**Đặc điểm của Mô hình MVC trong Spring Framework**

Mô hình MVC (Model-View-Controller) trên **Spring Framework** là một kiến trúc thiết kế phần mềm giúp phân tách rõ ràng các thành phần của một ứng dụng web, nhằm tăng tính linh hoạt, bảo trì và khả năng mở rộng. Các đặc điểm chính bao gồm:

**1. Phân tách trách nhiệm rõ ràng**

* **Model (M)**:
  + Đại diện cho dữ liệu và logic nghiệp vụ của ứng dụng.
  + Chịu trách nhiệm tương tác với cơ sở dữ liệu và xử lý các tác vụ liên quan đến nghiệp vụ.
  + Được hiện thực thông qua các lớp POJO (Plain Old Java Object) hoặc các mô hình ORM (Object-Relational Mapping) như JPA hoặc Hibernate.
* **View (V)**:
  + Xử lý việc hiển thị dữ liệu và giao diện người dùng.
  + Được triển khai thông qua các công nghệ như JSP, Thymeleaf, FreeMarker, hoặc các framework frontend như React, Angular.
  + Chỉ nhận dữ liệu từ Model và hiển thị mà không chứa logic nghiệp vụ.
* **Controller (C)**:
  + Đóng vai trò là cầu nối giữa Model và View.
  + Nhận và xử lý yêu cầu từ người dùng, gọi Model để xử lý logic và chuyển dữ liệu tới View.
  + Được định nghĩa bằng các lớp Java có annotation như @Controller, @RequestMapping, @GetMapping, @PostMapping.

**2. Hệ thống định tuyến linh hoạt**

* Sử dụng DispatcherServlet làm Front Controller để xử lý tất cả các yêu cầu HTTP.
* HandlerMapping xác định controller thích hợp dựa trên URL yêu cầu.
* ViewResolver xác định View thích hợp để trả về phản hồi cho người dùng.

**3. Hỗ trợ mạnh mẽ cho xử lý dữ liệu**

* **Data Binding**: Tự động chuyển đổi dữ liệu từ yêu cầu HTTP thành các đối tượng Java.
* **Validation**: Tích hợp cơ chế kiểm tra dữ liệu đầu vào thông qua các annotation như @Valid, @NotNull, @Size.

**4. Khả năng mở rộng và bảo trì cao**

* Tách biệt rõ ràng giữa logic nghiệp vụ và giao diện, giúp dễ dàng bảo trì và mở rộng ứng dụng.
* Cho phép thay đổi một thành phần mà không ảnh hưởng đến các thành phần khác.

**5. Xử lý ngoại lệ và bảo mật**

* **Exception Handling**: Sử dụng @ExceptionHandler và @ControllerAdvice để xử lý ngoại lệ một cách nhất quán.
* **Security**: Tích hợp dễ dàng với Spring Security để quản lý xác thực, phân quyền và bảo vệ ứng dụng khỏi các lỗ hổng bảo mật.

**6. Hỗ trợ đa dạng kiểu phản hồi**

* Có thể trả về nhiều kiểu dữ liệu như HTML, JSON, XML, hoặc plain text.
* Sử dụng @ResponseBody hoặc ResponseEntity để tùy chỉnh phản hồi.

**7. Tích hợp với RESTful API**

* Hỗ trợ xây dựng API REST với các annotation như @RestController, @RequestBody, @PathVariable.
* Tích hợp tốt với các công cụ như Swagger để tạo tài liệu API tự động.