



Java2Blog



[Home](#) ▶ [Spring Frameworks](#) ▶ [Spring Boot](#) ▶

Top 15 Spring Boot Interview questions

Last updated : June 8th, 2018



NO COMMENTS

Top 15 Spring Boot Interview questions



[Previous](#)

[Next](#)



In this post, we will see top 15 Spring Boot interview questions with answers. If you want to read more about Spring boot, you can go through [Spring boot tutorial](#).

Here is the list of Spring Boot interview questions.

1. What is Spring boot?

Spring Boot makes it easier for you to create production ready applications in no time. It is an opinionated view to create Spring application quickly. It follows convention over configuration. In simple terms, it comes with default configurations for most of the Spring projects, you don't need to do much to bootstrap any spring application.

2. Why did you use Spring boot in your application?

As discussed earlier, Spring boot makes it easier for you to create Spring application, it can save a lot of time and

efforts.

For example: Let's say you want to create Spring boot project with activeMQ. You can simply use "spring-boot-starter-activemq" as artifact Id, it will take all the defaults and create Spring application with ActiveMQ configured. Let's say you don't want to use inbuilt activeMQ, you can simply override "spring.activemq.broker-url" in application.properties to use external ActiveMQ.

3. Can you list advantages of Spring boot?

Advantages of Spring boot are:

- It provides a lot of default configurations which help you to create Spring application faster.
- It comes with embedded tomcat or jetty server, so you don't have to deploy jar.
- It reduces development code by avoiding a lot of boilerplate code.
- It increases productivity as you can create Spring application quickly.
- It provides a lot of starter project for easy maven integration. You don't have to worry about version mismatch.
- You can quickly create using sample project using [spring boot initializer](#)

4. What are disadvantages of Spring boot?

If you want to convert your old spring application to Spring boot application, it may not be straight forward and can be time consuming.

5. How can you override default properties in Spring boot Project?

Spring boot provides a lot of properties which can be overridden by specifying them in application.properties.

For example: You want to specify prefix and suffix in Spring MVC applications. You can simply do it by putting below properties in application.properties.

```
spring.mvc.view.prefix: /WEB-INF/
```

```
spring.mvc.view.suffix: .jsp
```

5. How can you run Spring boot application on custom port?

You can simply put server.port properties in application.properties.

For example:server.port=8050

6. What is Spring boot starter and how it is useful?

Spring boot comes with a lot of starters which is set of convenient dependency descriptors which you can include in your pom.xml.

For example: Let's say you want to work Spring MVC application, you can simply include “spring-boot-starter-web” as dependency in pom.xml .

7. Can we use Spring boot with applications which are not using Spring?

No, it is not possible as of now. Spring boot is limited to Spring applications only.

8. What is name of the configuration file which you use in Spring boot?

Configuration file used in Spring boot projects is application.properties. It is very important file as it is used to override all default configurations.

9. What is DevTools in Spring boot?

Spring boot comes with DevTools which is introduced to increase the productivity of developer. You don't need to redeploy your application every time you make the changes. Developer can simply reload the changes without restart of the server. It avoids pain of redeploying application every time when you make any change. This module will be disabled in production environment.

10. What is actuator in Spring boot?

Spring boot actuator is one of the most important features of Spring boot. It is used to access current state of running application in production environment. There are various metrics which you can use to check current state of the application.

Spring boot actuator provides restful web services end points which you can simply use and check various metrics.

For example:

/metrics : This restful end point will show you metrics such as free memory, processors, uptime and many more properties,

Spring boot actuator will help you to monitor your application in production environment. Restful end points can be sensitive, it means it will have restricted access and will be shown only to authenticated users. You can change this property by overriding it in application.properties.

11. How can you implement Spring security in Spring boot application?

Implementation of Spring security in Spring boot application requires very little configuration. You need to add **spring-boot-starter-security** starter in pom.xml. You need to create Spring config class which will extend

WebSecurityConfigurerAdapter and override required method to achieve security in Spring boot application.

You can read more about [Spring boot security example](#).

12. Have you used @SpringBootApplication annotation in Spring boot project?

@SpringBootApplication annotation was introduced in Spring Boot 1.2.0. This annotation is equivalent to declaring these 3 annotations.

- @Configuration
- @EnableAutoConfiguration
- @ComponentScan

For example:

When you create your main class with Spring boot, you have to use below annotations before Spring boot 1.2.0.

```
1
2 @Configuration
3 @EnableAutoConfiguration
4 @ComponentScan
5 public class SpringBootHelloWorldApplicat
6 ...
7 }
8
```

But after Spring boot 1.2.0, you just need to use @SpringBootApplication annotation which will cover above 3 annotations

```
1
2 @SpringBootApplication
3 public class SpringBootHelloWorldApplicat
4 ...
5 }
6
```

13. What are embedded containers which are supported by Spring Boot?

Spring boot contains embedded Tomcat, Jetty and undertow servers.

14. Have you used ActiveMQ in Spring Boot application? Do you know how to configure external ActiveMQ?

Spring Boot comes with embedded ActiveMQ. We need to use “spring-boot-starter-activemq” dependency in pom.xml and it will take care of all defaults and will configure ActiveMQ in the project.

If you want to configure external ActiveMQ then you need to just put “spring.activemq.broker-url” in application.properties and provide the URL of external ActiveMQ.

15. How can you configure logging in Spring boot application?

Spring Boot comes with support for Java Util Logging, Log4J2 and Logback and it will be pre-configured as Console output.

Hence, You can simply specify logging.level in application.properties.

```
1  
2 logging.level.spring.framework=Debug  
3
```

It will set Spring framework logs to debug level.

Let's say you want to put logs to the file. You can specify logger.file in application.properties.

```
1  
2 logging.file=${java.io.tmpdir}/applicatio
```

If you want to do logging configuration explicitly, You can also create logback.xml in main/java/resources folder and specify logging configuration in the file. Spring Boot will pick this file and configure logging accordingly.

That's all about Spring boot interview questions.

[Previous](#)[Next](#)

JOIN OUR NEWS LETTER – STAY UPDATED

Subscribe to Awesome Java Content.

Enter your email here

SUBSCRIBE

We respect your privacy and take protecting it seriously.

 Facebook

 Twitter

 Google+

 LinkedIn

 Pinterest

Related Posts

Spring Profile
annotation example

Circular dependencies in
Spring

Spring XML
configuration example