


```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help spring6 - UserDaoImpl.java [spring6-ioc-annotation]
spring6-ioc-annotation > src > main > java > com > atguigu > spring6 > autowired > dao > UserDaoImpl > add com.atguigu
UserController.java x UserService.java x UserServiceImpl.java x UserDao.java x UserDaoImpl.java x
1 package com.atguigu.spring6.autowired.dao;
2
3 import org.springframework.stereotype.Repository;
4
5 @Repository
6 public class UserDaoImpl implements UserDao{
7     @Override
8     public void add() {
9         System.out.println("dao.....");
10    }
11 }
12
```

```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help spring6 - UserController.java [spring6-ioc-annotation] - Administrator
spring6-ioc-annotation > src > main > java > com > atguigu > spring6 > autowired > controller > UserController > add com.atguigu.spring6.bean.TestUser
UserController.java x UserService.java x UserServiceImpl.java x UserDao.java x UserDaoImpl.java x
7 @Controller
8 public class UserController {
9
10    //注入service
11    //第一种方式 属性注入
12    1 usage
13    @Autowired //根据类型找到对应对象, 完成注入
14    private UserService userService;
15
16    public void add() {
17        System.out.println("controller.....");
18        userService.add();
19    }
20 }
```

三个对象完成创建:

```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help spring6 - UserController.java [spring6-ioc-annotation] - Administrator
spring6-ioc-annotation > src > main > java > com > atguigu > spring6 > autowired > controller > UserController > add com.atguigu.spring6.bean.TestUser
UserController.java x UserService.java x UserServiceImpl.java x UserDao.java x UserDaoImpl.java x TestUser.java x
Project
spring6
spring6-ioc-annotation
src
main
java
com.atguigu.spring6
autowired
controller
UserController
dao
UserDao
UserDaoImpl
service
UserService
UserServiceImpl
bean
1 package com.atguigu.spring6.autowired.controller;
2
3 import org.springframework.stereotype.Controller;
4
5 @Controller
6 public class UserController {
7 }
8
```

```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help spring6 - UserServiceImpl.java [spring6-ioc-annotation] - Administrator
spring6-ioc-annotation src main java com atguigu spring6 autowired service UserServiceImpl com.atguigu.spring6.bean.TestUser
UserController.java UserService.java UserServiceImpl.java UserDao.java UserDaoImpl.java TestUser.java
1 package com.atguigu.spring6.autowired.service;
2
3 import org.springframework.stereotype.Service;
4
5 @Service
6 public class UserServiceImpl implements UserService{
7 }
8
```

```
spring6 spring6-ioc-annotation src main java com atguigu spring6 autowired dao UserDaoImpl com.atguigu
UserController.java UserService.java UserServiceImpl.java UserDao.java UserDaoImpl.java
1 package com.atguigu.spring6.autowired.dao;
2
3 import org.springframework.stereotype.Repository;
4
5 @Repository
6 public class UserDaoImpl implements UserDao{
7 }
8
```

场景一：属性的注入

```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help spring6 - UserController.java [spring6-ioc-annotation] - Administrator
spring6-ioc-annotation src main java com atguigu spring6 autowired controller UserController UserService com.atguigu.spring6.bean.TestUser
UserController.java UserService.java UserServiceImpl.java UserDao.java UserDaoImpl.java
3 import com.atguigu.spring6.autowired.service.UserService;
4 import org.springframework.beans.factory.annotation.Autowired;
5 import org.springframework.stereotype.Controller;
6
7 @Controller
8 public class UserController {
9
10 //注入service
11 //第一种方式 属性注入
12 @Autowired //根据类型找到对应对象，完成注入
13 private UserService userService;
14
15
```

```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help spring6 - UserServiceImpl.java [spring6-ioc-annotation] - Administrator
spring6-ioc-annotation src main java com atguigu spring6 autowired service UserServiceImpl userDao com.atguigu.spring6.bean.TestUser

UserController.java x UserService.java x UserServiceImpl.java x UserDao.java x UserDaoImpl.java x
Project
3 import com.atguigu.spring6.autowired.dao.UserDao;
4 import org.springframework.beans.factory.annotation.Autowired;
5 import org.springframework.stereotype.Service;
6
7 @Service
8 public class UserServiceImpl implements UserService{
9
10 //注入dao
11 //第一种方式 属性注入
12 @Autowired //根据类型找到对应对象，完成注入
13 private UserDao userDao;
14
15 }
16
Bookmarks
are
```

测试:

```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help spring6 - TestUserController.java [spring6-ioc-annotation] - Administrator
spring6-ioc-annotation src main java com atguigu spring6 autowired TestUserController main com.atguigu.spring6.bean.TestUser

UserController.java x TestUserController.java x UserService.java x UserServiceImpl.java x UserDao.java x UserDaoImpl.java x
Project
3 import com.atguigu.spring6.autowired.controller.UserController;
4 import org.springframework.context.ApplicationContext;
5 import org.springframework.context.support.ClassPathXmlApplicationContext;
6
7 public class TestUserController {
8
9 public static void main(String[] args) {
10     ApplicationContext context =
11         new ClassPathXmlApplicationContext( configLocation: "bean.xml");
12     UserController controller = context.getBean(UserController.class);
13     controller.add();
14 }
15 }
16
Bookmarks
uture
```

