

Anatomia di un Bean

Cosa imparerai

- Com'è fatto un Bean

Anatomia di un bean

Il container IoC di Spring gestisce uno o più bean.

Ciascun bean viene creato utilizzando i metadati di configurazione forniti al container (tramite la definizione dell'elemento `<bean />` nel file XML di configurazione).

La definizione di un bean (via XML, annotation ecc...) è rappresentata nel container sotto forma di oggetto di tipo **BeanDefinition**. Ogni oggetto contiene tante informazioni, tra cui i seguenti metadati:

- ❑ il nome della classe (package compreso) da utilizzare in fase di creazione dell'istanza del bean;
- ❑ parametri di configurazione che specificano il comportamento del bean (lo scope, i metodi di callback del ciclo di vita ecc...);
- ❑ le dipendenze da altri bean necessari affinché il bean possa funzionare correttamente;
- ❑ altre configurazioni da impostare nell'istanza del bean (connection pool size, parametri di connessione al DB,...).

Anatomia di un bean

Questi metadati si traducono in una serie di proprietà che costituiscono ciascuna definizione di bean. La tabella seguente descrive queste proprietà:

Proprietà	Cosa fa
Class	Definisce il tipo di oggetto che verrà creato (es. <code>it.test.corso.ProdottoService</code>)
Name	Definisce il nome univoco da assegnare al bean dopo essere stato creato. Questo nome può essere utilizzato per accedere al bean nel context
Scope	Definisce l'ambito del bean (se deve esserci una sola istanza (singleton pattern), se è a livello di HTTP request o HTTP session (nel caso di web app), ecc...)
Constructor arguments	Specifica gli oggetti da istanziare durante la creazione dell'istanza del bean mediante la <i>Constructor Dependency Injection</i>
Attributes	Specifica i valori da impostare alle variabili di istanza o gli oggetti da istanziare durante la creazione dell'istanza del bean mediante la <i>Setter Dependency Injection</i>
Autowiring mode	Consente di definire le modalità di risoluzione delle dipendenze da altri bean
Lazy Initialization	Consente di specificare se un bean deve essere istanziato all'avvio dell'applicazione oppure alla prima richiesta di utilizzo (lazy mode)
Initialization method	Indica il metodo del bean da invocare subito dopo la creazione dell'istanza del bean
Destruction method	Indica il metodo del bean da invocare subito prima la distruzione dell'istanza del bean

Anatomia di un bean

Naming convention per gli attributi name e id

Per convenzione, questi due attributi possono contenere stringhe alfanumeriche ("prodottoService", ecc.). Possono contenere anche caratteri speciali.

L'id ed il name devono essere univoco altrimenti il container non riesce a identificare correttamente il bean e verrà segnalata l'eccezione di tipo *BeanDefinitionParsingException* che specificherà l'esistenza di due o più bean con lo stesso id.

```
11  
12 <bean id="ordine" class="it.test.service.OrdineService">  
13 </bean>  
14 <bean id="ordine" class="it.test.service.ProdottoService">  
15 </bean>  
16 </beans>
```

Console Problems Debug Shell

```
<terminated> Main [Java Application] C:\Users\Paolo\Downloads\sts-4.9.0.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_15.0.1.v20201027-0507\jre\bin\javaw.exe (8 gen 2021, 14:24:08)  
Exception in thread "main" org.springframework.beans.factory.parsing.BeanDefinitionParsingException: Configuration problem: Bean name 'ordine' is already  
used in this <beans> element  
Offending resource: class path resource [services.xml.xml]
```

Anatomia di un bean

Naming convention per gli attributi name e id

Non è obbligatorio specificare per un bean il name o l'id.

Se non vengono specificati, il container genererà un name univoco per il bean.

Tuttavia, in questo caso non è possibile far riferimento al bean per nome, tramite l'attributo ref ecc...

Il mio consiglio è di utilizzare sempre l'id (o il name)



Di cosa abbiamo parlato in questa lezione

- Com'è fatto un Bean