

Java Spring Boot & Microservices Interview Questions

1. Why Profiles in spring boot.?

To deal with multiple environments in development of application, ideally each environment has separate configuration/property files like Dev, QA, Pre-Prod and Prod. Spring has provision of Profiles to make to use of configuration specific to environment.

2. How do we define Properties in Spring Boot.?

You can define properties into a file called application. properties in a Spring Boot Application, you can create file manually or make use of Spring Initializer. If property files exist in class path by default spring boot loads the properties into an application.

3. What is Spring Boot Actuator.?

Actuator is mainly used to expose operational information about the running application like monitoring our application, gathering metrics, understanding traffic and the state of our database. Actuator brings production- grade tools to an application without explicit implementation.

4. How to enable Actuator in Spring boot application?

To enable the spring actuator feature, we need to add the dependency from Group Id: org.springframework.boot and Artifact Id: spring-boot-starter-actuator in pom.xml.

5. What is @SpringBootApplication Annotation in Spring Boot.?

@SpringBootApplication comes from the base package of org.springframework.boot.autoconfigure and the combination of @Configuration, @EnableAutoConfiguration, and @ComponentScan annotations with their default attributes.

6. How to change Embedded tomcat server in Spring Boot.?

Default Embedded tomcat server can be changed to any of Jetty, Undertow and Reactor servers by changing the default dependency to specific dependency in pom.xml and also by mentioning exclude option to particular dependency.

7. How to use Request Mapping and Get Mapping.?

Request Mapping can be used with GET, POST, PUT, and many other request methods using the method attribute on the annotation. Whereas Get Mapping is only an extension of Request Mapping which helps you to improve on performing the Get request.

8. What is the use of Spring Boot starters.?

Spring Boot starters are set of dependency descriptors can be added as dependencies to spring boot application and by default spring provides various starters which helps in rapid application development.

9. What is thyme leaf in Spring Boot.?

Thyme leaf is a server-side Java template engine for web applications which can integrate with spring framework by adding the spring-boot-starter-thymeleaf starter as dependency in pom file.

10. How to make use/available Custom Application Configuration in Spring Boot.?

Custom Application Configuration in Spring Boot by using the @Value annotation, but with this all the configuration values will be distributed throughout the application instead use @ConfigurationProperties by defining configuration component.

11. How to Change the Default Port in Spring Boot.?

Default port can be changed in 3 ways first, by defining server.port in application properties file second, by setting server.port in SpringBootApplication main class third, by setting server.port while application running as jar file from command line.

12. How to configure properties in Spring Boot Application.?

Configuration of properties in Spring Boot Application can be done either by using Property or YAML file. YAML is more readable for hierarchical configuration data.

13. What is the advantage with DevTools in Spring Boot.?

DevTools in Spring Boot will improve the development time while working on Spring Boot applications. While development spring boot application will restart automatically for any code changes, to get this feature spring-boot-devtools dependency needs to be added.

14. How to ignore null values in JSON response in spring boot?

To ignore null values in JSON response, use @JsonIgnore annotation. The field can be ignored while reading JSON into Java objects as well as while writing Java objects into JSON.

15. Basic Features of Spring Boot.?

Spring Boot has basic features like set of dependencies as starters, and the capability of providing Auto-Configuration and Spring Initializer from Spring core modules.

16. How to avoid/exclude auto-configuration classes in Spring Boot.?

Spring Boot provided the flexibility to avoid one or more auto-configuration classes with the help of exclude attribute or attribute Name(which are not in class path) and adding the property in property file like spring.autoconfigure.exclude.

17. What is the use of Spring Initializer in Spring Boot.?

Spring Boot Initializer helps in rapid application development by providing Spring Boot Project with all required project structure/skeleton including dependencies without creating the project structure explicitly.

18. What is Live Reload in Spring Boot?

Live Reload is a spring-boot-devtools module that includes Live Reload server to trigger a browser refresh when a resource is changed. Live Reload server extensions are available freeware for Firefox, Chrome, etc.

19. How to Handle Exceptions in Spring Boot Application.?

Exceptions in the Spring Boot Application can be handled by using `@ControllerAdvice` and `@ExceptionHandler` to create generic or application specific exceptions. `@ControllerAdvice` returns the created exception as response to represent the exception type.

