Spring框架中进行注入式，使用@Autowired.

@Autowired可以对成员变量、方法和构造函数进行标注，来完成自动装配的工作，这里必须明确：@Autowired是根据类型进行自动装配的，如果需要按名称进行装配，则需要配合@Qualifier[1] 使用；

@Autowired标注可以放在成员变量上，也可以放在成员变量的set方法上。前者，Spring会直接将UserDao类型的唯一一个bean赋值给userDao这个成员变量；后者，Spring会调用setUserDao方法来将UserDao类型的唯一一个bean装配到userDao这个属性。

Spring 2.5 引入了 @Autowired 注释，它可以对类成员变量、方法及构造函数进行标注，完成自动装配的工作。 通过 @Autowired的使用来消除 set ，get方法。

要实现我们要精简程序的目的。需要这样来处理：   
　　\* 在applicationContext.xml中加入：

**[c-sharp]** view plain[copy](http://baike.baidu.com/view/596598.htm)[print](http://baike.baidu.com/view/87232.htm)?

1. <!-- 该 BeanPostProcessor 将自动对标注 @Autowired 的 Bean 进行注入 -->
2. <bean class="org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor"/>

Spring 通过一个 BeanPostProcessor 对 @Autowired 进行解析，所以要让 @Autowired 起作用必须事先在 Spring 容器中声明 AutowiredAnnotationBeanPostProcessor Bean。

　　**\* 修改在原来注入spirng容器中的bean的方法。**   
　　在域变量上加上标签@Autowired,并且去掉 相应的get 和set方法

清单 6. 使用 @Autowired 注释的 Boss.java

**[java]** view plain[copy](http://baike.baidu.com/view/596598.htm)[print](http://baike.baidu.com/view/87232.htm)?

1. package com.baobaotao;
2. import org.springframework.beans.factory.annotation.Autowired;
3. public class Boss {
4. @Autowired
5. private Car car;
6. @Autowired
7. private Office office;
8. …
9. }

package com.baobaotao; import org.springframework.beans.factory.annotation.Autowired; public class Boss { @Autowired private Car car; @Autowired private Office office; … }

**\* 在applicationContext.xml中 把原来 引用的<property >标签也去掉。**

**[xhtml]** view plain[copy](http://baike.baidu.com/view/596598.htm)[print](http://baike.baidu.com/view/87232.htm)?

1. <?xml version="1.0" encoding="UTF-8" ?>
2. <beans xmlns="http://www.springframework.org/schema/beans"
3. xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4. xsi:schemaLocation="http://www.springframework.org/schema/beans
5. http://www.springframework.org/schema/beans/spring-beans-2.5.xsd">
6. <!-- 该 BeanPostProcessor 将自动起作用，对标注 @Autowired 的 Bean 进行自动注入 -->
7. <bean class="org.springframework.beans.factory.annotation.
8. AutowiredAnnotationBeanPostProcessor"/>
9. <!-- 移除 boss Bean 的属性注入配置的信息 -->
10. <bean id="boss" class="com.baobaotao.Boss"/>
11. <bean id="office" class="com.baobaotao.Office">
12. <property name="officeNo" value="001"/>
13. </bean>
14. <bean id="car" class="com.baobaotao.Car" scope="singleton">
15. <property name="brand" value=" 红旗 CA72"/>
16. <property name="price" value="2000"/>
17. </bean>
18. </beans>

<?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-2.5.xsd"> <!-- 该 BeanPostProcessor 将自动起作用，对标注 @Autowired 的 Bean 进行自动注入 --> <bean class="org.springframework.beans.factory.annotation. AutowiredAnnotationBeanPostProcessor"/> <!-- 移除 boss Bean 的属性注入配置的信息 --> <bean id="boss" class="com.baobaotao.Boss"/> <bean id="office" class="com.baobaotao.Office"> <property name="officeNo" value="001"/> </bean> <bean id="car" class="com.baobaotao.Car" scope="singleton"> <property name="brand" value=" 红旗 CA72"/> <property name="price" value="2000"/> </bean> </beans>

