Q1. Spring vs Spring Boot

Spring	Spring Boot
A web application framework based on Java	A module of Spring
Provides tools and libraries to create customized web applications	Used to create a Spring application project which can just run/ execute
Spring is more complex than Spring Boot	Spring Boot is less complex than the Spring framework
Takes an unopinionated view	Takes an opinionated view of a platform

Q2. What is Spring Boot and mention the need for it?

Spring Boot is a Spring module which aims to simplify the use of the Spring framework for Java development. It is used to a create stand-alone Spring-based applications which you can just run. So, it basically removes a lot of configurations and dependencies. Aiming at the Rapid Application Development, Spring Boot framework comes with the auto-dependency resolution, embedded HTTP servers, auto-configuration, management endpoints, and Spring Boot CLI.

So, if you ask me why should anybody use Spring Boot, then I would say, Spring Boot not only improves productivity but also provides a lot of conveniences to write your own business logic.

Q3. Mention the advantages of Spring Boot

The advantages of Spring Boot are as follows:

- Provides auto-configuration to load a set of default configuration for a quick start of the application
- Creates stand-alone applications with a range of non-functional features that are common to large classes of projects
- It comes with embedded tomcat, servlet containers jetty to avoid the usage of WAR files
- Spring Boot provides an opinionated view to reduce the developer effort and simplify maven configurations
- Provides CLI tool to develop and test applications
- Comes with Spring Boot starters to ensure dependency management and also provides various security metrics
- Consists of a wide range of APIs for monitoring and managing applications in dev and prod.
- Integrates with Spring Ecosystem like Spring JDBC, Spring ORM, Spring Data, Spring Security easily by avoiding boilerplate code.

Q4. Mention a few features of Spring Boot

Few important features of Spring Boot are as follows:

1. Spring CLI – Spring Boot CLI allows you to Groovy for writing Spring boot application and avoids boilerplate code.
2. Starter Dependency – With the help of this feature, Spring Boot aggregates common dependencies together and eventually improves productivity
3. Spring Initializer – This is basically a web application, which can create an internal project structure for you. So, you do not have to manually set up the structure of the project, instead, you can use this feature.
4. Auto-Configuration – The auto-configuration feature of Spring Boot helps in loading the default configurations according to the project you are working on. In this way, you can avoid any unnecessary WAR files.
5. Spring Actuator – This feature provides help while running Spring Boot applications.
6. Logging and Security – The logging and security feature of Spring Boot, ensures that all the applications made using Spring Boot are properly secured without any hassle.

Q5. Explain how to create Spring Boot application using Maven

Well, there are various approaches to create a Spring Boot application using maven, but if I have to name a few, then following are the ways to create a Spring Boot project/ application using maven:

- Spring Boot CLI
- Spring Starter Project Wizard
- Spring Initializr
- Spring Maven Project

Q6. Mention the possible sources of external configuration

There is no doubt in the fact that Spring Boot allows the developers to run the same application in different environments. Well, this is done with the support it provides for external configuration. It uses environment variables, properties files, command-line arguments, YAML files, and system properties to mention the required configuration properties. Also, the @value annotation is used to gain access to the properties. So, the most possible sources of external configuration are as follows:

- **Application Properties –** By default, Spring Boot searches for the application properties file or its YAML file in the current directory, classpath root or config directory to load the properties.
- **Command-line properties –** Spring Boot provides command-line arguments and converts these arguments to properties. Then it adds them to the set of environment properties.
- **Profile-specific properties –** These properties are loaded from the application-{profile}.properties file or its YAML file. This file resides in the same location as that of the non-specific property files and the{profile} placeholder refers to an active profile.

Q7. Can you explain what happens in the background when a Spring Boot Application is "Run as Java Application"?

When a Spring Boot application is executed as "Run as Java application", then it automatically launches up the tomcat server as soon as it sees, that you are developing a web application.

Q8. What are the Spring Boot starters and what are available the starters?

