|  |  |
| --- | --- |
| 1. Phân biệt Model, ModelMap, ModelAndView | Trong **Spring MVC**, cả 3 đều được dùng để truyền dữ liệu từ **Controller → View**, nhưng khác nhau:   * **Model**   + Là một **interface** do Spring cung cấp.   + Thường được dùng để thêm dữ liệu dạng key–value:   + model.addAttribute("key", value);   + Dễ dùng, đơn giản, phổ biến trong controller. * **ModelMap**   + Thực chất là một **class implement** từ Map.   + Cho phép lưu trữ dữ liệu dưới dạng cặp key–value, nhưng linh hoạt hơn Model (vì là Map).   + Có các phương thức tiện lợi như addAttribute(), addAllAttributes().   + Dùng khi muốn thao tác map-like. * **ModelAndView**   + Là một đối tượng **kết hợp cả dữ liệu (model) và tên view**.   + Dùng khi bạn muốn vừa trả về **dữ liệu** vừa **chỉ định view cụ thể**.   + Ví dụ:   + ModelAndView mv = new ModelAndView("home");   + mv.addObject("message", "Hello");   + return mv;   👉 Tóm gọn:   * Model: đơn giản để truyền data. * ModelMap: giống Model nhưng mạnh hơn (dựa trên Map). * ModelAndView: gói cả data + view. |
| 2. Ý nghĩa của @RequestMapping và các biến thể |  **@RequestMapping**   * Dùng để ánh xạ URL request vào phương thức hoặc controller. * Có thể gán **path**, **HTTP method**, **params**, **headers**, **produces/consumes**. * Ví dụ: * @RequestMapping(value="/users", method=RequestMethod.GET) * public String listUsers() { ... }    **Các biến thể (shortcut annotations):**   * @GetMapping("/path") → tương đương @RequestMapping(value="/path", method=RequestMethod.GET) * @PostMapping("/path") → POST * @PutMapping("/path") → PUT * @DeleteMapping("/path") → DELETE * @PatchMapping("/path") → PATCH |
| 3. Phân biệt @RequestParam và @PathVariable | * **@RequestParam**   + Lấy giá trị từ **query string** hoặc **form data**.   + Ví dụ: URL /user?id=10   + @GetMapping("/user")   + public String getUser(@RequestParam("id") int id) { ... }   + Thích hợp cho dữ liệu **tùy chọn, filter, search**. * **@PathVariable**   + Lấy giá trị từ **path segment trong URL**.   + Ví dụ: URL /user/10   + @GetMapping("/user/{id}")   + public String getUser(@PathVariable("id") int id) { ... }   + Thích hợp cho dữ liệu **định danh resource**.   👉 Tóm lại:   * @RequestParam: query/form data → /user?id=10 * @PathVariable: trong path → /user/10 |
| 4. Ý nghĩa của RedirectAttributes | * **RedirectAttributes** được dùng khi **redirect** từ một controller đến một URL khác. * Cho phép truyền dữ liệu trong redirect mà **không lộ ra query string**, tránh mất dữ liệu khi redirect.   Có 2 cách chính:   * **addAttribute()** → dữ liệu sẽ gắn vào query string. * redirectAttributes.addAttribute("id", 1); * return "redirect:/user/detail"; * // Redirect đến /user/detail?id=1 * **addFlashAttribute()** → dữ liệu chỉ tồn tại trong 1 request tiếp theo, dùng cho **flash message** (thông báo thành công, lỗi). * redirectAttributes.addFlashAttribute("message", "Thêm thành công"); * return "redirect:/user/list"; * // Ở /user/list có thể lấy message trong model   👉 Ý nghĩa: giúp **chuyển dữ liệu an toàn qua redirect** (thường để thông báo thành công/thất bại sau khi POST). |