

Spring Boot Interview Questions

In this post we will look at Spring Boot Interview questions. Examples are provided with explanation.

Q: What is Spring Boot?

A: Over the years spring has become more and more complex as new functionalities have been added. Just visit the page-<https://spring.io/projects> (<https://spring.io/projects>) and we will see all the spring projects we can use in our application for different functionalities. If one has to start a new spring project we have to add build path or add maven dependencies, configure application server, add spring configuration . **So a lot of effort is required to start a new spring project as we have to currently do everything from scratch. Spring Boot is the solution to this problem.** Spring boot has been built on top of existing spring framework. Using spring boot we avoid all the boilerplate code and configurations that we had to do previously. Spring boot thus helps us use the existing Spring functionalities more robustly and with minimum efforts.

More details and miscellaneous examples ([/spring/sprboot](#))

Q: What are advantages of Spring Boot ?

A: The advantages of Spring Boot are

- Reduce Developement, Testing time and efforts.
- Use of JavaConfig helps avoid usage of XML.
- Avoid lots of maven imports and the various version conflicts.
- Provide Opinionated Development approach.
- Quick start to development by providing defaults.
- No Separate Web Server Needed.Which means that you no longer have to boot up Tomcat, Glassfish, or anything else.
- Requires less configuration-Since there is no web.xml file. Simply add classes annotated with@Configuration and then you can add methods annotated with@Bean, and Spring will automagically load up the object and manage it like it always has. You can even add @Autowired to the bean method to have Spring autowire in dependencies needed for the bean.
- Environment Based Configuration-Using these properties, you can pass into the application which environment you are using with:-Dspring.profiles.active={enviornment}. Spring will then load up the

subsequent application properties file at (application-{environment}.properties) after loading up the main application properties file.

Q: Which build tool have you used to develop Spring Boot Application ?

A: Spring Boot application can be developed using Maven as well as Gradle.

Spring Boot application using Maven (/spring/SpringBoot_HelloWorld)

Spring Boot application using Gradle (/spring/SpringBoot_HelloWorld_gradle)

Q: What is JavaConfig?

A: Spring JavaConfig is a product of the Spring community that provides a pure-Java approach to configuring the Spring IoC Container. It thus helps avoid using XML configurations. The advantages of using JavaConfig are

The advantages of JavaConfig are

- Object-oriented configuration. Because configurations are defined as classes in JavaConfig, users can take full advantage of object-oriented features in Java. One configuration class may subclass another, overriding its @Bean methods, etc.
- Reduced or eliminated XML configuration. The benefits of externalized configuration based on the principles of dependency injection have been proven. However, many developers would prefer not to switch back and forth between XML and Java. JavaConfig provides developers with a pure-Java approach to configuring the Spring container that is conceptually similar to XML configuration. It is technically possible to configure the container using only JavaConfig configuration classes, however in practice many have found it ideal to mix-and-match JavaConfig with XML.
- Type-safe and refactoring-friendly. JavaConfig provides a type-safe approach to configuring the Spring container. Thanks to Java 5.0's support for generics, it is now possible to retrieve beans by type rather than by name, free of any casting or string-based lookups.

Added security configuration without xml using java config. (/spring/sprboot_sec)

Q:How to reload my changes on Spring Boot without having to restart server?

A: 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. It also provides H2-database console for better testing the application. The following dependency is used

```
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-devtools</artifactId>
<optional>true</optional>
</dependency>
```

The DevTool dependency usage for autorestart and H2 DB console is illustrated in this example ([/spring/SpringBootUsingPagination](#))

Q:What is Actuator in Spring Boot?

A: Spring boot actuator is one of the important feature in spring boot framework. Spring boot actuator helps you to access the current state of the running application in production environment. There are several metrics that has to be checked and monitored in the production environment. Even some external applications may be using those services to trigger the alert message to concerned person. Actuator module exposes set of REST endpoints that can be directly accessed as a HTTP URL to check the status.

Q:How to deploy Spring Boot application as a WAR?

A: Spring Boot WAR Deployment ([/spring/boot-war](#))

Q:What is Docker? How to deploy Spring Boot Applications to Docker?

A: What is Docker ([/devOps/docker](#))

Deploying Spring Based WAR Application to Docker ([/devOps/docker/docker-war](#))

Deploying Spring Based JAR Application to Docker ([/devOps/docker/docker-jar](#))

Q:How to disable Actuator endpoint security in Spring Boot?

A: By default all sensitive HTTP endpoints are secured such that only users that have an ACTUATOR role may access them. Security is enforced using the standard `HttpServletRequest.isUserInRole` method.

We can disable security using -

management.security.enabled=false

It is suggested to disable security only if the actuator endpoints are accessed behind firewall.

Q: How to run Spring boot application to custom port ?

