

Spring Framework Job Interview Questions And Answers



Interview Questions Answers

<https://interviewquestionsanswers.org/>

About Interview Questions Answers

Interview Questions Answers . ORG is an interview preparation guide of thousands of Job Interview Questions And Answers, Job Interviews are always stressful even for job seekers who have gone on countless interviews. The best way to reduce the stress is to be prepared for your job interview. Take the time to review the standard interview questions you will most likely be asked. These interview questions and answers on Spring Framework will help you strengthen your technical skills, prepare for the interviews and quickly revise the concepts.

If you find any **question or answer** is incorrect or incomplete then you can **submit your question or answer** directly with out any registration or login at our website. You just need to visit [Spring Framework Interview Questions And Answers](#) to add your answer click on the *Submit Your Answer* links on the website; with each question to post your answer, if you want to ask any question then you will have a link *Submit Your Question*; that's will add your question in Spring Framework category. To ensure quality, each submission is checked by our team, before it becomes live. This [Spring Framework Interview preparation PDF](#) was generated at **Wednesday 29th November, 2023**

You can follow us on FaceBook for latest Jobs, Updates and other interviews material.
www.facebook.com/InterviewQuestionsAnswers.Org

Follow us on Twitter for latest Jobs and interview preparation guides.
<https://twitter.com/InterviewQA>

If you need any further assistance or have queries regarding this document or its material or any of other inquiry, please do not hesitate to contact us.

Best Of Luck.

Interview Questions Answers.ORG Team
<https://InterviewQuestionsAnswers.ORG/Support@InterviewQuestionsAnswers.ORG>



Spring Framework Interview Questions And Answers Guide.

Question - 1:

What are the different types of AutoProxying?

Ans:

- * BeanNameAutoProxyCreator
- * DefaultAdvisorAutoProxyCreator
- * Metadata autoproxying

[View All Answers](#)

Question - 2:

What are the different advice types in spring?

Ans:

- * Around : Intercepts the calls to the target method
- * Before : This is called before the target method is invoked
- * After : This is called after the target method is returned
- * Throws : This is called when the target method throws an exception
- * Around : org.aopalliance.intercept.MethodInterceptor
- * Before : org.springframework.aop.BeforeAdvice
- * After : org.springframework.aop.AfterReturningAdvice
- * Throws : org.springframework.aop.ThrowsAdvice

[View All Answers](#)

Question - 3:

What are the different points where weaving can be applied?

Ans:

- * Compile Time
- * Classload Time
- * Runtime

[View All Answers](#)

Question - 4:

What is meant by Weaving?

Ans:

The process of applying aspects to a target object to create a new proxy object is called as Weaving. The aspects are woven into the target object at the specified joinpoints.

[View All Answers](#)

Question - 5:

What is a Proxy?

Ans:

A proxy is an object that is created after applying advice to a target object. When you think of client objects the target object and the proxy object are the same.

[View All Answers](#)

Question - 6:

What is a Target?

Ans:

A target is the class that is being advised. The class can be a third party class or your own class to which you want to add your own custom behavior. By using the concepts of AOP, the target class is free to center on its major concern, unaware to any advice that is being applied.



[View All Answers](#)

Question - 7:

What is an Introduction in AOP?

Ans:

An introduction allows the user to add new methods or attributes to an existing class. This can then be introduced to an existing class without having to change the structure of the class, but give them the new behavior and state.

[View All Answers](#)

Question - 8:

What is a Pointcut?

Ans:

A pointcut is something that defines at what joinpoints an advice should be applied. Advices can be applied at any joinpoint that is supported by the AOP framework. These Pointcuts allow you to specify where the advice can be applied.

[View All Answers](#)

Question - 9:

What is an Advice?

Ans:

Advice is the implementation of an aspect. It is something like telling your application of a new behavior. Generally, an advice is inserted into an application at joinpoints.

[View All Answers](#)

Question - 10:

What is a Joinpoint?

Ans:

A joinpoint is a point in the execution of the application where an aspect can be plugged in. This point could be a method being called, an exception being thrown, or even a field being modified. These are the points where your aspect's code can be inserted into the normal flow of your application to add new behavior.

[View All Answers](#)

Question - 11:

What is an Aspect?

Ans:

An aspect is the cross-cutting functionality that you are implementing. It is the aspect of your application you are modularizing. An example of an aspect is logging. Logging is something that is required throughout an application. However, because applications tend to be broken down into layers based on functionality, reusing a logging module through inheritance does not make sense. However, you can create a logging aspect and apply it throughout your application using AOP.