Spring Boot starters are a set of convenient dependency management providers which can be used in the application to enable dependencies. These starters, make development easy and rapid. All the available starters come under the org.springframework.boot group. Few of the popular starters are as follows:

- *spring-boot-starter:* – This is the core starter and includes logging, auto-configuration support, and YAML.
- *spring-boot-starter-jdbc* – This starter is used for HikariCP connection pool with JDBC
- *spring-boot-starter-web* – Is the starter for building web applications, including RESTful, applications using Spring MVC
- *spring-boot-starter-data-jpa* – Is the starter to use Spring Data JPA with Hibernate
- *spring-boot-starter-security* – Is the starter used for Spring Security
- *spring-boot-starter-aop:* This starter is used for aspect-oriented programming with AspectJ and Spring AOP
- *spring-boot-starter-test:* Is the starter for testing Spring Boot applications

Q9. Explain Spring Actuator and its advantages

Spring Actuator is a cool feature of Spring Boot with the help of which you can see what is happening inside a running application. So, whenever you want to debug your application, and need to analyze the logs you need to understand what is happening in the application right? In such a scenario, the Spring Actuator provides easy access to features such as identifying beans, CPU usage, etc. The Spring Actuator provides a very easy way to access the production-ready REST points and fetch all kinds of information from the web. These points are secured using Spring Security's content negotiation strategy.

Q10. What is Spring Boot dependency management?

Spring Boot dependency management is basically used to manage dependencies and configuration automatically without you specifying the version for any of that dependencies.

Q11. Mention the minimum requirements for a Spring boot System

Spring Boot 2.1.7.RELEASE requires

- Java 8 +
- Spring Framework 5.1.9 +

Explicit build support

- Maven 3.3+
- Gradle 4.4+

Servlet Container Support

- Tomcat 9.0 – Servlet Version 4.0
- Jetty 9.4 – Servlet Version 3.1
- Undertow 2.0 – Servlet Version 4.0

Q12. Explain what is thymeleaf and how to use thymeleaf?

Spring Framework Certification Training

Weekday / Weekend Ba

Thymeleaf is a server-side Java template engine used for web applications. It aims to bring natural template for your web application and can integrate well with Spring Framework and HTML5 Java web applications. To use Thymeleaf, you need to add the following code in the pom.xml file:

```
1    <dependency>
2    <groupId>org.springframework.boot</groupId>
3    <artifactId>spring-boot-starter-thymeleaf</artifactId>
4    </dependency>
```

Q13. Can we change the port of the embedded Tomcat server in Spring boot?

Yes, we can change the port of the embedded tomcat server by using the application properties file. In this file, you have to add a property of “server.port” and assign it to any port you wish to. For example, if you want to assign it to 8081, then you have to mention server.port=8081. Once you mention the port number, the application properties file will be automatically loaded by Spring Boot and the required configurations will be applied on to the application.

Q14. What is the need for Spring Boot DevTools?

Spring Boot Dev Tools are an elaborated set of tools and aims to make the process of developing an application easier. If the application runs in the production, then this module is automatically disabled, repackaging of archives are also excluded by default. So, the Spring Boot Developer Tools applies properties to the respective development environments. To include the DevTools, you just have to add the following dependency into the pom.xml file:

```
1    <dependency>
2    <groupId>org.springframework.boot</groupId>
```

```
<artifactId>spring-boot-devtools</artifactId>
</dependency>
```

Q15. Mention the steps to create a Spring Boot project using Spring Initializr

Spring Initializr is a web tool provided by Spring. With the help of this tool, you can create Spring Boot projects by just providing project details. The following steps need to be followed to create a Spring Boot project using Spring Initializr:

- Choose the maven project and the required dependencies. Then, fill the other required details like Group, Artifact, and then click on Generate Project.
- Once the project is downloaded, extract the project onto your system
- Next, you have to import this project using the import option on the Spring Tool Suite IDE
  - While importing the project, remember that you have to choose the project type to be Maven and the source project should contain the pom.xml file.

Once, all the above steps are followed you will see that the Spring Boot project is created with all the required dependencies.

Q16. Mention the steps to connect Spring Boot application to a database using JDBC

Spring Boot starter projects provide the required libraries to connect the application with JDBC. So, for example, if you just have to create an application and connect it with MySQL database, you can follow the below steps:

Step 1: Create a database in MySQL

```
CREATE DATABASE example;
```

Step 2: Then you have to create a table inside this database.

```
CREATE TABLE customers(customerid INT PRIMARY KEY NOT NULL AUTO_INCREMENT, customername VARCHAR(255));
```

Step 3: Now, create a Spring Boot project and provide the required details

Step 4: Add the JDBC, MySQL and web dependencies.