这样，当 Spring 容器启动时，AutowiredAnnotationBeanPostProcessor 将扫描 Spring 容器中所有 Bean，当发现 Bean 中拥有 @Autowired 注释时就找到和其匹配（默认按类型匹配）的 Bean，并注入到对应的地方中去。   
　　  
　　按照上面的配置，Spring 将直接采用 Java 反射机制对 Boss 中的 car 和 office 这两个私有成员变量进行自动注入。所以对成员变量使用 @Autowired 后，您大可将它们的 setter 方法（setCar() 和 setOffice()）从 Boss 中删除。   
　　  
　　当然，您也可以通过 @Autowired 对方法或构造函数进行标注，如果构造函数有两个入参，分别是 bean1 和 bean2，@Autowired 将分别寻找和它们类型匹配的 Bean，将它们作为 CountryService (Bean1 bean1 ,Bean2 bean2) 的入参来创建 CountryService Bean。来看下面的代码： 对方法

**[java]** view plain[copy](http://baike.baidu.com/view/596598.htm)[print](http://baike.baidu.com/view/87232.htm)?

1. package com.baobaotao;
2. public class Boss {
3. private Car car;
4. private Office office;
5. @Autowired
6. public void setCar(Car car) {
7. this.car = car;
8. }
9. @Autowired
10. public void setOffice(Office office) {
11. this.office = office;
12. }
13. …
14. }

package com.baobaotao; public class Boss { private Car car; private Office office; @Autowired public void setCar(Car car) { this.car = car; } @Autowired public void setOffice(Office office) { this.office = office; } … }

2标注[编辑](http://baike.baidu.com/link?url=_YOkQkAdhoq5DMf0wsw_XKDDDKEgEvm3-4iAizhk4H2118clEm6OjhYkdgxxijnXbFA_f6U7isG7CtbSPMkGlq" \o "编辑本段)

这时，@Autowired 将查找被标注的方法的入参类型的 Bean，并调用方法自动注入这些 Bean。而下面的使用方法则对构造函数进行标注：

**[java]** view plain[copy](http://baike.baidu.com/view/596598.htm)[print](http://baike.baidu.com/view/87232.htm)?

1. package com.baobaotao;
2. public class Boss {
3. private Car car;
4. private Office office;
5. @Autowired
6. public Boss(Car car ,Office office){
7. this.car = car;
8. this.office = office ;
9. }
10. …
11. }

package com.baobaotao; public class Boss { private Car car; private Office office; @Autowired public Boss(Car car ,Office office){ this.car = car; this.office = office ; } … }

由于 Boss() 构造函数有两个入参，分别是 car 和 office，@Autowired 将分别寻找和它们类型匹配的 Bean，将它们作为 Boss(Car car ,Office office) 的入参来创建 Boss Bean。

Autowired也可以手动指定按照byName方式注入，使用@Qualifier标签，例如：

@Autowired ()

 @Qualifier ( "baseDao" )

@Resource（这个注解属于J2EE的）的标签，默认是按照byName方式注入的

 (1).@Autowired 与@Resource都可以用来装配bean. 都可以写在字段上,或写在setter方法上;  
   (2).@Autowired 默认按类型装配，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;

把DAO实现类注入到action的service接口(注意不要是service的实现类)中，注入时不要new 这个注入的类，因为spring会自动注入，如果手动再new的话会出现错误，  
然后属性加上@Autowired后不需要getter()和setter()方法，Spring也会自动注入。

在接口前面标上@Autowired注释使得接口可以被容器注入，如：

**[java]** [view plaincopy](http://blog.csdn.net/zhang854429783/article/details/6785574)

1. @Autowired
2. @Qualifier("chinese")
3. **private** Man man;