

Definire i bean tramite le annotation

Cosa imparerai

- Come definire un bean senza utilizzare la notazione XML, con l'annotation *@Component*

Definire i bean tramite le annotation

Fin ad ora abbiamo visto che per definire un bean è possibile utilizzare la seguente notazione XML:

```
<bean="myBean" class="it.test.MyBean" />
```

```
<beans
xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:context="http://www.springframework.org/schema/context"
xsi:schemaLocation="
  http://www.springframework.org/schema/beans
  https://www.springframework.org/schema/beans/spring-beans.xsd
  http://www.springframework.org/schema/context
  https://www.springframework.org/schema/context/spring-context.xsd">

  <bean id="prodottoService" class="it.test.service.ProdottoService">
  </bean>

  <bean id="ordineService" class="it.test.service.OrdineService">
  </bean>

  <bean id="clienteService" class="it.test.service.ClienteService">
  </bean>
</beans>
```

```
public class ProdottoService {
    public ProdottoService() {
        super();
    }

    public String sayHello() {
        String ret = "TV 32''";

        return ret;
    }
}
```

Anche attraverso le annotation è possibile definire i bean e dare istruzioni al container di rilevarli automaticamente, senza definirli nel file XML.

Definire i bean tramite le annotation

Affinché il container rilevi automaticamente i bean definiti tramite annotation, è necessario utilizzare una notazione XML per specificare l'**autoscan**.

`<context:component-scan base-package="it.test.service" />`

```
<beans
xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:context="http://www.springframework.org/schema/context"
xsi:schemaLocation="
    http://www.springframework.org/schema/beans
    https://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/context
    https://www.springframework.org/schema/context/spring-context.xsd">

    <context:component-scan base-package="it.test.service" />
</beans>
```

1

All'elemento <beans /> bisogna aggiungere l'XML namespace context, affinché sia possibile utilizzare l'elemento <context... />

2

In questo caso diciamo al container che i bean si trovano nel package (e sotto-package) it.test.service

Definire i bean tramite le annotation

Le annotation che possono essere utilizzate per definire dei bean sono:

- ❑ **@Component**
- ❑ **@Repository**
- ❑ **@Service**
- ❑ **@Controller**

@Component definisce un generico componente gestito dal container.

@Repository, *@Service* e *@Controller* sono specializzazioni di *@Component* per casi d'uso più specifici (rispettivamente nel persistence layer, service layer e presentation layer).

Definire i bean tramite le annotation

@Component

È l'annotation che definisce un generico componente gestito dal container.

Se annotiamo una classe con l'annotation @Component, senza specificare altro, il container istanzierà un bean che avrà come name il nome della classe con la prima lettera lowercase.

```
1  @Component
2  public class OrdineService {
    public String sayHello() {
        return "CIAO sono un ordine del cliente ";
    }
}
```

```
public class Main {
    public static void main(String[] args) {
        ApplicationContext context = new ClassPathXmlApplicationContext("services.xml");

        OrdineService o = context.getBean("ordineService", OrdineService.class);
        System.out.println(o.sayHello());
    }
}
```

Definire i bean tramite le annotation

@Component

Se vogliamo dare un nome specifico ad un bean, nell'annotation dobbiamo utilizzare l'attributo **value** che conterrà il nome da dare al bean.

```
1
@Component(value = "ordine")
public class OrdineService {
    public String sayHello() {
        return "CIAO sono un ordine del cliente ";
    }
}
```

```
public class Main {
    public static void main(String[] args) {
        ApplicationContext context = new ClassPathXmlApplicationContext("services.xml");

        OrdineService o = context.getBean("ordine", OrdineService.class);

        System.out.println(o.sayHello());
    }
}
```

Di cosa abbiamo parlato in questa lezione

- Come definire un bean senza utilizzare la notazione XML, con l'annotation *@Component*