- Spring makes it easy to create Java enterprise applications. It provides everything you need to embrace the Java language in an enterprise environment

- At the heart are the modules of the core container, including a configuration model and a dependency injection mechanism

- The Spring Framework is modular and publishes 20+ different jars:

spring-aop spring-context-indexer spring-instrument spring-orm spring-webflux

spring-aspects spring-context-support spring-jcl spring-oxm spring-webmvc

spring-beans spring-core spring-jdbc spring-test spring-websocket

spring-beans-groovy spring-expression spring-jms spring-tx

spring-context spring-framework-bom spring-messaging spring-web

Một vài module phụ thuộc lẫn nhau như: spring-context phụ thuộc spring-beans, spring-beans lại phụ thuộc vào spring core

- The Spring Artifactory is a repository manager

Core: IoC container, Events, Resources, i18n, Validation, Data Binding, Type Conversion, SpEL, AOP.

- Quan trọng nhất trong số này là Spring Framework’s Inversion of Control (IoC) container

[1.1. Introduction to the Spring IoC container and beans](https://docs.spring.io/spring/docs/5.0.6.BUILD-SNAPSHOT/spring-framework-reference/core.html#beans-introduction)

- **IoC is also known as dependency injection (DI): là xử lý mà nhờ đó các đối tượng xác định được các dependencies của chúng hay những object chúng làm việc với** :

- 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 container then injects those dependencies when it creates the bean (bean được khởi tạo trước sau đó khi nơi nào cần sẽ được container injects)

Bean điều khiển việc khởi tạo và vị trí các dependencies của nó bằng việc using direct construction of classes, or a mechanism such as the Service Locator pattern

- The org.springframework.beans and org.springframework.context packages are the basis for Spring Framework’s IoC container

The [BeanFactory](https://docs.spring.io/spring-framework/docs/5.0.6.BUILD-SNAPSHOT/javadoc-api/org/springframework/beans/factory/BeanFactory.html) interface provides an advanced configuration mechanism capable of managing any type of object.[ApplicationContext](https://docs.spring.io/spring-framework/docs/5.0.6.BUILD-SNAPSHOT/javadoc-api/org/springframework/context/ApplicationContext.html) is a sub-interface of BeanFactory.

the BeanFactory provides the configuration framework and basic functionality, and the ApplicationContext adds more enterprise-specific functionality.

- **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 là 1 object mà được khởi tạo, nhúng hay do IoC quản lý.

[**1.2. Container overview**](https://docs.spring.io/spring/docs/5.0.6.BUILD-SNAPSHOT/spring-framework-reference/core.html#beans-basics)

The interface org.springframework.context.ApplicationContext represents the Spring IoC container

Container lấy các chỉ thị hướng dẫn trên các objects để khởi tạo, cấu hình và nhúng bằng việc đọc configuration metadata.

**The configuration metadata is represented in XML, Java annotations, or Java code.**

Several implementations of the ApplicationContext interface are [ClassPathXmlApplicationContext](https://docs.spring.io/spring-framework/docs/5.0.6.BUILD-SNAPSHOT/javadoc-api/org/springframework/context/support/ClassPathXmlApplicationContext.html) or [FileSystemXmlApplicationContext](https://docs.spring.io/spring-framework/docs/5.0.6.BUILD-SNAPSHOT/javadoc-api/org/springframework/context/support/FileSystemXmlApplicationContext.html)

While XML has been the traditional format for defining configuration metadata you can **instruct the container to use Java annotations** or code as the metadata format **by providing a small amount of XML configuration** to declaratively enable support.

*The Spring IoC container*



this configuration metadata represents how you as an application developer tell the Spring container to instantiate, configure, and assemble the objects in your application.