

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



Spring Boot Microservices Interview Guide 2021

1. Spring Boot Internals : 62 Questions
2. Spring Boot Data JPA : 32 Questions
3. Restful Webservices : 47 Questions
4. Spring MVC : 11 Questions

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

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

Website : www.javaexpresschannel.com

Contact : +91 8555929285

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
- 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 ?

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

- Spring Boot is the combination of Spring framework and embedded 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 evertng 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.
- 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 ?

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

- 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"})`

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)`

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

Website : www.javaexpresschannel.com

Contact : +91 8555929285

- 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.
- @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.

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

- 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.
- 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 then 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

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

- 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 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

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

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)`

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`

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

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

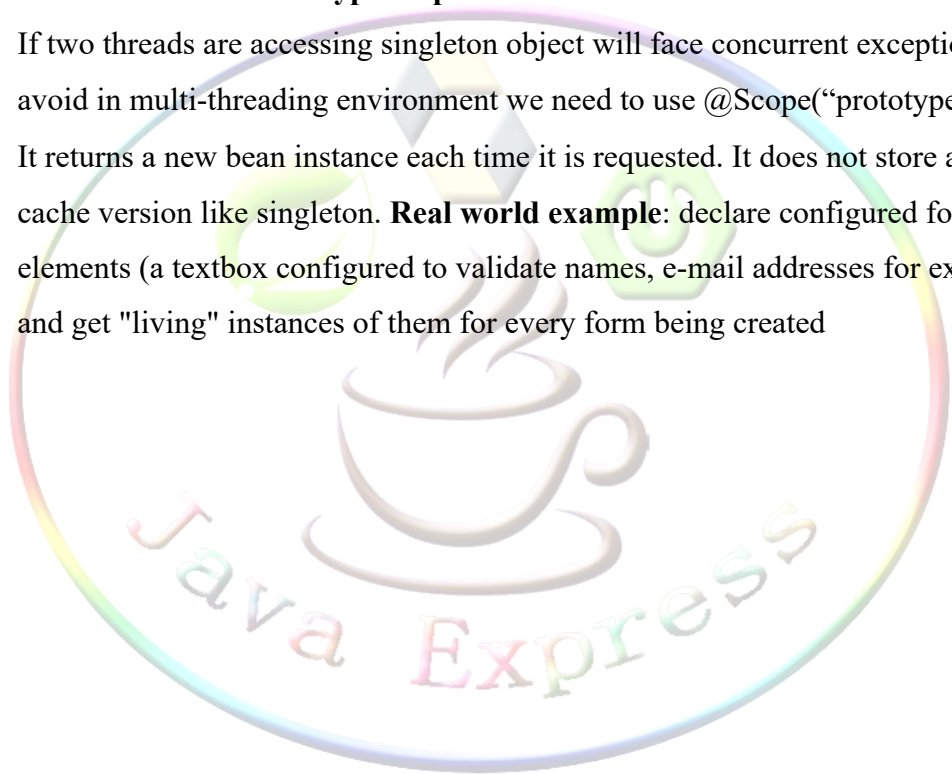
- Rightclick on project -> buildPath -> configureBuildPath -> Libraries tab ->add ExternalJars

