# Autowired和Resource

## 注解来源

@Autowired是Spring中的注解

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

@Resource是java提供的注解

import javax.annotation.Resource

## 自动装配方式

@Autowired会先按byType去找，如果没找到，则会按照byName去找

@Resource会先按byName去找，如果没找到则会byType去找。如果设置了name属性，则只会按byName去找，找不到就报错。

举例：

@Resource(name = "studentServiceImpl")

private PersonService personService;

## 适用场景

##Autowired：效率低下，先按类型查找，再按名字查找

##Resource：按名字查找，后面要跟参数name，好处：当有多个Impl实现类时，可以通name快速找到

注：1.当只有一个Impl实类的时候，随便用哪个都差不多，>=2的时候，最好用@Resource，比@Autowired@Qualifier()效率高。

2.不用名字使用autoware，使用名字的时候resource。

## @Autowired

### @Autowired默认按类型装配（这个注解是属业spring的），默认情况下必须要求依赖对象必须存在，如果要允许null值，可以设置它的required属性为false，如：@Autowired(required=false) ，如果我们想使用名称装配可以结合@Qualifier注解进行使用。

### @Autowired注入的时候要确保接口只有一个实现类。

### @Autowired @Qualifie("userService") 是直接按照名字进行搜索，也就是说，对于UserServiceImpl 上面@Service注解必须写名字，不写就会报错，而且名字必须是@Autowired @Qualifie("userService") 保持一致。如果@Service上面写了名字，而@Autowired @Qualifie() ，一样会报错。

### 一般@Autowired和@Qualifier一起用，@Resource单独用，当然没有冲突的话@Autowired也可以单独用。

## @Resource

### 一、@Resource 默认情况下是按照名称进行匹配，如果没有找到相同名称的Bean，则会按照类型进行匹配，有人可能会想了，这下好了，用这个是万能的了，不用管名字了，也不用管类型了，但这里还是有缺点。首先，根据这个注解的匹配效果可以看出，它进行了两次匹配，也就是说，如果你在UserService这个类上面这样写注解，@Service,它会怎么找呢，首先是找相同名字的，如果没有找到，再找相同类型的，而这里的@Service没有写名字，这个时候就进行了两次搜索，显然，速度就下降了许多。也许你还会问，这里的@Service本来就没有名字，肯定是直接进行类型搜索啊。其实不是这样的，UserServiceImpl 上面如果有@Service默认的名字 是这个userServiceImpl，注意看，就是把类名前面的大写变成小写，就是默认的Bean的名字了。 @Resource根据名字搜索是这样写@Resource("userService")，如果你写了这个名字叫userService，那么UserServiceImpl上面必须也是这个名字，不然还是会报错。