

01. What are the different bean scopes in spring?

There are 5 bean scopes in spring framework. No. Scope Description

- singleton: The bean instance will be only once and same instance will be returned by the IOC container. It is the default scope.
- prototype: The bean instance will be created each time when requested.
- request: The bean instance will be created per HTTP request.
- session: The bean instance will be created per HTTP session.
- globalsession: The bean instance will be created per HTTP global session. It can be used in portlet context only.

02. What is the difference Between IoC and DI?

IoC is a generic term meaning rather than having the application, call the methods in a framework, the framework calls implementations provided by the application.

DI is a form of IoC, where implementations are passed into an object through constructors/setters/service lookups, which the object will 'depend' on in order to behave correctly.

03. What are the types of IOC container in spring?

There are two types of IOC containers in spring framework.

1. BeanFactory
2. ApplicationContext

04. What is the difference between BeanFactory and ApplicationContext?

- a. BeanFactory is the basic container whereas ApplicationContext is the advanced container.
- b. ApplicationContext extends the BeanFactory interface.
- c. ApplicationContext provides more facilities than BeanFactory such as integration with spring AOP, message resource handling for i18n, load file resources, ResourceLoader support etc.

05. What is the Main objective of an ORM library?

The main objective of an ORM library is to close the gap between the relational data structure in the RDBMS and the OO model in Java so that developers can focus on programming with the object model and at the same time easily perform actions related to persistence.

06. What is JdbcTemplate?

The JdbcTemplate is the spring answer to plain JDBC access. It handles database connectivity. Users can specify the sql and a callback through its method. The template does the work of creating the connection and executing the query or update. The callback can be used to handle resultset (ResultSetExtractor) or create a prepared statement (PreparedStatementCreator)

7. What is Data Access Layer?

Data Access Layer provides to our application components with a standard mechanism for storing and retrieving data.

8. What is Interface-driven design?

Interface-driven design is a traditional OOP best practice, In Interface-driven design the main components of the application are defined in terms of interfaces rather than concrete classes.

09. What is the function of DispatcherServlet?

The Spring Web MVC framework is designed around a DispatcherServlet that handles all the HTTP requests and responses.

10. What is a Spring Framework?

- Spring is a powerful open source, application framework created to reduce the complexity of enterprise application development.
- It is light-weighted and loosely coupled.
- It has layered architecture, which allows you to select the components to use, while also providing a cohesive framework for J2EE application development.
- Spring framework is also called the framework of frameworks as it provides support to various other frameworks such as Struts, Hibernate, Tapestry, EJB, JSF etc.

11. List the advantages of Spring Framework.

- Because of Spring Frameworks layered architecture, you can use what you need and leave which you don't.
- Spring Framework enables POJO (Plain Old Java Object) Programming which in turn enables continuous integration and testability.
- JDBC is simplified due to Dependency Injection and Inversion of Control.
- It is open-source and has no vendor lock-in.

12. What are the different features of Spring Framework?

Following are some of the major features of Spring Framework :

- **Lightweight:** Spring is lightweight when it comes to size and transparency.
- **Inversion of control (IOC):** The objects give their dependencies instead of creating or looking for dependent objects. This is called Inversion Of Control.
- **Aspect oriented Programming (AOP):** Aspect oriented programming in Spring supports cohesive development by separating application business logic from system services.
- **Container:** Spring Framework creates and manages the life cycle and configuration of the application objects.
- **MVC Framework:** Spring Framework's MVC web application framework is highly configurable. Other frameworks can also be used easily instead of Spring MVC Framework.
- **Transaction Management:** Generic abstraction layer for transaction management is provided by the Spring Framework. Spring's transaction support can be also used in container less environments.
- **JDBC Exception Handling:** The JDBC abstraction layer of the Spring offers an exception hierarchy, which simplifies the error handling strategy.

13. What is Spring IOC Container?

At the core of the Spring Framework, lies the Spring container. The container creates the object, wires them together, configures them and manages their complete life cycle. The Spring container makes use of Dependency Injection to manage the components that make up an application. The container receives instructions for which objects to instantiate, configure, and assemble by reading the configuration metadata provided. This metadata can be provided either by XML, Java annotations or Java code.

14. What do you mean by Dependency Injection?

In Dependency Injection, you do not have to create your objects but have to describe how they should be created. You don't connect your components and services together in the code directly, but describe which services are needed by which components in the configuration file. The IoC container will wire them up together.

15. In how many ways can Dependency Injection be done?

In general, dependency injection can be done in three ways, namely :

- Constructor Injection
- Setter Injection
- Interface Injection

In Spring Framework, only constructor and setter injections are used.

16. Detail of MVC framework?

The Spring Web MVC framework provides Model-View-Controller (MVC) architecture and ready components that can be used to develop flexible and loosely coupled web applications. The MVC pattern results in separating the

different aspects of the application (input logic, business logic, and UI logic), while providing a loose coupling between these elements.

- The **Model** encapsulates the application data and in general they will consist of POJO.
- The **View** is responsible for rendering the model data and in general it generates HTML output that the client's browser can interpret.
- The **Controller** is responsible for processing user requests and building an appropriate model and passes it to the view for rendering.