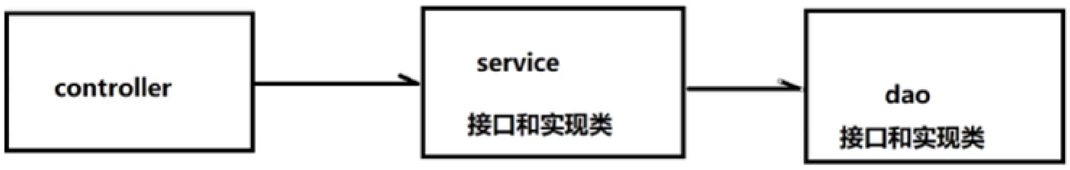
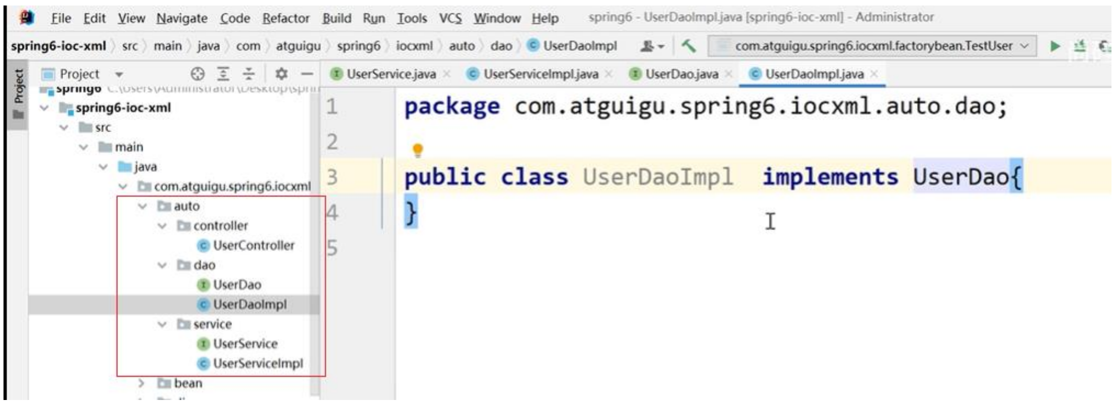
实验 13：基于 xml 的自动装配

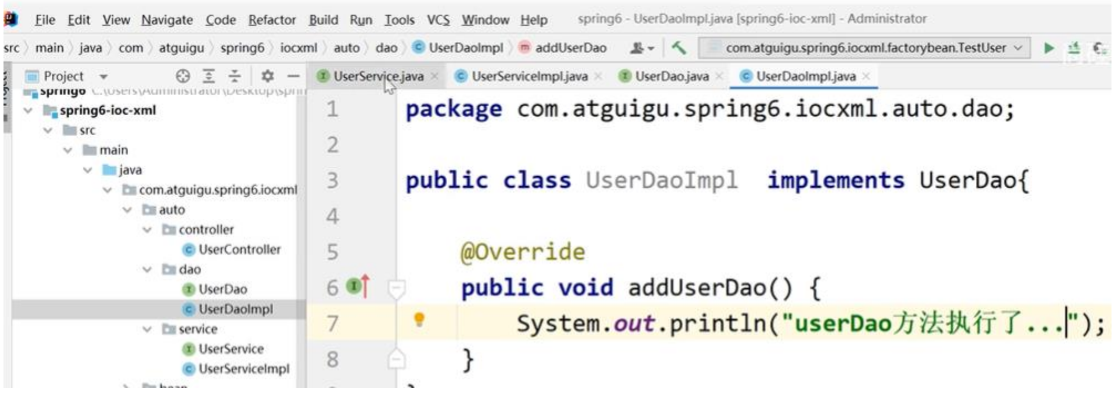
自动装配：根据指定的策略，在 IOC 容器中匹配某一个 bean，自动为指定的 bean 中所依赖的类类型或接口类型属性赋值



