|  |  |
| --- | --- |
| Framework là gì? | - Framework là các ứng dụng phần mềm mang tính trừu tượng, cung cấp các tính năng phổ biến và thông dụng, có thể tùy chỉnh để tạo ra các ứng dụng cụ thể khác nhau.  - Mỗi framework cung cấp một phương pháp riêng biệt để xây dựng và triển khai ứng dụng.  - Mỗi framework bao gồm một môi trường tổng thể, có khả năng tái sử dụng để cung cấp các chức năng và công cụ hỗ trợ cho quá trình phát triển ứng dụng. |
| Phân biệt Framework và library | - Điểm khác biệt lớn nhất giữa **Framework** và **Library** chính là cơ chế "Inversion of Control" (Đảo ngược quyền điều khiển):  **- Đối với Library**: Ứng dụng giữ quyền điều khiển (control), lập trình viên tự gọi và sử dụng các hàm của thư viện khi cần.  **- Đối với Framework**: Framework giữ quyền điều khiển, ứng dụng phải tuân theo luồng xử lý và cấu trúc mà framework định sẵn. |
| Các thành phần trong Spring Framework? | **- Core**: Bao gồm IoC container, Events, Resources, i18n (quốc tế hóa), Validation (xác thực), Data Binding, Type Conversion (chuyển đổi kiểu dữ liệu), SpEL (Spring Expression Language), và AOP (Aspect-Oriented Programming).  **- Testing**: Hỗ trợ Mock objects, TestContext framework, Spring MVC Test, và WebTestClient.  **- Data Access**: Gồm các tính năng như Transactions (giao dịch), hỗ trợ DAO, JDBC, ORM, và Marshalling XML.  **- Web Servlet**: Bao gồm Spring MVC, WebSocket, SockJS, và STOMP messaging.  **- Web Reactive**: Cung cấp Spring WebFlux, WebClient, và WebSocket.  **- Integration**: Hỗ trợ Remoting, JMS, JCA, JMX, Email, Tasks, Scheduling, và Cache.  **- Languages**: Tích hợp với các ngôn ngữ như Kotlin, Groovy, và các ngôn ngữ động |
| Các thành phần của mô hình MVC? Chức năng của mỗi thành phần? | **- MVC (Model-View-Controller)** là một kiến trúc phần mềm giúp tách biệt logic nghiệp vụ, giao diện, và điều khiển.  **- Các thành phần:**  **Model**:  Đại diện cho dữ liệu và logic nghiệp vụ.  Tương tác với cơ sở dữ liệu, quản lý trạng thái.  **View**:  Hiển thị giao diện người dùng.  Chỉ nhận dữ liệu từ **Controller** để render.  **Controller**:  Điều hướng luồng yêu cầu (request).  Nhận yêu cầu từ người dùng, xử lý, gọi **Model**, và trả về **View**. |
| Spring MVC là gì? luồng đi của 1 request trong Spring MVC? | **- Spring MVC** là một module trong Spring Framework, sử dụng mô hình MVC để xây dựng ứng dụng web.  Nó xử lý yêu cầu HTTP, kết nối với logic nghiệp vụ, và trả lại phản hồi (HTML, JSON, XML).  **- Luồng đi của một request trong Spring MVC:**  Người dùng gửi request tới **DispatcherServlet** (bộ điều phối trung tâm).  **- HandlerMapping** xác định Controller phù hợp để xử lý request.  Controller nhận request, xử lý logic và gọi **Model** (nếu cần).  Controller trả về tên View.  **- ViewResolver** tìm View tương ứng.  View render dữ liệu và trả lại phản hồi cho người dùng. |
| ViewResolver là gì? | **- ViewResolver** là thành phần trong Spring MVC chịu trách nhiệm ánh xạ tên View (trả về từ Controller) thành tệp giao diện thực tế.  **- Ví dụ**:  Nếu tên View là "home", ViewResolver có thể ánh xạ tới home.jsp hoặc home.html.  **- Các loại ViewResolver phổ biến:**  **+ InternalResourceViewResolver**:  Xử lý JSP (Java Server Pages).  Ví dụ: ánh xạ home thành /WEB-INF/views/home.jsp.  **+ ThymeleafViewResolver**:  Hỗ trợ Thymeleaf templates.  **+ FreeMarkerViewResolver**:  Hỗ trợ FreeMarker templates. |