1)What is Spring boot?

Sprint boot is a Java-based spring framework used for Rapid Application Development (to build stand-alone microservices).

It has extra support of auto-configuration and embedded application server like tomcat, jetty, etc.

2)Features of Spring Boot?

Creates stand-alone spring application with minimal configuration needed.

It has embedded tomcat, jetty which makes it just code and run the application.

Provide production-ready features such as metrics, health checks, and externalized configuration.

Absolutely no requirement for XML configuration.

3)What are the advantages of using Spring Boot?

Easy to understand and develop spring applications.

Spring Boot is nothing but an existing framework with the addition of an embedded HTTP server and annotation configuration which makes it easier to understand and faster the process of development.

Increases productivity and reduces development time.

Minimum configuration.

We donâ€™t need to write any XML configuration, only a few annotations are required to do the configuration

4)What are the Spring Boot key components?

Spring Boot auto-configuration.

Spring Boot CLI.

Spring Boot starter POMs.

Spring Boot Actuators

5)Why Spring Boot over Spring?

Below are some key points which spring boot offers but spring doesnâ€™t:

Starter POM.

Version Management.

Auto Configuration.

Component Scanning.

Embedded server.

InMemory DB.

Actuators.

6)What is the starter dependency of the Spring boot module?

Spring boot provides numbers of starter dependency, here are the most commonly used -

Data JPA starter.

Test Starter.

Security starter.

Web starter.

Mail starter.

Thymeleaf starter.

7)How does Spring Boot works?

Spring Boot automatically configures your application based on the dependencies you have added to the project by using annotation.

The entry point of the spring boot application is the class that contains @SpringBootApplication annotation and the main method.

Spring Boot automatically scans all the components included in the project by using @ComponentScan annotation.

8)What does the @SpringBootApplication annotation do internally?

The @SpringBootApplication 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. But, as we know, Spring provided loosely coupled features that we can use for each annotation as per our project needs.

9)What is the purpose of using @ComponentScan in the class files?

Spring Boot application scans all the beans and package declarations when the application initializes.

You need to add the @ComponentScan annotation for your class file to scan your components added to your project.

10)What are starter dependencies?

Spring boot starter is a maven template that contains a collection of all the relevant transitive dependencies that are needed to start a particular functionality.

Like we need to import spring-boot-starter-web dependency for creating a web application.

11) What is Spring Initializer?

Spring Initializer is a web application that helps you to create an initial spring boot project structure and provides a maven or gradle file to build your code.

It solves the problem of setting up a framework when you are starting a project from scratch.

12)What is Spring Boot CLI and what are its benefits?

Spring Boot CLI is a command-line interface that allows you to create a spring-based java application using Groovy.

Example: You donâ€™t need to create getter and setter method or access modifier, return statement. If you use the JDBC template, it automatically loads for you.

13)What Are the Basic Annotations that Spring Boot Offers?

@EnableAutoConfiguration â€“ to make Spring Boot look for auto-configuration beans on its classpath and automatically apply them.

@SpringBootApplication â€“ used to denote the main class of a Boot Application.

This annotation combines @Configuration, @EnableAutoConfiguration, and @ComponentScan annotations with their default attributes.

14)What is Spring Boot dependency management?

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

15)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

16) Is it possible to change the port of the embedded Tomcat server in Spring Boot?

Yes, it is possible. By using the server.port in the application.properties.

17)What is the default port of tomcat in spring boot?

The default port of the tomcat server-id 8080. It can be changed by adding sever.port properties in the application.property file.

18)Explain @RestController annotation in Sprint boot?

It is a combination of @Controller and @ResponseBody, used for creating a restful controller. It converts the response to JSON or XML. It ensures that data returned by each method will be written straight into the response body instead of returning a template.

19)What is the difference between @RestController and @Controller in Spring Boot?

Controller Map of the model object to view or template and make it human readable but @RestController simply returns the object and object data is directly written in HTTP response as JSON or XML.

20)What is Spring Actuator? What are its advantages?

An actuator is an additional feature of Spring that helps you to monitor and manage your application when you push it to production.

These actuators include auditing, health, CPU usage, HTTP hits, and metric gathering, and many more that are automatically applied to your application.

21)How to enable Actuator in Spring boot application?

To enable the spring actuator feature, we need to add the dependency of â€œspring-boot-starter-actuatorâ€ in pom.xml.

22) What is the difference between RequestMapping and GetMapping?

RequestMapping can be used with GET, POST, PUT, and many other request methods using the method attribute on the annotation.

Whereas getMapping is only an extension of RequestMapping which helps you to improve on clarity on request.

23)What is the use of Profiles in spring boot?

While developing the application we deal with multiple environments such as dev, QA, Prod, and each environment requires a different configuration. For eg., we might be using an embedded H2 database for dev but for prod, we might have proprietary Oracle or DB2. Even if DBMS is the same across the environment, the URLs will be different.

To make this easy and clean, Spring has the provision of Profiles to keep the separate configuration of environments.