Outlook

PrepQues

From Uddin, Md. Kamran (Cognizant) <Md.Kamran.Uddin@cognizant.com>
Date Tue 10/17/2023 10:12 PM
To Uddin, Md. Kamran (Cognizant) <Md.Kamran.Uddin@cognizant.com>

Type	Area	Details
Must Have	Strong Java Proficiency	Expertise in core Java programming, multi-threading, and memory management.
Must Have	Strong Java Proficiency	In-depth understanding of Java ecosystem and latest language features.
Must Have	Distributed Systems Concepts	Solid grasp of distributed systems principles, such as scalability, fault-tolerance, consistency, and concurrency control.
Must Have	Distributed Systems Concepts	Familiarity with distributed data storage and processing paradigms (e.g., distributed databases, message queues, caching mechanisms).
Must Have	Architectural Patterns	Ability to design and implement distributed systems using appropriate architectural patterns, such as microservices, event-driven architecture, and service-oriented architecture (SOA).
Must Have	Architectural Patterns	Strong understanding of the trade-offs between inheritance and composition, and the ability to select the appropriate pattern for a given scenario.
Must Have	Concurrency and Parallelism	Thorough understanding of concurrent programming in Java, using frameworks like Java Concurrency API and Executor framework.
Must Have	Concurrency and Parallelism	Experience in handling parallel processing and synchronization challenges in distributed environments.
Must Have	Networking and Communication	Proficiency in network protocols and communication mechanisms like HTTP, TCP/IP, UDP, RPC, and RESTful APIs.
Must Have	Networking and Communication	Understanding of how to optimize network performance and minimize latency.
Must Have	Data Consistency and Replication	Familiarity with data replication techniques, consistency models (e.g., eventual consistency, strong consistency), and data partitioning strategies.
Must Have	Fault Tolerance and Resilience	Knowledge of techniques for ensuring fault tolerance and high availability in distributed systems, such as replication, redundancy, and failover mechanisms
Must Have	Performance Monitoring and Optimization	Proficiency in identifying performance bottlenecks in distributed systems and employing tools to monitor and optimize system performance
Must Have	Security Considerations	Awareness of security principles in distributed systems, including authentication, authorization, and secure communication
Must Have	Testing and Debugging	Skill in writing effective unit tests, integration tests, and end-to-end tests for distributed systems.
Must Have	Testing and Debugging	Experience with debugging distributed systems issues using various tools and techniques.
Must Have	Collaboration and Leadership	Ability to mentor and lead a team, providing technical guidance and code reviews

Nice to have	Containerization and Orchestration	Familiarity with containerization technologies like Docker and container orchestration platforms like Kubernetes
Nice to have	Version Control and CI/CD	Proficiency in version control systems like Git and experience with Continuous Integration/Continuous Deployment (CI/CD) pipelines
Nice to have	Cloud Technologies	Experience with cloud platforms like AWS, Azure, or Google Cloud, and knowledge of relevant services for building distributed systems.
Nice to have	Cloud Technologies	Ability to deploy, manage, and scale distributed applications in a cloud environment.
Nice to have	Distributed Algorithms	Understanding of fundamental distributed algorithms like distributed consensus (e.g., Paxos, Raft), distributed locking, and distributed caching
Nice to have	Communication skills	Spoken English

==

1. Tell me about your project and technologies used? And raised questiones related to the project.

2. what is your recently working project?

3. What is entitlement?

4. How to create a spring boot application and Angular from scratch and expecting what are the other things which are done in the project?

5. Authentication and authorization in Spring Boot/secure an API?

6. Design pattern followed in your project?

7. How to create a Spring boot profiles?

8. How do you handle exception in your application?

9. How do you fetch top 100 records from a Database with complex query?

Explain Event driven Architecture

How to address data inconsistency issue with Event driven architecture

What Swagger is used for

Have you used JOOQ

What are the concerns when you develop UI code and Server side code by different teams ?
How do they work independently before knowing the data that is exchanged between two layers ?

How would you synchronize the data across Microservices ?

How to address concurrency issues with Microservices ?

Explain how Publish Subscribe model works

What is your experience with Docker and Kubernetes

How do you handle graceful shutdown of applications

Questions on last project.

JWT

Docker implementation

Kafka implementation

====================================

1. Rest API vs gRPC highlevel, Java vs GoLang highlevel, Kafka vs NATS/MQ highlevel, Reduce latency of app (for ex. CDN, query optimization, use of gRPC instead of Rest APIs etc. or anything else you could suggest)

2. Event Driven Architecture, SAGA pattern, CQRS pattern (it helps to reduce latency as well)
3. Java 17 features added, Java 21 features being added