Step 5: Once the project is created, you have to configure the database into application properties

```
spring.datasource.url=jdbc:mysql://localhost:3306/example
spring.datasource.username=root
spring.datasource.password=edureka
spring.jpa.hibernate.ddl-auto=create-drop
```

Step 6: The main application.java class should have the following code:

```
package com.edureka;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
@SpringBootApplication
public class SampleApplication {
public static void main(String[] args) {
SpringApplication.run(SampleApplication.class, args);
}
}
```

Step 7: Next, you have to create a controller to handle the HTTP requests, by mentioning the following code:

```
package com.edureka;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.jdbc.core.JdbcTemplate;
```

```
5   import org.springframework.web.bind.annotation.RestController;
6   @RestController
7   public class JdbcController {
8   @Autowired
9   JdbcTemplate jdbc;
10  @RequestMapping("/insert")
11  public String index(){
12  jdbc.execute("insert into customers(name)values('Aryya')");
13  return "Data Entry Successful";
14  }
15  }
```

Step 8: Finally, execute this project as a Java application.
Step 9: Next, open the URL (localhost:8080/insert), and you will see the output as Data Entry Successful. You can also go forward and check if the data is entered into the table.

Q17. How to enable HTTP/2 support in Spring Boot

You can enable the HTTP/2 support in Spring Boot by: server.http2.enabled=true

Q18. What are the @RequestMapping and @RestController annotation in Spring Boot used for?

@RequestMapping	@RestController
This annotation is used to provide the routing information and tells to Spring that any HTTP request must be mapped to the respective method.	This annotation is used to add the @ResponseBody and @Controller annotation to the class
To use this annotation, you have to import org.springframework.web. bind.annotation.RequestMapping;	To use this annotation, you have to import org.springframework.web. bind.annotation.RestController;

Example: Consider you have a method example() which should map with /example URL.

```
1   package com.edureka;
2   import org.springframework.web.bind.annotation.RequestMapping;
3   import org.springframework.web.bind.annotation.RestController;
4   @RestController
5   public class SampleController {
6   @RequestMapping("/example")
7   public String example(){
8   return"Welcome To Edureka";
9   }
10  }
```

Q19. What is Spring Boot CLI and how to execute the Spring Boot project using boot CLI?

Spring Boot CLI is a tool supported by the official Spring Framework. The steps to execute a Spring Boot project are as follows:

- Download the CLI tool from the official site and extract the zip file. The bin folder present in the Spring setup is used to execute the Spring Boot application.
- Since Spring Boot CLI executes groovy files, you need to create a groovy file for Spring Boot application. So, to do that, open terminal and change the current directory to the bin folder. Now, open a groovy file (for example Sample.groovy)
- In this file create a controller as follows:

```
@RestController   public class Sample {
 @RequestMapping("/example")
 String index(){
<h1>"Welcome To Edureka"</h1>;
}  }
```

- Then execute the groovy file by mentioning:

```
1  ./spring run Sample.groovy;
```

Once, the project is executed go to the URL(localhost:8080:/example) and you will see the output as **Welcome To Edureka**

Q20. Mention the differences between JPA and Hibernate

JPA	Hibernate
JPA is a Data Access Abstraction used to reduce the amount of boilerplate code	Hibernate is an implementation of Java Persistence API and offers benefits of loose coupling

Q21. How can we create a custom endpoint in Spring Boot Actuator?

To create a custom endpoint in Spring Boot 2.x, you can use the @Endpoint annotation. Spring Boot also exposes endpoints using @WebEndpointor, @WebEndpointExtension over HTTP with the help of Spring MVC, Jersey, etc.

Q22. Explain Spring Data

Spring Data aims to make it easy for the developers to use relational and non-relational databases, cloud-based data services, and other data access technologies. So, basically, it makes it easy for data access and still retains the underlying data.

Q23. What do you understand by auto-configuration in Spring Boot and how to disable the auto-configuration?

Auto-configuration is used to automatically configure the required configuration for the application. For example, if you have a data source bean present in the classpath of the application, then it automatically configures the JDBC template. With the help of auto-configuration, you can create a Java application in an easy way, as it automatically configures the required beans, controllers, etc.

To disable the auto-configuration property, you have to exclude attribute of @EnableAutoConfiguration, in the scenario where you do not want it to be applied.

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

If the class is not on the classpath, then to exclude the auto-configuration, you have to mention the following code:

```
1          @EnableAutoConfiguration(excludeName={Sample.class})
```

Apart from this, Spring Boot also provides the facility to exclude list of auto-configuration classes by using the spring.autoconfigure.exclude property. You can go forward, and add it either in the application.properties or add multiple classes with comma-separated.

Q24. What are the differences between @SpringBootApplication and @EnableAutoConfiguration annotation?