[View All Answers](#)

Question - 12:

What are different types of Autowire types?

Ans:

There are four different types by which autowiring can be done.

- * byName
- * byType
- * constructor
- * autodetect

[View All Answers](#)

Question - 13:

What are the different types of bean injections?

Ans:

There are two types of bean injections.

1. By setter
2. By constructor

[View All Answers](#)

Question - 14:

What are Inner Beans?

Ans:

When wiring beans, if a bean element is embedded to a property tag directly, then that bean is said to be an Inner Bean. The drawback of this bean is that it cannot be reused anywhere else.

[View All Answers](#)

**Question - 15:**

What are the important beans lifecycle methods?

Ans:

There are two important bean lifecycle methods. The first one is setup which is called when the bean is loaded in to the container. The second method is the teardown method which is called when the bean is unloaded from the container.

[View All Answers](#)

Question - 16:

How to integrate your Struts application with Spring?

Ans:

To integrate your Struts application with Spring, we have two options:

- * Configure Spring to manage your Actions as beans, using the ContextLoaderPlugin, and set their dependencies in a Spring context file.
- * Subclass Spring's ActionSupport classes and grab your Spring-managed beans explicitly using a getWebApplicationContext() method.

[View All Answers](#)

Question - 17:

What is Significance of JSF- Spring integration?

Ans:

Spring - JSF integration is useful when an event handler wishes to explicitly invoke the bean factory to create beans on demand, such as a bean that encapsulates the business logic to be performed when a submit button is pressed.

[View All Answers](#)

Question - 18:

Explain Bean lifecycle in Spring framework?

Ans:

1. The spring container finds the bean's definition from the XML file and instantiates the bean.
2. Using the dependency injection, spring populates all of the properties as specified in the bean definition.
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 post- ProcessBeforeInitialization() methods will be called.
6. If an init-method is specified for the bean, it will be called.
7. Finally, if there are any BeanPostProcessors associated with the bean, their postProcessAfterInitialization() methods will be called.

[View All Answers](#)

Question - 19:

What is XMLBeanFactory?

Ans:

BeanFactory has many implementations in Spring. But one of the most useful one is org.springframework.beans.factory.xml.XmlBeanFactory, which loads its beans based on the definitions contained in an XML file. To create an XmlBeanFactory, pass a java.io.InputStream to the constructor. The InputStream will provide the XML to the factory. For example, the following code snippet uses a java.io.FileInputStream to provide a bean definition XML file to XmlBeanFactory.

[View All Answers](#)

Question - 20:

What does a simple spring application contain?

Ans:

These applications are like any Java application. They are made up of several classes, each performing a specific purpose within the application. But these classes are configured and introduced to each other through an XML file. This XML file describes how to configure the classes, known as theSpring configuration file.

[View All Answers](#)

Question - 21:

What is Spring configuration file?

Ans:

Spring configuration file is an XML file. This file contains the classes information and describes how these classes are configured and introduced to each other.

[View All Answers](#)

Question - 22:

What is AOP Alliance?

Ans:

AOP Alliance is an open-source project whose goal is to promote adoption of AOP and interoperability among different AOP implementations by defining a common set of interfaces and components.

[View All Answers](#)

Question - 23:

What is web module?

**Ans:**

This module is built on the application context module, providing a context that is appropriate for web-based applications. This module also contains support for several web-oriented tasks such as transparently handling multipart requests for file uploads and programmatic binding of request parameters to your business objects. It also contains integration support with Jakarta Struts.

[View All Answers](#)

Question - 24:

What are object/relational mapping integration module?

Ans:

Spring also supports for using of an object/relational mapping (ORM) tool over straight JDBC by providing the ORM module. Spring provide support to tie into several popular ORM frameworks, including Hibernate, JDO, and iBATIS SQL Maps. Spring's transaction management supports each of these ORM frameworks as well as JDBC.

[View All Answers](#)

Question - 25:

What is JDBC abstraction and DAO module?

Ans:

Using this module we can keep up the database code clean and simple, and prevent problems that result from a failure to close database resources. A new layer of meaningful exceptions on top of the error messages given by several database servers is bought in this module. In addition, this module uses Spring's AOP module to provide transaction management services for objects in a Spring application.

[View All Answers](#)

Question - 26:

What is AOP module?

Ans:

The AOP module is used for developing aspects for our Spring-enabled application. Much of the support has been provided by the AOP Alliance in order to ensure the interoperability between Spring and other AOP frameworks. This module also introduces metadata programming to Spring. Using Spring's metadata support, we will be able to add annotations to our source code that instruct Spring on where and how to apply aspects.

