

6 Explain Spring Architecture with Diagram.

⇒ Spring Architecture mainly consist Four Part.

(1) Core Container

(2) Data Access / Integration

(3) Web

(4) Miscellaneous

Data Access /
Integration

JDBC ORM
OXM JMS
Transaction

Web (MVC)

WebSocket + Web
Servlet +
Portlet

AOP Aspects Instrumentation - Messaging

Core Container

Beans Core Context SpEL

Test

(1) Core Container:

Core Container is consist Beans, Core, Context and SpEL.

Core: Core provides the fundamental parts of the Framework.

Bean: Bean provides the ~~to~~ BeanFactory.

Context: Context module builds on the solid base provided by the Core and Beans.

SpEL: SpEL module provides a powerful expression language for Query.

(2) **Data Access / Integration**:

Data Access / Integration consists of JDBC, ORM, OXM, JMS and Transaction.

JDBC: JDBC module provides a JDBC Abstraction Layer.

ORM: ORM module provides integration layer for object relational mapping.

OXM: OXM module provides an abstraction layer that supports Object / XML mapping.

Jms: Jms module contains features for provide and consume the messages.

Transaction: Transaction modules support transaction management.

(3) Web (MVC):

Web layer consist of web, socket, portlet and MVC.

Web - web module provides basic web-oriented integration features.

Socket: Socket provides two way communication between client and server.

Portlet: Portlet module provides the MVC implementation.

MVC: MVC module contains Spring's Model-View-Controller implementation.

(4) Miscellaneous:

Miscellaneous layer consist of AOP, Aspects, Instrumentation, Messaging and Test.

AOP : AOP module provides an Aspect-Oriented programming implementation.

Aspects : Aspects module provides integration with AspectJ.

Instrumentation : Instrumentation module provide class instrumentation support.

Messaging : Messaging module provides support for STOMP.

Test : Test module supports the testing of Spring components.

7 Explain AOP with Spring:

AOP stands for Aspect-Oriented Programming.

AOP enables modularization of cross-cutting concerns.

such as logging, security, transaction management.

Spring Framework provides support for AOP which is improve the modularity and maintainability of code.

This are the basic words is use in AOP.

- Aspect : This Module Provides set of APIs which providing cross-cutting.
- Join Point : A Point in the execution of a program, where you can add AOP Aspect.
- Advice : This is actual action to be taken either or after the method of execution.
- Pointcut : This is set of one or more Join points where an advice should be executed.

- Weaving : Weaving is the process of linking aspects with other application.

Spring supports the two type of Approach for custom aspect implement.

(a) AspectJ Annotation

(b) XML Schema Based

(a) AspectJ Annotation:

AspectJ Annotation is used to define an aspect and other annotations are used to define advice and pointcuts.

(b) XML Schema Based

If we use XML Schema Based Configuration, than we have to define aspects, advice and pointcuts in spring Configuration File.

8 Explain difference between Spring Dependency Injection and Inversion of control.

=> Spring Dependency Injection Inversion of Control.

1 Spring Dependency Injection is a way to inject the dependency of a framework component. Spring Inversion of Control is the core of Spring Framework.

2 Used to create Loosely-coupled application. Used to help create objects, managing objects.

3 It is the method of providing the dependencies. It is achieved through Dependency Injection.

4 It is one of the subtypes of IoC principle. It is a design principle.

Q Write a note on Managing transaction in Spring.

⇒ Spring Framework provides a different transaction management APIs.

Spring Transaction Management can be implemented without using application server.

Spring Support two types of Transaction management.

- a) Programmatic
- b) Declarative

a) Programmatic:

In this method, we have to manage transaction with the help of program.

This way is gives extreme flexibility.

b) Declarative:

In this method, we have to

Use annotations and XML based configuration to manage the transactions.

This transaction is preferable over programmatic transaction management.

Spring Transaction Abstraction is defined using org.springframework.transaction.PlatformTransactionManager interface.

These are the different types of methods used in Spring Transaction.

(a) `commit(TransactionStatus status)`: This method is used to commit the transaction.

(b) `rollback(TransactionStatus status)`: This method is used to rollback the transaction.

(c) `getName()`: Method returns the name of the transaction.

(d) `getTimeout()`: Method returns the time in second in which the transaction must complete

(e) `isReadOnly()`: Method returns whether the transaction is read-only.

10 Briefly explain Spring bean Life cycle.

⇒ Spring Bean is any object in the spring framework which is initialize through Spring container.

Spring Bean is managed by Spring Inversion of Control.

Spring Bean life cycle is managed by the spring container.

This are the working step of Spring Bean.

1 First we have to run the Program than Spring container is start.

Container Started



Bean Instantiated



Dependencies Injected



Custom init() method



Custom Utility method



Custom destroy() method

- 2 After that Spring Container create the Instance of bean
- 3 After that Dependencies is Injected in Spring Bean.
- 4 After that IF we want to write code than we have to write code in init() method.
- 5 After that we have to execute the custom destroy() method.

init() method can execute the all its code as string

container start and bean is instantiated.

Destroy() method can execute all its code on closing the Spring container. ~~AAA~~