IoC e DI in Spring

Cosa imparerai

- Com’è implementato l’Inversion of Control in Spring tramite la Dependency Injection

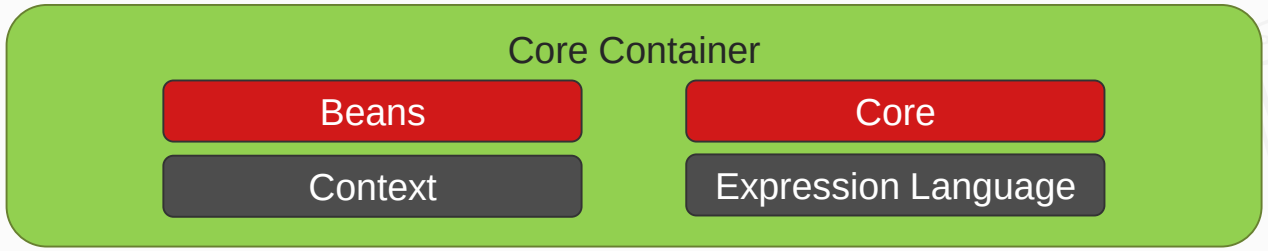


IoC e DI in Spring

Spring implementa l'Inversion of Control utilizzando il pattern DI per la gestione delle dipendenze!

I moduli che contengono le classi e le interfacce necessarie per implementare l'IoC in Spring sono:

- **spring-beans**
- **spring-core**



La componente Spring che si occupa di creare istanze e configurare gli oggetti si chiama **IoC container**.

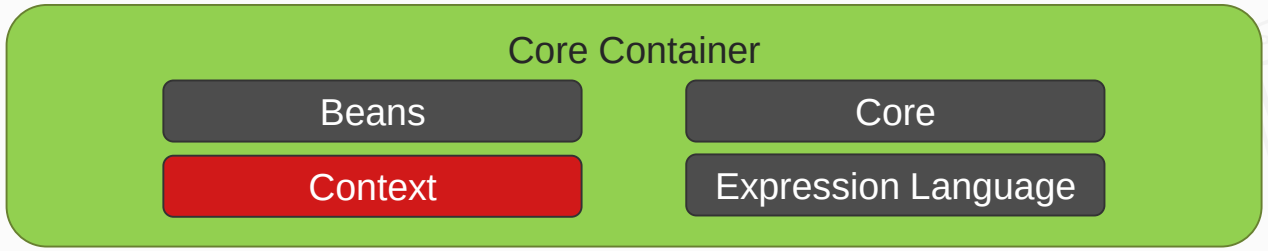
Gli oggetti creati dall'IoC Container sono chiamati **beans** e sono configurati utilizzando metadati che possono essere file XML, Java annotations o codice Java.

Il container si occupa di iniettare le dipendenze quando crea il bean.

IoC e DI in Spring

L'interfaccia **BeanFactory** mette a disposizione un meccanismo efficace di configurazione che consente di gestire oggetti Java di qualsiasi tipo.

Il modulo **spring-context**, basato sul modulo spring-beans, contiene l'interfaccia **ApplicationContext** che eredita le funzionalità della *BeanFactory* e ne aggiunge altre, tra cui:



- supporto per l'internazionalizzazione
- propagazione di eventi
- caricamento di risorse
- supporto a JEE

IoC e DI in Spring

Devo usare BeanFactory o ApplicationContext?

Poiché l'interfaccia *ApplicationContext* include tutte le funzionalità di *BeanFactory*, è consigliato utilizzare classi che implementano *ApplicationContext.*

È utile utilizzare classi che implementano *BeanFactory* solo quando è necessario il controllo completo sull'elaborazione dei bean.

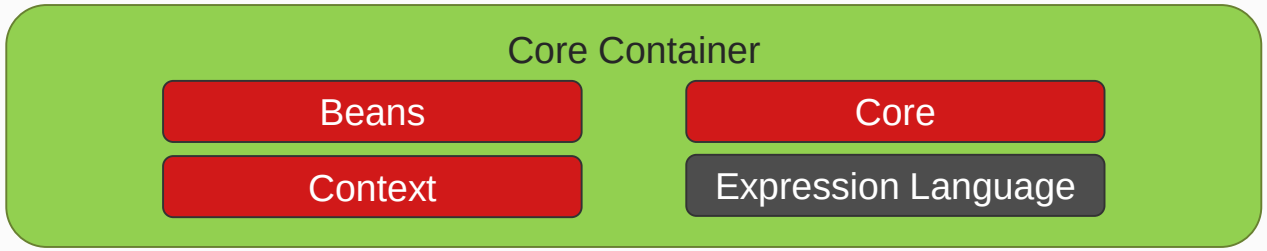
Esistono diverse classi che implementano le due interfacce:

- *GenericApplicationContext, ClassPathXmlApplicationContext, FileSystemXmlApplicationContext,…*
- *DefaultListableBeanFactory, …*

IoC e DI in Spring

Possiamo quindi concludere che i moduli necessari per realizzare l'IoC in Spring e che non devono mai mancare in un progetto Spring sono:

- **spring-beans**
- **spring-core**
- **spring-context**



Di cosa abbiamo parlato in questa lezione

- Com’è implementato l’Inversion of Control in Spring tramite la Dependency Injection