

# Spring MVC

Spring MVC Interview Questions (3+ Years Experience)

---

## 1) Architecture & Internals (VERY IMPORTANT ★★★)

**Q1. Explain Spring MVC architecture in detail (internal flow).**

Expected depth:

- DispatcherServlet initialization
  - HandlerMapping
  - HandlerAdapter
  - Controller execution
  - Model processing
  - ViewResolver
  - View rendering
- 

**Q2. Why is DispatcherServlet thread-safe?**

Answer:

- One instance handles multiple requests
  - Uses **stateless controllers**
  - Each request has its own thread & request scope objects
- 

**Q3. What happens internally when a request hits DispatcherServlet?**

Answer points:

- doDispatch()
  - HandlerMapping lookup
  - HandlerAdapter selection
  - invoke controller method
  - process return value
  - resolve view
  - render response
-

#### **Q4. Can we replace DispatcherServlet?**

**Answer:**

No. But we can extend it for customization.

---

#### **Q5. Difference between HandlerMapping and HandlerAdapter?**

**HandlerMapping HandlerAdapter**

Finds controller    Invokes controller

URL → method    Executes method

---

## **② Controller Design & Best Practices**

#### **Q6. Why controllers should be stateless?**

- Thread safety
  - Scalability
  - Avoid concurrency issues
- 

#### **Q7. What return types are allowed in controller methods?**

- String
  - Model
  - ModelMap
  - ModelAndView
  - ResponseEntity
  - void
- 

#### **Q8. Difference between returning String and ModelAndView?**

**String                  ModelAndView**

View name only    View + data

Lightweight        Explicit control

---

### **Q9. When do you use ResponseEntity?**

- Custom HTTP status
  - Headers control
  - REST + MVC hybrid
- 

### **Q10. How do you handle multiple request methods for same URL?**

```
@RequestMapping(value="/user", method={GET, POST})
```

---

### **3 Data Binding, Validation & Conversion ★★☆**

#### **Q11. Explain data binding in Spring MVC.**

- WebDataBinder
  - PropertyEditors
  - Type conversion
- 

#### **Q12. Difference between @ModelAttribute and @RequestBody?**

##### **@ModelAttribute @RequestBody**

Form data      JSON/XML

Query params    HTTP body

---

### **Q13. How validation works internally?**

#### **Answer flow:**

- @Valid triggers Validator
  - BindingResult captures errors
  - Validation annotations checked
- 

### **Q14. What is BindingResult and why must it be immediate next parameter?**

- Captures validation errors
  - Order matters to avoid exception
-

### **Q15. How to create custom validator?**

- Implement Validator interface
  - Or use custom annotations
- 

### **4 Exception Handling (VERY COMMON ★★★★)**

### **Q16. Difference between @ExceptionHandler, @ControllerAdvice, and HandlerExceptionResolver?**

Type	Scope
@ExceptionHandler	Single controller
@ControllerAdvice	Global
HandlerExceptionResolver	Low-level

---

### **Q17. How do you return custom error pages?**

- Error view mapping
  - Global exception handler
  - HTTP status mapping
- 

### **Q18. How do you handle REST + MVC exceptions together?**

- Use @ControllerAdvice
  - Return ResponseEntity conditionally
- 

### **5 View Layer, JSP & Performance**

### **Q19. How ViewResolver works internally?**

- Maps logical name → actual JSP
  - Prefix + suffix
  - Caches views
- 

### **Q20. Why JSP is not recommended in modern apps?**

- Tight coupling
- Limited reuse
- No SPA support

---

**Q21. How do you optimize view rendering performance?**

- View caching
  - Avoid heavy JSP logic
  - Use JSTL properly
- 

**6 Request Lifecycle & Scopes ★★★****Q22. Explain request, session, and application scope.****Scope      Lifetime**

Request    One request

Session    User session

Application App lifecycle

---

**Q23. How does Spring manage request scope in multi-threading?**

- ThreadLocal
  - RequestContextHolder
- 

**Q24. What is Flash Attribute?**

- Temporary data between redirects
- 

**7 Security, Filters & Interceptors ★★★****Q25. Difference between Filter, Interceptor, and AOP?****Filter                  Interceptor                  AOP**

Servlet level                  Spring level                  Method level

Before/after request   Pre/Post handle   Cross-cutting

---

**Q26. Use case of HandlerInterceptor?**

- Authentication
- Logging
- Authorization

---

**Q27. Order of execution?**

Filter → Interceptor → Controller → View → Response

---

**8 REST + MVC Integration****Q28. Can Spring MVC and REST coexist?**

Yes.

---

**Q29. Difference between @ResponseBody and ResponseEntity?**

**@ResponseBody** **ResponseEntity**

Body only      Body + status + headers

---

**Q30. How content negotiation works?**

- Accept header
  - HttpMessageConverters
- 

**9 Configuration & Deployment ★★★****Q31. XML vs Java config – what do you prefer and why?**

- Java config → type-safe
  - Less verbose
  - Better refactoring support
- 

**Q32. How Spring MVC loads at startup?**

- web.xml / initializer
  - DispatcherServlet initialization
  - ContextLoaderListener
- 

**Q33. How do you handle environment-specific configs?**

- Profiles
- Property files

---

 **REAL PRODUCTION SCENARIO QUESTIONS ★★★**

**Q34. Application works locally but fails on server – why?**

- Context path issue
  - ViewResolver config
  - Missing WAR config
- 

**Q35. Why do you get 404 even when controller exists?**

- Wrong mapping
  - DispatcherServlet mapping
  - Component scan issue
- 

**Q36. How to debug slow Spring MVC requests?**

- Logging
  - Interceptor timing
  - Thread dump
- 

**Q37. How do you prevent duplicate form submission?**

- PRG pattern
  - Tokens
  - RedirectAttributes
- 

**Q38. How do you handle file upload?**

- MultipartResolver
  - @RequestParam MultipartFile
- 

**1 [0] Comparison Questions (ALWAYS ASKED)**

**Q39. Spring MVC vs Struts?**

**Q40. Spring MVC vs Spring Boot?**

**Q41. JSP vs Thymeleaf?**