1.What are the advantages of using Spring Boot?

The advantages of Spring Boot are listed below:

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 don’t need to write any XML configuration, only a few annotations are required to do the configuration.

2. What are the Spring Boot key components?

Below are the four key components of spring-boot:

Spring Boot auto-configuration.

Spring Boot CLI.

Spring Boot starter POMs.

Spring Boot Actuators.

3. Why Spring Boot over Spring?

Below are some key points which spring boot offers but spring doesn’t:

Starter POM.

Version Management.

Auto Configuration.

Component Scanning.

Embedded server.

InMemory DB.

Actuators

4. 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.

5. 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.

6. 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.

7. 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.

8. What is @restcontroller in spring boot?

@RestController was introduced by Spring 4.0 as a specialized version and convenience-based annotation of the controller. It is responsible for adding the @ResponseBody and @Controller annotations. When the controller class is annotated with @RestController annotation, the @ResponseBody annotation may not be included in the request mapping methods. Do note that the @ResponseBody annotation will be active by default.

9. What is @requestmapping in spring boot?

@RequestMapping refers to the commonest annotation in the repository of Spring Web applications. It maps HTTP requests for handling the methods of REST and MVC

controllers.

10. What is JPA

Spring Boot JPA is a Java specification for managing relational data in Java applications. It allows us to access and persist data between Java object/ class and relational database. JPA follows Object-Relation Mapping (ORM). It is a set of interfaces.

11. What is the difference between RequestMapping and GetMapping?

Ans: RequestMapping is generic - you can use with GET, POST, PUT or any of the other request methods using the method attribute on the annotation.

GetMapping is specific to GET request method. It’s just an extension of RequestMapping to improve clarity.

12. What is the difference between JPA and Hibernate?

Ans: JPA is a specification/Interface

Hibernate is one of JPA implementations

When we use JPA, we use the annotation and interfaces from javax.persistence package, without using the hibernate import packages.

13. What Is Spring Boot DevTools Used For?

Spring Boot Developer Tools, or DevTools, is a set of tools making the development process easier.

To include these development-time features, we just need to add a dependency to the pom.xml file:

<dependency>

<groupId>org.springframework.boot</groupId>

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

</dependency>

The spring-boot-devtools module is automatically disabled if the application runs in production. The repackaging of archives also excludes this module by default. So, it won't bring any overhead to our final product.

14. Why Do We Need Spring Profiles?

When developing applications for the enterprise, we typically deal with multiple environments such as Dev, QA and Prod. The configuration properties for these environments are different.

For example, we might be using an embedded H2 database for Dev, but Prod could have the proprietary Oracle or DB2. Even if the DBMS is the same across environments, the URLs would definitely be different.

To make this easy and clean, Spring has the provision of profiles to help separate the configuration for each environment. So, instead of maintaining this programmatically, the properties can be kept in separate files such as application-dev.properties and application-prod.properties. The default application.properties points to the currently active profile using spring.profiles.active so that the correct configuration is picked up.

15. 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.