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>
```

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

Spring Boot Data JPA Interview Questions – Part 2

1. What are the technologies available to develop persistence layer ?

- JDBC
- ORM
- Spring JDBC
- Spring ORM
- Spring Data JPA

2. What are the advantages of using ORM over JDBC ?

JDBC	Hibernate/JPA
JDBC is Persistence Technology	Hibernate is Persistence/ORM Framework
It will not represent data in the form of object.	It will not represent data in the form of object.
In JDBC user is responsible for creating and closing the connection	Hibernate Runtime system takes care of creating and closing the connections
Does not support Associations	Support Associations
Native SQL	HQL, Native SQL
In JDBC, mapping between two tables needs to write lot of codes and query. If there is change in database or table, you need to change the code and query as well.	Mapping between two tables done through XML or annotations. If there is a change in database or in any table, you need to change annotated model class.
Every line throw SQL Exception	No need to handle any exceptions

3. Which design pattern using internally for JdbcTemplate and HibernateTemplate ?

- Template Design Pattern

4. Difference between CRUD Repository vs JPA Repository ?

- Both are predefined interfaces

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

Website : www.javaexpresschannel.com

Contact : +91 8555929285

- CRUD Repository interface we can perform curd operations in our application
 - JPA Repository interface is superior to CRUD Repository ?
5. When will get access Denied exception while connecting to databases in Spring Boot Application?
- Wrong password or wrong username
6. When will get unknow database exception while connecting to database in SB?
- If schema is not available in database, then will get this exception
 - Solution : Need to create schema manually in mysql DB
7. when will get Communications link failure or Connection refused in SB ?
- Wrong URL in application.properties
8. Which properties need to enable database properties in SB ?
- `spring.datasource.url=jdbc:mysql://localhost:3306/user_dev_07`
 - `spring.datasource.username= root`
 - `spring.datasource.password= India@123`
 - `spring.jpa.hibernate.ddl-auto=create`
9. Which dependency need to enable for MySQL in SB ?
- `<dependency>`
`<groupId>mysql</groupId>`
`<artifactId>mysql-connector-java</artifactId>`
`<scope>runtime</scope>`
 - `</dependency>`
10. What are the parameters required to pass for CRUD Repository ?
- First Parameter represents Entity class Name
 - Second Parameter represents primary key datatype of entity class
 - `interface UserRepository extends CrudRepository<User, Integer> { }`
11. What are the mandatory annotations to apply in entity class ?
- `@Entity`
 - `@Id`
12. Difference between `@Entity` and `@Table` annotation ?
- The java class which represents DB table structure is called as Entity Class
 - It is used to map class name to particular table
 - If `@Table` is not mentioned then SB Data considers our class name as table name

13. What is @GeneratedValue in entity class ?

- Generators are used to generate value for primary key column when record is inserting to table.
- @GeneratedValue

14. Which property need to enable for schema generation in SB ?

- spring.jpa.hibernate.ddl-auto=**update/create/validate/create-drop**

15. How many ways can execute queries using Spring Data JPA ?

- By using predefined methods (save(), findById(), findAll())
- By writing findBy methods (findByIdByCustomerName(.), findByIdByEmail(.))
- By writing custom queries (@Query)
- Ex : "from User where userEmail=:email"

16. What are the methods available to perform curd operations and please mention below

- save(entity)
- delete(id)
- findAll()

17. What is Optional in Java ?

- It is introduced in java 8 and to order to avoid null check manually.

18. Which design pattern using internally to perform CURD operations ?

- Proxy Design pattern

19. Which property enable to display sql statements in console ?

- `spring.jpa.show-sql=true`

