

SPRING - IOC CONTAINERS

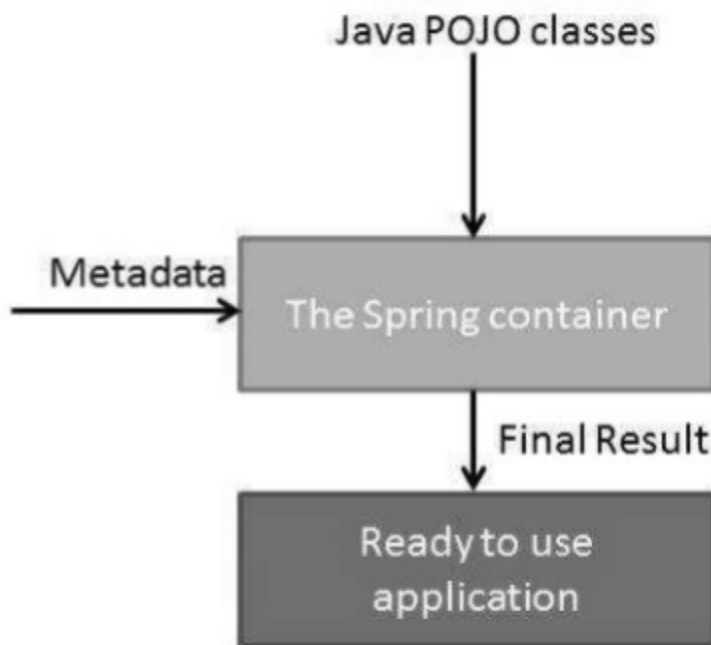
https://www.tutorialspoint.com/spring/spring_ioc_containers.htm

Copyright © tutorialspoint.com

Advertisements

The Spring container is at the core of the Spring Framework. The container will create the objects, wire them together, configure them, and manage their complete life cycle from creation till destruction. The Spring container uses DI to manage the components that make up an application. These objects are called Spring Beans, which we will discuss in the next chapter.

The container gets its instructions on what objects to instantiate, configure, and assemble by reading the configuration metadata provided. The configuration metadata can be represented either by XML, Java annotations, or Java code. The following diagram represents a high-level view of how Spring works. The Spring IoC container makes use of Java POJO classes and configuration metadata to produce a fully configured and executable system or application.



Spring provides the following two distinct types of containers.

Sr.No.	Container & Description
1	<u>Spring BeanFactory Container</u> This is the simplest container providing the basic support for DI and is defined by the <i>org.springframework.beans.factory.BeanFactory</i> interface. The BeanFactory and related interfaces, such as BeanFactoryAware, InitializingBean, DisposableBean, are still present in Spring for the purpose of backward compatibility with a large number of third-party frameworks that integrate with Spring.

The *ApplicationContext* container includes all functionality of the *BeanFactory* container, so it is generally recommended over *BeanFactory*. *BeanFactory* can still be used for lightweight applications like mobile devices or applet-based applications where data volume and speed is significant.