

WEDNESDAY, MARCH 7, 2018

Top 10 Spring Interview Questions Answers for Java JEE Developers

Spring framework interview questions are on the rise on Java EE and core Java interviews. Spring, which is obvious given Spring is the best framework available for Java application development and now *Spring IOC container* and Spring MVC framework are used as a de-facto framework for all new Java development. Because of its popularity, interview questions from spring framework are top on any list of [Core Java Interview questions](#). I thought to put together some spring interview questions and answers which have appeared on many Java and J2EE interviews and useful for practicing before appearing on any Java Job interview. I first wrote this article a long back and even its content is still relevant I thought to update it, especially after finishing my list of [Spring Boot Interview Questions](#) recently.

This list of Spring interview questions and answers contains questions from Spring fundamentals e.g. Spring IOC and Dependency Injection, **Spring MVC Framework**, Spring Security, Spring AOP etc, because of length of this post I haven't included Spring interview questions from Spring JDBC and JMS which is also a popular topic in core Java and [J2EE interviews](#). I suggest preparing those as well.

Anyway, these Spring questions are not very difficult and based on fundamentals like what is default scope of Spring bean? which is mostly asked during the first round or the telephonic round of Java interview. Although you can find answers to these **Spring interview questions** by doing Google I have also included answers for most of the questions for your quick reference.

As I have said it before both Spring and Spring MVC, and now Spring Boot are fantastic Java frameworks and if you are not using it then it's a good time to start using them, these questions will give you some head start as well. Spring MVC can be used to develop both standalone Java web application as well as RESTful Web Services using Spring.

Btw, if you are completely new to Spring framework, I suggest you to first take a look at [Spring Framework 5: Beginner to Guru](#) course, otherwise, most of these question will not make any sense to you.

Top Spring Interview Questions and Answers

Now let's start with questions, these **Spring Interview Questions** are not very tricky or tough and based upon primary concepts of spring framework. If you are developing an application using Spring framework then you may be, already familiar with many of these Java and Spring interview questions and answers. Nevertheless, it's a good recap before appearing in any Spring and Java interview.

You can also check out the [Java Programming Interview exposed](#) book from Wrox publication, apart from various Java topics it also contains some really good Spring framework, Hibernate, and Spring MVC questions with detailed explanation.

Also, Hibernate is mostly used in Spring, so don't forget to prepare some [Hibernate interview](#)

[questions](#) along with Spring.

Spring Security Interview Questions Answer

Some of the readers requested to provide Spring Security interview questions and answer, So I thought to update this article with a few of Spring security question I came across.

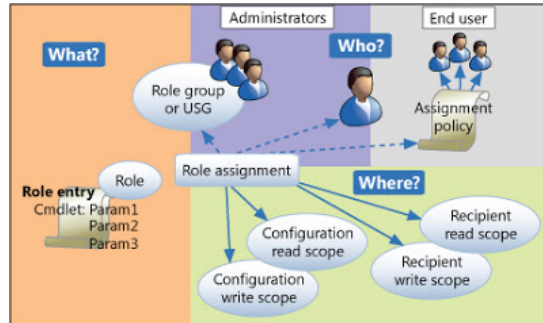
Here are those:

How do you control concurrent Active session using Spring Security? ([answer](#))

Another Spring interview question which is based on Out of box feature provided by Spring framework. You can easily control How many active session a user can have with a Java application by using Spring Security. Apart from that Spring Security also provides the "remember me" feature which you can use to provide easier access for your users by remembering their login details on their personal computer.

How do you set up LDAP Authentication using Spring Security? ([answer](#))

This is a very popular Spring Security interview question as Spring provides out of the box support to connect Windows Active Directory for LDAP authentication and with few configurations in Spring config file you can have this feature enabled. You can further see [Learn Spring Security](#) course by Eugen Paraschiv to learn more about advanced details of Spring Security.



Question1: What is IOC or inversion of control? ([answer](#))

Answer: This *Spring interview question* is the first step towards the Spring framework and many interviewers start Spring interview from this question. As the name implies **Inversion of the control** means now we have inverted the control of creating the object from our own using new operator to container or framework.

Now it's the responsibility of container to create an object as required. We maintain one XML file where we configure our components, services, all the classes and their property. We just need to mention which service is needed by which component and container will create the object for us.

This concept is known as [dependency injection](#) because all object dependency (resources) is injected into it by the framework.

Example:

```
<bean id="createNewStock"
class="springexample.stockMarket.CreateNewStockAccount">
    <property name="newBid"/>
</bean>
```

In this example, `CreateNewStockAccount` class contains getter and setter for `newBid` and container will instantiate `newBid` and set the value automatically when it is used. This whole process is also called wiring in Spring and by using annotation it can be done automatically by Spring, referred as auto-wiring of bean in Spring.