20. Difference between HQL vs SQL queries ?

- In HQL queries we will use entity className and entity class variable names.
 - i. Native SQL works on column names and table Name
- HQL queries can't be executed in DB directly because databases knows only Sql.
 - i. NATIVE SQL queries are DB dependent (DB specific)
- When we write HQL queries in the application, those queries should be converted into sql queries to execute.

21. How to log Sql statements in SpringBoot ?

- `spring.jpa.properties.hibernate.format_sql=true`

22. What is the use of @Param tag in Spring data JPA ?

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

- Method parameter names should match with positional parameter names then Spring Data will bind method parameter values to positional parameters names in run time to execute that query.

23. Write HQL Query using Named Parameters in Repository interfaces ?

- `@Query("select u from User u where u.userName = :name and u.emailId = :uEmail")`
- `User findByUserNameAndEmailId1(@Param("name")String username, @Param("uEmail")String emailId);`

24. Write HQL Query for – input: List of Names - output: List of Users ?

- `@Query(value="select u from User u where u.userName IN :names")`
- `List<User> findUsersByNamesList(@Param("names")Collection<String> usernames);`

25. Write HQL Query to update record in DB ?

- `@Modifying`
- `@Query("update User u set u.userName = :uName where u.emailId = :uEmail")`
- `@Transactional`
- `int updateUserEmailforName(@Param("uName")String username, @Param("uEmail")String userEmail);`

26. Write HQL Query for Positional Parameters in Repository interface ?

- `@Query("select u from User u where u.userName = ?1 and u.emailId= ?2")`

27. Which method need to use for pagination in JPA Repository ?

- `Page<User> userPagination = userRepository.findAll(PageRequest.of(0, 2, Direction.ASC, "emailId"));`

28. How to sort based on column value using JPA Repository ?

- `List<User> userList1 = userRepository.findAll(Sort.by("emailId").descending());`

29. How to read sql queries in interface using Properties file

- 1) Create META-INF Folder in src/main/resources
- 2) create a file jpa-named-queries.properties
- 3) provide key and values
- `Contact.fetchAllUsers=SELECT t FROM User t`

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

Website : www.javaexpresschannel.com

Contact : +91 8555929285

- `Contact.fetchAllUsersByEmailId=SELECT t FROM User t where t.emailId = :emailId`
- 4) Add below methods in Repository
- `List<User> fetchAllUsers();`
- `List<User> fetchAllUsersByEmailId(@Param("emailId")String emailId);`

30. Write syntax for NamedQuery in Entity class ?

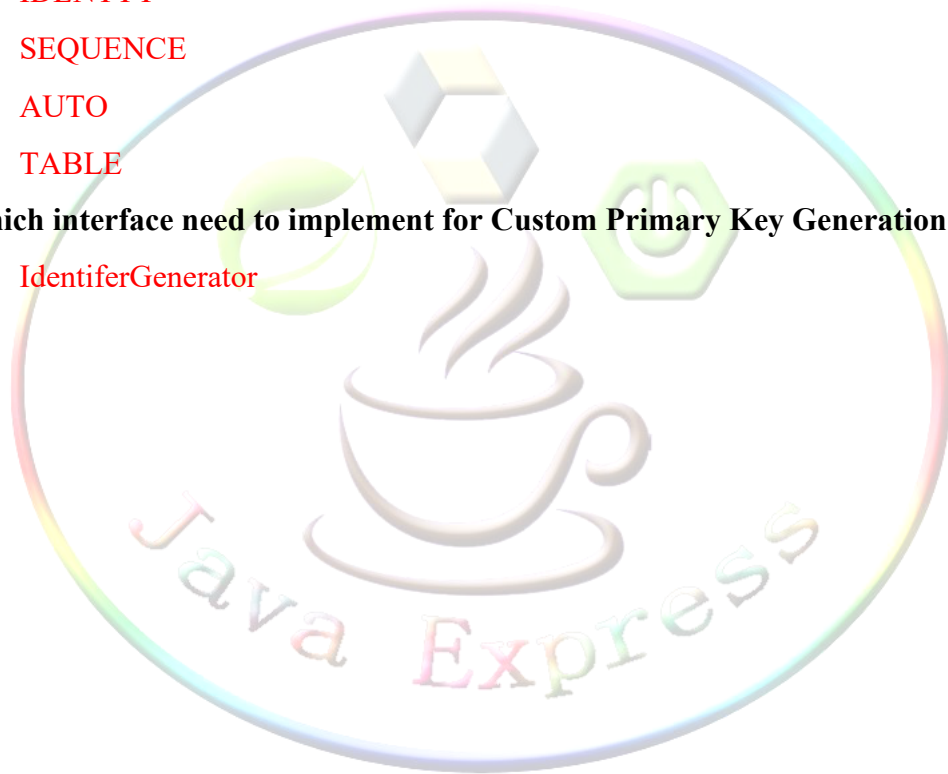
- `@NamedQuery(name="User.findUserDetailsByNameAndEmail2",
query = "select u from User u where u.userName=?1 and u.emailId=?2")`

31. What are the generators available for Primary Key Generation ?

- IDENTITY
- SEQUENCE
- AUTO
- TABLE

32. Which interface need to implement for Custom Primary Key Generation?

- `IdentifierGenerator`



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

Spring Boot Restful Interview Questions – Part 3

1. What is a web service ?
 - Webservices is a technology which is used to develop distributed applications with interoperability.
2. What is Interoperable Applications ?
 - Irrespective of platform, irrespective of programming language if two applications are communicating with each other then they are called as Interoperable applications.
3. How many ways we can develop Soap Based Webservices?
 - JAX-RPC
 - JAX-WS
4. What are the implementations available for JAX-RPC ?
 - SI
 - Apache Axis
 - WebLogic
5. What are the implementations available for JAX-WS ?
 - RI
 - Apache Axis2
 - CXF
6. What are the problems available if we use Soap Based Webservices ?
 - Adoptability issue (only soap people can access soap based webservices)

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

- Interoperability issue (only support xml)

7. What are the implementations available for JAX-RS ?

- Jersey(SUN)
- Rest Easy(JBOSS)

8. Difference between Producer vs Consumer in Soap-Based Webservices?

