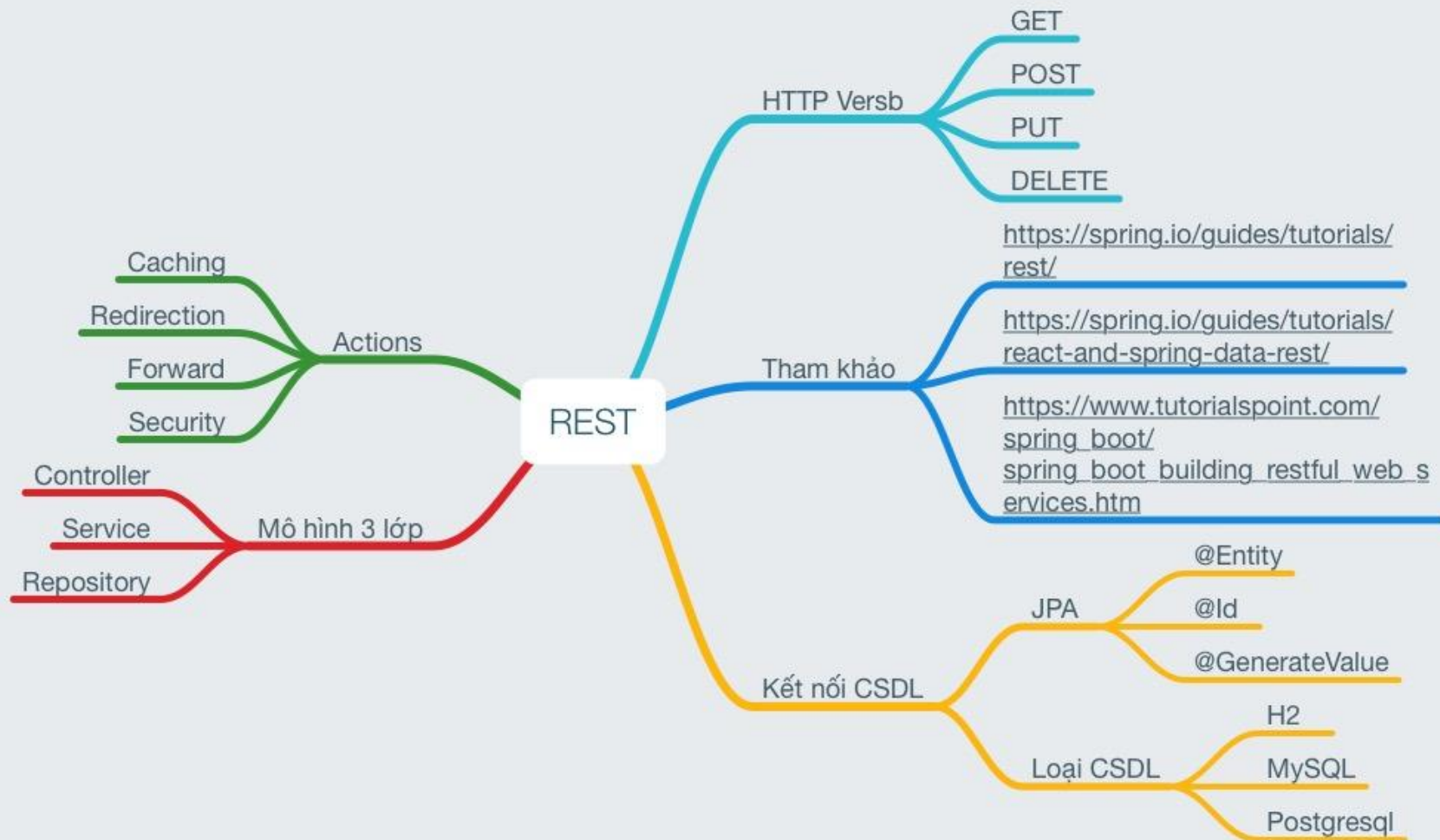




03: Spring And Restful API

Trần Văn Thịnh





Rest & Spring Restful Services

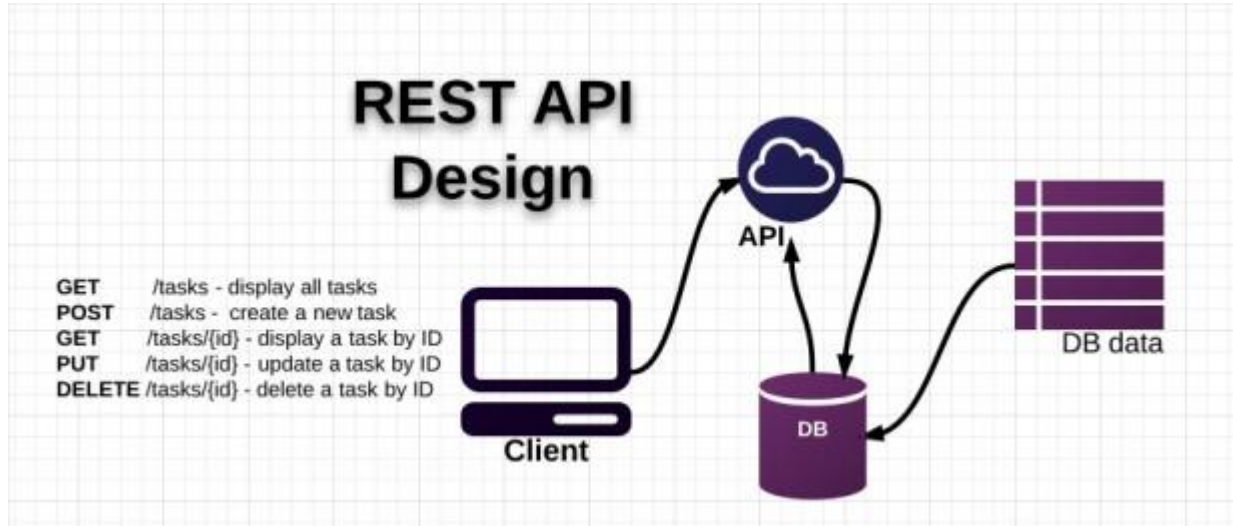
Rest là gì

Http Methods (Verbs)
trong Spring Restful
Services

Rest là gì

- **Representational state transfer (REST)** là một kiểu kiến trúc phần mềm xác định một bộ các ràng buộc để tạo các web service. REST không phải là một tiêu chuẩn.
- Các web service tuân theo kiểu kiến trúc REST, được gọi là các Web RESTful service (hoặc đơn giản là các RESTful service).
- REST dựa trên HTTP1.1
- Mọi thứ trong Rest đều được coi là tài nguyên (resource) và được biểu diễn qua các dạng xml, Json,...

Rest là gì



Http Methods (Verbs) trong Spring Restful Services

- Các RESTful service cho phép phát triển bất kỳ loại ứng dụng nào liên quan đến các hoạt động CRUD (create, retrieve, update, delete - tạo, truy xuất, cập nhật, xóa)
- Nên sử dụng các HTTP verbs(methods) khác nhau tương ứng với các hoạt động CRUD.
- Các HTTP methods chính hoặc được sử dụng phổ biến nhất là GET, POST, PUT, PATCH và DELETE.
- Khi thực hiện các hoạt động này trong các RESTful services, có các hướng dẫn hoặc nguyên tắc đề xuất sử dụng HTTP method cụ thể trên một loại lệnh gọi cụ thể được thực hiện tới server.

