Search...

Spring - Difference Between Inversion of Control and Dependency Injection

Last Updated : 23 Jul, 2025

Understanding the **difference between Inversion of Control (IoC) and Dependency Injection (DI)** is very important for mastering the Spring framework. Both concepts are closely related, they serve different purposes in the context of Spring. The main difference between IoC and DI is listed below:

- **Inversion of Control (IOC)**: It is a design principle where the control of object creation and lifecycle is managed by a framework or container rather than by the developer. Spring IOC Container is responsible for creating, configuring, and managing the lifecycle of objects called beans.
- **Dependency Injection:** It is a design pattern and a part of IOC container. It allows objects to be injected with their dependencies rather than creating those dependencies themselves

Spring IOC vs Spring DI

The table below demonstrates the difference between Spring IOC and Spring DI

Spring IoC (Inversion of Control)	Spring Dependency Injection (DI)
Spring IoC Container is the core of the Spring Framework. It creates the objects, configures and assembles	Spring Dependency Injection is a way to inject the dependency of a framework component by the

Spring IoC (Inversion of Control)	Spring Dependency Injection (DI)
their dependencies, and manages their entire life cycle.	following ways of spring: Constructor Injection and Setter Injection
Spring helps in creating objects, managing objects, configurations, etc. because of IoC (Inversion of Control).	Spring framework helps in the creation of loosely-coupled applications because of Dependency Injection.
Spring IoC is achieved through Dependency Injection.	Dependency Injection is the method of providing the dependencies, and Inversion of Control is the end result of Dependency Injection.
IoC is a design principle where the control flow of the program is inverted.	Dependency Injection is one of the subtypes of the IOC principle.
Aspect-Oriented Programming is one way to implement Inversion of Control.	In case of any changes in business requirements, no code change is required.

Spring IoC (Inversion of Control)

The Spring IoC Container is the core of the Spring Framework. It creates and manages objects (beans) and injects dependencies. The IoC Container retrieves object configuration from:

- XML Configuration Files
- Java Configuration Classes
- Java Annotations

Since object creation and lifecycle management are handled by the IoC Container, developers do not need to manually instantiate dependencies. This reduces tight coupling in the application.

Spring Dependency Injection

Dependency Injection (DI) is a key feature provided by Spring IoC. The Spring Core module injects dependencies into objects via different injection methods, ensuring that components are loosely coupled.

There are two types of DI in Spring:

- Setter Dependency Injection (SDI)
- Constructor Dependency Injection (CDI)

1. Setter Dependency Injection (SDI)

In Setter Injection, dependencies are injected using setter methods. The property to be injected is declared inside the <property> tag in the XML configuration file.

Example:


```
<bean id="myBean" class="com.example.MyClass">
    <property name="dependency" ref="myDependency"/>
</bean>
```

2. Constructor Dependency Injection (CDI)

In Constructor Injection, dependencies are passed via the class constructor. The dependency is set using the <constructor-arg> tag in the XML configuration file.

Example:

```
<bean id="myBean" class="com.example.MyClass">
    <constructor-arg ref="myDependency"/>
</bean>
```