@SpringBootApplication	@EnableAutoConfiguration
Used in the main class or bootstrap class	Used to enable auto-configuration and component scanning in your project
It is a combination of @Configuration, @ComponentScan and @EnableAutoConfiguration annotations.	It is a combination of @Configuration and @ComponentScan annotations

Q25. What are the steps to deploy Spring Boot web applications as JAR and WAR files?

To deploy a Spring Boot web application, you just have to add the following plugin in the pom.xml file:

```
<plugin>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-maven-plugin</artifactId>
</plugin>
```

By using the above plugin, you will get a JAR executing the package phase. This JAR will contain all the necessary libraries and dependencies required. It will also contain an embedded server. So, you can basically run the application like an ordinary JAR file.
Note: The packaging element in the pom.xml file must be set to **jar** to build a JAR file as below:

```
<packaging>jar</packaging>
```

Similarly, if you want to build a WAR file, then you will mention

```
<packaging>war</packaging>
```

Q.26 Can you give an example for ReadOnly as true in Transaction management?

Example for ReadOnly as TRUE in transaction management could be as follows:

Consider a scenario, where you have to read data from the database. For example, let us say you have a customer database, and you want to read the customer details such as customerID, and customername. To do that, you will set **read-only on the transaction** as we do not want to check for the changes in the entities.

Q27. Can you explain how to deploy to a different server with Spring Boot?

To deploy a different server with Spring Boot, follow the below steps:

- Generate a WAR from the project
- Then, deploy the WAR file onto your favorite server

Note: The steps to deploy the WAR file on the server is dependent on the server you choose.

Q28: What is the best way to expose custom application configuration with Spring Boot?

One way to expose the custom application configuration in Spring Boot is by using the **@Value annotation**. But, the only problem with this annotation is that all the configuration values will be distributed throughout the application. Instead, you can use a centralized approach.

By centralized approach, I mean that you can define a configuration component using the @ConfigurationProperties as follows:

```
@Component
@ConfigurationProperties("example")
public class SampleConfiguration {
private int number;
private boolean value;
private String message;
```

According to the above snippet, the values configured in application.properties will be as follows:

```
example.number: 100
example.value: true
example.message: Dynamic Message
```

Q29. Can we create a non-web application in Spring Boot?

Yes, we can create a non-web application by removing the web dependencies from the classpath along with changing the way Spring Boot creates the application context.

Q 30. What are the steps to connect an external database like MySQL or Oracle?

To connect an external database, you have to follow the below steps:

- Start by adding the dependency for MySQL Connector to pom.xml
- Then remove H2 Dependency from pom.xml
- Now, set up your MySQL database and configure your connection to the MySQL database
- Restart your project

Q31. Mention the advantages of the YAML file than Properties file and the different ways to load YAML file in Spring boot

The advantages of the YAML file than a properties file is that the data is stored in a hierarchical format. So, it becomes very easy for the developers to debug if there is an issue. The SpringApplication class supports the YAML file as an alternative to properties whenever you use the SnakeYAML library on your classpath. The different ways to load a YAML file in Spring Boot is as follows:

- Use YamlMapFactoryBean to load YAML as a Map
- Use YamlPropertiesFactoryBean to load YAML as Properties

Q32. How is Hibernate chosen as the default implementation for JPA without any configuration?

When we use the Spring Boot Auto Configuration, automatically the spring**-boot-starter-data-jpa** dependency gets added to the pom.xml file. Now, since this dependency has a transitive dependency on JPA and Hibernate, Spring Boot automatically auto-configures Hibernate as the default implementation for JPA, whenever it sees Hibernate in the classpath.

Q33. What do you understand by Spring Data REST?

Spring Data REST is used to expose the RESTful resources around Spring Data repositories. Consider the following example:

```
1    @RepositoryRestResource(collectionResourceRel = "sample", path = "sample")
2                public interface SampleRepository
3              extends CustomerRepository<sample, Long> {
```

Now, to expose the REST services, you can use the POST method in the following way:

```
1                {
2              "customername": "Rohit"
3                }
```

Response Content

```
1                {
2              "customername": "Rohit"
3                "_links": {
4                 "self": {
5            "href": "http://localhost:8080/sample/1"
6                },
7               "sample": {
8            "href": "http://localhost:8080/sample/1"
9                }
10               }
```

Observe that the response content contains the href of the newly created resource.

Q34. What is the difference between RequestMapping and GetMapping?

The @GetMapping is a composed annotation that acts as a shortcut for @RequestMapping(method = RequestMethod.GET). Both these methods support the consumes. The consume options are :

consumes = “text/plain”
consumes = {“text/plain”, “application/*”}

Q35: In which layer, should the boundary of a transaction start?

