

# **Spring MVC Complete Roadmap (Basic to Advanced – Industry Level)**

## **LEVEL 1: Spring MVC Basics**

- Introduction to Spring MVC
- Spring MVC vs Struts
- Spring MVC vs Spring Boot MVC
- Use cases in industry

## **LEVEL 2: MVC Architecture**

- Model, View, Controller
- Front Controller Pattern
- DispatcherServlet overview
- Request–Response lifecycle

## **LEVEL 3: Project Setup (Non-Boot)**

- Maven dependencies
- web.xml configuration
- DispatcherServlet mapping
- spring-mvc.xml
- Component scanning

## **LEVEL 4: DispatcherServlet Internals**

- HandlerMapping
- Controller invocation
- ViewResolver
- View rendering flow

## **LEVEL 5: Controllers & Request Mapping**

- @Controller vs @RestController
- @RequestMapping
- @GetMapping, @PostMapping
- Class-level vs method-level mapping

## **LEVEL 6: Request Data Handling**

- @RequestParam
- @PathVariable
- @RequestBody
- @ModelAttribute
- @RequestHeader, @CookieValue

## **LEVEL 7: Model & View**

- Model
- ModelMap
- ModelAndView
- Redirect vs Forward

## **LEVEL 8: View Technologies**

- JSP & JSTL
- Thymeleaf integration
- ViewResolver types

## **LEVEL 9: Form Handling & Data Binding**

- Form submission flow

- POJO binding
- Custom editors & formatters

#### **LEVEL 10: Validation**

- @Valid & @Validated
- BindingResult
- JSR-303 annotations
- Custom validators

#### **LEVEL 11: Exception Handling**

- @ExceptionHandler
- @ControllerAdvice
- @RestControllerAdvice

#### **LEVEL 12: REST with Spring MVC**

- REST principles
- @ResponseBody
- ResponseEntity
- Status codes
- Pagination & sorting

#### **LEVEL 13: Interceptors, Filters & Listeners**

- HandlerInterceptor
- Servlet Filters
- Context & Session listeners

#### **LEVEL 14: File Handling & Localization**

- File upload & download
- MultipartResolver
- i18n & localization

#### **LEVEL 15: Async, Caching & Performance**

- Async controllers
- @Cacheable
- Performance best practices

#### **LEVEL 16: Testing Spring MVC**

- MockMvc
- Controller testing
- Validation testing

#### **LEVEL 17: Industry Best Practices**

- Layered architecture
- DTO usage
- Logging standards
- API documentation (Swagger)