- Once Producer development is completed, provider will share WSDL file to consumer
 - i. Note: Email, SharePoint, repository, UDDI
- Once we got the WSDL file, we can start Consumer Development
- Once Consumer development is completed, consumer will send SOAP Xml request to provider
- Producer will process consumer request and will send response to consumer.

9. What is a Contract or what is WSDL ?

- Contract means WSDL
- WSDL -> Webservices Description Language
- WSDL is a special XML which describes how provider is providing business services to consumers

10. Difference between Contract First Approach vs Contract Last Approach ?

- Contract First Approach means WSDL file will be created first then Development will be started.
- Contract Last Approach means First we will develop the service then we will prepare WSDL for that.

11. What are the REST Principles available ?

- 1) Unique Addressability
- 2) Uniform Constraint Interfaces
- 3) Message Oriented Representation
- 4) Communication Stateless
- 5) HATEOS

12. If webserver failed to start then How to change different tomcat port number ?

- Server.port = 8097

13. How to change JDK version in Spring Boot Project ?

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

Website : www.javaexpresschannel.com

Contact : +91 8555929285

- Buildpath->configurebuildPath -> Libraries -> Add Library ->JRE
SystemLibrary

14. Difference between @RestController vs @Controller ?

- @RestController = @Controller + @ResponseBody
- @RestController = develop distributed based applications
- @Controller = develop web based applications

15. Difference between Idempotent vs Non-idempotent ?

- When we send multiple identical(same) request to server if it is producing same result then it is called as Idempotent.
 - i. Ex: GET, PUT, HEAD methods are called as idempotent.
- When we send multiple identical(same) request to server if it is producing different result then it is called as non-idempotent.
 - i. Ex: POST

16. What is HTTP Protocol ?

- To Establish communication between Client and Resource we will use HTTP protocol
- For HTTP Protocol we have 2 versions they are
 - i. 1) HTTP 1.0 (status codes support is not available)
 - ii. 2) HTTP 1.1

17. How many types of Http methods available ?

- 1) GET (read only request)
- 2) POST
- 3) PUT
- 4) DELETE
- 5) HEAD
- 6) OPTIONS
- 7) CONNECT
- 8) TRACE
- 9) PATCH

18. Difference between POST vs PUT ?

- POST method is used to send information to server in request body (File upload, HTML form) - it creates new record at server.

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

- PUT Method is used to replace all the current representation of record/resource.(current data will be updated with new data)

19. Can I use PUT instead of POST ?

- Yes, but as per standards we need to use only POST for creating new record

20. Difference between PUT vs PATCH ?

- PATCH method is used to update partial content of target resource at server.
- PUT Method is used to replace all the current representation of record/resource.(current data will be updated with new data)

21. Difference between @RequestParam vs @PathVariable ?

- **Query Parameters : - @RequestParam**
 - a. Query Parameters should present only at end of url.
 - b. Query Parameters will start with ?
 - c. Query Parameters represents data in the form of key-value pair
`http://localhost:9090/welcome?id=40`
 - d. We can send more than one query Parameter also in URL using & Separator
`http://localhost:9090/fetchProduct?id=1&name=oneplus`
- **Path Params :**
 - i. Path Params contains values directly (no keys)
 - ii. To Read Path Params we will use @PathVariable("name") in Spring MVC Controller
 - iii. Path Params can present any where in the URL.
`http://localhost:8080/product/{id}`

22. To retrieve custom headers from postman, which annotation do we need to use in controller ?

- @RequestHeader("my-number") int num

23. Difference between Rest Resource vs Rest Client ?

- Rest Resource is a distributed component which provides business services to other over a network.
- Rest Client can be a human, can be a device, can be an application and can be a tool.

