Last updated : June 15th, 2018 NO COMMENTS

Spring interview questions and answers

« Previous Next »

In this post, we will see Spring interview interview questions. If you are java/j2ee developer and have some experienced on Spring, you are going to face Spring interview questions.

Spring Interview Questions and Answers

Here I am sharing a list of important Spring interview questions.

Hot Spring interview questions

- Circular dependencies in Spring
- Injecting prototype bean into Singleton bean

1. What is Spring framework?

Spring framework is an open source framework created to solve 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. Main module for Spring are Spring core,Spring AOP and Spring MVC.

2. What are main features of Spring frameworks?

- **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):**
  The basic concept of the Dependency Injection or Inversion of Control is that, programmer do not need to create the objects, instead just describe how it should be created.
- **Aspect oriented (AOP):**

  Spring supports Aspect oriented programming .
  Aspect oriented programming refers to the programming paradigm which isolates secondary or supporting functions from the main program's business logic. AOP is a promising technology for separating crosscutting concerns, something usually hard to do in object-oriented programming. The application's modularity is increased in that way and its maintenance becomes significantly easier.
- **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.
- **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.

- **JDBC Exception Handling:**
  The JDBC abstraction layer of the Spring offers a meaningful exception hierarchy, which simplifies the error handling strategy. Integration with Hibernate, JDO, and iBATIS: Spring provides best Integration services with Hibernate, JDO and iBATIS

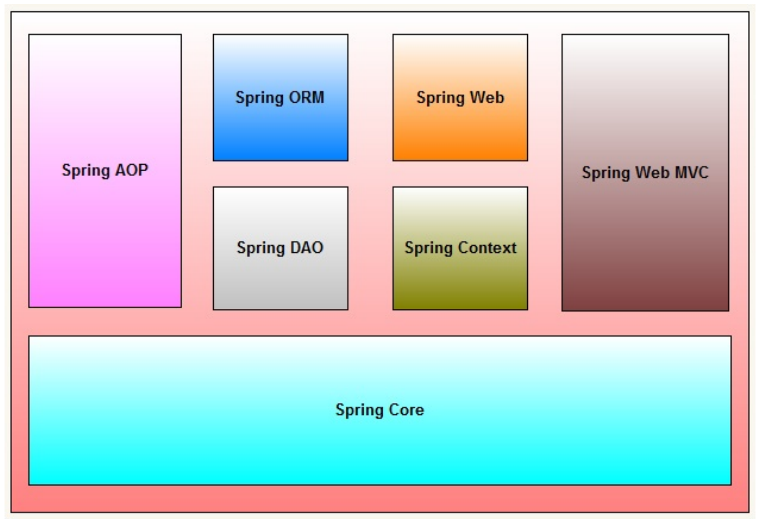
3. Explain main modules of Spring ?

- **Spring AOP**:
  One of the key components of Spring is the *AOP framework*. AOP is used in Spring:

  - To provide declarative enterprise services, especially as a replacement for EJB declarative services. The most important such service is *declarative transaction management*, which builds on Spring's transaction abstraction.
  - To allow users to implement custom aspects, complementing their use of OOP with AOP
- **Spring ORM**:
  The *ORM* package is related to the database access. It provides integration layers for popular object-relational mapping APIs, including JDO, Hibernate and iBatis.
- **Spring Web**:
  The Spring Web module is part of Spring?s web application development stack, which includes Spring MVC.
- **Spring DAO:**
  The DAO (Data Access Object) support in Spring is primarily for standardizing the data access work using the technologies like JDBC, Hibernate or JDO.
- **Spring Context**:
  This package builds on the beans package to add support for message sources and for the Observer

design pattern, and the ability for application objects to obtain resources using a consistent API.

- **Spring Web MVC**:
  This is the Module which provides the MVC implementations for the web applications.
- **Spring Core**:
  The *Core* package is the most import component of the Spring Framework.
  This component provides the Dependency Injection features. The BeanFactory provides a factory pattern which separates the dependencies like initialization, creation and access of the objects from your actual program logic.



