

Channel Link : www.youtube.com/c/javaexpress
Website : www.javaexpresschannel.com
Contact : +91 8555929285



Spring Boot Microservices Interview Guide 2021
Spring Boot Internals Interview Questions – Part 1

1. What is a Framework ?

- Framework is a special kind of Software which provides some common logics which are required for application development in less time.

2. How many types of framework available in Market ?

- ORM Frameworks
- Web Frameworks
- Application Frameworks

3. Difference between Web application and Distributed Application ?

Web Application	Distributed Application
Web applications are meant for C 2 B (Customer to Business)	Distributed applications are meant for B 2 B
A web application is a software or program which is accessible using any web browser	A distributed application is software that is executed or run on multiple computers
To develop web applications we can use Spring MVC	To develop distributed applications we will use Webservices

4. What is Spring Framework ?

- Spring is a alternative framework for EJB Technology.
- Spring is a application development framework

Channel Link : www.youtube.com/c/javaexpress

Website : www.javaexpresschannel.com

Contact : +91 8555929285

- Using Spring we can develop all types of applications (end to end)
- The main feature of Spring is Dependency Injection & IOC Container.
- Spring is non-invasive framework.

5. Which technology is alternative for Spring ?

- EJB

6. How many types of applications we can develop in Spring ?

- Standalone applications
- Web Applications
- Distributed Applications

7. Please explain modules available in Spring Framework ?

- Spring core is the base module for all other modules
- Spring DAO/Spring JDBC module provides abstraction layer on plain JDBC to develop persistence logic.
- Spring ORM Module provides abstraction layer on other ORM flows like Hibernate & Ibatis
- Spring AOP module is given to apply aspects on Spring Applications
- Spring Web Module is given to make spring application communication from other web framework software applications like Struts, JSF e.t.c
- Spring Web-MVC is given spring own web framework develop MVC Architecture based web applications.
- Spring Context/JEE Module provides abstraction layer on multiple core technologies like EJB, RMS, JMS, Java Mail to develop Distributed applications & other applications.

8. What is Spring Boot ?

- Spring Boot is the combination of Spring framework and embeded servers
- Spring Boot is a framework for RAD build using spring framework with extra support of auto-configuration and embedded application server(like tomcat, jetty)
- It helps us in creating efficient fast stand-alone applications which you can just run it basically removes a lot of configurations and dependencies

9. Why Spring Boot ?

- To Design any spring based application, a developer needs to be taken care on manual setup on Hibernate data source, Entity manager, Session Factory and Transaction management everything.
- To Design all those common setup, a developer doesn't need to define everything individually, @SpringBootConfiguratioin annotation enough to manage everting at the time of deployment.

10. What is the latest version of Spring Boot ?

- 2.6.0 – Snapshot
- 2.3.12 - Release

11. What are the advantages of using Spring Boot ?

- Develop Spring Application Faster
- Avoid Boiler Plate code
- Convention over configuration
- Embedded Tomcat
- Actuators
- Embedded Databases

12. What is GroupId and ArtifactId while creating spring boot project ?

- **GroupId** : A unique base name of the company or group that created the project
- **ArtifactId** : A unique name of the project

13. How to create standalone application using Spring Boot ?

- Using Spring Initializer
- Using STS or Eclipse or IntelliJ Idea

14. What is Spring Boot Starter ?

- Spring Boot starters can help to reduce the number of manually added dependencies just by adding one dependency. So instead of manually specifying the dependencies just add one starter.

15. What is the use of parent pom and child pom.xml file ?

- **Maven parent POM** (or super POM) is used to structure the project to **avoid redundancies or duplicate configurations** using *inheritance between pom files*. It helps in easy maintenance in long term.

Channel Link : www.youtube.com/c/javaexpress
Website : www.javaexpresschannel.com
Contact : +91 8555929285

- Now child POM need to refer the parent POM using parent tag and specifying groupId/artifactId/version attributes

16. How to monitor Spring Boot Application ?

- Actuators
- /health – predefined endpoint to check monitor of the application

17. Which starter is required to develop web applications and distributed applications ?

- Spring-boot-starter-web

18. How many steps follow when spring boot application starts ?

- This run() method contains boot strapping logic to start spring boot application.

Start Stopwatch

Start Listeners

Prepares Environment

Create IOC Container

Print Banner

Print Application Startup Time

Call Runners

Return Context.

