Spring Boot Interview Questions and Answers

1. What is Spring Boot?

Ans-First of all Spring Boot is not a framework, it is a way to ease to create stand-alone application with minimal or zero configurations. It is approach to develop spring based application with very less configuration. It provides defaults for code and annotation configuration to quick start new spring projects within no time. It leverages existing spring projects as well as Third party projects to develop production ready applications. It provides a set of Starter Pom's or gradle build files which one can use to add required dependencies and also facilitate auto configuration.

Spring Boot automatically configures required classes depending on the libraries on its classpath. Suppose your application want to interact with DB, if there are Spring Data libraries on class path then it automatically sets up connection to DB along with the Data Source class.

2. What are the advantages of using Spring Boot?

Ans-It is very easy to develop Spring Based applications with Java or Groovy.

It reduces lots of development time and increases productivity.

It avoids writing lots of boilerplate Code, Annotations and XML Configuration.

It is very easy to integrate Spring Boot Application with its Spring Ecosystem like Spring JDBC, Spring ORM, Spring Data, Spring Security etc.

It follows "Opinionated Defaults Configuration" Approach to reduce Developer effort

It provides Embedded HTTP servers like Tomcat, Jetty etc. to develop and test our web applications very easily.

It provides CLI (Command Line Interface) tool to develop and test Spring Boot (Java or Groovy) Applications from command prompt very easily and quickly.

It provides lots of plugins to develop and test Spring Boot Applications very easily using Build Tools like Maven and Gradle

It provides lots of plugins to work with embedded and in-memory Databases very easily.

3. What are the disadvantages of using Spring Boot?

It is very tough and time consuming process to convert existing or legacy Spring Framework projects into Spring Boot Applications. It is applicable only for brand new/Greenfield Spring Projects.

4. Why is it “opinionated”?

Ans-It follows “Opinionated Defaults Configuration” Approach to reduce Developer effort. Due to opinionated view of spring boot, what is required to get started but also we can get out if not suitable for application.

• Spring Boot uses sensible defaults, “opinions”, mostly based on the classpath contents.

• For example

– Sets up a JPA Entity Manager Factory if a JPA implementation is on the classpath.

– Creates a default Spring MVC setup, if Spring MVC is on the classpath.

• Everything can be overridden easily

– But most of the time not needed

5. How does it work? How does it know what to configure?

Ans-• Auto-configuration works by analyzing the classpath

– If you forget a dependency, Spring Boot can’t configure it

– A dependency management tool is recommended

– Spring Boot parent and starters make it much easier

• Spring Boot works with Maven, Gradle, Ant/Ivy

– Our content here will show Maven

6. How are properties defined? Where?

Ans-In spring boot, we have to define properties in the application.properties file exists in classpath of application as follow.

Example: configure default DataSource bean

database.host=localhost

database.user=admin

7. What is the difference between an embedded container and a WAR?

Ans-There is no force to go container less

– Embedded container is just one feature of Spring Boot

• Traditional WAR also benefits a lot from Spring Boot

– Automatic Spring MVC setup, including DispatcherServlet

– Sensible defaults based on the classpath content

– Embedded container can be used during development

8. What embedded containers does Spring Boot support?

Ans-Spring Boot includes support for embedded Tomcat, Jetty, and Undertow servers.

By default the embedded server will listen for HTTP requests on port 8080.

9. What does @EnableAutoConfiguration do? What about @SpringBootApplication?

Ans-@EnableAutoConfiguration annotation on a Spring Java configuration class

– Causes Spring Boot to automatically create beans it thinks you need

– Usually based on classpath contents, can easily override

```
@Configuration
@EnableAutoConfiguration
public class MyAppConfig {
public static void main(String[] args) {
SpringApplication.run(MyAppConfig.class, args);
}
}
```

@SpringBootApplication was available from Spring Boot 1.2

It is very common to use @EnableAutoConfiguration, @Configuration, and @ComponentScan together.

```
@Configuration
@ComponentScan
```

```
@EnableAutoConfiguration
public class MyAppConfig {
...
}
```

With @SpringBootApplication annotation

```
@SpringBootApplication
public class MyAppConfig {
...
}
```

10. What is a Spring Boot starter POM? Why is it useful?

Ans-Starters are a set of convenient dependency descriptors that you can include in your application. The starters contain a lot of the dependencies that you need to get a project up and running quickly and with a consistent, supported set of managed transitive dependencies.

The starter POMs are convenient dependency descriptors that can be added to your application's Maven. In simple words, if you are developing a project that uses Spring Batch for batch processing, you just have to include spring-boot-starter-batch that will import all the required dependencies for the Spring Batch application. This reduces the burden of searching and configuring all the dependencies required for a framework.

11. Spring Boot supports both Java properties and YML files. Would you recognize and understand them if you saw them?

Ans-spring boot application java property file name is application.properties

spring boot application YML file name is application.yml

12. Can you control logging with Spring Boot? How?

Ans-Yes, we can control logging with spring boot.

Customizing default Configuration for Logging:

By adding logback.xml file to the application we can override the default logging configuration providing by the Spring Boot. This file place in the classpath (src/main/resources) of the application for Spring Boot to pick the custom configuration.

Spring Boot can control the logging level

– Just set it in application.properties

• Works with most logging frameworks

– Java Util Logging, Logback, Log4J, Log4J2

logging.level.org.springframework=DEBUG

logging.level.com.acme.your.code=INFO

13. How to reload my changes on Spring Boot without having to restart server?

Ans-Include following maven dependency in the application.