The boundary of the transaction should start from the Service Layer since the logic for the business transaction is present in this layer itself.

Q36. How does path=”sample”, collectionResourceRel=”sample” work with Spring Data Rest?

```
1  @RepositoryRestResource(collectionResourceRel = "sample", path = "sample")
2  public interface SampleRepository extends
3  PagingAndSortingRepository<Sample, Long>
```

- path – This section is used to mention the segment under which the resource is to be exported.
- collectionResourceRel – This value is used to generate links to the collection resource.

Q37. Explain how to register a custom auto-configuration.

In order to register an auto-configuration class, you have to mention the fully-qualified name under the @EnableAutoConfiguration key META-INF/spring. factories file. Also, if we build the with maven, then this file should be placed in the resources/META-INT directory.

Q38. How do you Configure Log4j for logging?

Since Spring Boot supports Log4j2 for logging a configuration, you have to exclude Logback and include Log4j2 for logging. This can be only done if you are using the starters project.

Q39. Mention the differences between WAR and embedded containers

WAR	Embedded Containers
WAR benefits a considerable measure from Spring Boot	Only one component of Spring Boot and is utilized during improvements

Q40. What do you think is the need for Profiles?

Profiles are used to provide a way to segregate the different parts of the application configuration and make it available for various environments. So, basically, any @Component or a @Configuration can be marked with a @Profile to limit as it is loaded. Consider you have multiple environments,

- Dev
- QA
- Stage
- Production

Now, let’s say, you want to have different application configuration in each of the environments, you can use profiles to have different application configurations for different environments. So, basically, Spring and Spring Boot provide features through which you can specify:

- The active profile for a specific environment
- The configuration of various environments for various profiles.

Q41. What are the steps to add a custom JS code with Spring Boot?

The steps to add a custom JS code with Spring Boot are as follows:

- Now, create a folder and name it **static** under the resources folder
- In this folder, you can put the static content in that folder

Note: Just in case, the browser throws an unauthorized error, you either disable the security or search for the password in the log file, and eventually pass it in the request header.

Q42. How to instruct an auto-configuration to back off when a bean exists?

To instruct an auto-configuration class to back off when a bean exists, you have to use the @ConditionalOnMissingBean annotation. The attributes of this annotation are as follows:

- **value:** This attribute stores the type of beans to be checked
- **name:** This attribute stores the name of beans to be checked

Q43. Why is Spring Data REST not recommended in real-world applications?

Spring Data REST is not recommended in real-world applications as you are exposing your database entities directly as REST Services. While designing RESTful services, the two most important things that we consider is the domain model and the consumers. But, while using Spring Data REST, none of these parameters are considered. The entities are directly exposed. So, I would just say, you can use Spring Data REST, for the initial evolution of the project.

Q44. What is the error you see if H2 is not in the classpath?

If H2 is not present in the classpath, then you see the following error:

Cannot determine embedded database driver class for database type NONE

To resolve this error, add H2 to the pom.xml file, and restart your server.
The following code snippet can be added to add the dependency:

```
1    <dependency>
2    <groupId>com.h2database</groupId>
3    <artifactId>h2</artifactId>
4    <scope>runtime</scope>
5    </dependency>
```

Q45. What is the way to use profiles to configure the environment-specific configuration with Spring Boot?

Since it is a known fact that a Profile is nothing but a key to identify an environment lets consider the following two profiles in the example:

- dev
- prod
- Consider the following properties present in the application properties file:

example.number: 100
example.value: true
example.message: Dynamic Message

Now, say you want to customize the application.properties for dev profile, then you need to create a file with name application-dev.properties and override the properties that you want to customize. You can mention the following code:

example.message: Dynamic Message in Dev
Similarly, if you want to customize the application.properties for prod profile, then you can mention the following code snippet:

example.message: Dynamic Message in Prod
Once you are done with the profile-specific configuration, you have to set the active profile in an environment. To do that, either you can

- Use -Dspring.profiles.active=prod in arguments
- Use spring.profiles.active=prod in application.properties file

Q46. Mention the dependencies needed to start up a JPA Application and connect to in-memory database H2 with Spring Boot?

The dependencies are needed to start up a JPA Application and connect to in-memory database H2 with Spring Boot

- web starter
- h2
- data JPA starter
- To include the dependencies refer to the following code:

```
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
<groupId>com.h2database</groupId>
<artifactId>h2</artifactId>
<scope>runtime</scope>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
```

Q47. What do you understand by Spring Boot supports relaxed binding?