Spring Framework Architecture

4. What is dependency injection(IOC) in Spring?

The basic concept of the dependency injection (also known as Inversion of Control pattern) is that you do not create your objects but describe how they should be created. You don't directly connect your components and services together in code but describe which services are needed by which components in a configuration file. A container (in the

case of the Spring framework, the IOC container) is then responsible for hooking it all up.
i.e., Applying IoC, objects are given their dependencies at creation time by some external entity that coordinates each object in the system. That is, dependencies are injected into objects. So, IoC means an inversion of responsibility with regard to how an object obtains references to collaborating objects.

Don't call us,We shall call you!!!

5. What are ways to inject dependency in Spring?

There are two ways to do dependency injection in Spring.

- Dependency injection via setter method
- Dependency injection via constructor.

6. What is Bean in Spring?

A normal POJO class managed by Spring IOC container are called Spring beans. It is core part of Spring application.
Example:

```

<bean id="countryBean" class="org.arpit.j
      <property name="countryName" value=
      <property name="capital" ref="Capit
</bean>

```

7. How can you configure Spring in your application?

There are 3 ways to do it.

- XML based configuration

- Java based configuration
- Annotation based configuration.

8. What is Spring XML based configuration?

In Spring XML based configuration, you define all dependency in an XML file. You define all your beans with tag in XML file and all dependencies are read using this XML file.
for example :
Sample ApplicationContext.xml file

```

<!--?xml version="1.0" encoding="UTF-8"?
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework
xmlns:xsi="http://www.w3.org/2001/XMLSch
xsi:schemaLocation="http://www.springfra

   <bean id="CountryBean" class="org.arpi
       <property name="countryName" value
       <property name="capital" ref="Capi
   </bean>
   <bean id="CapitalBean" class="org.arpi
       <property name="capitalName" value
   </bean>
</beans>

```

You can read this ApplicationContext.xml using:

```

ApplicationContext appContext = new Clas

```

9. What is Spring java based configuration?

In Spring Java based configuration, you inject all dependencies using java class only. You can use @Configuaration and @Bean annotations to do it.

```
package org.arpit.java2blog.config;
import org.arpit.java2blog.model.Country
import org.springframework.context.annot
import org.springframework.context.annot

@Configuration
public class ApplicationConfiguration {

 @Bean(name="countryObj")
 public Country getCountry()
 {
  return new Country("India");
 }

}
```

Above file is equivalent to below spring configuration xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework
    xmlns:xsi="http://www.w3.org/2001/XM
    xmlns:context="http://www.springfram
    xsi:schemaLocation="http://www.sprin
    http://www.springframework.org/schem
    http://www.springframework.org/schem
    http://www.springframework.org/schem
<context:annotation-config/>
 <bean id="countryObj" class="org.arpit.
  <property name="countryName" value="In
 </bean>
</beans>
```

To get above bean to application context, you need to use below code

```
ApplicationContext appContext = new Annot
 Country countryObj = (Country) appContex
```

You can refer Spring java based configuration for complete example.

10. What is spring annotation based configuration?

You can do dependency injection via annotation also instead of XML configuration. You can define bean autowiring using annotations. You can use @Component,@Repository,@Service and @Controller annotation to configure bean in Spring application.
Annotations wiring is not turned on by default. You need to turn it on using :

```

<?xml version="1.0" encoding="UTF-8"?>
<bean>
<context:annotation-config/>
</beans>

```

Once you put above code, you can start using annotation on class , fields or methods.

11. What are different bean scopes in Spring?

There are 5 types of bean scopes supported in spring

- **singleton** – Scopes a single bean definition to a single object instance per Spring IoC container.
- **prototype** – Return a new bean instance each time when requested
- **request** – Return a single bean instance per HTTP request.
- **session** – Return a single bean instance per HTTP session.
- **globalSession** – Return a single bean instance per global HTTP session.

12. What is default scope of bean in Spring?

singleton is default scope of a bean in Spring. You have to explicitly change scope of a bean if you want different scope.This is one of most asked spring interiew questions.

13. What is ApplicationContext and what are its functions?

ApplicationContext is an central interface for providing configuration information to an application.
An ApplicationContext provides the following functionalities:

- Bean factory methods, inherited from ListableBeanFactory. This avoids the need for applications to use singletons.
- The ability to resolve messages, supporting internationalization. Inherited from the MessageSource interface.
- The ability to load file resources in a generic fashion. Inherited from the ResourceLoader interface.
- The ability to publish events. Implementations must provide a means of registering event listeners.
- Inheritance from a parent context. Definitions in a descendant context will always take priority. This means, for example, that a single parent context can be used by an entire web application, while each servlet has its own child context that is independent of that of any other servlet.

14. How do you injection collection in Spring?

You can initialize collection using list and value tag as below:

```

 <bean id="CountryBean" class="org.arpit
  <property name="listOfStates">
   <list>
    <value>Himachal Pradesh</value>
    <value>West Bengal</value>
    <value>Gujrat</value>
   </list>
  </property>
</bean>

```

15. What do you mean by bean autowiring in Spring?

In Spring framework, you can wire beans automatically with auto-wiring feature. To enable it, just define the "**autowire**" attribute in .The Spring container can **autowire** relationships between collaborating beans without using and elements which helps cut down on the amount of XML configuration

```

<bean id="countryBean" class="org.arpit.j

```

16. What are different modes of autowiring supported by Spring?

There are following autowiring modes which can be used to instruct Spring container to use autowiring for dependency injection.
no:
Default, no auto wiring, set it manually via "ref" attribute as we have done in dependency injection via settor method post.
byName:
Autowiring by property name. Spring container looks at the properties of the beans on which *autowire* attribute is set to *byName* in the XML configuration file and it tries to match it with name of bean in xml configuration file.
byType:
Autowiring by property datatype. Spring container looks at the properties of the beans on

which *autowire* attribute is set to *byType* in the XML configuration file. It then tries to match and wire a property if its **type** matches with exactly one of the beans name in configuration file. If more than one such beans exists, a fatal exception is thrown.

contructor:

byType mode in constructor argument.

autodetect:

Spring first tries to wire using autowire by *constructor*, if it does not work, Spring tries to autowire by *byType*.

17. What is Spring AOP?

Aspect oriented Programming is programming paradigm which is analogous to object oriented programming. Key unit of object oriented programming is class, similarly key unit for AOP is Aspect. Aspect enable modularisation of concerns such as transaction management, it cut across multiple classes and types. It also refers as a crosscutting concerns.

18. What is Aspect, Advice, Join point and pointcut in Spring AOP?

Aspect: An Aspect is a class that implements concerns that cut across different classes such as logging. It is just a name.

Joint Point : It is a point in execution of program such as execution of method. In Spring AOP, a join point always represents a method execution.

Advice : Action taken by aspect at particular join point. For example: Before execution of getEmployeeName() method, put logging. So here, we are using before advice.

Pointcut : Pointcut is an expression that decides execution of advice at matched joint point. Spring uses the AspectJ pointcut expression language by default.

19. What is @Qualifier annotation in Spring?

You can have more than one bean of same type in your XML configuration but you want to autowire only one of them ,so @Qualifier removes confusion created by @Autowired by declaring exactly which bean is to autowired.
You can read about Spring @Qualifier annotation for more details.

20. What is @Required annotation in Spring?

This annotation simply indicates that the affected bean property must be populated at configuration time: either through an explicit property value in a bean definition or through autowiring. The container will throw an exception if the affected bean property has not been populated; this allows for eager and explicit failure, avoiding NullPointerExceptions or the like later on.
Suppose you have very large application and you get NullPointerExceptions because required dependency has not been injected then it is very hard to find out what goes wrong.So this annotation helps us in debugging the code.
You can read about Spring @Required annotation for more details.

21. How will you handle Circular dependencies in Spring?

refer Circular dependencies in Spring for more details.

22. What happens when you inject prototype bean into singleton bean? Can you explain the behavior?

refer injecting prototype bean into singleton bean in Spring for more details.

That's all about Spring interview questions.

You may also like:

- web services interview questions
- restful web services interview questions
- Data structure and algorithm Interview Questions
- Hibernate interview questions
- Core java interview questions
- Java Collections interview questions
- Java String interview questions
- OOPs interview questions in java
- Java Multithreading interview questions
- Exceptional handling interview questions in java
- Java Serialization interview questions in java

« 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.

Pinterest

Related Posts

Java 8 interview questions

Java interview questions for 5 years experience

No qualifying bean of type in Spring or Spring Boot

Longest Substring Without Repeating Characters

Spring XML configuration example

Java interview questions

Add Comment

Comment Text*

Name*

Email*

Website

POST COMMENT