地址：https://blog.csdn.net/shuair/article/details/78824261

正常情况下，注入bean时，如果存在多个资源，就会出错，可以用@Qualifier指定名字，如下

比如有一个Speak接口

package shuai.spring.study.zhujie;

public interface Speak {

public void speak();

}

有两个实现Hello，Nihao

package shuai.spring.study.zhujie;

public class SpeakHello implements Speak {

@Override

public void speak() {

System.out.println("hello");

}

}

package shuai.spring.study.zhujie;

public class SpeakNihao implements Speak {

@Override

public void speak() {

System.out.println("nihao");

}

}

配置文件可以这么写

<?xml version="1.0" encoding="UTF-8"?>

<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"

xmlns:aop="http://www.springframework.org/schema/aop"

xsi:schemaLocation="

http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/aop

http://www.springframework.org/schema/aop/spring-aop.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context.xsd">

<context:annotation-config/>

<bean id="speakHello" class="shuai.spring.study.zhujie.SpeakHello"/>

    <bean id="speakNihao" class="shuai.spring.study.zhujie.SpeakNihao"/>

    <bean id="hello" class="shuai.spring.study.HelloImpl6"/>

</beans>

然后需要注入Speak的HelloImpl6,java可以这样写

package shuai.spring.study;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.beans.factory.annotation.Qualifier;

import shuai.spring.study.zhujie.Speak;

public class HelloImpl6 implements HelloApi {

private Speak speak;

public Speak getSpeak() {

return speak;

}

@Autowired

public void setSpeak(@Qualifier("speakNihao") Speak speak) {

this.speak = speak;

}

@Override

public void sayHello() {

speak.speak();

}

}

使用@Autowired自动注入，使用@Qualifier("speakNihao")确定注入的是具体的是什么，因为speakNihao和speakHello都是Speak类型的。

其实可以自定义一个注解，就可以不用@Qualifier("speakNihao")这种形式了。

先定义两个自定义注解

package shuai.spring.study.zhujie;

import java.lang.annotation.ElementType;

import java.lang.annotation.Retention;

import java.lang.annotation.RetentionPolicy;

import java.lang.annotation.Target;

import org.springframework.beans.factory.annotation.Qualifier;

@Target({ ElementType.TYPE, ElementType.FIELD, ElementType.PARAMETER })

@Retention(RetentionPolicy.RUNTIME)

@Qualifier

public @interface Hello {

}

package shuai.spring.study.zhujie;

import java.lang.annotation.ElementType;

import java.lang.annotation.Retention;

import java.lang.annotation.RetentionPolicy;

import java.lang.annotation.Target;

import org.springframework.beans.factory.annotation.Qualifier;

@Target({ ElementType.TYPE, ElementType.FIELD, ElementType.PARAMETER })

@Retention(RetentionPolicy.RUNTIME)

@Qualifier

public @interface Nihao {

}

上面的两个注解@Target、@Retention的意思可以看下面的连接，感觉每个自定义注解都得有

http://blog.csdn.net/sw5131899/article/details/54947192

配置文件修改下

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:aop="http://www.springframework.org/schema/aop"

xsi:schemaLocation="

http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/aop

http://www.springframework.org/schema/aop/spring-aop.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context.xsd">

<context:annotation-config/>

<bean class="shuai.spring.study.zhujie.SpeakHello">

<qualifier type="shuai.spring.study.zhujie.Hello"/>

</bean>

<bean class="shuai.spring.study.zhujie.SpeakNihao">

<qualifier type="shuai.spring.study.zhujie.Nihao"/>

</bean>

<bean id="hello" class="shuai.spring.study.HelloImpl6"/>

</beans>

HelloImpl6修改下

package shuai.spring.study;

import org.springframework.beans.factory.annotation.Autowired;

import shuai.spring.study.zhujie.Nihao;

import shuai.spring.study.zhujie.Speak;

public class HelloImpl6 implements HelloApi {

private Speak speak;

public Speak getSpeak() {

return speak;

}

@Autowired

public void setSpeak(@Nihao Speak speak) {

this.speak = speak;

}

@Override

public void sayHello() {

speak.speak();

}

}

注意那个注解@Nihao

测试一下

package shuai.spring.test;

import org.junit.Test;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import shuai.spring.study.HelloApi;

public class HelloTest {

@Test

public void testHelloWorld() {

@SuppressWarnings("resource")

ApplicationContext context = new ClassPathXmlApplicationContext("HelloWorld.xml");

HelloApi helloApi = context.getBean("hello", HelloApi.class);

helloApi.sayHello();

