

## Spring Boot Interview Questions- Part 1

[Part 1](#) | [Part 2](#) | [Part 3](#)



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

- **What is Spring Boot?**

Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that you can “just run”.

You can use Spring Boot to create Java applications that can be started using `java -jar` or more traditional war deployments.

- **What are the advantages of spring boot application?**

Provide a range of non-functional features that are common to large classes of projects (e.g. embedded servers, security, metrics, health checks, externalized configuration).

Opinionated view of the Spring platform and third-party libraries so you can get started with minimum fuss.

Absolutely no code generation and no requirement for XML configuration.

Provide a radically faster and widely accessible getting started experience for all Spring development.

- **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 +)

Read more [Spring boot system requirements](#)

**What is the latest version of spring boot and its System requirement?**

As per July 2018 Spring boot latest version is 2.1.

Spring boot needs Java 8+ version and Spring 5 framework as minimum version.

Gradle 4 + version

- **Spring boot 2 new features?**

Refer : [Spring boot 2 new features](#)

- **What are the use of @EnableAutoConfiguration annotation?**

This annotation tells Spring Boot to “guess” how you will want to configure Spring, based on the jar dependencies that you have added.

Read more [@enableautoconfiguration spring boot example](#)

- **What is Spring Boot Starter?**

Spring Boot provides a number of “Starters” that make easy to manage dependencies for your project.

Read More [Spring boot starter](#)

- **What is spring-boot-starter-parent?**

The spring-boot-starter-parent is a special starter that makes Maven or Gradle dependency-management easier by adding jars to your classpath.

Read More [spring-boot-starter-parent](#)

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

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

Read More [spring-boot-starter-web](#)

- **How to create an executable jar using spring boot?**

Add this below plugin to pom.xml

```
<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
  </plugins>
</build>
```

Use mvn clean package to create executable jar file

Read More [spring boot executable jar example](#)

- **How do you run and stop spring boot executable jar?**

Open cmd or shell window and use java -jar as shown below

```
$ java -jar myproject-0.0.1-SNAPSHOT.jar
```

To stop use ctrl+C

Read More [run spring boot jar from command line](#)

- **How do you change JDK version in spring boot?**

Java 1.6 as the default compiler level.

You can overwrite it by adding java.version property tag as shown below

```
<properties>
  <java.version>1.8</java.version>
</properties>
```

- **What is the use of @SpringBootApplication annotation?**

The @SpringBootApplication annotation is equivalent to using @Configuration, @EnableAutoConfiguration and @ComponentScan with their default attributes

Since many spring Boot developers have their main class annotated with @Configuration, @EnableAutoConfiguration and @ComponentScan, spring boot provides you a new annotation @SpringBootApplication as replacement.

- **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.

- **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. Any @Component or @Configuration can be marked with @Profile to limit when it is loaded

- **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.

- **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.

- **How to disable specific auto-configuration in spring boot?**