20. How do you generate a Spring Boot Application.?

One of the ways to generate the Spring Boot based Project is using Spring Initializer by visiting spring.io by selecting appropriate parameters and another way is to use Spring Tools 4 for Eclipse (IDE) and generate a Spring Boot Project direct inside IDE.

21. How to create a login page in spring boot?

You can create a simple and default login page in spring boot by using Spring security and it secures all HTTP endpoints where the user has to login into the default HTTP form provided by spring. We need to add spring-boot-starter-security dependency in your pom.xml or build.gradle and a default username and password can be generated with which you can log in.

22. How the bootstrapping happens in Spring Boot.?

Bootstrapping of a Spring Boot application can be possible by using Spring Initializer, where creating the spring boot project with required dependencies and execution of application in IDE.

23. Why spring boot is used for microservices?

In microservices, you can write code for your single functionality. You can use different technology stacks for different microservices as per the skill set. You can develop this type of microservices with the help of Spring boot very quickly as spring boot gives priority to convention over configuration which increases the productivity of your application development.

24. What is Filter in Spring Boot Application.?

Filters will help in intercepting all HTTP incoming requests and outgoing responses. Filters are available by implementing Filter Interface, using FilterRegistrationBean and using MVC controller.

25. What is CORS in Spring Boot?

CORS stands for Cross-Origin Resource Sharing is a mechanism implemented by browsers and helps users to authorize cross-domain requests. This mechanism serves as an alternative to less secure and less powerful hacks of the kinds of IFrame or JSONP.

26. What are the advantages of Microservices Architecture.?

There are plenty of advantages that are offered by Microservices architecture like Microservices can adapt easily to other frameworks or technologies, Failure of a single process does not affect the entire system, provides support to big enterprises as well as small teams, can be deployed independently and in relatively less time, etc.

27. Why Reports and Dashboards in Microservices.?

Reports and dashboards are mainly used to monitor and track microservices and can be used to find out which microservices expose what resources, find out the services which are impacted whenever changes in a component occur, provide an easy point which can be accessed whenever documentation is required, etc.

28. Mention Few drawbacks in using Microservices.?

Microservices Requires heavy investment in synchronize the services well all the time, Heavy architecture set up needed for distributed or high-end systems, Excessive planning for handling operations overhead, Autonomous staff selection, etc.

29. What is the use of Containers for Microservices.?

To manage a microservice based application, containers are the easiest alternative. It helps the user to individually deploy and develop. You can also use Docker to encapsulate the microservice in the image of a container. Without any additional dependencies or effort, microservices can use these elements.

30. What is Semantic Monitoring in Microservices.?

The primary benefit of Semantic Monitoring is to find out the factors which are more profitable to your business. Semantic monitoring along with service layer monitoring approaches monitoring of microservices from a business point of view. It combines monitoring of the entire application along with automated tests.

31. How to access Restful Microservices.?

Restful Microservices can be accessed in ways by using a rest template that is load balanced and by using multiple microservices.

32. How to perform Security Testing for Microservices.?

Security Testing for Microservices can be possible in different ways like ensure that any line of code is bug-free and can be replicated, security solution should be flexible so that it can be adjusted as per the requirements of the system, security protocols should be flexible and updated to cope up with the new threats by hackers or security breaches, etc.

33. What are the Challenges while testing Microservices Application.?

While testing microservices application test engineers should have a thorough understanding of all the inbound and outbound processes before he starts writing the test cases for integration testing. When independent teams are working on different functionalities, collaboration can prove to be quite a struggling task. It can be tough to find an idle time-window to perform a complete round of regression testing.

34. Explain the inter communication between Microservices.?

One Microservice will communicate with other Microservice by implementing Restful services for incoming request and for outgoing response, by using third party server programs will helps in better communication, by using web sockets for streaming.

35. What is Distributed Transaction in Microservices.?

Transactions are distributed to several services to complete the transaction in sequence with so many moving parts, it is very complex and prone to failure. Distributed transactions are an outdated approach in today's microservice architecture that leaves the developer with severe scalability issues.

36. How to test end-to-end Microservices Application?

Usually, end-to-end (E2E) microservice testing is used to ensure that all components work together for a complete user journey. Usually, it is done through the user interface application from test environment. It also ensures all processes in the workflow are working properly.

37. what is Eureka Server in Microservices.?

