|  |  |
| --- | --- |
| Controller | * Là 1 thành phần trong mô hình MVC, có nhiệm vụ điều hướng cho request: gọi model để lấy dữ liệu sau đó xử lý và chuyển dữ liệu sang View * Tạo Controller bằng cách dùng Annotation   + @Controller: dùng để khai báo trong web thông thường  + @RestController: dùng để khai báo Controller trong ứng dụng WebService   * Để tự động phát hiện Controller trong Spring thì cần khai báo:   + Java config: @ComponentScan(“com.codegym.\*”)  + XML: <context:component-scan base-package=”come.codegym.\*”/>   * Để ánh xạ các request tới action tương ứng của Controller dùng: @RequestMapping (có thể ứng dụng được cho lớp hoặc phương thức) * @RequestMapping có 1 số biến thể như sau:   + @GetMapping  + @PostMapping  + @PutMapping  + @DeleteMapping  + @PatchMapping   * 1 số thuộc tính request ánh xạ tới @RequestMapping:   + consumes: ánh xạ tới Content-Type của request  + produces: ánh xạ đến accept của request  + headers: ánh xạ đến header của request |
| * Truyền tham số | * Có 2 cách truyền tham số phổ biến   + Parameter  + Path variable   * Giống nhau: đều thực hiện nhiệm vụ lấy tham số từ người dùng truyền lên server. Sử dụng cái nào tùy vào mục đích của mình, tùy vào thiết kế hệ thống * Khác nhau: RequestParam có thể lấy dữ liệu từ form còn Path variable chỉ có thể lấy dữ liệu từ URL |
| Handler method | * Là phương thức trong controller dùng để xử lý cho 1 request tương ứng * Một số tham số hay gặp:   + HttpServletRequest  + HttpServletRespone  + PathVariable  + RequestParam  + HttpSection  + …   * Giá trị trả về thường gặp: String, ModelAndView, @ReponseBody… |
| Forward vs Redirect |  |
| Chuẩn bị | * Data binding là cơ chế liên kết dữ liệu đầu vào hoặc đầu ra với các đối tượng model, giúp cho việc tương tác với dữ liệu trở nên dễ dàng * Cơ chế hoạt động và cách của data binding trong spring: khi tương tác với form, dữ liệu trên form đc tự động chuyển đổi thành các thuộc tính của đối tượng liên kết với nó, khi thay đổi dữ liệu của đối tượng thì dữ liệu trên các điều khiển cũng thay đổi theo * Ràng buộc dữ liệu có thể là 1 chiều hoặc 2 chiều:   + Chiều đi: chuyển dữ liệu từ các điều khiển trên form vào các thuộc tính của đối tượng dữ liệu  + Chiều về: hiển thị dữ liệu từ các thuộc tính của đối tượng lên các điều khiển của form |