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

- Anybody/anything which is capable of sending http request can be treated as Rest Client

24. Difference between Accept-Type and Content-Type ?

- When client is making request to Rest API he will send Accept and Content-Type headers in request.
- Accept header represents in which format client is expecting response data from Rest API.
 - i. Accept = application/json
- Content-Type header represents in which format client is sending input data to rest api.
 - i. Content-Type = application/xml

25. What is the syntax to support json and xml in controller http methods ?

- `@GetMapping(value="/products", produces= { "application/json" , "application/xml" })`

26. When to bind our rest controller method to PUT Request ?

- If our rest controller method is responsible for updating existing record data then we will bind that method to PUT Request method.

27. When to bind our rest controller method to Delete Request ?

- If our rest controller method is responsible for delete a record at server then we will bind that method to DELETE Request method.

28. Which HTTP Status codes to use for Every CRUD App ?

- Mostly 200,201,204 – success status codes

29. Which status codes you encounter as part of your projects and explain ?

- 200 OK - Our Request processed successfully
- 201 - Created
- 204 - No Content
- 400 - Bad Request
- 401 – Unauthorized
- 403 - Forbidden
- 405 - Method Not Allowed
- 406 - Not Acceptable
- 500 – Internal Server Error

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

30. Which library is required to perform object to json and json to object data ?

- Jackson-databind

31. Which library is required to perform object to xml and xml to object data ?

- JAX-B

32. Which tag need to enable in controller method to support json and xml for post request ?

- JSON – no need to use any annotation by default it will support SpringBoot
- @XmlRootElement – to bind our POJO class with XML

33. Which headers are required while sending post request to controller method ?

- Accept-Type = response value – application/xml
- Content-Type = input value – application/json

34. Which annotations are required to support global exception handling in Spring boot Application ?

- @ControllerAdvice
- @ExceptionHandler

35. To enable Swagger framework in Spring boot, which annotation is required ?

- @EnableSwagger2

36. Which design pattern using for Docket object in Swagger ?

- Builder Design Pattern

37. How to access swagger UI and apidocs ?

- <http://localhost:8090/swagger-ui.html>
- <http://localhost:8090/v2/api-docs>

38. What are the annotations used in Swagger Documentation ?

- Models :
 - i. @ApiModelProperty
 - ii. @ApiModelProperty
- Controllers :
 - i. @ApiOperation
 - ii. @ApiResponse

39. Which maven command (goal) need to generate jar file as below ?

- Output : Spring-boot-dev.jar
- Command Prompt : mvn package -Dspring.profiles.active=dev

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

- `<finalName>applicationName-${spring.profiles.active}</finalName>`

40. How to configure Query parameters as optional in Spring Boot Application ?

- `@RequestParam(name="pageNumber", required=false, defaultValue="10")`

41. Which command need to configure spring profiles in application.properties ?

- `spring.profiles.active=dev/prod/uat/test`

42. Difference between RestTemplate vs Web Client ?

- RestTemplate supports Synchronous communication
 - i. Web Client supports both Synchronous and Asynchronous Communication.(introduced in spring 5)
- Rest Template After Making a request it will block the thread until we receive the response)
 - i. Asynchronous communication means after making a request it won't wait for the response it will continue its execution further.

43. Is this possible to change port of Embedded Tomcat server in Spring Boot ?

- Yes , using `server.port = 8090`

44. Can we override or replace the Embedded Tomcat server in Spring Boot ?

- Yes, we can replace using different starter like `spring-boot-starter-jetty` and exclude tomcat from web starter.