Eureka Server also known as Netflix Service Discovery Server, is an application that keeps track of all client-service applications. As every Microservice registers to Eureka Server, Eureka Server knows all the client applications running on the different ports and IP addresses. It generally uses Spring Cloud and is not heavy on the application development process.

38. What is the use of docker in microservices?

Docker generally provides a container environment, in which any application can be hosted. This is accomplished by tightly packaging both the application and the dependencies required to support it. These packaged products are referred as Docker Containers. Docker allows you to containerize your microservices and manage these microservices more easily.

39. What are Reactive Extensions in Microservices?

Reactive extensions are also known as Rx, is basically a design approach that calls multiple services and then generates a single response by combining the results. The calls can be synchronous or asynchronous. A popular tool in distributed systems, Rx works exactly opposite to legacy flows.

40. What is the advantage of using Eureka Server in Microservices.?

Netflix's Eureka is the most efficient in setting up service discovery. This is a hassle-free procedure that doesn't add much weight to the application. It also supports a wide range of web applications. Several annotations are provided by Spring Cloud to make its use as simple as possible and to hide complex concepts in building service discovery.

41. Explain few main features of Microservices.?

Few main features of microservices like services within a system are largely decoupled. So, the application can be easily built, altered, and scaled. Continuous delivery allows frequent releases of software, through systematic automation of software creation, testing, and approval. Microservices do not focus on applications as projects. Instead, they treat applications as products for which they are responsible.

42. What are the most significant benefits of using microservices?

The most significant benefit of using microservices is that it builds an application to collect small autonomous services developed for a business domain. So, if the business needs to change constantly, the development teams can rapidly build new apps components to meet the requirement. Each microservice runs a unique process and communicates through a well-defined, lightweight mechanism, such as a container, to serve a business goal.

43. How does a Microservice architecture work.?

The Microservice architecture of an application can be simplified into multiple modules that independently perform the single precise standalone task. Each application module is an independent service/process that can be replaced, updated, or deleted without disrupting the rest of the application. Under microservice architecture, an application can grow along with its requirements.

44. How many ways we can deploy Microservices Application.?

Microservices Application can be deployed with Serverless Deployment, it packages the service as a ZIP file and uploads it to the Lambda function. The Lambda function is a stateless service that automatically runs enough micro-services to handle all requests. And another way with Multiple Service Instance per Host, it is used to run single or multiple service instances of the application on a single or multiple physical/virtual host.

45. Explain difference between SOA and the Microservices Architecture?

SOA stands for Service Oriented Architecture. It is a collection of services used to communicate with each other through simple data passing or activity coordination. On the other hand, the Microservices Architecture is a collection of small functional modules that are independently deployable, scalable, target specific business goals, and communicate over standard protocols.

46. What are the few challenges while working Microservice Architectures?

Developing several smaller microservices sounds easy, but the challenges often faced while developing them are Automate the Components is difficult to automate because there are several smaller components. So, for each component we must follow the stages of Build, Deploy and, Monitor. Debugging is difficult to find out each service for an error. It is essential to maintain centralized logging and dashboards to debug problems.

47. What are the other challenges while working Microservice Architectures?

Other challenges are Perceptibility in maintaining many components together becomes difficult to deploy, maintain, monitor and identify problems. It requires great perceptibility around all the components. Configuration Management for the components across the various environments becomes tough sometimes.

48. What are the types of tests used in Microservices?

Tests used in Microservice architecture into three main categories as the top-level tests are used to conduct acceptance tests, mostly fewer in numbers. These types of tests make stakeholders know about different software features, the middle-level tests are used to perform exploratory tests such as the stress test and usability test and the bottom-level tests perform general tests such as performance tests and unit tests. These kinds of tests are entirely automated.

49. How to implement security in Microservices Application.?

Security in Microservices application can be implemented either using OAuth2 or JWT. Develop a Spring Boot Application to expose a Simple REST GET API with request URL mapping. Configure Spring Security for JWT, Expose REST POST API with authentication URL mapping using which User will get a valid JSON Web Token And then allow the user access to the API with request only if it has a valid token.

50. When to use Microservice Architecture.?

Microservice Architecture can be best option When you want your monolithic application to accommodate scalability, agility, manageability and delivery speed. When you have standalone business applications or modules that have to be reused across diverse channels and some good examples would be login services, search options, authentication facilities and if you are building a highly agile application (product or service) that demands swift speed of delivery, innovation, etc.

codewitharrays.in 8007592194