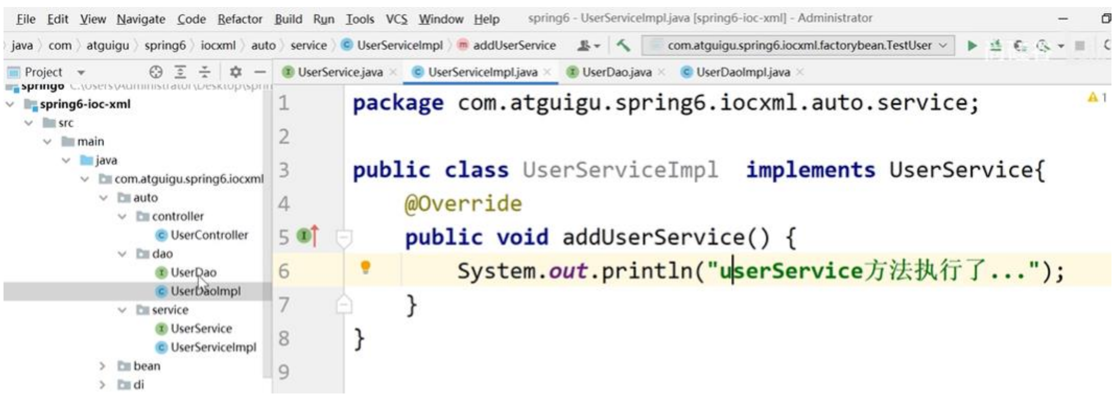
环境准备：



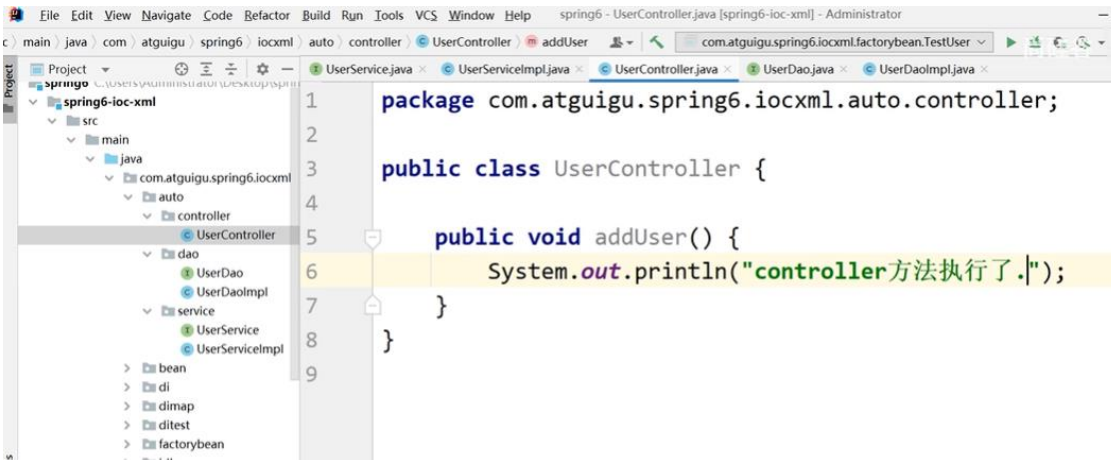
```
package com.atguigu.spring6.iocxml.auto.dao;

public class UserDaoImpl  implements UserDao{
}
```



```
package com.atguigu.spring6.iocxml.auto.dao;

public class UserDaoImpl  implements UserDao{

    @Override
    public void addUserDao() {
        System.out.println("userDao方法执行了...");
    }
```

