**1. What is Spring Boot? Why should we use it?**

- Spring Boot is a Java framework from Sring umbrella which is developed to simplify the use of Spring Framework for Java development. It helps in automation, configuration, and dependencies.

- Why should we use it? It provides a lot of convenience by auto-configuration which helps improve productivity because it allows developers and programmers to focus on writing the business logic.

For example, you don't need to setup a Tomcat server to run your web application. You can just write code and run it as Java application because it comes with an embedded Tomcat server. You can also create a JAR file or WAR file for deployment based on your convenience.

In fact, it's now the standard way to develop Java application with Spring framework.

**2. What is spring-boot-starter-web?**

- This starter will add Tomcat and Spring MVC dependency to our application and its default configuration.

**3. What is spring-boot-devtools ?**

- 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.

**4. What are Profiles in spring boot?**

- Spring Profiles provide a way to segregate parts of your application configuration and make it only available in certain environments

**5. Spring Boot Actuator?**

- Spring Boot Actuator help you monitor and manage your application health when you push it to production.

You can choose to manage and monitor your application by using HTTP endpoints.

**6. How do you Change tomcat or jetty HTTP port?**

- You can change the tomcat http port by changing default http property in application.properties file.

**7. What are the requirements of Spring boot System?**

Spring Boot 1.5.9.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 +)

8. What is Spring Boot AutoConfigurator?

- Spring Boot AutoConfigurator is used by Spring Boot Framework to provide “Auto-Configuration”.

9. Mention Spring Boot Components

Spring Boot Framework has the following components:

Spring Boot Starter

Spring Boot AutoConfigurator

Spring Boot Actuator

Spring Boot CLI

Spring Boot Initilizr.

**Spring boot 2 new features list**

* A Java 8 baseline, and Java 9 support.
* Reactive web programming support with Spring WebFlux/WebFlux.fn.
* Auto-configuration and starter POMs for reactive Spring Data Cassandra, MongoDB, Couchbase and Redis.
* Support for embedded Netty.
* HTTP/2 for Tomcat, Undertow and Jetty.  
  Kotlin support.
* A brand new actuator architecture, with support for Spring MVC, WebFlux and Jersey.  
  Micrometer based metrics with exporters for Atlas, Datadog, Ganglia, Graphite, Influx, JMX, New Relic, Prometheus, SignalFx, StatsD and Wavefront.
* Quartz scheduler support.
* Greatly simplified security auto-configuration.

**Spring Boot Questions- Part 2**

Recently many interview has a set of**questions on Spring Boot**when it comes to**advanced Java Interviews**. Here are frequently asked **Spring Boot questions** with answers.

* **What is LiveReload?**

The spring-boot-devtools module includes an embedded LiveReload server that can be used to trigger a browser refresh when a resource is changed. LiveReload browser extensions are freely available for Chrome, Firefox and Safari from livereload.com.

* **How to exclude auto restart for static files?**

By default changing resources in /META-INF/maven, /META-INF/resources, /resources, /static, /public or /templates will not trigger a restart  
But If you want to customize these exclusions you can use the spring.devtools.restart.exclude property  
if you want to keep those defaults and add additional exclusions, use the spring.devtools.restart.additional-exclude property instead

* **How to start spring boot application in debug mode?**

java -jar myproject-0.0.1-SNAPSHOT.jar –debug

* **What are the advantages of YAML file than Properties file?**

YAML is a superset of JSON, and as such is a very convenient format for specifying hierarchical configuration data. The SpringApplication class will automatically support YAML as an alternative to properties whenever you have the SnakeYAML library on your classpath.

* **What are the different ways to load YAML file in Spring boot?**

1. YamlPropertiesFactoryBean will load YAML as Properties  
2. YamlMapFactoryBean will load YAML as a Map

* **What are the advantages of spring Externalized Configuration?**

Externalize your configuration to work with the same application code in different environments. You can use properties files, YAML files, environment variables and command-line arguments to externalize configuration.

* **How to write custom log configuration in spring boot?**

You can force Spring Boot to use a particular logging system using the org.springframework.boot.logging.LoggingSystem system property. The value should be the fully-qualified class name of a LoggingSystem implementation. You can also disable Spring Boot’s logging configuration entirely by using a value of none.

* **How do you customize Favicon in spring boot web application?**

Spring Boot looks for a favicon.ico in the configured static content locations and the root of the classpath (in that order). If such file is present, it is automatically used as the favicon of the application.

* **How spring boot handles error in application?**

Spring Boot provides an /error mapping by default that handles all errors in a sensible way, and it is registered as a ‘global’ error page in the servlet container.

* **How do you Create a deployable war file in spring boot?**

Step1: Extend SpringBootServletInitializer and override its configure method  
Step 2: Change packing type to war in pom.xml or in build.gradle  
Step 3: Mark the embedded servlet container dependency as provided

## Questions on Spring Boot – Part 3

Recently many interview has a set of**questions on Spring Boot**when it comes to**advanced Java Interviews**. Here are frequently asked **Spring Boot questions** with answers.

* **What is Hot swapping in spring boot?**

Reloading the changes without restarting the server is called hot swapping, Modern IDEs (Eclipse, IDEA, etc.) all support hot swapping of bytecode,  so if you make a change that doesn’t affect class or method signatures it should reload cleanly with no side effects.

* **How do you Switch off the Spring Boot security configuration?**

If you define a @Configuration with @EnableWebSecurity anywhere in your application it will switch off the default webapp security settings in **Spring Boot**.

* **How to execute Spring Batch jobs on startup?**

Spring Batch auto-configuration is enabled by adding @EnableBatchProcessing (from Spring Batch) somewhere in your context. By default it executes all Jobs in the application context on startup

* **Does spring boot need Logging? What is the default one?**

Spring Boot has no mandatory logging dependency, except for the Commons Logging API.

* **How do you configure Configure Logback for logging?**

If you put a logback.xml in the root of your classpath it will be picked up from there

* **How do you Configure Log4j for logging?**

Spring Boot supports Log4j 2 for logging configuration if it is on the classpath. If you are using the starters for assembling dependencies that means you have to exclude Logback and then include log4j 2 instead

* **How do you write a Write a JSON REST service in spring boot?**

Any Spring @RestController in a Spring Boot application should render JSON response by default as long as Jackson2 is on the classpath

* **How do you Write an XML REST service in spring boot?**

If you have the Jackson XML extension (jackson-dataformat-xml) on the classpath, it will be used to render XML responses

* **What is the default Multipart File Uploads size in spring boot?**

By default Spring Boot configures Spring MVC with a maximum file of 1MB per file and a maximum of 10MB of file data in a single request.

* **How do you Enable HTTP response compression in spring boot?**

HTTP response compression is supported by Jetty, Tomcat, and Undertow. It can be enabled by adding server.compression.enabled=true in application.properties

* **How do you add Add a Servlet, Filter or Listener to an application ?**

There are two ways to add Servlet, Filter, ServletContextListener and the other listeners supported by the Servlet spec to your application. You can either provide Spring beans for them, or enable scanning for Servlet components.