

# Gestione delle dipendenze tra bean – Parte III

Cosa imparerai

---

- Come si definiscono e gestiscono le dipendenze tra oggetti in Spring tramite la Dependency Injection

# Gestione delle dipendenze tra bean

Come abbiamo visto, quando il bean A dipende dal bean B, generalmente significa che il bean B è una variabile di istanza del bean A.

Nei metadati XML-based, la definizione della dipendenza tra bean si effettua con l'elemento **<ref />**.

```
<bean id="ordineService" class="it.test.service.OrdineService">
  <property name="cliente">
    <ref bean="clienteService"/>
  </property>
  <property name="statoPredefinito">
    <value>COMPLETATO</value>
  </property>
</bean>
```

```
public class OrdineService {
    private ClienteService cliente;
    private String statoPredefinito;

    public void setCliente(ClienteService cliente) {
        this.cliente = cliente;
    }

    public void setStatoPredefinito(String statoPredefinito) {
        this.statoPredefinito = statoPredefinito;
    }

    public String sayHello() {
    }

    public void saluto() {
    }
}
```

# Gestione delle dipendenze tra bean

Ci sono dei casi in cui le dipendenze tra bean sono meno dirette, ovvero il bean B non è una variabile di istanza del bean A.

Un tipico esempio è il caso in cui nel bean B c'è un blocco di codice statico utilizzato per inizializzare delle variabili (il classico caso d'uso è la registrazione del driver del DB).

Utilizzando l'attributo **depends-on** nell'elemento `<bean ... />`, chiediamo al container di inizializzare un bean prima che inizi il nostro.

Esempio:

```
<bean id="beanA" class="it.test.MyBeanA" depends-on="beanB" />  
<bean id="beanB" class="it.test.MyBeanB" />
```

In questo caso, il beanB verrà inizializzato prima che venga inizializzato il beanA.

# Gestione delle dipendenze tra bean

L'attributo `depends-on` può specificare anche la fase di destroy dei bean.

In questo caso, i bean che dipendono da un altro bean vengono distrutti prima che il bean stesso venga distrutto.

Esempio:

```
<bean id="beanA" class="it.test.MyBeanA" depends-on="beanB" />  
<bean id="beanB" class="it.test.MyBeanB" />
```

In questo caso, il beanB verrà distrutto dopo la distruzione del beanA.

# Di cosa abbiamo parlato in questa lezione

- Come si definiscono e gestiscono le dipendenze tra oggetti in Spring tramite la Dependency Injection