# CAPITOLO 2: Spring Bean LifeCycle and Configuration

**Configuration**

La CONFIGURAZIONE dell’APPLICATON CONTEXT (contesto applicativo) può essere fornita:

1. Via XML
2. XML + Annotations
3. Java

APPLICATION CONTEXT:

Rappresenta l’istanza del contesto applicativo. L’Application Context viene creato attraverso l’istanziazione di classi specifiche che implementano l’interfaccia **org.springframework.context.ApplicationContext**

La classe specifica dipende dal tipo di configurazione che si sta fornendo:

1. XML Configuration 🡪 **org.springframework.context.support.ClassPathXmlApplicationContext**
2. JAVA Configuration 🡪**org.springframework.context.annotation.AnnotationConfigApplicationContext**

**Spring IoC e Dependency Injection**

Può avvenire in due modi differenti:

1. Tramite **COSTRUTTORE:**

Da preferire quando la dipendenza è OBBLIGATORIA, ovvero quando il bean non può esistere senza le sue dipendenze oppure quando il BEAN deve essere immutabile, ovvero quando il suo stato interno non può cambiare.

Dichiarazione XML Classica:

**<bean id=”…” class=”…”>**

**<constructor-arg ref=”…”/>** 🡪 Riferimento a tipi “Complessi” ovvero altri BEAN definiti nell’Application Context

**<constructor-arg value=”…”/>** 🡪 Riferimento a tipi “Scalari” come primitive e altri tipi base

**</bean>**

Dichiarazione XML abbreviata (tramite namespace c):

**<bean id="…" class="…" c:{NOME\_PARAMETRO\_COSTRUTTORE}-ref="…" c:{NOME\_PARAMETRO\_COSTRUTTORE}="…"/> 🡪** Il primo per riferimento ad altri BEAN, il secondo per riferimenti a

variabili scalari.

1. Tramite **METODO SETTER:**

Da preferire quando la dipendenza è OPZIONALE.

Dichiarazione XML Classica:

**<bean id=”…” class=”…”>**

**<property name=”…” ref=”…”/>**

**<property name=”…” value=”…”/>**

**</bean>**

L’attributo NAME si riferisce al nome della variabile di istanza della classe “class”.

Dichiarazione XML abbreviata (tramite namespace p):

**<bean id="…" class="…" p:{NOME\_CAMPO\_CLASSE}-ref="…" p:{NOME\_CAMPO\_CLASSE"/> 🡪** Il primo per riferimenti ad altri BEAN, il secondo per riferimenti a variabili scalari.

1. Tramite **FIELD** o **ANNOTATION** (solo tramite JAVA Configuration):

**COLLECTIONS:**

**Internamente ad altre configurazione può non essere usato il namespace “util”. Altrimenti è obbligatorio.**

|  |  |
| --- | --- |
| **Inizializzazione con Collections vuote** | **Inizializzazione con collections predefinite** |
| **<bean id=”…” class=”…”>**  **<property name=”aList”>**  **<list/>**  **</property>**  **<property name=”aSet”>**  **<set/>**  **</property>**  **<property name=”aMap”>**  **<map/>**  **</property>**  **</bean>**  **<util:list>**  **</util:list>**  **<util:set>**  **</util:set>**  **<util:map>**  **</util:map>** | **<bean id=”…” class=”…”>**  **<property name=”aList”>**  **<list>**  **<ref bean=”anotherBean”/>**  **<bean class=””/>**  **<null/>**  **</list>**  **</property>**  **<property name=”aSet”>**  **<set>**  **<ref bean=”anotherBean”/>**  **<bean class=””/>**  **<null/>**  **</set>**  **</property>**  **<!—Non accetta Null e non è possibile usare <bean> per creare istanze ‘on the fly’ -->**  **<property name=”aMap”>**  **<map>**  **<entry key="one" value-ref=" anotherBean"/>**  **<entry key="two" value” ="simpleScalar"/>**  **</map>**  **</property>**  **</bean>** |
|  |  |