Http Methods (Verbs) trong Spring Restful Services

- Xem bảng liệt kê ở bên dưới

Sử dụng postman

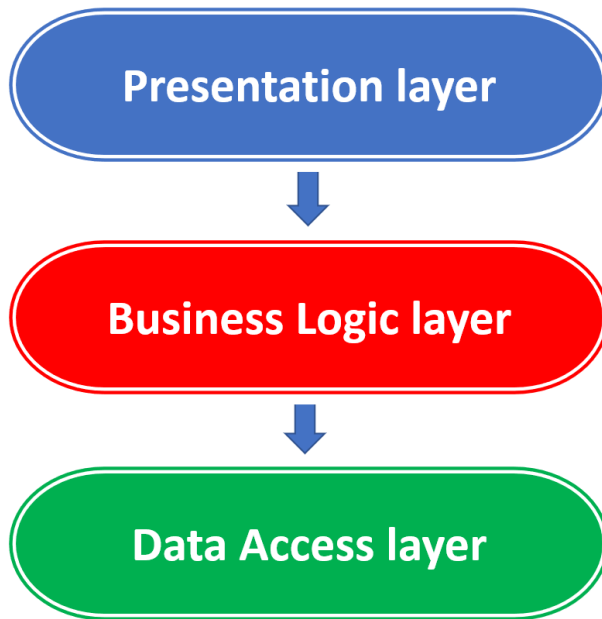
<https://viblo.asia/p/huong-dan-su-dung-postman-cho-test-api-aWj53Lb1K6m>



Mô hình 3 lớp (3 layer Architecture)

Mô hình 3 lớp

Three-Tier (or Three-Layer) Architecture



Mô hình 3 lớp

- Phân chia ra các lớp để đảm nhận các nhiệm vụ khác nhau (ở đây là 3), tầng làm việc với giao diện, tầng làm công việc tính toán, tầng làm việc với dữ liệu
- Việc phân chia này giúp code tường minh hơn, dễ bảo trì nâng cấp sửa chữa hơn.

Mô hình 3 lớp

- **Presentation Layer (Controller)**

- Là tầng giao tiếp với người dùng cuối (client) để hiển thị dữ liệu, thu thập dữ liệu, kiểm tra tính đúng đắn của dữ liệu trước khi gọi tầng service.

- **Business Logic Layer (Service)**

- Đây là nơi kiểm tra các ràng buộc, tính toàn vẹn và hợp lệ của dữ liệu. Đây là cũng là nơi tính toán, xử lý theo các nghiệp vụ và logic của yêu cầu.

- **Data Access Layer (Repository)**

- Đây là nơi thực hiện giao tiếp với cơ sở dữ liệu, xử lý truy vấn và trả về dữ liệu service mong muốn



Các annotation hay dùng trong Spring restful API

Các annotation hay dùng trong Spring restful API

- **@Controller**

- Khai báo Class đó là một controller và có nhiệm vụ mapping request trên url (điều hướng các request) vào các method tương ứng trong controller

- **@RestController**

- Hiểu đơn giản **@RestController = @Controller + @ResponseBody**
- REST services có thể sử dụng với JSON hoặc XML, JSON thì phổ biến hơn, và Spring @RestController trong spring boot application sẽ mặc định trả về JSON response khi có Jackson2 dependency ở trong class-path
- Để trả về dữ liệu xml, cần
 - Jackson xml trong classpath (com.fasterxml.jackson.dataformat -> jackson-dataformat-xml dependency)
 - Client gửi thông tin 'Accept' trong header với value text/xml hoặc application/xml
 - Bỏ produces = MediaType.APPLICATION_JSON_VALUE hoặc cho phép xử lý XML produces = { "application/json", "application/xml" }
- Các đối tượng trả về dưới dạng Object sẽ được **Spring Boot** chuyển thành JSON, mặc định dùng Jackson converter. Nếu muốn tùy biến hơn dữ liệu trả về thì sử dụng ResponseEntity để wrapper các object trả về

Các annotation hay dùng trong Spring restful API

- **@RequestMapping**: Map các http request đến các method của controller, tất cả mọi loại request
- **@GetMapping**: = @RequestMapping + method GET
- **@PostMapping**: = @RequestMapping + method POST
- **@PutMapping**: = @RequestMapping + method PUT
- **@PatchMapping**: = @RequestMapping + method PATCH
- **@DeleteMapping**: = @RequestMapping + method DELETE