Relaxed binding, is a way in which, the property name does not need to match the key of the environment property. In Spring Boot, relaxed binding is applicable to the type-safe binding of the configuration properties. For example, if a property in a bean class with the @ConfigurationPropertie annotation is used sampleProp, then it can be bounded to any of the following environment properties:

- sampleProp
- sample-Prop
- sample_Prop
- SAMPLE_PROP

Q48. Where is the database connection information specified and how does it automatically connect to H2?

Well, the answer to this question is very simple. It is because of the Spring Boot auto-configuration that, configures the dependencies of the application. So, the database connection information, and automatically connecting the database to H2 is done by the auto-configuration property.

Q49. What is the name of the default H2 database configured by Spring Boot?

The name of the default H2 database is **testdb. Refer below:**

spring.datasource.name=testdb # Name of the datasource.
Note: Just incase if you are using H2 in-memory database, then exactly that is the name of Spring Boot which is used to setup your H2 database.

Q50. Do you think, you can use jetty instead of tomcat in spring-boot-starter-web?

Yes, we can use jetty instead of tomcat in spring-boot-starter-web, by removing the existing dependency and including the following:

```
<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>
```

What is Spring boot?

Spring boot is java-based framework for creating Web services. Whether you are interviewing for the first job or for the job, which you going to switch, you can always feel that nervousness and tension. The developers are always thinking about codes and especially when there is an interview, they can feel they are going nuts over it. In recent time, demand for spring boot developers has increased a lot as spring boot offers quick application advancement structure to the spring system.

Prepare enough by having different ideas of spring boot including various features, venture, maven project, starter venture wizard, application, annotations, dm, properties, starters, and actuator. There are various spring boot jobs in the market, however, to be chosen in that specific post, you must clear the interview. In the current time, there is exceptionally outrageous competition in the market when it comes to interviewing and getting a break in MNC.

Also, Read: Spring Interview Questions

We have provided you some of the questions, which are normally asked in Spring boot tough interview and which will give you a head start. You never know which of these questions you may get in your spring boot interview:

Q1. What does Spring Boot mean?

Spring Boot is a system from "The Spring Team" to facilitate the bootstrapping and development of new Spring Applications. It gives defaults to code, and annotation configurations to snappy begin new spring projects at no time. It takes after the "Opinionated Defaults Configuration" Approach to escape from a lot of standard code and configuration to enhance Development, Unit Test, and Integration Test Process.

Q2. What are the various Advantages Of Using Spring Boot?

Here are some of the various advantages of using Spring Boot:

It is quite easy to create Spring Based applications with Java or Groovy. It lessens lots of improvement time and expands profitability. It abstains from writing lots of standard Codes, Annotations, and XML Configuration. It is quite easy to coordinate Spring Boot Application with its Spring Ecosystem like Spring JDBC, Spring ORM, Spring Data, Spring Security and so forth. It takes after the "Opinionated Defaults Configuration" Approach to diminish Developer effort It gives Embedded HTTP servers like Tomcat, Jetty and more to create and test our web applications effectively. It gives CLI (Command Line Interface) tool to create and test Spring Boot (Java or Groovy) Applications from commanding prompt very easily and rapidly.

It gives lots of modules to create and test Spring Boot Applications effectively utilizing Build Tools like Maven and Gradle It provides loads of plug-ins to work with implanted and in-memory Databases effortlessly.

Q3. What are the various features of Spring Boot?

Various Spring Boot Features are as follows:

Web Development Spring Application Application occasions and listeners Admin highlights Externalized Configuration Properties Files YAML Support Type-safe Configuration Logging Security

Q4. What is the reason to have a spring-boot-maven module?

The reason behind to have Spring-boot-maven module is it gives a couple of charges which empower you to package the code as a container or run the application

spring-boot: run operates your Spring Boot application. spring-boot: repackage it repackages your jug/war to be executable. spring-boot: start and spring-boot: stop to deal with the lifecycle of your Spring Boot application (i.e., for joining tests). spring-boot: build-data creates build data that can be utilized by the Actuator.

Q5. How to make Spring Boot venture utilizing Spring Initializer?

The Spring Initializer is a web application that can produce a Spring Boot project structure for you. It doesn't create any application code. However, it will give you an essential project structure and either a Maven or a Gradle build specification to fabricate your code with. You should simply compose the application code.

