INTERVIEW QUESTIONS - Spring Boot (Part- 2) of 5

Part 02:

41. Explain the advantages of using Spring Boot for application development.

Spring Boot helps to create stand-alone applications which can be started using java.jar (Doesn't require configuring WAR files).
Spring Boot also offers pinpointed 'started' POMs to Maven configuration.
Has provision to embed Undertow, Tomcat, Jetty, or other web servers directly.
Auto-Configuration: Provides a way to automatically configure an application based on the dependencies present on the classpath.
Spring Boot was developed with the intention of lessening the lines of code.
It offers production-ready support like monitoring and apps developed using spring boot are easier to launch.

42. Differentiate between Spring and Spring Boot.

The Spring Framework provides multiple features like dependency injection, data binding, aspect-oriented programming (AOP), data access, and many more that help easier development of web applications whereas Spring Boot helps in easier usage of the Spring Framework by simplifying or managing various loosely coupled blocks of Spring which are tedious and have a potential of becoming messy.

Spring boot simplifies commonly used spring dependencies and runs applications straight from a command line.
It also doesn't require an application container and it helps in monitoring several components and configures them externally.

43. What are the features of Spring Boot?

Spring Boot CLI – This allows you to Groovy / Maven for writing Spring boot application and avoids boilerplate code.
Starter Dependency – With the help of this feature, Spring Boot aggregates common dependencies together and eventually improves productivity and reduces the burden on
Spring Initializer – This is a web application that helps a developer in creating an internal project structure. The developer does not have to manually set up the structure of the project while making use of this feature.
Auto-Configuration – This helps in loading the default configurations according to the project you are working on. In this way, unnecessary WAR files can be avoided.
Spring Actuator – Spring boot uses actuator to provide "Management EndPoints" which helps the developer in going through the Application Internals, Metrics etc.
Logging and Security – This ensures that all the applications made using Spring Boot are properly secured without any hassle.

44. What does @SpringBootApplication annotation do internally?

As per the Spring Boot documentation, the @SpringBootApplication annotation is one point replacement for using @Configuration,@EnableAutoConfiguration and @ComponentScan annotations alongside their default attributes.

@SpringBootApplication = @Configuration + @EnableAutoConfiguration + @ComponentScan

This enables the developer to use a single annotation instead of using multiple annotations thus lessening the lines of code. However, Spring provides loosely coupled features which is why we can use these annotations as per our project needs.

45. What are the effects of running Spring Boot Application as "Java Application"?

The application automatically launches the tomcat server as soon as it sees that we are running a web application.

46. What is Spring Boot dependency management system?

It is basically used to manage dependencies and configuration automatically without the need of specifying the version for any of that dependencies.

47. What are the possible sources of external configuration?

Spring Boot allows the developers to run the same application in different environments by making use of its feature of external configuration.
This uses environment variables, properties files, command-line arguments, YAML files, and system properties to mention the required configuration properties for its corresponding environments.

Following are the sources of external configuration:
Command-line properties – Spring Boot provides support for command-line arguments and converts these arguments to properties and then adds them to the set of environment properties.

Application Properties – By default, Spring Boot searches for the application properties file or its YAML file in the current directory of the application, classpath root, or config directory to load the properties.

Profile-specific properties – 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 or an environment.

48.Can we change the default port of the embedded Tomcat server in Spring boot?
Yes, we can change it by using the application properties file by adding a property of server.port and assigning it to any port you wish to.

For example, if you want the port to be 8081, then you have to mention server.port=8081.
Once the port number is mentioned, the application properties file will be automatically loaded by Spring Boot and the specified configurations will be applied to the application

49. Can you tell how to exclude any package without using the basePackages filter?
We can use the exclude attribute while using the annotation @SpringBootApplication as follows:

```
@SpringBootApplication(exclude= {Student.class})
public class MainAppConfiguration {

}
```

50. How to disable specific auto-configuration class?

You can use the exclude attribute of @EnableAutoConfiguration for this purpose as shown below:
@EnableAutoConfiguration(exclude = {MainAutoConfiguration.class})
If the class is not specified on the classpath, we can specify the fully qualified name as the value for the excludeName.

```
//By using "excludeName"
@EnableAutoConfiguration(excludeName={Foo.class})
```
You can add into the application.properties and multiple classes can be added by keeping it comma-separated.

51.Can the default web server in the Spring Boot application be disabled?

Yes! application.properties is used to configure the web application type, by mentioning spring.main.web-application-type=none.

