

## Sl4 - Các annotation trong spring boot

### 1. Giới thiệu:

Giống như các chỉ thị để hệ thống hiểu được các request, luôn bắt đầu bằng ký tự @

Ví dụ:

**@RestController:** Đứng trước controller và chỉ ra rằng các request thuộc RESTApi

**@Controller:** Chỉ ra rằng các request sẽ trả về trang html hoặc JSP

**@RequestMapping:** đứng trước controller để chỉ ra lời gọi hàm sau <http://localhost:8080>

Ví dụ: @RequestMapping("/slot4")

=> chúng ta cần gọi: <http://localhost:8080/slot4>

**@GetMapping("/bai1")** => đây là API sử dụng phương thức GET : lấy dữ liệu

=> lời gọi hàm: <http://localhost:8080/slot4/bai1>

**@PostMapping("/bai1")**

=> API POST, sử dụng để thêm dữ liệu

=> <http://localhost:8080/slot4/bai1>

**@PutMapping("/bai1/{id}")**

=> API PUT, sử dụng để sửa dữ liệu theo Id cho trước

<http://localhost:8080/slot4/bai1/1>

**@DeleteMapping("/bai1/{id}")**

=> API Delete, sử dụng để xóa dữ liệu theo Id cho trước

<http://localhost:8080/slot4/bai1/1>

## 2. Cú pháp viết giao diện html, sử dụng thư viện Thymeleaf

\*Trước khi sử dụng, ta cần khai báo thư viện trong thẻ html

```
<html xmlns:th="http://www.thymeleaf.org">
```

\* sử dụng vòng lặp

Thông thường, sẽ đưa vào 1 table

```
<tr th:each="sp : ${listSinhVien}>: mỗi sv lặp trong list
```

\*Truy xuất vào 1 thuộc tính của đối tượng:

```
<span th:text="${doiTuong.thuocTinh}"></span>
```

\*truyền đường dẫn: bắt đầu bằng ký tự @

ví dụ: 

```
<a th:href="@{....}"
```

\*để ứng dụng tham chiếu đúng file giao diện, thì cần khai báo trong application.properties

```
spring.thymeleaf.prefix=classpath:/template/
```

```
spring.thymeleaf.suffix=.html; .jsp
```

=====

Ví dụ1: Liệt kê danh sách sản phẩm với Spring Boot

Ví dụ2: Xem chi tiết sản phẩm

=====