Spring Initializer can be utilized a few different ways, including:

An online interface. Via Spring Tool Suite. Using the Spring Boot CLI.

Q6. What do Dev Tools in Spring boot mean?

Spring boot accompanies Dev Tools, which is acquainted with increase the profitability of designer. You don't have to redeploy your application each time you influence the changes. The developer can reload the progressions without restart of the server. It maintains a strategic distance from the agony of redeploying application each time when you roll out any improvement. This module will can't be utilized in a production environment.

Q7. What does Spring Boot Starter Pom mean? Why Is It Useful?

Starters are an arrangement of advantageous reliance descriptors that you can incorporate into your application. The starters contain a considerable amount of the dependencies that you have to get a task up and running rapidly and with a steady, supported a set of managed transitive conditions.

The starter POMs are helpful reliance descriptors that can be added to your application's Maven. In another word, if you are building up a project that utilizes Spring Batch for batch preparing, you need to incorporate spring-boot-starter-bunch that will import all the required conditions for the Spring Batch application. This decreases the burden of looking at and designing all of the conditions required for a structure.

Q8. What does Actuator in Spring Boot mean?

Spring Boot Actuator is a sub-task of Spring Boot. It adds a few creation review administrations to your application with little exertion on your part. There are also has numerous features added to your application outof-the-case for dealing with the administration in a production (or other) condition. They're basically used to uncover diverse kinds of data about the running application – health, measurements, information, dump, env and so on.

Q9. What Is the Configuration File Name Used By Spring Boot?

The configuration record utilized as a part of spring boot ventures is an application. Properties. This record is imperative where we would overwrite all the default designs. Regularly we need to hold this document under the assets envelope of the project.

Q10. Why in spring boot "Opinionated " is used?

It takes after "Opinionated Defaults Configuration" Approach to lessen Developer exertion. Because of the Opinionated perspective of spring boot, what is required to begin yet additionally we can get out if not appropriate for the application. Spring Boot utilizes sensible defaults, "opinions," for the most part in light of the classpath substance.

Q11. What are esteem properties of Spring Boot?

Spring Boot gives different properties, which can be indicated in our project's application. Properties record. These properties have default values, and you can set that inside the properties record. Properties are utilized to set qualities like a server-port number, database association configuration and much more.

Q12. What Is the Configuration File Name, which is used By Spring Boot?

The configuration file name, which is utilized as a part of spring boot projects is application.properties. This document is very important where we would overwrite all the default setups. Ordinarily, we need to hold this document under the assets folder of the project.

Q13. Would we be able to Use Spring Boot with Applications Which Are Not Using Spring?

No, it isn't conceivable starting at now. Spring boot is restricted to Spring applications only.

Q14. What Is Name Of The Configuration File, Which You Use In Spring Boot?

Configuration file name which is utilized as a part of Spring boot ventures is known as application. Properties. It is vital to document as it is utilized to abrogate all default configurations.

Q15. How Might You Implement Spring Security In Spring Boot Application?

Usage of spring security in Spring boot application requires quite a little configuration. You have to include spring-boot-starter-security starter in pom.xml. You need to make spring config class, which will expand WebSecurity Configure Adapter and override expected strategy to accomplish security in Spring boot application.

Q16. Would you be able to Control Logging with Spring Boot? How?

Yes, we can control logging with spring boot.

Q17. Differentiate Between An Embedded Container And A War?

There is no force to go containerless

The embedded container is only one component of Spring Boot Traditional WAR additionally benefits a considerable measure from Spring Boot Automatic Spring MVC setup, including Dispatcher Servlet Sensible defaults in light of the class-path content The embedded container can be utilized during improvement.

Q18. What does Spring Security mean?

Spring Security is a groundbreaking and very adjustable authentication and access-control structure. It is the true standard for securing Spring-based applications. Spring Security is a system that spotlights on giving both authentication and approval to Java applications. Like all spring ventures, the genuine power of Spring Security is found in how effectively it can be reached out to meet custom prerequisites.

Q19. What does Aspect-Oriented Programming (AOP) mean?

Aspect Oriented Programming (AOP) supplements Object-Oriented Programming (OOP) by giving another mindset about program structure. The key unit of measured quality in OOP is the class, while in AOP the unit of particularity is the viewpoint. Angles empower the modularization of concerns, for example, transaction management that cut over numerous sorts and questions.

Q20. Describe some of the spring sub-projects briefly?

