

## @Resource注解给引用类型属性赋值

该注解是由JDK提供的，并不是Spring框架的，但是Spring支持这个注解给对象赋值。

具体使用方式如下面代码所示

```
package com.studymyself.bao03;

import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Component;
import javax.annotation.Resource; //这里很明显是JDK的注解类

@Component("student03")
public class Student {

    public Student() {
        System.out.println("Student03无参构造方法执行!");
    }

    @Value("终南山") //value可省略
    private String name;
    @Value(value = "5214")
    private int age;

    /**
     * 引用类型注解赋值
     * @Resource: JDK提供的注解，实现引用类型的赋值
     * 注解实现引用类型的赋值，使用的也是自动注入原理，方式有byType和byName
     * @Autowired注解默认使用byName方式，正好和@Autowired注解默认相反
     *
     * 使用方式：
     *     1、放在在引用类型属性定义上面，该方式不使用set方法，推荐使用
     *     2、放在属性的set方法上，调用set方法赋值
     *
     * @Autowired注解中有一个属性name，String类型的
     * 一般我们使用@Resource注解时，不指定该属性的值时，
     *     Spring底层会先使用byName的方式自动注入，如果找不到和引用类型变量名
     *     同名的Java对象，改用byType的方式实现自动注入
     * 当我们将name的值指定一个名称时，Spring就只会使用byType的方式进行自动注入
     * Spring底层就会查找id为name属性中名称的bean，找不到就报错
     *
     */
    //默认使用@Resource
    @Resource
    private School school;

    //指定只能用byName方式
    //@Resource(name = "myschool")
    //private School school;

    public void setName(String name) {
```

```

        this.name = name;
    }

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

    public void setSchool(School school) {
        this.school = school;
    }

    @Override
    public String toString() {
        return "Student{" +
            "name='" + name + '\'' +
            ", age=" + age +
            ", school=" + school +
            '}';
    }
}

```

## 基于XML和基于注解的选择

使用注解的方式快速、高效，能很直观的看到代码的所有信息，一般以注解方式为主，xml配置文件为辅。实在不能用注解的，在用配置文件的方式。

有时候我们不能再代码中修改属性所赋的值，可以使用 \${属性配置文件中的key} 来代替真实值，达到在配置文件中修改值和使用注解结合。

### 1)、第一步现在resources目录下新建一个test.properties属性配置文件

```

name=你xx
age=26

```

### 2)、在Spring配置文件中声明一个标签，如下

```

<?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"

    <context:property-placeholder location="com.test.propertis"/>

</beans>

```

### 3) 、在Java类中的属性上方使用注解，使用\${}代替真实值，如下

```
package com.studymyself.bao03;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Component;

@Component("student03")
public class Student {

    public Student() {
        System.out.println("Student03无参构造方法执行!");
    }

    //使用占位符${}
    @Value("${name}")
    private String name;
    @Value(value = "${age}")
    private int age;

    @Autowired
    private School school;

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

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

    public void setSchool(School school) {
        this.school = school;
    }

    @Override
    public String toString() {
        return "Student{" +
            "name='" + name + '\'' +
            ", age=" + age +
            ", school=" + school +
            '}';
    }
}
```