**🌱 Week 1: Spring Core Fundamentals**

**Goal**: Master IoC, DI, AOP, and bean lifecycle.

| **Day** | **Topic** | **Key Tasks** | **Resources** |
| --- | --- | --- | --- |
| 1 | **IoC Container** | Create ApplicationContext with ClassPathXmlApplicationContext | [Spring IoC](https://www.baeldung.com/inversion-control-and-dependency-injection-in-spring) |
| 2 | **Dependency Injection** | Wire beans using: - XML config (<bean>) - @Autowired + @Component | [Spring DI](https://www.baeldung.com/spring-dependency-injection) |
| 3 | **Bean Scopes** | Test singleton vs prototype scopes | [Bean Scopes](https://www.baeldung.com/spring-bean-scopes) |
| 4 | **Bean Lifecycle** | Implement InitializingBean, DisposableBean, @PostConstruct | [Lifecycle](https://www.baeldung.com/spring-bean-lifecycle) |
| 5 | **AOP Basics** | Create @Aspect for logging method executions | [Spring AOP](https://www.baeldung.com/spring-aop) |
| 6 | **SpEL** | Use #{...} for dynamic bean properties | [SpEL Guide](https://www.baeldung.com/spring-expression-language) |
| 7 | **FactoryBean** | Build custom factory for complex object creation | [FactoryBean](https://www.baeldung.com/spring-factorybean) |

**Deliverable**:  
✅ A standalone Spring app (no Boot!) with:

* XML + annotation-based bean wiring
* Custom aspect logging method calls
* Prototype-scoped beans

**🚀 Week 2-3: Spring MVC Deep Dive**

**Goal**: Build web apps with classic Spring MVC.

| **Day** | **Topic** | **Key Tasks** | **Resources** |
| --- | --- | --- | --- |
| 8 | **DispatcherServlet** | Configure in web.xml with WebApplicationContext | [DispatcherServlet](https://www.baeldung.com/spring-dispatcherservlet) |
| 9 | **Controllers** | Create @Controller with @RequestMapping | [Spring MVC Controllers](https://www.baeldung.com/spring-controllers) |
| 10 | **Handler Mappings** | Test BeanNameUrlHandlerMapping vs @RequestMapping | [Handler Mappings](https://www.baeldung.com/spring-handler-mappings) |
| 11 | **View Resolvers** | Setup InternalResourceViewResolver for JSPs | [View Resolvers](https://www.baeldung.com/spring-mvc-view-resolver-tutorial) |
| 12 | **Form Handling** | Bind @ModelAttribute to JSP forms | [Spring Forms](https://www.baeldung.com/spring-mvc-form-tutorial) |
| 13 | **Validation** | Add @Valid with BindingResult | [Spring Validation](https://www.baeldung.com/spring-mvc-validation) |
| 14 | **File Upload** | Handle MultipartFile with CommonsMultipartResolver | [File Upload](https://www.baeldung.com/spring-file-upload) |
| 15 | **Interceptors** | Log requests with HandlerInterceptor | [Interceptors](https://www.baeldung.com/spring-mvc-handlerinterceptor) |
| 16 | **REST Support** | Convert @Controller to @RestController | [Spring REST](https://www.baeldung.com/spring-controller-vs-restcontroller) |
| 17 | **Content Negotiation** | Return XML/JSON with @ResponseBody | [Content Negotiation](https://www.baeldung.com/spring-mvc-content-negotiation-json-xml) |
| 18 | **Exception Handling** | Use @ExceptionHandler and SimpleMappingExceptionResolver | [Exception Handling](https://www.baeldung.com/exception-handling-for-rest-with-spring) |

**Deliverable**:  
✅ A **pure Spring MVC** web app (no Boot!) with:

* JSP views + form binding
* REST endpoints returning JSON
* Global exception handling

**🔧 Week 4: Advanced Core & Integration**

**Goal**: Master niche but critical features.

| **Day** | **Topic** | **Key Tasks** | **Resources** |
| --- | --- | --- | --- |
| 19 | **Property Sources** | Load config with @PropertySource | [PropertySource](https://www.baeldung.com/properties-with-spring) |
| 20 | **Event Handling** | Publish/listen to custom ApplicationEvent | [Spring Events](https://www.baeldung.com/spring-events) |
| 21 | **Task Execution** | Schedule tasks with TaskExecutor | [Task Execution](https://www.baeldung.com/spring-task-executor) |
| 22 | **JDBCTemplate** | CRUD ops without Hibernate | [JDBCTemplate](https://www.baeldung.com/spring-jdbc-jdbctemplate) |
| 23 | **Testing** | Write SpringJUnit4ClassRunner tests | [Spring Testing](https://www.baeldung.com/integration-testing-in-spring) |
| 24 | **Profiles** | Switch configs with @Profile | [Spring Profiles](https://www.baeldung.com/spring-profiles) |

**Deliverable**:  
✅ A console app with:

* Custom events (e.g., UserRegisteredEvent)
* JdbcTemplate-based DAO layer
* Profile-specific DB configurations

**🎯 Key Interview Questions (Core + MVC)**

1. **How does ApplicationContext work?**
2. **Explain DispatcherServlet request flow**
3. **Difference between @Component and @Bean?**
4. **How to handle exceptions in Spring MVC?**
5. **What is a ViewResolver?**

**📚 Resources (No Spring Boot!)**

1. **Books**:
   * *Spring 5 Design Patterns* (Dinesh Rajput)
   * *Expert Spring MVC and Web Flow* (Rob Harrop)
2. **Tutorials**:
   * [Spring Framework Docs](https://docs.spring.io/spring-framework/docs/current/reference/html/)
   * [Mkyong Spring Tutorials](https://mkyong.com/tutorials/spring-tutorials/)

**💡 Final Project Idea**

Build a **library management system** with:

* XML-based bean configuration
* Spring MVC + JSP frontend
* JdbcTemplate for database ops
* Custom LibraryEvent publishing

This roadmap ensures you **deeply understand Spring's core mechanics** before moving to Spring Boot. Want to adjust any topics? Let me know! 🛠️

New chat