

WALMART INTERVIEW QUESTIONS

Round-1: DSA+Java Basics:

1. Write a program to reverse a linked list and calculate time complexity
2. How to reverse a string using stack
3. What is enqueue and dequeue operations in Queue
4. How does memory management work in Java
5. Garbage collection
6. Difference between Interface and Abstract class
7. What is REST API and difference between PUT and PATCH

DSA Reference: <https://www.youtube.com/@mycodeschool>

Memory Management in Java: <https://www.geeksforgeeks.org/java-memory-management/>

Garbage Collection: <https://newrelic.com/blog/best-practices/java-garbage-collection>

Round-2: Backend:

1. Explain the entire architecture of your project
2. Have you used design patterns in your project
3. What is factory and builder patterns
4. What is reverse proxy and how can you implement it
5. What are different types of authentication methods
6. How do you secure your http/rest endpoints
7. Features of microservice
8. How can you make your microservice more resilient to failure
9. How can you make sure that your microservice is highly available and scalable. Horizontal and Vertical scaling
10. Can we insert a data in database using GET request
11. What is spring IOC container
12. Features of spring boot
13. What authentication mechanism are you using in your project
14. How do you manage transactions in your spring boot application
15. What is database sharding
16. How to avoid data redundancy in database

Design Patterns in Java: High level- <https://medium.com/@saygiligozde/design-patterns-in-java-5251032ca244> In Depth - <https://refactoring.guru/design-patterns/java>

Types of Authentication Methods: <https://medium.com/@robert7kariuki/types-of-authentication-methods-comprehensive-guide-2023-387b720a63c1>

Microservices Features, Benefits and Limitations: <https://medium.com/hprog99/microservices-architecture-adc013129f01#:~:text=Microservices%20is%20an%20architectural%20style%20that%20structures%20an%20application%20as,flexibility%20and%20faster%20development%20cycles.>

<https://medium.com/microservicegeeks/an-introduction-to-microservices-a3a7e2297ee0>

Handling Failures in Microservices: <https://dip-mazumder.medium.com/building-resilient-microservices-strategies-for-handling-failures-in-stock-price-notification-629a3e206f41>

<https://www.linkedin.com/advice/0/what-best-practices-ensuring-resilience-fault-tolerance-ef1fe>

Scaling Microservices: <https://medium.com/cloud-native-daily/scaling-microservices-a-comprehensive-guide-200737d75d62>

<https://dev.to/somadevtoo/horizontal-scaling-vs-vertical-scaling-in-system-design-3n09>

Spring IoC: <https://www.digitalocean.com/community/tutorials/spring-ioc-bean-example-tutorial>

Spring JPA Transaction Management: <https://www.geeksforgeeks.org/spring-boot-transaction-management-using-transactional-annotation/>

Database Sharding: <https://www.geeksforgeeks.org/database-sharding-a-system-design-concept/>

Reduce Data Redundancy in DB: <https://dbsyncseo.medium.com/ways-to-reduce-data-redundancy-f45e2194a23f>

<https://medium.com/@yg17381/normalization-in-dbms-297269fe7e9f>

System Design: <https://www.youtube.com/@gkcs> (System Design Playlist)

Round-3: Frontend:

1. What is difference between class and functional component in react
2. React component lifecycle
3. What is hooks
4. What is useMemo hook
5. What is useCallback hook and how does it different from useMemo hook
6. What is useRef hook
7. What is Client-Side Rendering (CSR) and Server-Side Rendering (SSR) and in which scenarios do you prefer using these
8. What is Suspense in react
9. **Use Case:** Which type of rendering (Server side or Client side) will you prefer using in Web-based News Application and why.

React JS Frequently Asked Interview Questions: <https://github.com/sudheerj/reactjs-interview-questions>

JavaScript Frequently Asked Interview Questions: <https://github.com/sudheerj/javascript-interview-questions>

React Component Lifecycle: <https://www.freecodecamp.org/news/react-component-lifecycle-methods>

ReactJS Reconciliation: <https://www.geeksforgeeks.org/reactjs-reconciliation/>

How React Works Under the Hood: <https://medium.com/@ruchivora16/react-how-react-works-under-the-hood-9b621ee69fb5>

Client-side vs Server-side Rendering: <https://medium.com/@olayidecodes/client-side-rendering-vs-server-side-rendering-3c41f03c700d>

<https://medium.com/@vamsee.l.krishna/server-side-rendering-vs-client-side-rendering-how-why-and-when-c8838bf87537>