

bean中的引用类型字段自动注入

创建一个bao03包，将bao02包装的类复制过来，然后在resources目录中创建bao3目录，新建一个Spring配置文件，其中需要把引用类型字段进行注入的改为自动注入

自动注入包括两种方式：

1)、byName方式

2)、byType方式

这两中方式，前者是按照bean中的id和另一个bean中引用类型属性的变量名一致进行赋值注入；后者则是按照bean中class的值和另一个bean中引用类型的类型一致、或者是该类的子类、异或是该类的实现类

两种方式实现的配置文件如下

byName方式：

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">

    <!--声明School类的对象-->
    <bean id="school" class="com.studymyself.bao02.School">
        <property name="name" value="北京大学"/>
        <property name="addr" value="北京"/>
    </bean>

    <!--
        javabean中引用类型属性自动注入：Spring框架根据某些规则给Java类中的引用类型
        字段自动赋值
        1、下面是按名称（byName）自动注入：
            需要赋值的javabean类中引用类型的属性变量名和Spring容器（配置文件）中某
            个被<bean>标签声明的bean类的id属性的名称一样，并且该引用类型和这个bean
            类的类型一样，这个bean就可以被赋值到引用变量中了
            语法：
            在需要赋值的javabean类的bean标签中添加 autowire="byName" autowire表示自动
            注入

            上面有一个Student类被声明对象，id是school
            下面有一个Student02类被声明对象，这个类中有一个属性是：private School
            school;

            看到类型一致，名称一致，所以可以赋值
        -->
```

```

<bean id="student02" class="com.studymyself.bao02.Student02"
autowire="byName">
    <property name="name" value="智控"/>
    <property name="age" value="25"/>
    <!--<property name="school" ref="school"/>-->
</bean>
<!--  执行流程: -->
<!--  Spring扫描这个配置文件-->
<!--  扫到上面第一个bean标签, 给Student类创建对象, 通过set注入给对象赋值-->
<!--  扫描到第二个bean标签, 给Student02类创建对象, 发现有autowire="byName"-->
<!--  通过set注入给对象的简单类型属性赋值的同时, 获取引用类型的类型和字段-->
<!--  名称,通过字段名称到存储的对象的集合中找到与该名称一样的对象, 再确认-->
<!--  类型是否一致, 一致, 就将找到的bean赋值给引用类型-->

</beans>

```

测试文件MyTest02.java

```

package com.studymyself;

import com.studymyself.bao03.Student03;
import org.junit.Test;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MyTest02 {

    /**
     * 测试自动注入
     */
    @Test
    public void testStudent01(){

        //定义配置文件路径
        String config = "bao03/applicationContext.xml";

        //获取容器对象
        ApplicationContext ac = new ClassPathXmlApplicationContext(config);

        //获取容器中的对象, 强制转换Student01类型
        Student03 student03 = (Student03) ac.getBean("student03");
        //输出对象
        System.out.println(student03);

    }

}

```

从结果可以看出, 自动注入底层调用了该引用类型属性的set方法, 这是肯定的, 不然私有的值就没法赋值了

byType方式:

```

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"

```

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

```
<!--声明School类的对象-->
```

```
<bean id="school" class="com.studymyself.bao03.School">
    <property name="name" value="北京大学3"/>
    <property name="addr" value="北京3"/>
</bean>
```

```
<!--
```

javabean中引用类型属性自动注入：**Spring**框架根据某些规则给**Java**类中的引用类型字段自动赋值

1、下面是按名称（**byName**）自动注入：

需要赋值的**javabean**类中引用类型的属性变量名和**Spring**容器（配置文件）中某个被**<bean>**标签声明的**bean**类的**id**属性的名称一样，并且该引用类型和这个**bean**类的类型一样，这个**bean**就可以被赋值到引用变量中了

语法：

在需要赋值的**javabean**类的**bean**标签中添加 **autowire="byName"** **autowire**表示自动

注入

上面有一个**Student**类被声明对象，**id**是**school**

下面有一个**Student02**类被声明对象，这个类中有一个属性是：**private School**

school;

看到类型一致，名称一致，所以可以赋值

2、下面是按类型（**byType**）自动注入：

Java类中引用数据类型的属性的类型和**Spring**容器（配置文件）中**<bean>**标签中的**class**属性是同源关系，这样就能自动找同源对象赋值给这个引用变量。

同源关系，就是同一类：

1. **Java**类中需要赋值的引用类型的数据类型和容器中**bean**的**class**的值一致，就是同一个类

2. **Java**类中需要赋值的引用类型的数据类型是容器器中**bean**类的父类

3. **Java**类中需要赋值的引用类型的数据类型是接口类，容器中**bean**类实现了该接口

语法：

将**autowire**的值改为：**autowire="byType"**

注意：这种方式要保证容器中只能有一个与引用类型的数据类型同源的对象，如果配置文件声明了这个引用类型的数据类型的对象，在声明该类的子类对象，会报错

```
-->
```

```
</beans>
```