<dependency>

<groupId>org.springframework</groupId>

<artifactId>springloaded</artifactId>

<version>1.2.6.RELEASE</version>

</dependency>

Automatic restart

Applications that use spring-boot-devtools will automatically restart whenever files on the classpath change. This can be a useful feature when working in an IDE as it gives a very fast feedback loop for code changes. By default, any entry on the classpath that points to a folder will be monitored for changes.

<dependency>

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

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

<optional>true</optional>

</dependency>

This can be achieved using DEV Tools. With this dependency any changes you save, the embedded tomcat will restart. Spring Boot has a Developer tools (DevTools) module which helps to improve the productivity of developers. One of the key challenge for the Java developers is to auto deploy the file changes to server and auto restart the server. Developers can reload changes on Spring Boot without having to restart my server. This will eliminates the need for manually deploying the changes every time. Spring Boot doesn't have this feature when it has released it's first version. This was a most requested features for the developers. The module DevTools does exactly what is needed for the developers. This module will be disabled in the production environment.

14. What is Actuator in Spring Boot?

Ans-pring Boot Actuator is a sub-project of Spring Boot. It adds several production grade services to your application with little effort on your part. There are also has many features added to your application out-of-the-box for managing the service in a production (or other) environment. They're mainly used to expose different types of information about the running application – health, metrics, info, dump, env etc.

15. How to run Spring boot application to custom port ?

Ans-In application.properties, add following property.

server.port = 8181

16. How to implement security for Spring boot application ?

Ans-Add spring security starter to the boot application

<dependency>

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

<artifactId>spring-boot-starter-security</artifactId>

</dependency>

17. What is the configuration file name used by Spring Boot?

Ans-The configuration file used in spring boot projects is application.properties. This file is very important where we would over write all the default configurations. Normally we have to keep this file under the resources folder of the project.

18. How to implement Spring web using Spring boot?

Ans-Web Application Convenience

• Boot automatically configures

– A DispatcherServlet & ContextLoaderListener

– Spring MVC using same defaults as @EnableWebMvc

• Plus many useful extra features:

– Static resources served from classpath

• /static, /public, /resources or /META-INF/resources

– Templates served from /templates

• If Velocity, Freemarker, Thymeleaf, or Groovy on classpath

– Provides default /error mapping

• Easily overridden

– Default MessageSource for I18N

19. How to configure datasource using Spring boot?

Ans-• Use either spring-boot-starter-jdbc or spring-boot-starterdata-jpa and include a JDBC driver on classpath

• Declare properties

spring.datasource.url=jdbc:mysql://localhost/test

spring.datasource.username=dbuser

spring.datasource.password=dbpass

spring.datasource.driver-class-name=com.mysql.jdbc.Driver

– Spring Boot will create a DataSource with properties set

– Will even use a connection pool if the library is found on the classpath!

20. What is YAML?

Ans-Yaml Ain’t a Markup Language

– Recursive acronym

• Created in 2001

• Alternative to .properties files

– Allows hierarchical configuration

• Java parser for YAML is called SnakeYAML

– Must be in the classpath

– Provided by spring-boot-starters

YAML for Properties

• Spring Boot support YAML for Properties

– An alternative to properties files

application.properties

database.host = localhost

database.user = admin

application.yml

database:

  host: localhost

  user: admin

• YAML is convenient for hierarchical configuration data

– Spring Boot properties are organized in groups

– Examples: server, database, etc

21.What are the various features of Spring Boot?

Ans-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

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

Ans-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.

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

Ans-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.

23.What do Dev Tools in Spring boot mean?

Ans-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.

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

Ans-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.

24.What does Actuator in Spring Boot mean?

Ans-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 out-of-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.

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

Ans-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.

26.Why in spring boot “Opinionated ” is used?

Ans-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.

27.What are esteem properties of Spring Boot?

Ans-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.

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

Ans-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.

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

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

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

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

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

Ans-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.

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

Ans-Yes, we can control logging with spring boot.

32.Differentiate Between An Embedded Container And A War?

Ans-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.

33.What does Spring Security mean?

Ans-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.

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

Ans-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.

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

Ans-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
- **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 web-arranged 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

37.Explain the difference between JPA and Hibernate?

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

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

Ans-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.

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

Ans-`/src/main/resources/static. is the suggested folder for static content.`

```
You can create a JS file by sending an alert
/src/main/resources/static/js/custom.js
alert("I'm active");
```

40.List minimum requirements for Spring boot System?

Ans-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 +)

41.What is Auto Configuration?

Ans-*Spring Boot Auto Configuration* comes with these features. Auto-Configuration will attempt to automatically try to set up our application with default behavior based on the jars in the classpath. For example, if *Spring Boot* finds `HSQLDB` in out classpath, it will automatically configure an in-memory database for us. Think of the *auto-Configuration* as an intelligent system which can provide ready to use the application to us based on the configured jars in our classpath.

In our sample application, we never defined any dispatcher servlet or configured tomcat for the deployment, however, we can still find `Mapping servlet: 'dispatcherServlet' to [/]` in the console, this is

happening because we added spring-boot-starter-web in our application using `POM.xml`

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

This single entry demonstrates power and feature of *auto-configuration*. On adding this starter in our application, Spring Boot auto configuration understands that we are building an MVC application and it added all required dependencies in our classpath for a Spring MVC web application.
This went one step ahead and automatically configured `DispactherServlet, CharacterEncodingFilter, RequestContextFilter` and even error page for us (We never did any configuration for these). Spring Boot will add similar configurations based on the Spring Boot Starters added in the `POM` file.