

Introduction to the Spring IoC Container and Beans

This chapter covers the Spring Framework implementation of the Inversion of Control (IoC) principle. Dependency injection (DI) is a specialized form of IoC, whereby objects define their dependencies (that is, the other objects they work with) only through constructor arguments, arguments to a factory method, or properties that are set on the object instance after it is constructed or returned from a factory method. The IoC container then injects those dependencies when it creates the bean. This process is fundamentally the inverse (hence the name, Inversion of Control) of the bean itself controlling the instantiation or location of its dependencies by using direct construction of classes or a mechanism such as the Service Locator pattern.

[Spring Framework](#) / [Core Technologies](#) / [The IoC Container](#) / Introduction to the Spring IoC Container and Beans

sub-interface of `BeanFactory`. It adds:

- Easier integration with Spring's AOP features
- Message resource handling (for use in internationalization)
- Event publication
- Application-layer specific contexts such as the `WebApplicationContext` for use in web applications.

In short, the `BeanFactory` provides the configuration framework and basic functionality, and the `ApplicationContext` adds more enterprise-specific functionality. The `ApplicationContext` is a complete superset of the `BeanFactory` and is used exclusively in this chapter in descriptions of Spring's IoC container. For more information on using the `BeanFactory` instead of the `ApplicationContext`, see the section covering the [BeanFactory API](#).

In Spring, the objects that form the backbone of your application and that are managed by the Spring IoC container are called beans. A bean is an object that is instantiated, assembled, and managed by a Spring IoC container. Otherwise, a bean is simply one of many objects in your application. Beans, and the dependencies among them, are reflected in the configuration metadata used by a container.

[Prev](#)
< [The IoC Container](#)

[Next](#)
[Container Overview](#) >

 [Edit this Page](#)
 [GitHub Project](#)
 [Stack Overflow](#)

Spring Framework

6.2.7

Q Search CTRL + k

- Overview
- ▼ Core Technologies
 - ▼ The IoC Container
 - Introduction to the Spring IoC Container and Beans**
 - Container Overview
 - Bean Overview
 - Dependencies
 - Bean Scopes
 - Customizing the Nature of a Bean
 - Bean Definition Inheritance
 - Container Extension Points
 - Annotation-based Container Configuration
 - Classpath Scanning and Managed Components
 - Using JSR 330 Standard Annotations
 - Java-based Container Configuration
 - Environment Abstraction
 - Registering a LoadTimeWeaver
 - Additional Capabilities of the ApplicationContext
 - The BeanFactory API
 - Resources
 - Validation, Data Binding, and Txoe