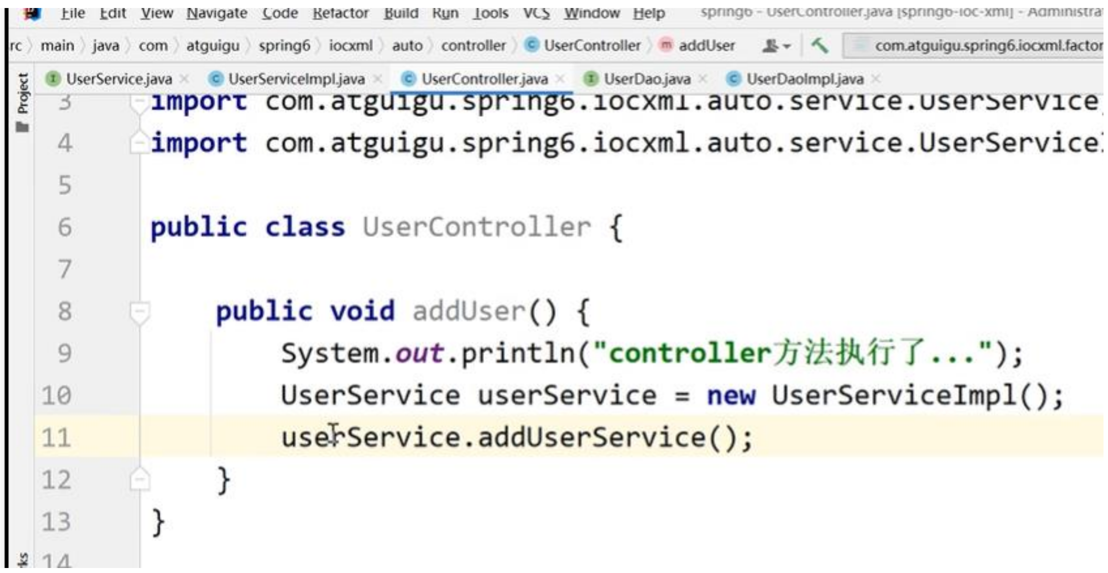


```
package com.atguigu.spring6.iocxml.auto.service;

public class UserServiceImpl  implements UserService{
    @Override
    public void addUserService() {
        System.out.println("userService方法执行了...");
    }
}
```

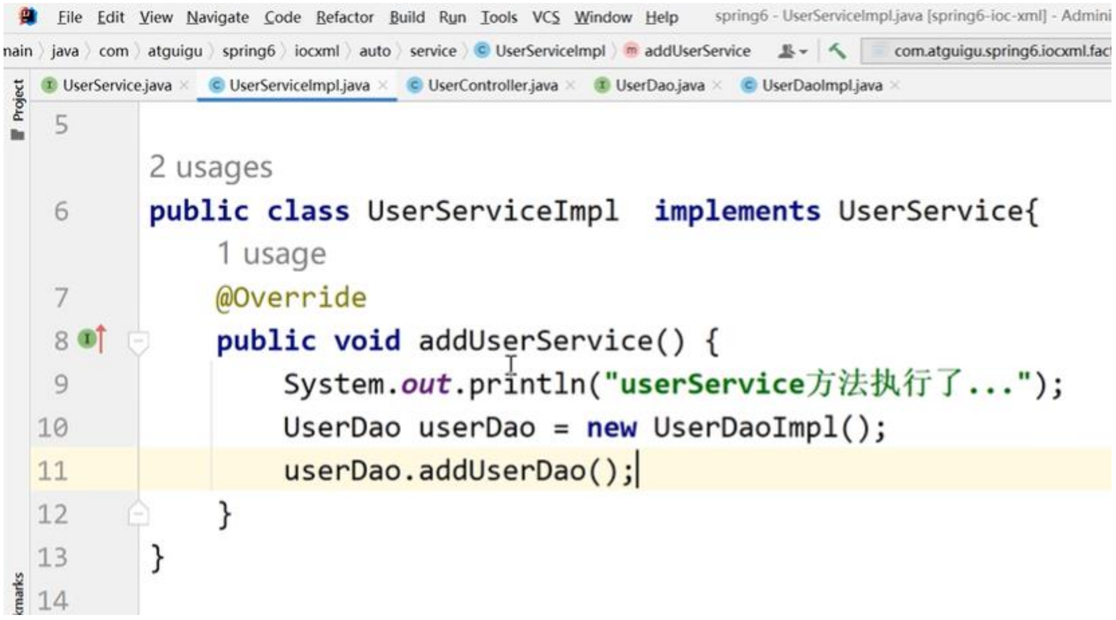


```java
package com.atguigu.spring6.iocxml.auto.controller;

public class UserController {

    public void addUser() {
        System.out.println("controller方法执行了.");
    }
}
```

