|  |  |
| --- | --- |
| Data Binding là gì? | Data Binding là quá trình tự động ánh xạ dữ liệu từ các input trong giao diện (ví dụ: form HTML) vào các đối tượng trong ứng dụng. Trong Spring Framework, cơ chế này thường được sử dụng để ánh xạ dữ liệu từ các yêu cầu HTTP (request) đến các biến hoặc đối tượng Java. |
| Cơ chế hoạt động của Data Binding trong Spring ? | Spring sử dụng một thành phần gọi là DataBinder để thực hiện quá trình binding dữ liệu. Các bước cơ bản:   1. **Yêu cầu từ client**: Người dùng gửi dữ liệu từ form HTML hoặc thông qua API (POST, PUT, GET...). 2. **Handler Mapping**: Spring định tuyến yêu cầu đến controller thích hợp. 3. **Data Binding**:    * Dữ liệu trong yêu cầu được ánh xạ đến các tham số hoặc đối tượng trong controller method (thông qua các annotation như @RequestParam, @PathVariable, hoặc @ModelAttribute).    * DataBinder chịu trách nhiệm ánh xạ dữ liệu từ request vào các đối tượng Java.    * Nếu dữ liệu không hợp lệ, Spring có thể sử dụng các cơ chế kiểm tra và báo lỗi. 4. **Binding Result**: Kết quả của binding được lưu trữ trong đối tượng BindingResult. |
| thuộc tính modelAttribute dùng để làm gì và sử dụng như thế nào? | **Chức năng**: @ModelAttribute được sử dụng để:   * Ánh xạ dữ liệu từ request vào đối tượng model. * Đưa đối tượng vào model để hiển thị trong view.   **Cách sử dụng**:  **a. Trong controller method**:   * Ánh xạ dữ liệu từ request:   @PostMapping("/save")  public String saveUser(@ModelAttribute User user) {  // 'user' sẽ tự động chứa dữ liệu từ form  return "success";  }  **b. Dùng để thêm đối tượng vào model**:  @ModelAttribute("user")  public User populateUser() {  return new User();  }  Đối tượng user sẽ có sẵn trong model để hiển thị trên view. |

|  |  |
| --- | --- |
| Các bước thực hiện data binding trong Spring? |  **Khai báo các đối tượng Model hoặc DTO**:   * Tạo lớp Java với các thuộc tính và getter/setter tương ứng với dữ liệu từ form.   java  Copy code  public class User {  private String name;  private int age;  // Getters and Setters  }   **Xử lý yêu cầu trong Controller**:   * Sử dụng @ModelAttribute để ánh xạ dữ liệu từ form.   @PostMapping("/submit")  public String handleSubmit(@ModelAttribute User user, BindingResult result) {  if (result.hasErrors()) {  return "error";  }  // Xử lý logic  return "success";  }   **Kết hợp với view**:   * Trong file HTML (hoặc JSP), các field input được đặt theo tên thuộc tính của đối tượng:   <form action="/submit" method="post">  <input type="text" name="name" />  <input type="number" name="age" />  <button type="submit">Submit</button>  </form>   **Xử lý lỗi binding (nếu có)**:   * Dùng BindingResult để kiểm tra và xử lý lỗi.   if (result.hasErrors()) {  // Xử lý lỗi  }   **Kết hợp Validator (nếu cần)**:   * Sử dụng các annotation như @NotNull, @Size để kiểm tra dữ liệu đầu vào. |
|  |  |