<https://www.cnblogs.com/think-in-java/p/5474740.html>

@Resource和@Autowired都是做bean的注入时使用，其实@Resource并不是Spring的注解，它的包是javax.annotation.Resource，需要导入，但是Spring支持该注解的注入。

1、共同点

两者都可以写在字段和setter方法上。两者如果都写在字段上，那么就不需要再写setter方法。

2、不同点

（1）@Autowired

@Autowired为Spring提供的注解，需要导入包org.springframework.beans.factory.annotation.Autowired;只按照byType注入。

[复制代码](javascript:void(0);)

public class TestServiceImpl {

// 下面两种@Autowired只要使用一种即可

@Autowired

private UserDao userDao; // 用于字段上

@Autowired

public void setUserDao(UserDao userDao) { // 用于属性的方法上

this.userDao = userDao;

}

}

[复制代码](javascript:void(0);)

@Autowired注解是按照类型（byType）装配依赖对象，默认情况下它要求依赖对象必须存在，如果允许null值，可以设置它的required属性为false。如果我们想使用按照名称（byName）来装配，可以结合@Qualifier注解一起使用。如下：

public class TestServiceImpl {

@Autowired

@Qualifier("userDao")

private UserDao userDao;

}

（2）@Resource

@Resource默认按照ByName自动注入，由J2EE提供，需要导入包javax.annotation.Resource。@Resource有两个重要的属性：name和type，而Spring将@Resource注解的name属性解析为bean的名字，而type属性则解析为bean的类型。所以，如果使用name属性，则使用byName的自动注入策略，而使用type属性时则使用byType自动注入策略。如果既不制定name也不制定type属性，这时将通过反射机制使用byName自动注入策略。

[复制代码](javascript:void(0);)

public class TestServiceImpl {

// 下面两种@Resource只要使用一种即可

@Resource(name="userDao")

private UserDao userDao; // 用于字段上

@Resource(name="userDao")

public void setUserDao(UserDao userDao) { // 用于属性的setter方法上

this.userDao = userDao;

}

}

[复制代码](javascript:void(0);)

注：最好是将@Resource放在setter方法上，因为这样更符合面向对象的思想，通过set、get去操作属性，而不是直接去操作属性。

@Resource装配顺序：

①如果同时指定了name和type，则从Spring上下文中找到唯一匹配的bean进行装配，找不到则抛出异常。

②如果指定了name，则从上下文中查找名称（id）匹配的bean进行装配，找不到则抛出异常。

③如果指定了type，则从上下文中找到类似匹配的唯一bean进行装配，找不到或是找到多个，都会抛出异常。

④如果既没有指定name，又没有指定type，则自动按照byName方式进行装配；如果没有匹配，则回退为一个原始类型进行匹配，如果匹配则自动装配。

@Resource的作用相当于@Autowired，只不过@Autowired按照byType自动注入。