}

}

测试结果：

nihao

带参数的自定义注解

上面的例子需要写两个自定义注解，也可以写到一起，用参数做区分就行了

定义一个自定义注解

package shuai.spring.study.zhujie;

import java.lang.annotation.ElementType;

import java.lang.annotation.Retention;

import java.lang.annotation.RetentionPolicy;

import java.lang.annotation.Target;

import org.springframework.beans.factory.annotation.Qualifier;

@Target({ ElementType.TYPE, ElementType.FIELD, ElementType.PARAMETER })

@Retention(RetentionPolicy.RUNTIME)

@Qualifier

public @interface MySpeak {

String name();

String note();

}

也可以给参数默认值

String note() default "good";

修改xml文件

<?xml version="1.0" encoding="UTF-8"?>

<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"

xmlns:aop="http://www.springframework.org/schema/aop"

xsi:schemaLocation="

http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/aop

http://www.springframework.org/schema/aop/spring-aop.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context.xsd">

<context:annotation-config/>

<bean class="shuai.spring.study.zhujie.SpeakHello">

<qualifier type="shuai.spring.study.zhujie.MySpeak">

<attribute key="name" value="hello"/>

<attribute key="note" value="well"/>

</qualifier>

</bean>

<bean class="shuai.spring.study.zhujie.SpeakNihao">

<qualifier type="shuai.spring.study.zhujie.MySpeak">

<attribute key="name" value="nihao"/>

<attribute key="note" value="good"/>

</qualifier>

</bean>

<bean id="hello" class="shuai.spring.study.HelloImpl6"/>

</beans>

修改HelloImpl6

package shuai.spring.study;

import org.springframework.beans.factory.annotation.Autowired;

import shuai.spring.study.zhujie.MySpeak;

import shuai.spring.study.zhujie.Speak;

public class HelloImpl6 implements HelloApi {

private Speak speak;

public Speak getSpeak() {

return speak;

}

@Autowired

public void setSpeak(@MySpeak(name = "hello", note = "well") Speak speak) {

this.speak = speak;

}

@Override

public void sayHello() {

speak.speak();

}

}

注解的参数值要跟配置文件对应上。

如果使用有默认值，那个参数就可以不写了

@Autowired

public void setSpeak(@MySpeak(name = "nihao") Speak speak) {

this.speak = speak;

}

用上面的测试测试结果是一样的。

配置文件还可以这么写

<?xml version="1.0" encoding="UTF-8"?>

<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"

xmlns:aop="http://www.springframework.org/schema/aop"

xsi:schemaLocation="

http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/aop

http://www.springframework.org/schema/aop/spring-aop.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context.xsd">

<context:annotation-config/>

<bean class="shuai.spring.study.zhujie.SpeakHello">

<meta key="name" value="hello"/>

<meta key="note" value="well"/>

</bean>

<bean class="shuai.spring.study.zhujie.SpeakNihao">

<meta key="name" value="nihao"/>

<meta key="note" value="good"/>

</bean>

<bean id="hello" class="shuai.spring.study.HelloImpl6"/>

</beans>

【摘抄】开发者也可以启用<meta/>元素，配置示例如下。当在同一受管Bean中同时指定<qualifier/>和<meta/>元素时，DI容器会优先使用<qualifier/>

貌似不用指定自定义注解类的位置

http://book.51cto.com/art/201004/193381.htm

上面的自定义注解类我们都加了@Qualifier，如果不加，可以在配置文件里注册

比如把MySpeak上面的@Qualifier去掉，然后修改配置文件

<?xml version="1.0" encoding="UTF-8"?>

<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"

xmlns:aop="http://www.springframework.org/schema/aop"

xsi:schemaLocation="

http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/aop

http://www.springframework.org/schema/aop/spring-aop.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context.xsd">

<context:annotation-config/>

<bean class="shuai.spring.study.zhujie.SpeakHello">

<meta key="name" value="hello"/>

<meta key="note" value="well"/>

</bean>

<bean class="shuai.spring.study.zhujie.SpeakNihao">

<meta key="name" value="nihao"/>

<meta key="note" value="good"/>

</bean>

<bean id="hello" class="shuai.spring.study.HelloImpl6"/>

<bean id="customAutowireConfigurer" class="org.springframework.beans.factory.annotation.CustomAutowireConfigurer">

<property name="customQualifierTypes">

<set>

<value>shuai.spring.study.zhujie.MySpeak</value>

</set>

</property>

</bean>

</beans>

就是在后面加一个bean，【引用】注解没有应用类一级的@Qualifier注解时，开发者必须借助如下对象注册它。（就是最下面那个bean）

---------------------

作者：shuair

来源：CSDN

原文：https://blog.csdn.net/shuair/article/details/78824261

版权声明：本文为博主原创文章，转载请附上博文链接！