原始方法：



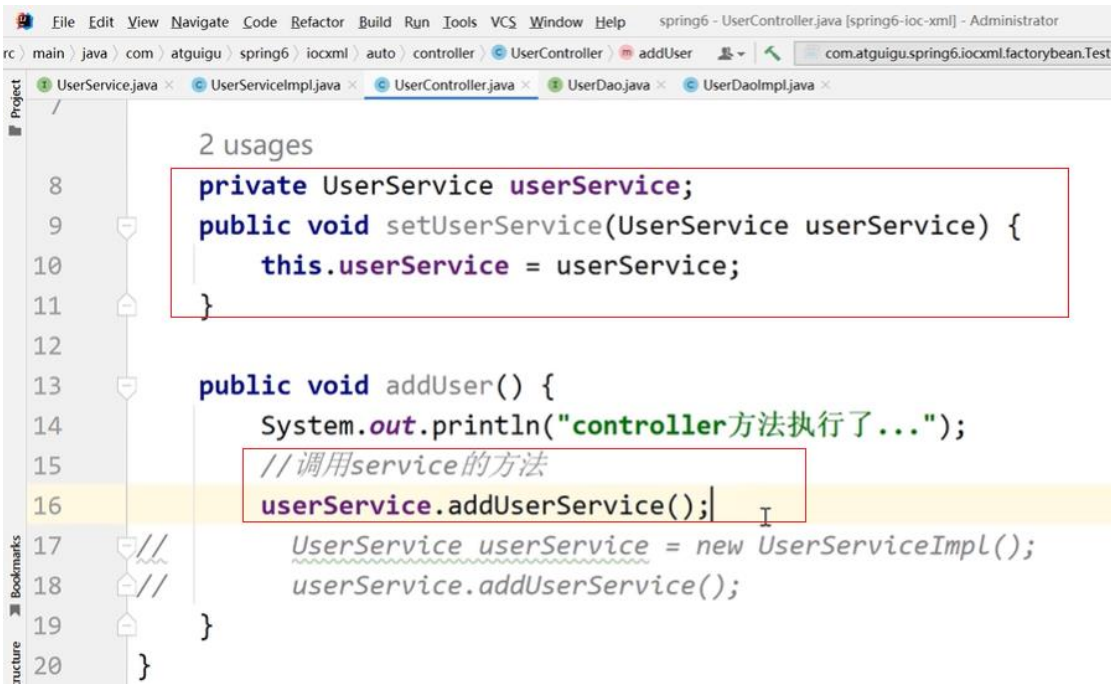
```java
import com.atguigu.spring6.iocxml.auto.service.UserService
import com.atguigu.spring6.iocxml.auto.service.UserService

public class UserController {

    public void addUser() {
        System.out.println("controller方法执行了...");
        UserService userService = new UserServiceImpl();
        userService.addUserService();
    }
}
```


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
spring6 - UserServiceImpl.java [spring6-ioc-xml] - Admini
UserService.java
UserServiceImpl.java
UserController.java
UserDao.java
UserDaoImpl.java
2 usages
1 usage