```
<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>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-jetty</artifactId>
</dependency>
```

45. Can we disable the default web server in the Spring Boot Application ?

- It is providing flexibility to build you application loosely coupled.
- Yes, we can disable by using below property
 - i. `spring.main.web-application-type=none`

46. How to display current running port in console ?

- Using `@Value("${server.port}")`

47. How to use a property defined in application.properties file into your java class ?

- Using `@Value("${server.port}")`

Spring MVC Interview Questions – Part 4

1. What is Spring MVC ?

- Spring MVC is a module available in Spring framework
- Using Spring MVC we can develop MVC based web application & we can develop distributed applications
- MVC is an universal design pattern
- Using MVC design pattern we can develop loosely coupled applications
- We can develop application in layered architecture in fashion

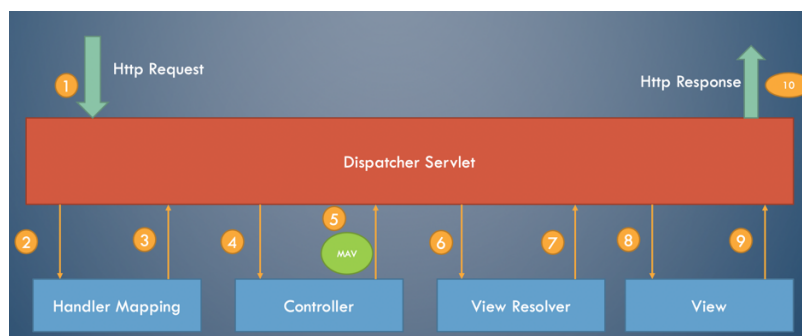
2. What are the advantages of using Spring MVC ?

- In Spring MVC, Components roles are very clearly separated.
- Spring MVC supports for multiple presentation technologies
- Spring MVC supports for Forms development
- They provided their own form tag library to simplify forms development
- Spring MVC supports form Form binding object.
- Spring MVC supports for I18N applications development
- Spring MVC supports for Interceptors to intercept every request

3. Explain Components available in Spring MVC ?

- Front Controller (DispatcherServlet)
- Handler Mapper
- Controller
- ModelAndView
- ViewResolver
- View

4. Explain Spring MVC Flow ?



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

Website : www.javaexpresschannel.com

Contact : +91 8555929285

- Incoming Http Request will be received by DispatcherServlet. DS is a predefined Servlet class in Spring MVC and it is acting as Front Controller.
- DispatcherServlet will send requested URL to Handler Mapper
- Handler Mapper will identify request handler which is responsible to handle this request and will send request handler details to DispatcherServlet.
- DispatcherServlet will call respective Controller class method
