Spring Framework FAQs

Chaper#1 Spring Core

1. What is Spring?

A) Spring is framework which has in-built design patterns and technologies provided supports for faster application development (RAD=Rapid Application Development).

2. What are the benefits of Spring?

A)

1. It provides pre-defined Template for code reducing,
2. Loosely coupled Design Pattern,
3. Lightweight container (IoC),
4. Rapid Application development
5. Supports XML, Java and Annotations configuration
6. Life Cycle methods configuration
7. Easy ways of Scope definitions

3. What are different modules/chapters in Spring

A) Spring Core, JDBC, ORM, WEB MVC, EMAIL, JMS(Java Message Service), restful, Scheduling, JAAS, AOP.

4. What is the difference between DI and IoC?

A) DI= Dependency Injection : It is a process of injecting child object into its parent object at runtime using configuration file (ex: XML,Java..)

IoC = Inversion of Control is a processing creating objects with data from parent to child using DI.

In simple, DI is theory and IoC means Program. It is used to achieve loosely coupling between classes

(specially in Project Layers)

5. What are different types of Container (or) IoCs in Spring?

A) Spring f/w has provided two types of IoCs or containers. Those are

BeanFactory (I) : Only supports XML Configuration has one Impl class XmlBeanfactory

ApplicationContext(I): it is extension to BeanFactory, it is also called as new Container.

It supports XML, java with Annotation Configuration. It is faster and light weight compared to BeanFactory.

6. What is the work of Spring container or IoC?

A) Creating Objects, Providing data, linking one object with another object, calling life cycle methods finally destroy the objects.

7. What are different lifecycle methods supports by Spring ? Explain them?

A) These are two optional life cycle methods provided by Spring container. Those are init() and destroy(). init() called after creating object with data and destroy() gets called by container before destroying the object.

1. XML Configuration : <bean .. Init-method="\_\_" destroy-method="\_\_"
2. Java Configuration: InitializingBean(C)-afterPropertiesSet(),DisposableBean(C)-destroy()
3. Annotations : @PostConstruct and @PreDestroy

8. What are different scopes provided by Spring container and explain LMI?

1. singleton: one Object per one <bean>(default scope)
2. prototype: Every time new object when requested.
3. request: New object for every HTTP request.
4. session: One object for one HTTP session.
5. global session: works for portlet only

LMI = When Parent is singleton and child is prototype then container will not look for new child object to inject with parent.

9. What are different types of Injection ?

A) 4 types. Those are:

1. Setter Injection = Provide data using set method (generally uses default constructor)
2. Constructor Injection = Provide data using Parameter constructor
3. Lookup Method Injection = only used Parent bean is singleton and child bean is prototype
4. Interface Injection = Not supported by Spring container.

10. What is the difference between constructor injection and setter injection?

|  |  |
| --- | --- |
| Setter Injection | Constructor Injection |
| We set pass few values(Parital injection) | All values must be passed |
| Provides data using set method | Provide data using param constructor |
| Use only if few values need to be set | Use only if all values need to be set |
| Order not required | Order is followed |
| Use <property> tag | Use <constructor-arg> tag |
| It will create a new instance if any modification is done. | It will not create new instance if any modification is done |

11. Explain Java Configuration Steps with Example?

A) We need to define one public class which gets annotated with @Configuration.

Define Methods inside class , that one method behaves as one object in it.

@Bean annotation must be applied over method. By default method name behaves as object name.

12. What is Bean Externalization ?

A) To remove hard coding values in Configuration , data is loaded from Properties file using either XML or Java Configuration, this process is called as Bean Externalization.

To load properties file using XML : <context:property-placeholder location="\_\_.properties"/>

Java : @PropertySource("\_\_.properties").

13. What are stereotype Annotation in Spring?

A) An annotation which detect the class and create the object is called as sterotype Annotation.

These are 5 types :

1. @Component : Create object to any non-abstract class
2. @Repository : Creating Object + Supports JDBC/ORM Operations setup
3. @Service : Creating Object + Transaction Management + Logics, calculations
4. @Controller : Creating Object + Web Application Setup.
5. @RestController : Creating Object + restful webservices support (@ResposenBody support)

14. What are Stand Alone Collections ? What are different ways of creating them?

A) An independent and Re-usable collection is called as Stand Alone Collection (SAC).

Using XML with util schema: <util:collectionname ..> </util:collectionname>

Using Java : @Bean over method returns Any Collection Type.

