Một class không nên phụ thuộc và class kế thừa cấp thấp mà chỉ nên phụ thuộc và lớp Abstraction hoặc Interface.

Khắc phục:

Graphical user interface, text, application, email

Description automatically generated

**Inversion Control:**

**Lombok:**

Ko cần getter và setter:

Graphical user interface, text, application

Description automatically generated

* **controller**: các bean tiếp nhận request và trả về response cho client.
* **repository**: các bean thao tác trực tiếp với cơ sở dữ liệu. Là nơi lấy data về
* **service**: các bean xử lý logic, trung gian giữa repository và controller
* **resources**: các tài nguyên tĩnh như hình ảnh, css, javascript.
* Khi thêm **@Component** trên clas, ta đã đánh dấu class này là một Bean.
* Khi chạy chương trình, Spring sẽ dò tìm toàn bộ các class cùng package hoặc ở trong các package cùng cấp với class Application. Các class được đánh dấu sẽ được khởi tạo bằng contructor không tham số và tạo ra một instance rồi đưa vào **ApplicationContext** quản lý. ApplicationContext ở đây chính là **IoC Container**.

Diagram

Description automatically generated

Xử lý logic ở service, trong controller gọi service.

* **@Controller** là một annotation đánh dấu một class là **@Component** với ý nghĩa đây là Bean đón nhận các **Request** và trả về **Response** cho client. Bean này thường sẽ sử dụng các **Service** là dependency hoặc có thể sử dụng trực tiếp các **Bean Repository**
* **@RequestMapping** được đặt trên **phương thức** sẽ xử lý request hoặc trên **class** Controller để xác định đường dẫn request sẽ được xử lý. Tạo Controller:

Graphical user interface, text, application, email

Description automatically generated

* Khi muốn trả về nội dung một cách trực tiếp ta sẽ thêm annotation @ResponseBody trên phương thức hoặc trên class Controller nếu muốn áp dụng cho toàn bộ các phương thức nội bộ
* Dữ liệu trả về với @ResponseBody sẽ có dạng **JSON**
* Nếu muốn tạo một Controller dạng Restful API thì nên sử dụng **@RestController**. Đây cũng là một @Controller nhưng được bổ sung thêm **@ResponseBody**
* **@GetMapping** sẽ thay thế cho @RequestMapping với method **GET**, điều này cũng áp dụng cho các method khác.

Diagram

Description automatically generated

Demo Project:

Biểu đồ lớp.

Diagram

Description automatically generated

**Thêm swagger ui:**

* Pom.xml:

<dependency>  
 <groupId>io.springfox</groupId>  
 <artifactId>springfox-swagger2</artifactId>  
 <version>2.9.2</version>  
</dependency>  
<dependency>  
 <groupId>io.springfox</groupId>  
 <artifactId>springfox-swagger-ui</artifactId>  
 <version>2.9.2</version>  
</dependency>

* Ở config: tạo SwaggerConfig:

@EnableSwagger2

public class SwaggerConfig {

@Bean

public Docket docket() {

return new Docket(DocumentationType.SWAGGER\_2)

.select()

.build();

}

}