A: In order to run a spring boot application on a custom port you can specify the port in `application.properties`.

server.port=8090

Q: Have you written Test cases using Spring Boot ?

A: Spring Boot provides the `@SpringBootTest` for writing Unit Test Cases

Spring Boot Unit Test Simple Example ([/spring/springboot_testcases](#))

Q: What is YAML ?

A: YAML is a human-readable data serialization language. It is commonly used for configuration files. Compared to properties file, YAML file is much more structured and less confusing in case we want to add complex properties in the configuration file. As can be seen YAML has hierarchical configuration data. Use YAML properties in Spring Boot (/spring/bootyaml)

Q: How to implement security for Spring boot application ?

A: For Implementing security for Spring Boot we use the **spring-boot-starter-security** dependency and have to add the Security config. It requires very little code. Config class will have to extend WebSecurityConfigurerAdapter and override its methods.

Spring Boot Security example and explanation (/spring/sprboot_sec)

Q: Have you integrated Spring Boot and ActiveMQ ?

A: For integrating Spring Boot and ActiveMQ we use the **spring-boot-starter-activemq** dependency. It requires very little configuration and no boilerplate code.

Spring Boot ActiveMQ explanation (/spring/sprboot_activemq)

Q: Have you integrated Spring Boot and Apache Kafka ?

A: For integrating Spring Boot and Apache Kafka we use the **spring-kafka** dependency.

Spring Boot + Apache Kafka Example (/spring/spring-boot-apache-kafka-hello-world)

Q: How to implement Pagination and Sorting with Spring Boot ?

A: Using Spring Boot achieving pagination is very simple. Using the Spring Data-JPA this is achieved passing pageable org.springframework.data.domain.Pageable to the repository methods.

Spring Boot pagination explanation (/spring/SpringBootUsingPagination)

Q: What is Swagger ? Have you implemented it using Spring Boot ?

A: Swagger is widely used for visualizing APIs, and with Swagger UI it provides online sandbox for frontend developers. For the tutorial, we will use the Springfox implementation of the Swagger 2 specification. Swagger is a tool, a specification and a complete framework implementation for producing the visual representation of RESTful Web Services. It enables documentation to be updated at the same pace as the server. When properly defined via Swagger, a consumer can understand and interact with the remote service with a minimal amount of implementation logic. Thus Swagger removes the guesswork in calling the service.

Spring Boot + Swagger2 (/spring/boot_swagger)

Q: What is Spring Profiles ? How do you implement it using Spring Boot ?

A: Spring Profiles allows users to register beans depending on the profile(dev, test, prod etc). So when the application is running in DEVELOPMENT only certain beans can be loaded and when in PRODUCTION certain other beans can be loaded. Suppose our requirement is that the Swagger documentation be enabled only for the QA environment and disabled for all others. This can be done using Profiles. Spring Boot makes using Profiles very easy.

Spring Boot + profiles (/spring/boot_swagger_profile)

Q: What is Spring Batch ? How do you implement it using Spring Boot ?

A: Spring Boot Batch provides reusable functions that are essential in processing large volumes of records, including logging/tracing, transaction management, job processing statistics, job restart, skip, and resource management. It also provides more advanced technical services and features that will enable extremely high-volume and high performance batch jobs through optimization and partitioning techniques. Simple as well as complex, high-volume batch jobs can leverage the framework in a highly scalable manner to process significant volumes of information.

Spring Boot + Batch (/spring/bootbatch)

Q: What is FreeMarker Template? How do you implement it using Spring Boot ?

A: FreeMarker is a Java-based Template Engine, originally focusing on dynamic web page generation with MVC software architecture. The major advantage of using Freemarker is the complete separation of the Presentation layer and the Business Layer. The Programmers can work on the application code while the designers can work on the html page design. Finally using freemarker these can then be combined to give the final output page.

Spring Boot + FreeMarker Example (/spring/spring-boot-freemarker-hello-world)

Q: How to implement Exception Handling using Spring Boot ?

A: Spring provides a very useful way to handle exceptions using ControllerAdvice.

We will be implementing a ControllerAdvice class which will handle all exceptions thrown by the controller class.

Spring Boot Exception Handling (/spring/boot-exception-handling)

Q: What is caching? Have you used any caching framework with Spring Boot ?

A: A cache is an area of local memory that holds a copy of frequently accessed data that is otherwise expensive to get or compute. Have used Hazelcast for caching.

Spring Boot + Hazelcast Example (/spring/spring-boot-hazelcast)

Q: Have you exposed a SOAP webservice endpoint using Spring Boot?

A: Yes. Using Spring Boot exposed a web service to be consumed. Used Contract first approach to generate the classes from wsdl.

Spring Boot + SOAP Web Service Example (/spring/springbootsoapwebservice)

Q: How did you perform database operations using Spring Boot ?