52. What are the uses of @RequestMapping and @RestController annotations in Spring Boot?

@RequestMapping:
This provides the routing information and informs Spring that any HTTP request matching the URL must be mapped to the respective method.
org.springframework.web.bind.annotation.RequestMapping has to be imported to use this annotation.

@RestController:
This is applied to a class to mark it as a request handler thereby creating RESTful web services using Spring MVC. This annotation adds the @ResponseBody and @Controller annotation to the class.
org.springframework.web.bind.annotation.RestController has to be imported to use this annotation.

53.What Are the Basic Annotations that Spring Boot Offers?

The primary annotations that Spring Boot offers reside in its org.springframework.boot.autoconfigure and its sub-packages. Here are a couple of basic ones:

@EnableAutoConfiguration – to make Spring Boot look for auto-configuration beans on its classpath and automatically apply them.
@SpringBootApplication – used to denote the main class of a Boot Application. This annotation combines @Configuration, @EnableAutoConfiguration, and @ComponentScan annotations with their default attributes.

54. 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.

55. 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.

56. 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.

57. 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.

58.How to enable Actuator in Spring boot application?

To enable the spring actuator feature, we need to add the dependency of "spring-boot-starter-actuator" in pom.xml.

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

59. What are the actuator-provided endpoints used for monitoring the Spring boot application?

Actuators provide below pre-defined endpoints to monitor our application -

Health
Info
Beans
Mappings
Configprops
Httptrace
Heapdump
Threaddump
Shutdown

60. How to get the list of all the beans in your Spring boot application?

Spring Boot actuator “/Beans” is used to get the list of all the spring beans in your application.

61. How to check the environment properties in your Spring boot application?

Spring Boot actuator “/env” returns the list of all the environment properties of running the spring boot application.

62. How to enable debugging log in the spring boot application?

Debugging logs can be enabled in three ways -
- We can start the application with --debug switch.
- We can set the logging.level.root=debug property in application.property file.
- We can set the logging level of the root logger to debug in the supplied logging configuration file.

63. Where do we define properties in the Spring Boot application?

You can define both application and Spring boot-related properties into a file called application.properties.
You can create this file manually or use Spring Initializer to create this file.
You don’t need to do any special configuration to instruct Spring Boot to load this file, If it exists in classpath then spring boot automatically loads it and configure itself and the application code accordingly.

64. 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+

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

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

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:

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

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

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

67. Mention the differences between JPA and Hibernate.

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

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

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

69. 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.

70. 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.

71. Explain different phases of RAD model.

Various phases of RAD mode are:
1. Business Modeling: Based on the flow of information and distribution between various business channels, the product is designed.
2. Data Modeling: The information collected from business modeling is refined into a set of data objects that are significant for the business.
3. Application Generation: Automated tools are used for the construction of the software, to convert process and data models into prototypes.

72. What is RAD model?

RAD or Rapid Application Development process is an adoption of the waterfall model; it targets developing software in a short period.
RAD follow the iterative SDLC RAD model has the following phases:
1. Business Modeling
2. Data Modeling
3. Process Modeling
4. Application Generation
5. Testing and Turnover

73.What is Spring Boot starter? How is it useful?

Spring Boot has many starters. They are a set of convenient dependency descriptors. Starter allows you to include these descriptors in your pom.xml. For example, If you want to work with Spring MVC, you can include "spring–boot–starter–web" as a dependency in pom.xml.

74. Can you use Spring Boot with applications which are not using Spring?

No, it is not possible as Spring Boot is limited to Spring application only.

75. What is DevTools in Spring Boot?

Spring Boot DevTools helps you to increase the productivity of the developer. So, you don't require to redeploy your application every time you make the changes.
It allows the developer to reload changes without the need of restarting of the server.

76. How can you access a value defined in the application? What is properties file in Spring Boot?

Use the @Value annotation to access the properties which is defined in the application .properties file.

```
@Value("${custom.name}")
private String customName;
```

77.Explain Spring Boot Admin?

Spring Boot admin is a community project which helps you to manage and monitor your Spring Boot applications.

78. What is @pathVariable?

@PathVariable annotation helps you to extract information from the URI directly.

79. What is Swagger2?

Swagger is used to describing the structure of APIs. Swagger 2 is an open-source service provided in Spring Boot which makes it easier for the machines to find out the structure of APIs
like RESTful Web services.

80. What are different environments for enterprise application development?

Dev
QA
Stage
Production

To be continued………..