|  |  |
| --- | --- |
| View | * Là 1 thành phần của MVC giúp hiển thị dữ liệu và cung cấp giao diện cho người dùng tương tác với server. |
| ViewResolver | * Là cơ chế để xử lý view, giúp ánh xạ tên của view sang đối tượng view tương ứng * Quyết định Front Controller sẽ trả về view thực tế nào |
| Thymeleaf | * Là 1 bộ xử lý view được sử dụng cho các ứng dụng web * Được xây dựng phù hợp với các tiêu chuẩn của web (HTML5) * Có thể xử lý HTML, XML, JS, CSS và text * Cung cấp cách thức xây dựng template hiện đại, dễ bảo trì * Cho phép xử lý 6 loại template (template mode): HTML, XML, TEXT, JS, CSS, RAW |
| Biểu thức của Thymeleaf | * Biến: ${…} * Thuộc tính: \*{…} * Message: #{…} * URL: @{…} * Phân đoạn: ~{…} |
| Vòng lặp | * th:each * Cú pháp: th:each=”object, iter:${list}” * Đối tượng iter cung cấp các thuộc tính   + index: chỉ số, bắt đầu từ 0  + count: chỉ sô, bắt đầu từ 1  + size: tổng số lượng phần tử trong list  + current: phần tử hiện tại (object)  + even hoặc odd: vòng lặp chẵn hay lẻ  + first hoặc last: vòng lặp đầu tiên hay cuối cùng |
| Biểu thức điều kiện | • th:if trả về true trong các trường hợp:  • Một giá trị boolean là true.  • Một số có giá trị là non-zero (khác 0)  • Một ký tự có giá trị là non-zero (khác 0)  • Một chuỗi có giá trị khác với “false”, “off” hoặc“no”  • Nếu một giá trị không phải là boolean, số, ký tự hoặc chuỗi. |
| Các bước cấu hình Thymeleaf | Import thư viện  Tạo các bean config: templateResolver, templateEngine, viewResolver  Sau đó khi sử dụng cần import <html xsml:thymeleaf…>  Sử dung thẻ th: để gọi thymeleaf |
| Fragment | * Define 1 biểu thức phân đoạn   th:fragment=”tên bt phân đoạn”   * Cách sử dụng:   th:insert=”~{path.html:: tên bt phân đoạn}” |