19. What is start class of Spring Boot ?

- Start class is the entry point for Spring Boot Application Execution.
- Start class contain main method to start program execution
- As part of Spring Boot start class main method we are calling SpringApplication.run(..) method

20. What is the importance of @SpringBootApplication ?

- This annotation is equivalent to using @Configuration, @EnableAutoConfiguration and @ComponentScan with their default attributes.
- Spring Boot enables the developer to use a single annotation instead of using multiple
- Spring provided loosely coupled features that we can use for each individual annotation as per our project needs.

21. How to override custom package using Component Scanning ?

- @ComponentScan(basePackages = {"org.javaexpress","com.javaexpress"})

Channel Link : www.youtube.com/c/javaexpress

Website : www.javaexpresschannel.com

Contact : +91 8555929285

22. Which annotation trigger auto-configuration functionalities in Spring Boot App ?

- `@EnableAutoConfiguration` annotation triggers auto-configuration functionalities for our application

23. Which annotation need to use if we create custom object in Spring Boot App ?

- `@Bean`

24. How to check predefined autoconfiguration classes in Spring Boot ?

- Spring-boot-autoconfigure jar file contain all configuration classes and we can in this path META-INF/spring.factories

25. Which annotation internally using `@SpringBootConfiguration` ?

- `@Configuration`

26. Which design pattern following while fetching IOC Container Object?

- Abstract Factory Design Pattern

27. How to represent java class as Spring Bean ?

- We can represent spring bean by using Stereotype Annotations
- `@Component`
- `@Service`
- `@Repository`

28. What is Spring Bean ?

- Any normal java class is initialized by Spring IOC container is called Spring Bean.
- Spring IOC manages the life cycle of Spring bean cycle, bean scopes and injecting any required dependencies in the bean

29. How to prove spring is providing singleton classes ?

- `Product p1 = context.getBean(Product.class)`
- `Product p2 = context.getBean(Product.class)`
- Print p1 and p2

30. How to change singleton classes to prototype and which annotation need to use ?

- `@Scope("prototype")` apply for Spring Bean classes

31. Difference between `@Configuration` and `@Bean` ?

- The class in which you want to provide the configuration about other beans, that class is called configuration class and you need to annotation with `@Configuration`.

Channel Link : www.youtube.com/c/javaexpress
Website : www.javaexpresschannel.com
Contact : +91 8555929285

- @Bean annotation is used to create custom objects by IOC Container

32. Difference between @Component and @Service ?

- This act as a more generic stereotype annotation to manage any component by spring where as the other annotations are specialized for the specific use cases
- Business layer class are annotated with @Service annotation even though the current release of the spring doesn't has any impact on using it(other than exposing that class as spring bean)

33. Can I use @Component instead of @Service ?

- Yes, good practice we need to use @Component for generic classes and utility classes.
- @Service is used for business classes.

34. Difference between @Service and @Repository ?

- @Repository : This is used for exposing a DAO class as a spring bean. It helps you in applying exception translations on these @Repository classes
- Business layer class are annotated with @Service annotation even though the current release of the spring doesn't has any impact on using it(other than exposing that class as spring bean)

35. When will get NoSuchBeanDefinitionException in Spring ?

- If we miss @Component annotation in Spring bean classes

36. What is IOC Container ?

- Giving control to the container to get instance of object is called IOC.
- IOC is a principle which is used to manage and collaborate dependencies among the objects in the application.
- IOC Container is a mechanism to achieve loose coupling between object dependencies.
- IOC will use dependency injection to manage

37. Difference between Dependency Injection and IOC Container ?

- IOC will use dependency injection to manage
- The process of injecting one class object into another class is called as Dependency Injection
- Dependency Injections is a design pattern in order to remove dependency from the programming code.

Channel Link : www.youtube.com/c/javaexpress

Website : www.javaexpresschannel.com

Contact : +91 8555929285

- In Spring Framework IOC will perform Dependency Injection

38. How many Types of Dependency Injection ?

- Setter Injection
- Constructor Injection
- Autowired

39. How to enable setter injection in Spring Boot ?

```
private Engine engine;  
  
@Autowired  
public void setEngine(Engine engine) {  
    this.engine = engine;  
}
```

40. Is it Mandatory @Autowired annotation in Constructor Injection ?

- If we have more than one constructor we have to specify @Autowired else it is optional.

41. What is Field Injection or Interface Injection and how it is working internally ?

- If we use @Autowired at variable level then IOC will perform field Injection
- When we use Field Injection, Internally IOC will use reflection api to inject dependent object into target object.
- Field Injection is not recommended because there is a chance to violate Single Responsibility Principle.