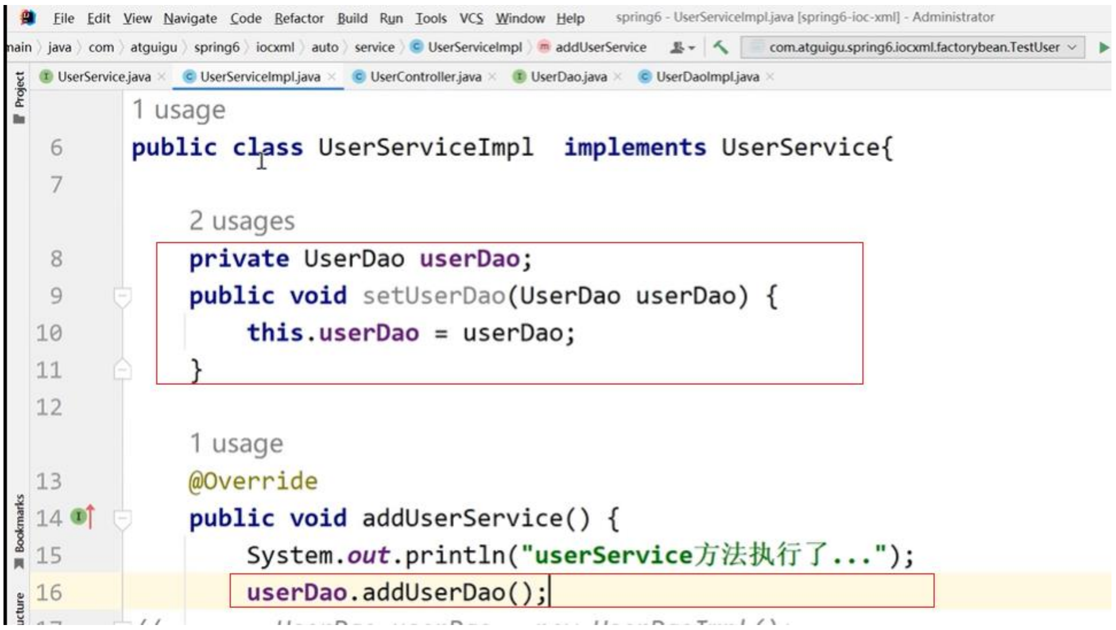
```java
public class UserServiceImpl  implements UserService{
    @Override
    public void addUserService() {
        System.out.println("userService方法执行了...");
        UserDao userDao = new UserDaoImpl();
        userDao.addUserDao();
    }
}
```

通过配置完成自动装配：

准备:

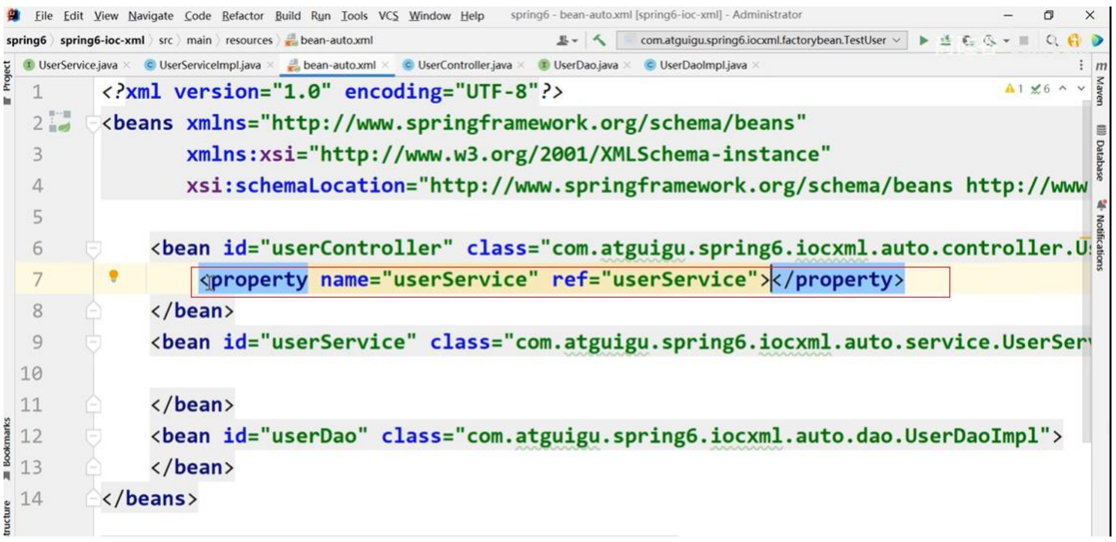

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
spring6 - UserController.java [spring6-ioc-xml] - Administrator
rc main java com atguigu spring6 iocxml auto controller UserController addUser
com.atguigu.spring6.iocxml.factorybean.Test
UserService.java
UserServiceImpl.java
UserController.java
UserDao.java
UserDaoImpl.java

```java
    2 usages
    private UserService userService;
    public void setUserService(UserService userService) {
        this.userService = userService;
    }

    public void addUser() {
        System.out.println("controller方法执行了...");
        //调用service的方法
        userService.addUserService();
//        UserService userService = new UserServiceImpl();
//        userService.addUserService();
    }
}
```


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
spring6 - UserServiceImpl.java [spring6-ioc-xml] - Administrator
main java com atguigu spring6 iocxml auto service UserServiceImpl addUserService
com.atguigu.spring6.iocxml.factorybean.TestUser
UserService.java
UserServiceImpl.java
UserController.java
UserDao.java
UserDaoImpl.java

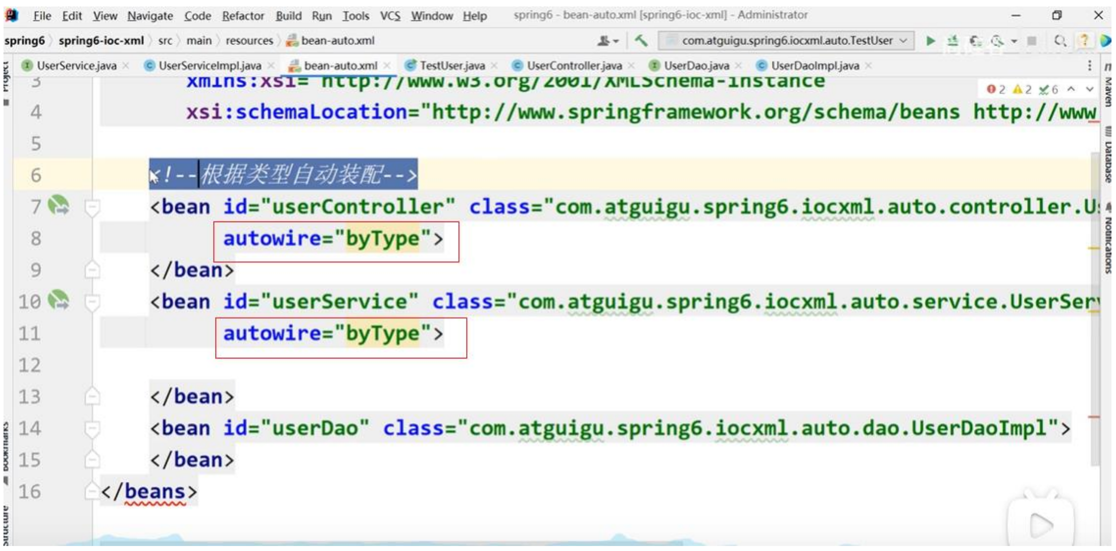
```java
1 usage
public class UserServiceImpl  implements UserService{

    2 usages
    private UserDao userDao;
    public void setUserDao(UserDao userDao) {
        this.userDao = userDao;
    }

    1 usage
    @Override
    public void addUserService() {
        System.out.println("userService方法执行了...");
        userDao.addUserDao();
```

之前的方法:



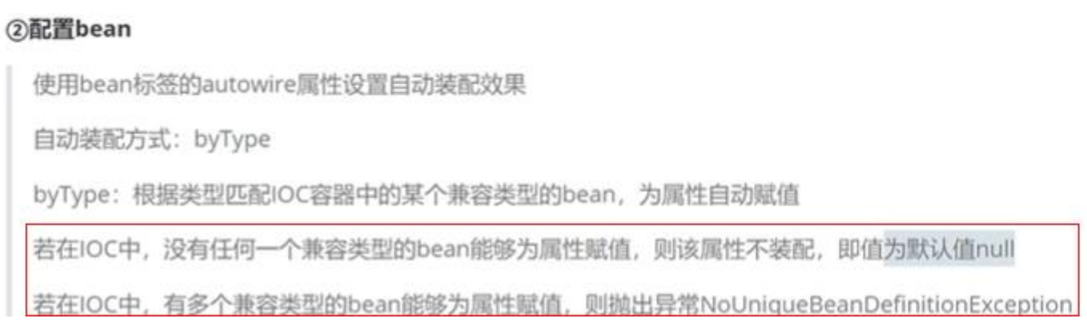
```xml
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans http://www

    <bean id="userController" class="com.atguigu.spring6.iocxml.auto.controller.U
        <property name="userService" ref="userService"></property>
    </bean>
    <bean id="userService" class="com.atguigu.spring6.iocxml.auto.service.UserSer

    </bean>
    <bean id="userDao" class="com.atguigu.spring6.iocxml.auto.dao.UserDaoImpl">
    </bean>
</beans>
```

基于类型自动装配:



```xml
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance
       xsi:schemaLocation="http://www.springframework.org/schema/beans http://www

    <!--根据类型自动装配-->
    <bean id="userController" class="com.atguigu.spring6.iocxml.auto.controller.U
          autowire="byType">
    </bean>
    <bean id="userService" class="com.atguigu.spring6.iocxml.auto.service.UserSer
          autowire="byType">

    </bean>
    <bean id="userDao" class="com.atguigu.spring6.iocxml.auto.dao.UserDaoImpl">
    </bean>
</beans>
```

基于名称自动装配:



②配置bean

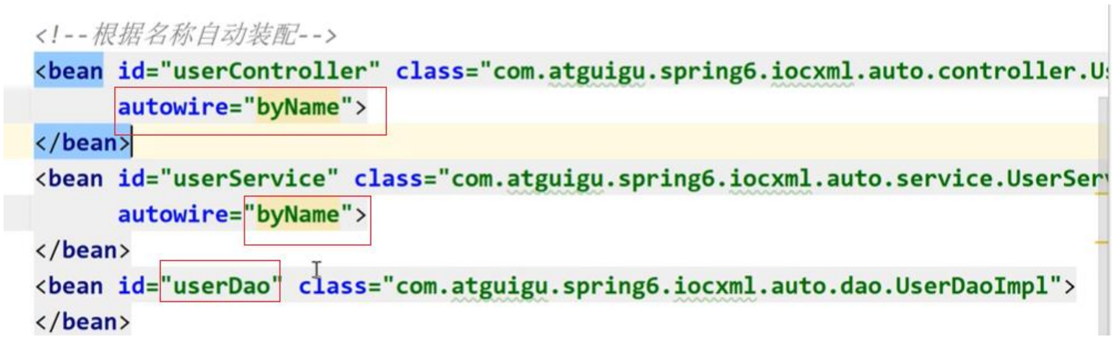
使用bean标签的autowire属性设置自动装配效果

自动装配方式：byType

byType：根据类型匹配IOC容器中的某个兼容类型的bean，为属性自动赋值

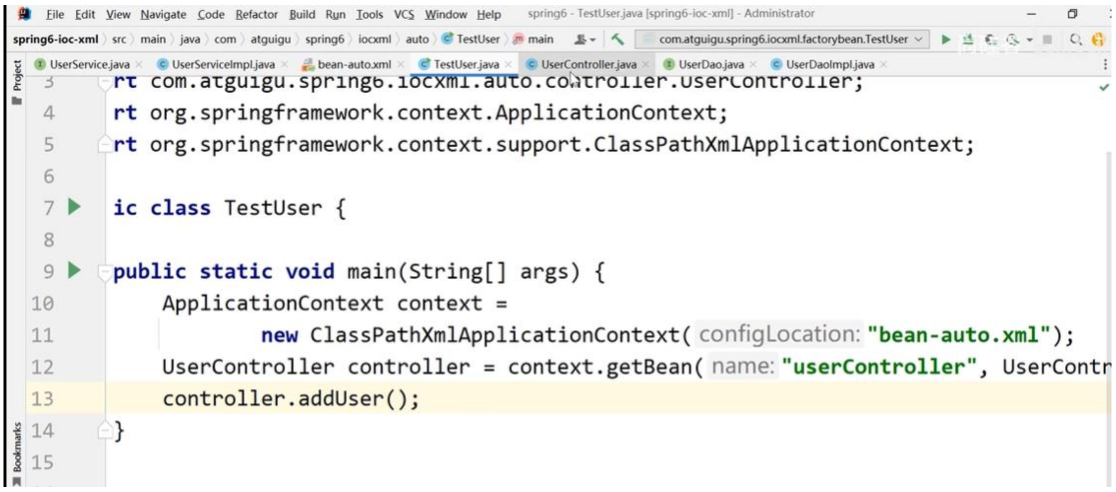
若在IOC中，没有任何一个兼容类型的bean能够为属性赋值，则该属性不装配，即值为默认值null

若在IOC中，有多个兼容类型的bean能够为属性赋值，则抛出异常NoUniqueBeanDefinitionException

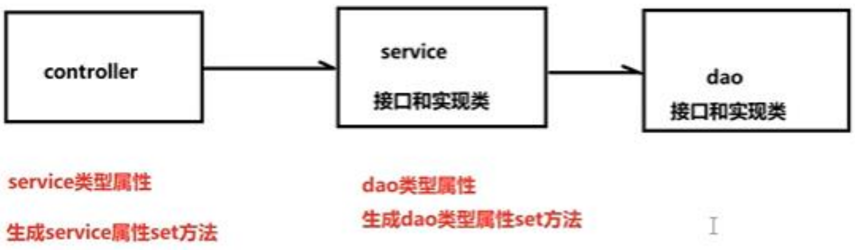


```xml
<!--根据名称自动装配-->
<bean id="userController" class="com.atguigu.spring6.iocxml.auto.controller.U
      autowire="byName">
</bean>
<bean id="userService" class="com.atguigu.spring6.iocxml.auto.service.UserSer
      autowire="byName">
</bean>
<bean id="userDao" class="com.atguigu.spring6.iocxml.auto.dao.UserDaoImpl">
</bean>
```

测试:


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
spring6 - TestUser.java [spring6-ioc-xml] - Administrator
spring6-ioc-xml 〉 src 〉 main 〉 java 〉 com 〉 atguigu 〉 spring6 〉 iocxml 〉 auto 〉 TestUser 〉 main
com.atguigu.spring6.iocxml.factorybean.TestUser
UserService.java
UserServiceImpl.java
bean-auto.xml
TestUser.java
UserController.java
UserDao.java
UserDaoImpl.java

```java
rt com.atguigu.spring6.iocxml.auto.controller.UserController;
rt org.springframework.context.ApplicationContext;
rt org.springframework.context.support.ClassPathXmlApplicationContext;

ic class TestUser {

public static void main(String[] args) {
    ApplicationContext context =
            new ClassPathXmlApplicationContext( configLocation: "bean-auto.xml");
    UserController controller = context.getBean( name: "userController", UserContr
    controller.addUser();
}

```


controller
service
接口和实现类
dao
接口和实现类
service类型属性
生成service属性set方法
dao类型属性
生成dao类型属性set方法