Various spring sub-projects are as follows:

JDBC: this module empowers a JDBC-deliberation layer that evaluates the need to do JDBC coding for particular vendor databases Read Best JDBC Interview Questions Core: a key module that gives basic parts of the system, as IoC or DI Web: a web-situated joining module, giving multipart document upload, listeners members, and webarranged application context functionalities ORM integration: gives mix layers to well-known protest object-relational mapping APIs, for example, JPA, JDO, and Hibernate AOP module: perspective oriented programming execution is permitting the meaning of clean strategy interceptors and pointcuts.

MVC system: a web module executing the Model View Controller configuration design

Q21. Explain the difference between JPA and Hibernate?

JPA is a specification/Interface whereas Hibernate is one of the JPA implementations.

Q22. How to connect to an external database like MSSQL or oracle with Spring boot?

It is done in the following steps.

Step 1

The first step to connect the database like Oracle or MySql is adding the dependency for your database connector to pom.xml.

Step 2

The next step is the elimination of H2 Dependency from pom.xml

Step 3

Step 3 includes the schema and table to establish your database.

Step 4

The next step is configuring of the database by using Configure application.properties to connect to your database.

Step 5

And the last step is to restart your device and your connection is ready to use.

Q23. How to add custom JS code in Spring Boot?

/src/main/resources/static is the suggested folder for static content in Spring boot.

You can create a JS file for sending an alert by creating a custom file named custom.js in /src/main/resources/static/js/ directory with below code

alert("I'm active");

Q24. List minimum requirements for Spring boot System?

Spring Boot 1.5.10. RELEASE requires

Java 7 + Spring 4.3.13 +

For build support

Maven 3.2+ Gradle 2.9+

Container Support

Tomcat 7+ Jetty 8+ (Jetty 9.3 requires JDK 8 +)

Q25. What is Auto Configuration in Spring boot?

Autoconfiguration is way in Spring Boot to configure a spring application automatically on the basis of dependencies that are present on the classpath. It makes development easier and faster.

You can create a custom configuration for a MySQL data source in spring boot as

@Configuration public class MySQLAutoconfiguration {

//... }

Q26. How can you enable auto reload of application with Spring Boot?

You can enable auto-reload/LiveReload of spring boot application by adding the spring-boot-devtools dependency in the pom.xml file.

<dependency> <groupId>org.springframework.boot</groupId> <artifactId>spring-boot-devtools</artifactId> <optional>true </dependency>

Note: please restart your application for immediate effects.

Q27. How to enable HTTP/2 support in Spring Boot?

You can enable HTTP/2 support in Spring Boot as follows: server.http2.enabled=true

Example:

@Bean public ConfigurableServletWebServerFactory tomcatCustomizer() { TomcatServletWebServerFactory factory = new TomcatServletWebServerFactory(); factory.addConnectorCustomizers(connector -> connector.addUpgradeProtocol(new Http2Protocol())); return factory; }

Q28. How do you Enable HTTP response compression in spring boot?

To enable HTTP response compression in spring boot using GZIP you have to add below settings in your application.properties file.

Enabling HTTP response compression in spring boot server.compression.enabled=true server.compression.min-response-size=2048 server.compression.mime-types=application/json,application/xml,text/html,text/xml,text/plain

Q29. What is the difference between RequestMapping and GetMapping in Spring Boot?

Both @GetMapping and @RequestMapping are annotations for mapping HTTP GET requests onto specific handler methods in Spring boot. @GetMapping is a composed annotation that acts as a shortcut for @RequestMapping. @GetMapping is the newer annotation.

Q30. What are Actuator in Spring Boot?

Actuator is a tool in Spring Boot for monitoring and managing our application. Actuator Monitors our app, gathers metrics, understands traffic or the state of our database. It uses HTTP endpoints or JMX beans to enable us to interact with it. An actuator is available to use from the first release of Spring Boot.

Here is a youtube video by Java Brains to understand Spring Boot Actuator

Q31. What is the use of @SpringBootApplication annotation?

@SpringBootApplication annotation was introduced in Spring Boot version 1.2.0. It is a convenience annotation which is used in spring boot application to enable additions of beans using the classpath definitions.

Q32. How do you configure error logging/debugging in Spring boot application?

You can configure error logging/debugging in Spring boot application by applying the following settings in application.properties or application.yml file.

logging.level.org.springframework.web: DEBUG logging.level.org.hibernate: ERROR

Q33. What is the Spring Boot Initilizr?

Spring Boot Initilizr is a web interface which to rapidly create spring boot projects. Using this tool you can create Maven and Gradle projects. You can find Spring Boot Initilizr tool on https://start.spring.io/