42. What are the solid principles available in java ?

- Single Responsibility principle
 - i. A class should have one and only one reason to change,, meaning that a class should have only one job
- Open Closed principle
 - i. You should be able to extend classes behaviour, without modifying it.
- LisKov Substitution principle
 - i. Derived classes must be Substitutable for their base classes
- Interface Segregation principle
- Dependency Inversion principle

43. Difference between Setter Injection and Constructor Injection ?

- **Partial dependency:** can be injected using setter injection but it is not possible by constructor. Suppose there are 3 properties in a class, having 3 arg constructor

Channel Link : www.youtube.com/c/javaexpress
Website : www.javaexpresschannel.com
Contact : +91 8555929285

and setters methods. In such case, if you want to pass information for only one property, it is possible by setter method only.

- **Overriding:** Setter injection overrides the constructor injection. If we use both constructor and setter injection, IOC container will use the setter injection.
- **Changes:** We can easily change the value by setter injection. It doesn't create a new bean instance always like constructor. So setter injection is flexible than constructor injection.

44. When will get NoUniqueBeanDefinitionException ?

- If one interface has multiple implementation classes, then if we use autowire at interface level then will get this exception.
- To overcome this issue, we need to use @Primary or @Qualifier

45. When to use @Primary for Spring bean classes ?

- If we have one interface has multiple implementation classes, then if we want to provide default configuration then will use @Primary

46. Difference between @Primary and @Qualifier ?

- Both are same but if we use @Qualifier we can customize alias names for component names.

47. Difference between Application Runner and Command Line Runner ?

- Runners are used to execute the logic only one time once application got started
- Runners will be executed only once after SpringBoot Application got started
- Above two runners are functional interfaces and contain single method – run method
- Both are same only arguments are different

48. Can we write main method in interface ?

- Yes

49. What are use cases for Application Runner or Command Line Runner ?

- Loading data to Cache
- Configuring CRON jobs
- Spring Batch framework will trigger jobs using using Runners only

50. How to provide priority for Spring Bean classes if we implement multiple Runners?

- @Order(1)

Channel Link : www.youtube.com/c/javaexpress

Website : www.javaexpresschannel.com

Contact : +91 8555929285

51. How to implement runners using Lambda Expression ?

```
ApplicationRunner fetchUserData() {  
    return runner -> {  
        System.out.println("ApplicationRunner executed");  
    };  
}
```

52. Which property need to enable to customize Spring Boot Banner mode ?

- Create banner.txt file in resources folder(default filename)
- `spring.banner.location=classpath:banner_dev.txt`

53. How to disable a specific auto-configuration class ?

- `@EnableAutoConfiguration(exclude= {DataSourceAutoConfiguration.class})`

54. How to enable debug logs in Spring Boot Application ?

- `logging.level.root=trace`

55. How to generate jar file in SpringBoot Application ?

- `java -jar applicationName`

56. How to append timestamp for SpringBoot Jar File ?

- `${project.build.finalName}-${maven.build.timestamp}`

57. How to customize maven time format while generating jar file ?

- `<maven.build.timestamp.format>yyyy-MM-dd-HH-mm</maven.build.timestamp.format>`

58. How to configure project logs in application.properties ?

- `logging.level.com.javaexpress=debug`

59. How to configure log pattern or file pattern in application.properties?

- `logging.pattern.console=%d{yyyy-MM-dd HH:mm:ss} [%thread] %-5level
%logger{36} - %msg%n`

60. How can we add external jar file in Spring Boot Project ?

- Rightclick on project -> buildPath -> configureBuildPath -> Libraries tab -> add ExternalJars

61. How to exclude module in Pom.xml file ?

```
<dependency>  
    <groupId>org.springframework.boot</groupId>  
    <artifactId>spring-boot-starter-web</artifactId>  
    <exclusions>  
        <exclusion>  
            <groupId>org.springframework.boot</groupId>  
            <artifactId>spring-boot-starter-tomcat</artifactId>  
        </exclusion>  
    </exclusions>  
</dependency>
```

Channel Link : www.youtube.com/c/javaexpress

Website : www.javaexpresschannel.com

Contact : +91 8555929285

62. Real time scenario of Prototype scope ?

- If two threads are accessing singleton object will face concurrent exception to avoid in multi-threading environment we need to use @Scope("prototype")
- It returns a new bean instance each time it is requested. It does not store any cache version like singleton. **Real world example:** declare configured form elements (a textbox configured to validate names, e-mail addresses for example) and get "living" instances of them for every form being created



Channel Link : www.youtube.com/c/javaexpress

Website : www.javaexpresschannel.com

Contact : +91 8555929285