- Controller method will process request and will send ModelAndView object to DispatcherServlet
- DispatcherServlet will send view name to ViewResolver
- ViewResolver will identify view location & extension and sends data to DispatcherServlet.
- DispatcherServlet will give model and view details to View Component
- Model data will be rendered on view sends back to DispatcherServlet
- DispatcherServlet will send response back for the received request

5. Can we store one object as a value in model objects ?

- Yes, we can store.

6. Can we access object data in view file ?

- Yes, we can access object data in view file using variable name.

7. How to access model data in View File?

- Using Expression language we can access data available in model scope
- `${key}`

8. Can we return direct response from Controller method?

- Yes, using `@ResponseBody`
- When we use `@ResponseBody` annotation for controller method, then DispatcherServlet will not execute ViewResolver.

9. How many ways we can send data UI to Controller ?

- we can send data UI to Controller in below 3 ways
 - i. Query Parameters
 - ii. Path Parameters
 - iii. Forms

10. Why Thymeleaf is required in Spring Boot ?

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

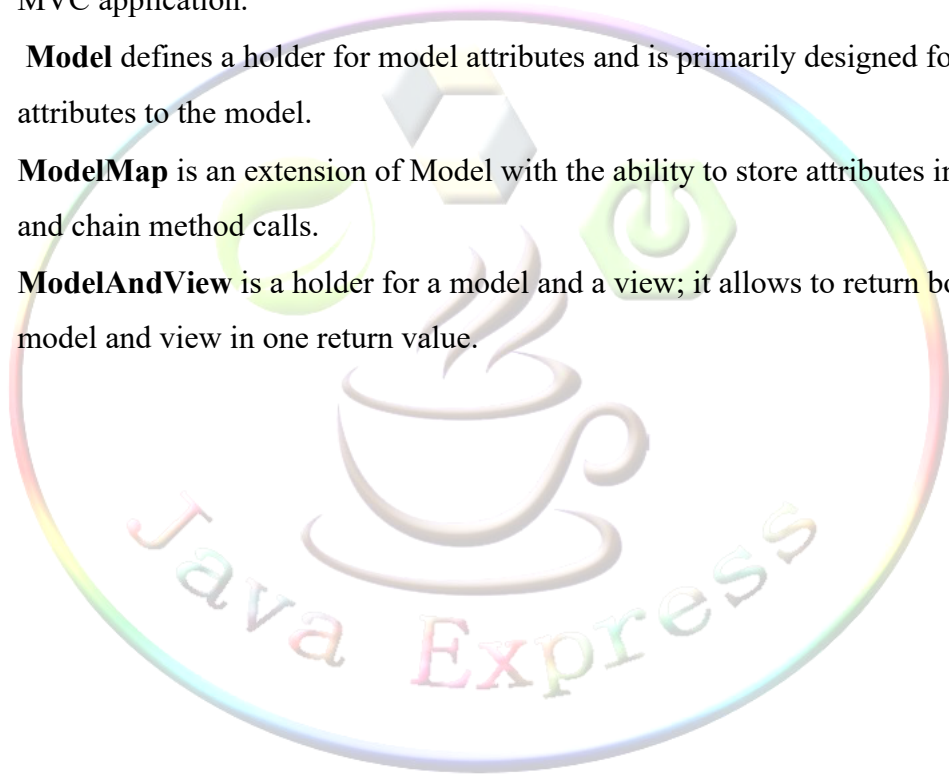
Website : www.javaexpresschannel.com

Contact : +91 8555929285

- Thymeleaf is a true template engine which takes HTML file, parse it and then produce web content on browser directly.
- Thymeleaf is more like HTML View
- Using Thymeleaf we can display dynamic data in HTML files.
- Spring Boot supports for Thymeleaf as a presentation technology
- Spring Boot provided below starter
 - i. spring-boot-starter-thymeleaf

11. Difference between Model, ModelMap & ModelAndView ?

- Model, ModelMap, and ModelAndView are used to define a model in a Spring MVC application.
- **Model** defines a holder for model attributes and is primarily designed for adding attributes to the model.
- **ModelMap** is an extension of Model with the ability to store attributes in a map and chain method calls.
- **ModelAndView** is a holder for a model and a view; it allows to return both model and view in one return value.



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

Website : www.javaexpresschannel.com

Contact : +91 8555929285

