nối.
3. @RestController(Repersentational State Tranfer): Bản than nó được annotation : @Controller và @RepoonseBody.

* Sự khác nhau giữa @Controller và @Restcontroller: Controller trả về giá trị chuỗi được xem ở dạng xem là để hiển thị chứ không phải nguyên trạng. Restcontroller trả về giá trị được sử dụng phản hồi.

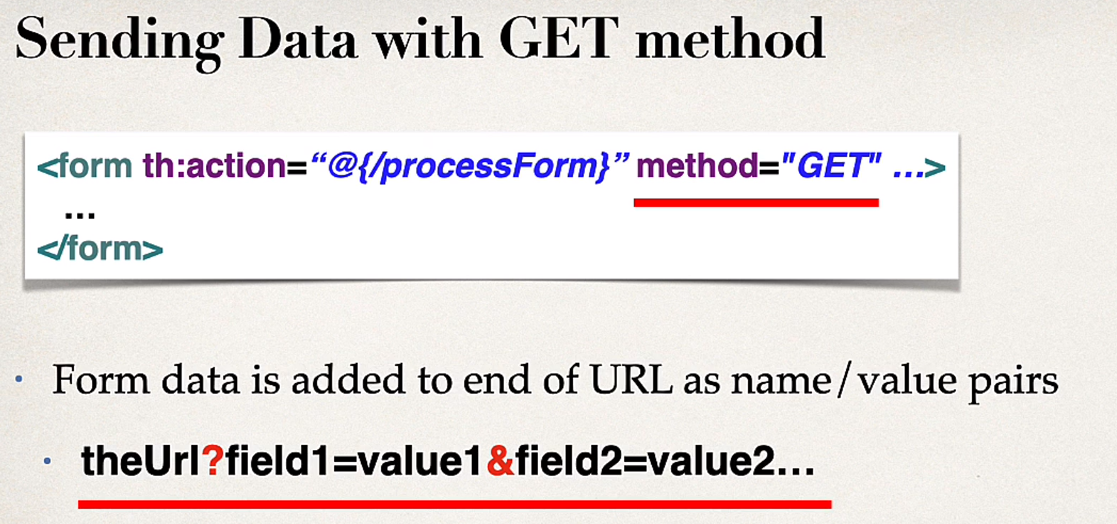
1. @Transactional: được đánh dấu ở tầng service để có thể rollback ().

* Những trường hợp sử dụng: Nên sử dụng ở tầng method.
* Nhưng khi mình sử dụng JpaRepository của Spring hỗ trợ thì mình không cần phải @ Transactionalb vì nó đã hỗ trợ.

Phần 6: NEW - Spring MVC

1.Get:

- Hiển thị được dữ liệu trên thành URL và nhờ đó có thể truyền dữ liệu qua lại được với nhau.

- Nếu sử dụng request GET bạn chỉ cần sử dụng @GetMapping 

2. POST:

- Không thể gửi qua URL.

- Có thể gửi dữ liệu binary data.

3. @RequestMapping:

- Hỗ trợ mọi yêu cầu GET, POST….

- StringTrimmerEditor: Cắt bớt loại bỏ khoảng trắng

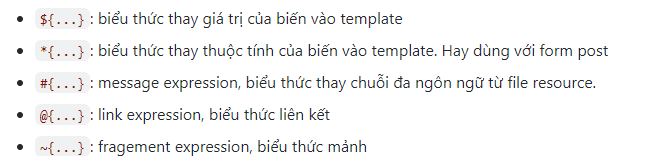


(true có nghĩa là được cắt thành null)

* @Path Variable: Lấy giá trị ở đường dẫn và lấy giá trij tuyệt đối.

Phần 7: NEW - Spring MVC CRUD

* @ModelAttribute: liên kết toàn bộ đối tượng Java hỗ trợ được nhiều tham số.
* @RequestParam: Liên kết một tham số yêu cầu duy nhất phù hợp với số lượng nhỏ thông số.
* Th:field là thuộc tính tạo ra cặp name và value.
* Th:object để hiển thị dữ liệu hoặc xử lí dữ liệu ( nó tự quản lí dữ liệu được input khi form gửi đi) và khi nhận thì @ModelAttribute sẽ là thằng nhận nhưng phải trùng tên với th:object.
* Th:text: Để hiển thị dữ liệu.
* Th:value: dung để ánh xạ dữ liệu vào 1 trường nhập liệu.



Phần 9: NEW-JPA/HIBERNATE ADVANCED