A: Spring Boot Tutorial-Spring Data JPA (/spring/SpringBoot_DataJPA)

Spring Boot JDBC Example (/spring/bootjdbc)

Q: How to upload a file using Spring ?

A: Spring Boot + File Upload Example (/spring/bootupload)

Q: How to implement interceptors with Spring Boot ?

A: Using Spring MVC HandlerInterceptor with Spring Boot (/spring/bootInterceptor)

Q: How to use schedulers with Spring Boot ?

A: Spring Boot Task Scheduler Example (/spring/bootTask)

Q: Which all starter maven dependencies have you used ?

A: Have used different starter dependencies like **spring-boot-starter-activemq dependency, spring-boot-starter-security dependency, spring-boot-starter-web dependency.**

This helps in adding less number of dependencies and also in reducing version conflicts.

Spring Boot Security example and explanation (/spring/sprboot_sec)

Q: What is CSRF attack? How to enable CSRF protection against it?

A: CSRF stands for Cross-Site Request Forgery. It is an attack that forces an end user to execute unwanted actions on a web application in which they are currently authenticated. CSRF attacks specifically target state-changing requests, not theft of data, since the attacker has no way to see the response to the forged request.

Spring Boot Security - Enabling CSRF Protection (/spring/boot_security_csrf)

Q: How to use Form Login Authentication using Spring Boot?

A: Spring Boot Form Security Login Hello World Example (/spring/boot_form_security)

Q: What is OAuth2? How to implement it using Spring Boot?

A: Spring Boot + OAuth2 implementation (/spring/spring-boot-oauth-introduction)

Q: What is GZIP?How to implement it using Spring Boot?

A: gzip is a file format and a software application used for file compression and decompression.

Spring Boot + GZIP Compression (/spring/boot-zip)

Q: Have you used any integration framework with Spring Boot?

A: Have integrated Apache Camel with Spring Boot. Made use of Apache Camel Spring Boot starter dependency. Spring Boot +Apache Camel (/spring/bootcamel)

Q: What is Apache Freemarker? When to use it instead of JSP? How to integrate it with Spring Boot?

A:JSP is tailor made for Web pages, Freemarker template is a more generic templating language - it can be used to generate html, plain text, emails, etc.

Spring Boot + FreeMarker Example (/spring/spring-boot-freemarker-hello-world)

Q: When will you use WebSockets? How to implement it using Spring Boot?

A:WebSocket is a computer communications protocol, providing full-duplex communication channels over a single TCP connection.

- **WebSocket are bi-directional** - Using WebSocket either client or server can initiate sending a message.
- **WebSocket are Full Duplex** - The client and server communication is independent of each other.
- **Single TCP connection** - The initial connection is using HTTP, then this connection gets upgraded to a socket based connection. This single connection is then used for all the future communication
- **Light** - The WebSocket message data exchange is much lighter compared to http.

Spring Boot + WebSockets Example (/spring/boot-websocket)

Q: What is AOP? How to use it with Spring Boot?

A:During software development, functions that span multiple points of an application are called cross-cutting concerns. These cross-cutting concerns differ from the application's main business logic. Hence ,separating these cross-cutting concerns from the business logic is where aspect-oriented programming (AOP) comes into picture.

Spring Boot + AOP Example (/spring/spring-boot-aop)

Q: What is Apache Kafka? How to integrate it with Spring Boot?

A: Apache Kafka is a distributed publish-subscribe messaging system. It is a scalable, fault-tolerant, publish-subscribe messaging system which enables us to build distributed applications. It is an Apache Top Level project. Kafka is suitable for both offline and online message consumption.

Spring Boot + Apache Kafka Example (/spring/spring-boot-apache-kafka-hello-world)

Q: How can we monitor all the Spring Boot Microservices?

A:Spring Boot provides actuator endpoints to monitor metrics of individual microservices. These endpoints are very helpful for getting information about applications like if they are up, if their components like database etc are working good. But a major drawback or difficulty about using actuator endpoints is that we have to individually hit the endpoints for applications to know their status or health. Imagine microservices involving 50 applications, the admin will have to hit the actuator endpoints of all 50 applications. To help us deal with this situation, we will be using open source project located at <https://github.com/codecentric/spring-boot-admin> ().

Built on top of Spring Boot Actuator, it provides a web UI to enable us visualize the metrics of multiple applications.

Spring Boot Admin (/spring/boot-admin)

Q: Have you used any Spring Cloud Components with Spring Boot?

A: Have used Spring Cloud components like Netflix Eureka for Service Registration, Ribbon for Load Balancing.

Spring Boot + Cloud Components (/spring/springcloud)

Spring Cloud interview Questions (/spring/spring-cloud-interview-questions)