Question 2: Explain the Spring Bean-LifeCycle.

Ans: Spring framework is based on IOC so we call it as IOC container also. So Spring beans reside inside the IOC container. Spring beans are nothing but Plain old java object (POJO). Following steps explain their life cycle inside the container.

1. The container will look the bean definition inside configuration file (e.g. `bean.xml`).
2. Using reflection container will create the object and if any property is defined inside the bean definition then it will also be set.
3. If the bean implements the `BeanNameAware` interface, the factory calls `setBeanName()` passing the bean's ID.
4. If the bean implements the `BeanFactoryAware` interface, the factory calls `setBeanFactory()`, passing an instance of itself.
5. If there are any `BeanPostProcessors` associated with the bean, their `processBeforeInitialization()` methods will be called before the properties for the Bean are set.
6. If an `init()` method is specified for the bean, it will be called.
7. If the Bean class implements the `DisposableBean` interface, then the method `destroy()` will be called when the Application no longer needs the bean reference.
8. If the Bean definition in the Configuration file contains a 'destroy-method' attribute, then the corresponding method definition in the Bean class will be called.

These were just some of the Spring Fundamentals I can cover here if you are interested to learn more I suggest you take a look at [Spring Master Class - Beginner to Expert](#), an end-to-end course to learn Spring.

Question 3: What is Bean Factory, have you used XmlBeanFactory?

Ans: `BeanFactory` is a factory Pattern which is based on IOC [design principles](#). It is used to make a clear separation between application configuration and dependency from actual code. The `XmlBeanFactory` is one of the implementations of Bean Factory which we have used in our project.

The `org.springframework.beans.factory.xml.XmlBeanFactory` is used to create bean instance defined in our XML file.

```
BeanFactory factory = new XmlBeanFactory(new  
FileInputStream("beans.xml"));
```

Or

```
ClassPathResource resorce = new ClassPathResource("beans.xml");  
XmlBeanFactory factory = new XmlBeanFactory(resorce);
```

Question 4: What are the difference between BeanFactory and ApplicationContext in Spring? ([answer](#))

Answer: This one is a very popular Spring interview question and often asks in an entry-level interview. `ApplicationContext` is the preferred way of using spring because of functionality provided by it and the interviewer wanted to check whether you are familiar with it or not.

ApplicationContext.	BeanFactory
Here we can have more than one config files possible	In this only one config file or .xml file
Application contexts can publish events to beans that are registered as listeners	Don't support.
Support internationalization (I18N) messages	It's not
Support application life-cycle events, and validation.	Doesn't support.
Supports many enterprise services such as JNDI access, EJB integration, remoting	Doesn't support.

Questions 5: What is the difference between @Controller and @RestController in Spring MVC? ([answer](#))

Even though both are used to indicate that a Spring bean is a Controller in Spring MVC setup, `@RestController` is better when you are developing RESTful web services using Spring MVC framework. It's a combination of `@Controller` + `@ResponseBody` annotation which allows the controller to directly write the response and bypassing the view resolution process, which is not required for RESTful web service.

It also instructs `DispatcherServlet` to use different `HttpMessageConverters` to represent the response in the format client is expecting e.g. `HttpMessageJackson2Convert` to represent response in JSON format and JAXB based message converts to generate XML response.

You can further see [REST with Spring](#) course by Baeldung to learn more about developing RESTful Web Services using Spring 4 and Spring 5.

Question 6: What is the difference between a singleton and prototype bean?

Ans: This is another popular *spring interview questions* and an important concept to understand. Basically, a bean has scopes which define their existence on the application.

Singleton: means single bean definition to a single object instance per Spring IOC container.

Prototype: means a single bean definition to any number of object instances.

Whatever beans we defined in spring framework are singleton beans.

There is an attribute in bean tag named 'singleton' if specified true then bean becomes singleton and if set to false then the bean becomes a prototype bean. By default, it is set to true. So, all the beans in spring framework are by default singleton beans.

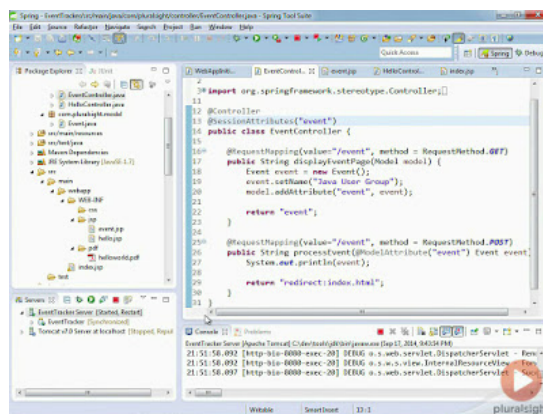
```
<bean id="createNewStock"
      class="springexample.stockMarket.CreateNewStockAccount"
      singleton="false">
    <property name="newBid"/>
</bean>
```

Question 7: What is the role of DispatcherServlet in Spring MVC? ([answer](#))

The DispatcherServlet is very important from Spring MVC perspective, it acts as a FrontController i.e. all requests pass through it. It is responsible for routing the request to the controller and view resolution before sending the response to the client. When Controller returns a Model or View object, it consults all the view resolvers registered to find the correct type of ViewResolver which can render the response for clients.

