

# **Java Backend Engineer: Sammlung von Interviewfragen**

## **Projektbezogen**

1. Selbstvorstellung
2. Projektvorstellung
3. Projektumfang
4. Thread- und Nebenläufigkeitskenntnisse
5. Unterschiede zwischen Lock und synchronized-Lock
6. Unterschiede zwischen MYSQL und MONGODB
7. Welche Komponenten und Versionen von SpringCloud werden verwendet

## **Spring-Ökosystem**

1. Was ist der Kern von SpringBoot
2. SpringBoot-Version
3. Welche SpringCloud-Komponenten werden verwendet
4. Unterschiede zwischen SpringBoot und SpringCloud
5. Spring-Bean-Injektionsmethoden
6. Benennungskonventionen für Beans bei der Konfiguration mehrerer Datenquellen

## **Microservice-Architektur**

1. Wie man Microservice-Module aufteilt
2. Bei der Kopplung: Was hat Vorrang, Geschäfts- oder technische Kopplung?
3. Welches Microservice-Muster wird verwendet?
4. Unterschiede zwischen zustandslosen und zustandsbehafteten APIs im Design, wie man wählt
5. Standardanzahl von Knoten für Cloud-Bereitstellung

## **Java-bezogen**

1. Verwendete JDK-Version
2. Stream-Implementierung auf unterster Ebene
3. Wie man Probleme im Stream-Code debuggt

4. Stream-Komponenten und zugrunde liegende Struktur
5. Wie man Nachrichtenaustausch zwischen Threads implementiert
6. Was ist ein wiedereintrittsfähiges Schloss (Reentrant Lock)
7. Erläutern Sie Ihr Verständnis von Sperren (Locks)
8. Unterschiede zwischen JDK8 und JDK17

## **Server-Konfiguration**

1. Standard-Thread-Anzahl in Tomcat
2. Migration von Tomcat zu anderen Webservern
3. Nginx Reverse-Proxy-Konfiguration

## **Datenbank und Cache**

1. Wie man eine Redis-Verbindung in einem Projekt erstellt, erforderliche Abhängigkeiten
2. Wie man eine Verbindung mit Jedis herstellt
3. Redis-Transaktionsbefehle
4. SQL-Szenario: Join-Abfrage zwischen Abteilungs- und Mitarbeitertabellen für das höchste Gehalt pro Abteilung
5. Vorteile eines Left Joins

## **Programmier-Szenarien**

1. Implementieren Sie die Sortierung von Person-Objekten nach Alter und Name mit der Java8 API
2. Geben Sie gemeinsame Zeichen zwischen zwei Zeichenketten aus

## **CI/CD**

1. Erfahrung im Schreiben von Dockerfiles
2. Groovy-Closure-Funktionen in Jenkins-Pipelines
3. Zweck des Nexus-Tools
4. Praktische Erfahrungen mit DevOps

## **Kernkonzepte**

1. Konzepte und Funktionen des Java Memory Model (JMM)
2. Implementierungsprinzipien von HashMap und Konfliktlösung
3. Implementierung des Proxy-Musters (statisch und dynamisch)
4. Prinzipien von IoC und AOP
5. Warum Redis mit einem einzelnen Thread hohe Leistung beibehält
6. Konfiguration der Parameter des Executor-Thread-Pools

## **Schwerpunkte indischer Interviewer**

1. Grundlegende Java-Konzepte (mit Beispielen)
2. Verwendung von Polymorphismus
3. Leistung und Unterschiede von LinkedList
4. Unterschiede zwischen Maps
5. Lösungen zur Array-Verarbeitung
6. Linux-Befehle
7. Projekterfahrung (Tech-Stack: Python, Angular, Control-M)
8. Englischkenntnisse

## **Englischkenntnisse: Niveaustufen**

1. Kann sich nicht ausdrücken - Schwache Grundlagen, unfähig, sich selbst vorzustellen oder Projekte zu präsentieren
2. Einfache Grundgespräche - Grundlegende Kenntnisse, kann vorbereitete Selbstvorstellungen abliefern
3. Alltägliche Kommunikation - Gute Grundlagen, kann an englischen Meetings teilnehmen und den Großteil des Inhalts verstehen
4. Arbeitsfähig - Starke Grundlagen, kann fließend mit dem Interviewer kommunizieren
5. Arbeitsfähig & technischer Austausch - Hervorragende Sprachkenntnisse, kann sich barrierefrei über technische Themen austauschen

## **Weitere Bewertungskriterien im Interview**

1. Lernmethoden
2. Fähigkeit zur technischen Kommunikation
3. Erfahrung in der Kundenkommunikation
4. Projektherausforderungen und Lösungen
5. Karriereentwicklungsplan
6. Technische Tiefe (Multithreading, Datenvalidierung, Frontend-Entwicklung usw.)