15. Explain @Value annotation usage in Configuration?

A)

1. It is used to provide static data to any primitive variable,
2. Supports SpEL (Spring Expression Language)
3. Supports Reading Data from properties file (${key})
4. Supports reading SAC into Collection Type Dependencies #{key}
5. Supports Linking Parent with child object (#{childobject})

16. What are different Types of dependencies in Spring?

A) 3 Types. Those are Primitive Types(8+1), Collection Types(4) and Reference types(HAS-A)

17. How to inject null value using XML Configuration file?

A) using <null/> or <null></null> tag

18. What are different ways of providing primitive data using XML?

A) value as tag ( <value>\_</value>),

Value as attribute (<property name="" value=""/>)

P-schema/p-namespace (<bean .. P:variablename="data"..)

19. What are different ways of writing reference type (link) using XML?

A) ref as tag (<ref bean="childbeanname"/>)

Ref as attribute (<property name=" " ref="childobjname"/>)

P-schema/p-namespace (<bean .. P:variablename-ref="childobjname"..)

Inner Bean (Child bean inside Parent Bean)

20. What are different ways of providing entry for a map using XML?

A) 4 ways.

1. Key and Values as Tag , (<entry> <key>..</key> <value>..</value> </entry>)
2. Key and value as attributes(<entry key="" value="" />)
3. Key as tag and value as attribute(<entry value=""> <key>..</key> </entry>),
4. Key as attribute and value as tag (<entry key=""> <value>..</value> </entry>)

21. What happens if same key is repeated in Map Configuration?

A) Value is overridden with last combination. Even for Properties also same.

22. What are default implementation used by container for Collections?

A) List - ArrayList, Set-LinkedHashSet, Map-LinkedHashMap.

23. When <list> or <set> tags are optional in XML Configuration?

A) If List or Set contains only one value.

24. What is dependency check?

A)Making one dependency (variable) mandatory (@Required over its set method)

25. What is wiring and auto-wiring ? Explain types of Autowiring?

A) Wiring is a process of linking parent bean and child bean by writing <ref/> tag manually.

Autowiring process creates links between them using container rules.

Types of Autowiring:

1. no(none) No Autowiring (default value)
2. byName: based on HAS-A variable name and child <bean name="") (setter injection)
3. byType: based on child classname name and child <bean class="") (setter injection)
4. constructor: It injects the child using Param constructor.

@Autowired : This annotation behaves by default like bytype, if multiple found then by name.

It follows internally required=true.

26. What is the use of @Qualifier?

A)This is used to choose one object in case of multiple child beans found for injection with Parent bean.

27. What are few Common Exceptions found in Spring Application?

A) NoSuchBeanDefinitionException, NoUniqueBeanDefinitionException, TypeMismatchException,..etc

28. Write XML, Java and Annotation Configuration using all types of Dependencies.

29. Write Example for lifecycle methods

30. Write Example for Inner Bean using XML Configuration

31. Write Example for sacs using XML and Java Configs.

32. Write Example for Loading and Reading Properties file in 3 configurations.

33. Writer Example for Scopes and LMI Solution

34. Explain Circular Dependency using XML with steps.

35. Write Example for Map configuration in XML using different <entry> styles.

36. What are the rules to write Spring Bean?

37. Is Spring Bean Thread Safe?

A)No by default, because scope is singleton. To make it thread safe make it is prototype or session.

38. Can we define multiple configuration file? How to handle them ?

A) Yes. But Import one into another finally.

Using XML <import resource="\_\_.xml"/>

Using Java @Import(\_\_\_.class)

39. What are in-built design Patterns used by Spring?

A) Few are given below

1. Singleton Pattern: Singleton-scoped beans
2. Factory Pattern: Bean Factory classes
3. Prototype Pattern: Prototype-scoped beans
4. Adapter Pattern: Spring Web and Spring MVC
5. Proxy Pattern: Spring Aspect Oriented Programming support
6. Template Method Pattern: jdbctemplate, hibernatetemplate, etc.
7. Front Controller: Spring MVC dispatcherservlet
8. Data Access Object: Spring DAO support
9. Model View Controller: Spring MVC

40. Which Layers Spring f/w supports coding (or implementation)?

A) Spring supports 4 layers. Those are Presentation Layer, Service Layer, Data Access Layer and Integration Layer.