

Spring的依赖注入 (Dependency Injection)

一、定义 (也是控制反转)

Spring 容器在创建被调用者的实例时，会自动将调用者需要的对象实例注入给调用者，这样，调用者通过Spring容器获得被调用者实例，这称为依赖注入。

二、注入的数据类型：

1. 基本类型和String
2. 其他bean类型
3. 复杂类型/集合类型

三、注入方式

1、基于构造函数的依赖注入

(1)使用的标签:constructor-arg

(2) 标签出现的位置: bean标签的内部

(3) 标签中的属性

type: 用于指定要注入的数据的数据类型，该数据类型也是构造函数中某个或某些参数的类型

index: 用于指定要注入的数据给构造函数中指定索引位置的参数赋值。索引的位置是从0开始

name: 用于指定给构造函数中指定名称的参数赋值

常用的

=====以上三个用于指定给构造函数中哪个参数赋值

value: 用于提供基本类型和String类型的数据

ref: 用于指定其他的bean类型数据。它指的就是在spring的Ioc核心容器中出现过的bean对象

(4)优势:

在获取bean对象时，注入数据是必须的操作，否则对象无法创建成功。

弊端：

改变了bean对象的实例化方式，使我们在创建对象时，如果用不到这些数据，也必须提供。

(5) 实例

```
public AccountServiceImpl(String name,Integer age,Date birthday){
    this.name = name;
    this.age = age;
    this.birthday = birthday;
}
```

```
public AccountServiceImpl(String name,Date birthday){
    this.name = name;
    this.birthday = birthday;
}
```

创建bean标签里的参数数目 至少要等于一个构造函数的参数

```
<bean id="accountService" class="com.itheima.service.impl.AccountServiceImpl">
    <constructor-arg name="name" value="泰斯特"></constructor-arg>
    <constructor-arg name="birthday" ref="now"></constructor-arg>
</bean>
```

可以不用<constructor-arg name="age" value="18"></constructor-arg>

2、 基于设值函数的依赖注入（set方法注入 更常用的方式）

(1)使用的标签:property

(2) 标签出现的位置： bean标签的内部

(3) 标签中的属性

name： 用于指定注入时所调用的set方法名称

value： 用于提供基本类型和String类型的数据

ref： 用于指定其他的bean类型数据。它指的就是在spring的Ioc核心容器中出现过的bean对象

(4) 优势

创建对象时没有明确的限制，可以直接使用默认构造函数

弊端：

如果有某个成员必须有值，则获取对象是有可能set方法没有执行。

(5) 实例

```
<bean id="accountService2"
class="com.itheima.service.impl.AccountServiceImpl2">
    <property name="name" value="TEST" ></property>
```

```
<property name="age" value="21"></property>
<property name="birthday" ref="now"></property>
</bean>
```

//如果是经常变化的数据，并不适用于注入的方式

```
private String name;
private Integer age;
private Date birthday;

public void setName(String name) {
    this.name = name;
}

public void setAge(Integer age) {
    this.age = age;
}

public void setBirthday(Date birthday) {
    this.birthday = birthday;
}
```

3、复杂类型的注入/集合类型的注入

用于给List结构集合注入的标签：

```
list array set
```

用于个Map结构集合注入的标签：

```
map props
```

结构相同，标签可以互换

```
<bean id="accountService3"
class="com.itheima.service.impl.AccountServiceImpl3">
    <property name="myStrs">
        <set>
            <value>AAA</value>
            <value>BBB</value>
            <value>CCC</value>
        </set>
    </property>

    <property name="myList">
        <array>
            <value>AAA</value>
            <value>BBB</value>
            <value>CCC</value>
```

```
        </array>
    </property>

    <property name="mySet">
        <list>
            <value>AAA</value>
            <value>BBB</value>
            <value>CCC</value>
        </list>
    </property>

    <property name="myMap">
        <props>
            <prop key="testC">ccc</prop>
            <prop key="testD">ddd</prop>
        </props>
    </property>

    <property name="myProps">
        <map>
            <entry key="testA" value="aaa"></entry>
            <entry key="testB">
                <value>BBB</value>
            </entry>
        </map>
    </property>
</bean>
```

3、使用注解