



SLIDE 1 — Cover Page

Preparing for a Java Backend Interview?

A quick guide to must-know questions
& topics

#Java #SpringBoot #InterviewPrep



SLIDE 2 – Introduction

Over the last few weeks, I've attended multiple interviews for Java Backend roles – and noticed certain questions came up repeatedly.

Here's a curated checklist to help you crack your next interview! 📌

SLIDE 3 — Core Java (Part 1)

- ✓ **Java 8 Features (Lambdas, Streams, Functional Interfaces)**
- ✓ **Streams API — map, filter, reduce, collect**
 - (💡 **Expect coding questions!**)
- ✓ **Immutable Class — how to create & its advantages**
- ✓ **Create a Singleton Class — steps & best practices**
- ✓ **Marker Interfaces — what and why?**



SLIDE 4 —



Core Java (Part 2)

- ✓ **HashSet Working — internal implementation**

- ✓ **Internal Working of HashMap**

- ✓ **HashMap vs TreeMap — when to use which?**

- ✓ **Fail-safe vs Fail-fast Iterators**

- ✓ **JVM Memory Areas — Heap, Stack, Method Area, etc.**

- ✓ **Reflection in Java — uses and best practices**



SLIDE 5 — ☕ Core Java (Part 3)

✓ **volatile vs Atomic variables —**

Visibility guarantees & differences

✓ **Thread Creation — methods & lifecycle**

✓ **Fixing**

ConcurrentModificationException

✓ **Future vs CompletableFuture —
async programming differences**

✓ **FlatMap vs Map — when to use
which?**

✓ **String, StringBuilder & StringBuffer
— performance and use cases**



SLIDE 6 – 🌱 Spring & Spring Boot

- ✓ **Bean Scopes – singleton, prototype, request, session**
- ✓ **IOC Container & DI – constructor vs setter vs field injections**
 - ✓ **Exception Handling – using @ExceptionHandler & @ControllerAdvice**
- ✓ **CRUD & JPA Repositories – best practices and differences**
- ✓ **JWT & Auth Token – implementation & flow**
 - ✓ **Securing Endpoints – best practices**
- ✓ **Spring Security – common questions & use cases**
 - ✓ **Redis Caching – why & how?**
 - ✓ **@Transactional – when and why?**
 - ✓ **Annotations in Spring – common and advanced usage**




SLIDE 7 — Microservices

- ✓ Saga Pattern — managing distributed transactions across services**
- ✓ Circuit Breaker Pattern — resilience design (using Resilience4J)**
- ✓ Deployment Models — Docker & Kubernetes best practices**
- ✓ ELK Stack — how it works & benefits for logging and monitoring**



SLIDE 8 — Design Patterns & Best Practices

 **SOLID Principles — especially Dependency Inversion & why it matters**

 **How to print "Hello World" WITHOUT using the main method?
(One of the unique questions I came across!)**

 **Prepare questions on DSA (Medium & Easy Level) — Take reference from Striver's Sheet for practice**



SLIDE 9 — 💡 Final Advice

If you're actively preparing for interviews:



Save this post for later!



Share it with someone on the same path.



Stay consistent, practice daily, and crack that next interview! 💪🔥