[View All Answers](#)

Question - 27:

What is Application context module?

Ans:

The Application context module makes spring a framework. This module extends the concept of BeanFactory, providing support for internationalization (I18N) messages, application lifecycle events, and validation. This module also supplies many enterprise services such JNDI access, EJB integration, remoting, and scheduling. It also provides support to other framework.

[View All Answers](#)

Question - 28:

What is the Core container module?

Ans:

This module is provides the fundamental functionality of the spring framework. In this module BeanFactory is the heart of any spring-based application. The entire framework was built on the top of this module. This module makes the Spring container.

[View All Answers](#)

Question - 29:

What are the different modules in Spring framework?

Ans:

- * The Core container module
- * Application context module
- * AOP module (Aspect Oriented Programming)
- * JDBC abstraction and DAO module
- * O/R mapping integration module (Object/Relational)
- * Web module
- * MVC framework module

[View All Answers](#)

Question - 30:

What is DelegatingVariableResolver?

Ans:

Spring provides a custom JavaServer Faces VariableResolver implementation that extends the standard Java Server Faces managed beans mechanism which lets you use JSF and Spring together. This variable resolver is called as DelegatingVariableResolver

[View All Answers](#)

Question - 31:



What do you mean by Auto Wiring?

Ans:

The Spring container is able to autowire relationships between collaborating beans. This means that it is possible to automatically let Spring resolve collaborators (other beans) for your bean by inspecting the contents of the BeanFactory. The autowiring functionality has five modes.

- * no
- * byName
- * byType
- * constructor
- * autodirect

[View All Answers](#)

Question - 32:

What do you mean by Bean wiring?

Ans:

The act of creating associations between application components (beans) within the Spring container is referred to as Bean wiring.

[View All Answers](#)

Question - 33:

What is the typical Bean life cycle in Spring Bean Factory Container?

Ans:

Bean life cycle in Spring Bean Factory Container is as follows:

- * The spring container finds the bean's definition from the XML file and instantiates the bean.
- * Using the dependency injection, spring populates all of the properties as specified in the bean definition
- * If the bean implements the BeanNameAware interface, the factory calls setBeanName() passing the bean's ID.
- * If the bean implements the BeanFactoryAware interface, the factory calls setBeanFactory(), passing an instance of itself.
- * If there are any BeanPostProcessors associated with the bean, their post- ProcessBeforeInitialization() methods will be called.
- * If an init-method is specified for the bean, it will be called.
- * Finally, if there are any BeanPostProcessors associated with the bean, their postProcessAfterInitialization() methods will be called.

[View All Answers](#)

Question - 34:

How is a typical spring implementation look like?

Ans:

For a typical Spring Application we need the following files:

- * An interface that defines the functions.
- * An Implementation that contains properties, its setter and getter methods, functions etc.,
- * Spring AOP (Aspect Oriented Programming)
- * A XML file called Spring configuration file.
- * Client program that uses the function.

[View All Answers](#)

Question - 35:

What is the difference between Bean Factory and Application Context?

Ans:

On the surface, an application context is same as a bean factory. But application context offers much more..

- * Application contexts provide a means for resolving text messages, including support for i18n of those messages.
- * Application contexts provide a generic way to load file resources, such as images.
- * Application contexts can publish events to beans that are registered as listeners.
- * Certain operations on the container or beans in the container, which have to be handled in a programmatic fashion with a bean factory, can be handled declaratively in an application context.
- * ResourceLoader support: Spring's Resource interface us a flexible generic abstraction for handling low-level resources. An application context itself is a ResourceLoader, Hence provides an application with access to deployment-specific Resource instances.
- * MessageSource support: The application context implements MessageSource, an interface used to obtain localized messages, with the actual implementation being pluggable

[View All Answers](#)

Question - 36:

What is Application Context?

Ans:

A bean factory is fine to simple applications, but to take advantage of the full power of the Spring framework, you may want to move up to Springs more advanced container, the application context. On the surface, an application context is same as a bean factory.Both load bean definitions, wire beans together, and dispense beans upon request. But it also provides:

- * A means for resolving text messages, including support for internationalization.
- * A generic way to load file resources.
- * Events to beans that are registered as listeners.

[View All Answers](#)

Question - 37:

What is Bean Factory?

**Ans:**

A BeanFactory is like a factory class that contains a collection of beans. The BeanFactory holds Bean Definitions of multiple beans within itself and then instantiates the bean whenever asked for by clients.

* BeanFactory is able to create associations between collaborating objects as they are instantiated. This removes the burden of configuration from bean itself and the beans client.