In case of RESTful Web Services, the DispatcherServlet is also responsible for using `HttpMessageConverter` to represent the response in the JSON, XML, or TEXT format, depending on the content negotiation between Client and Server e.g. if client send request with HTTP accept header as "application/json" then DispatcherServlet will ask the `HttpMessageJackson2Converter` to convert the response into JSON format.

You can further see the free [Introduction to Spring MVC](#) course from Pluralsight to learn more about Spring MVC and DispatcherServlet.



Question 8: What is AOP?

Answer: The core construct of AOP is the aspect, which encapsulates behaviors affecting multiple classes into reusable modules. AOP is a programming technique that allows a developer to modularize crosscutting concerns, that cuts across the typical divisions of responsibility, such as **logging and transaction management**. Spring AOP, aspects are implemented using regular classes or regular classes annotated with the `@Aspect` annotation.

You can also check out these [Spring MVC interview questions](#) for more focus on Java web development using Spring framework.

Question 9: Explain Advice?

Answer: It's an implementation of aspect; advice is inserted into an application at join points. Different types of advice include "around," "before" and "after" advice

Question 10: What are the joint Point and point cut?

Ans: This is not really a spring interview questions I would say an AOP one. Similar to [Object-oriented programming](#), AOP is another popular programming concept which complements OOPS. A join point is an opportunity within the code for which we can apply an aspect. In Spring AOP, a join point always represents a method execution.

Pointcut: a predicate that matches join points. A pointcut is something that defines what join-points an advice should be applied.

Here are few more Spring fundamental interview questions for practice

Question 11: Difference between the setter and constructor injection in Spring? ([answer](#))

Setter injection is more flexible than constructor injection because you must remember the type and order of constructor parameter. Also, constructor injection is generally used to inject the mandatory dependency, while setter can be used to inject optional dependency.

Question 12: How to implement Role Based Access Control (RBAC) using Spring Security? ([answer](#))

Spring Security provides a couple of ways to implement Role based access control e.g. by using GrantedAuthority. See the article to learn more about it.

Question 13: How to call the stored procedure from Java using Spring Framework? ([answer](#))

Question 14: How to Setup JDBC Database connection pool in Spring Web application? ([answer](#))

Question 15: Difference between Factory Pattern and Dependency Injection in Java? ([answer](#))

Even though both allow you to reduce coupling in code, dependency injection is much more flexible and easier to test than Factory pattern.

Starting with Spring 4.0, the framework has provided several useful annotations like `@RestController` to simplify the development of RESTful Web Services in Spring. You can also see [REST with Spring course](#) by Eugen Paraschiv to learn more about developing and security real world RESTful web services using Spring framework.

Question 16: What are the different modules in spring?

Answer: spring has seven core modules

1. The Core container module
2. Application context module
3. AOP module (Aspect Oriented Programming)
4. JDBC abstraction and DAO module
5. O/R mapping integration module (Object/Relational)
6. Web module
7. MVC framework module

Question 17: What type of transaction Management Spring support?

Ans: This spring interview question is a little difficult as compared to previous questions just because **transaction management** is a complex concept and not every developer familiar with it. Transaction management is critical in any applications that will interact with the database.

The application has to ensure that the data is consistent and the integrity of the data is maintained. Following two types of transaction management is supported by spring:

1. Programmatic transaction management
2. Declarative transaction management.

These **spring interview Questions and answers** are not very difficult and focused on spring fundamentals rather than focusing on an advanced feature of session management, spring security, authentication etc. we will cover of those question on some other interview article. I would also suggest that share some spring questions asked to you guys during the interview and then I can put together those with their answers for quick reference of everybody.

Other **Spring Articles and Interview Questions** you may like

[3 Best Practices Java Devs can Learn from Spring](#)

[How Spring MVC works internally in Java?](#)

[10 Things Java Developer should learn in 2018?](#)

[10 Frameworks Java and Web Developer Should learn](#)

[Top 5 Spring Microservice Courses for Java developers](#)

[10 Tips to become a better Java Programmer in 2018](#)

[5 Courses to Learn Spring Boot in 2018](#)

[5 Spring Books Experienced Java Developers Should Read](#)

Thanks for reading this article, if you like this article then please share with your friends and colleagues.

You May Like

New Site Finds the Cheapest Flights in Seconds!

FlightFinder

People from India cannot believe these flight prices

Trips Shop

Sponsored Links