* BeanFactory also takes part in the life cycle of a bean, making calls to custom initialization and destruction methods.

[View All Answers](#)

Question - 38:

What are the types of Dependency Injection Spring supports?

Ans:

* Setter Injection:

Setter-based DI is realized by calling setter methods on your beans after invoking a no-argument constructor or no-argument static factory method to instantiate your bean.

* Constructor Injection:

Constructor-based DI is realized by invoking a constructor with a number of arguments, each representing a collaborator.

[View All Answers](#)

Question - 39:

How many modules are there in Spring? What are they?

Ans:

Spring comprises of seven modules. They are..

* The core container:

The core container provides the essential functionality of the Spring framework. A primary component of the core container is the BeanFactory, an implementation of the Factory pattern. The BeanFactory applies the Inversion of Control (IOC) pattern to separate an application's configuration and dependency specification from the actual application code.

* Spring context:

The Spring context is a configuration file that provides context information to the Spring framework. The Spring context includes enterprise services such as JNDI, EJB, e-mail, internationalization, validation, and scheduling functionality.

* Spring AOP:

The Spring AOP module integrates aspect-oriented programming functionality directly into the Spring framework, through its configuration management feature. As a result you can easily AOP-enable any object managed by the Spring framework. The Spring AOP module provides transaction management services for objects in any Spring-based application. With Spring AOP you can incorporate declarative transaction management into your applications without relying on EJB components.

[View All Answers](#)

Question - 40:

What are features of Spring?

Ans:

* Lightweight:

spring is lightweight when it comes to size and transparency. The basic version of spring framework is around 1MB. And the processing overhead is also very negligible.

* Inversion of control (IOC):

Loose coupling is achieved in spring using the technique Inversion of Control. The objects give their dependencies instead of creating or looking for dependent objects.

* Aspect oriented (AOP):

Spring supports Aspect oriented programming and enables cohesive development by separating application business logic from system services.

* Container:

Spring contains and manages the life cycle and configuration of application objects.

* MVC Framework:

Spring comes with MVC web application framework, built on core Spring functionality. This framework is highly configurable via strategy interfaces, and accommodates multiple view technologies like JSP, Velocity, Tiles, iText, and POI. But other frameworks can be easily used instead of Spring MVC Framework.

* Transaction Management:

Spring framework provides a generic abstraction layer for transaction management. This allowing the developer to add the pluggable transaction managers, and making it easy to demarcate transactions without dealing with low-level issues. Spring's transaction support is not tied to J2EE environments and it can be also used in container less environments.

* JDBC Exception Handling:

The JDBC abstraction layer of the Spring offers a meaningful exception hierarchy, which simplifies the error handling

[View All Answers](#)

Question - 41:

What are the advantages of Spring framework?

Ans:

The advantages of Spring are as follows:

* Spring has layered architecture. Use what you need and leave you don't need now.

* Spring Enables POJO Programming. There is no behind the scene magic here. POJO programming enables continuous integration and testability.

* Dependency Injection and Inversion of Control Simplifies JDBC

* Open source and no vendor lock-in.

[View All Answers](#)

Question - 42:

What is Spring?



Ans:

Spring is an open source framework created to address the complexity of enterprise application development. One of the chief advantages of the Spring framework is its layered architecture, which allows you to be selective about which of its components you use while also providing a cohesive framework for J2EE application development.

[View All Answers](#)

Interview Questions Answers.ORG

Java Programing Most Popular & Related Interview Guides

- 1 : [Core Java Interview Questions and Answers.](#)
- 2 : [Hibernate Interview Questions and Answers.](#)
- 3 : [IBM WebSphere Interview Questions and Answers.](#)
- 4 : [Advanced Java Interview Questions and Answers.](#)
- 5 : [Full Stack Developer \(Java\) Interview Questions and Answers.](#)
- 6 : [JSF Interview Questions and Answers.](#)
- 7 : [JDBC Interview Questions and Answers.](#)
- 8 : [Java Swing Programming Interview Questions and Answers.](#)
- 9 : [Java JSP Programming Interview Questions and Answers.](#)
- 10 : [J2EE Interview Questions and Answers.](#)

Follow us on FaceBook

www.facebook.com/InterviewQuestionsAnswers.Org

Follow us on Twitter

<https://twitter.com/InterviewQA>

For any inquiry please do not hesitate to contact us.

Interview Questions Answers.ORG Team

[https://InterviewQuestionsAnswers.ORG/
support@InterviewQuestionsAnswers.ORG](https://InterviewQuestionsAnswers.ORG/support@InterviewQuestionsAnswers.ORG)