场景二：set 注入

The screenshot shows an IDE window titled 'spring6 - UserController.java [spring6-ioc-annotation] - Administrator'. The file 'UserController.java' is open, showing lines 13 to 27. A red box highlights the following code:

```
//第二种方式 set方法注入
2 usages
private UserService userService;

@Autowired
public void setUserService(UserService userService) {
    this.userService = userService;
}

public void add() {
    System.out.println("controller.....");
    userService.add();
}
```

The screenshot shows an IDE window titled 'spring6-ioc-annotation'. The file 'UserController.java' is open, showing lines 10 to 24. A red box highlights the following code:

```
//注入dao
//第一种方式 属性注入
// @Autowired //根据类型找到对应对象，完成注入
private UserDao userDao;

//第二种方式 set方法注入
2 usages
private UserDao userDao;

@Autowired
public void setUserDao(UserDao userDao) {
    this.userDao = userDao;
}

@Override
public void add() {
```

场景三：构造方法注入


```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help spring6 - UserController.java [spring6-ioc-annotation] - Administrator
spring6 spring6-ioc-annotation src main java com atguigu spring6 autowired controller UserController TestUserController
UserController.java UserServiceImpl.java UserDaoImpl.java TestUserController.java
19 // public void setUserService(UserService userService) {
20 //     this.userService = userService;
21 // }
22
23 // 第三种方式 构造方法注入
24 // 2 usages
25 private UserService userService;
26
27 @Autowired
28 public UserController(UserService userService) {
29     this.userService = userService;
30 }
31
32 public void add() {
```

```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help spring6 - UserServiceImpl.java [spring6-ioc-annotation] - Administrator
spring6 spring6-ioc-annotation src main java com atguigu spring6 autowired service UserServiceImpl TestUserController
UserController.java UserServiceImpl.java UserDaoImpl.java TestUserController.java
22
23 // 第三种方式 构造方法注入
24 // 2 usages
25 private UserDao userDao;
26
27 @Autowired
28 public UserServiceImpl(UserDao userDao) {
29     this.userDao = userDao;
30 }
31
32 @Override
33 public void add() {
34     System.out.println("service.....");
35     userDao.add();
36 }
37
```

场景四：形参上注入

```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help spring6 - UserController.java [spring6-ioc-annotation] - Administrator
spring6-ioc-annotation src main java com atguigu spring6 autowired controller UserController UserController TestUserController
UserController.java UserServiceImpl.java UserDaoImpl.java TestUserController.java
28 // this.userService = userService;
29 // }
30
31 // 第四种方式 形参上注入
32 // 2 usages
33 private UserService userService;
34
35 public UserController(@Autowired UserService userService) {
36     this.userService = userService;
37 }
38
39 public void add() {
40     System.out.println("controller.....");
41     userService.add();
42 }
```

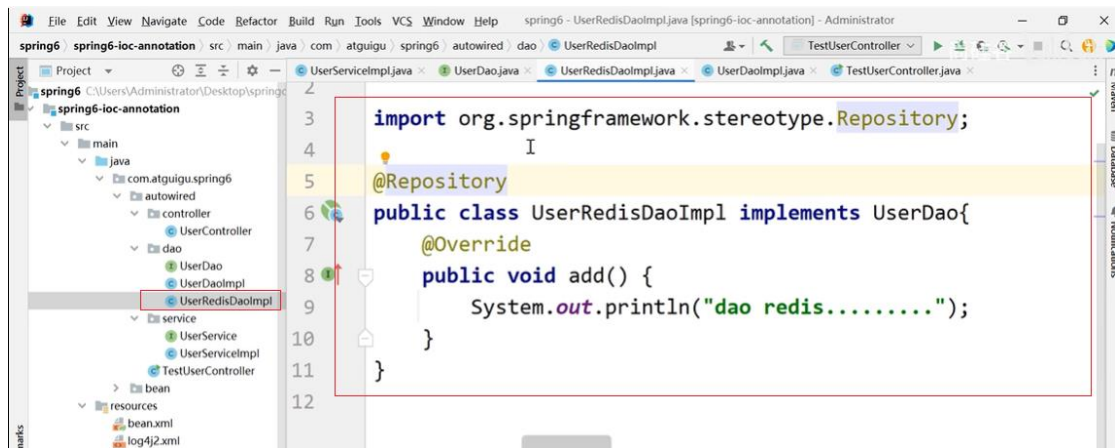
```
25 //
26 // @Autowired
27 // public UserServiceImpl(UserDao userDao) {
28 //     this.userDao = userDao;
29 // }
30
31 //第四种方式 形参上注入
32 // 2 usages
33 private UserDao userDao;
34 public UserServiceImpl(@Autowired UserDao userDao) {
35     this.userDao = userDao;
36 }
37
```

场景五：只有一个构造函数，无注解

```
34 // public UserController(@Autowired UserService userService) {
35 //     this.userService = userService;
36 // }
37
38 //第五种方式 只有一个有参数构造函数，无注解
39 // 2 usages
40 private UserService userService;
41 public UserController(UserService userService) {
42     this.userService = userService;
43 }
44
45 public void add() {
46     System.out.println("controller.....");
47     userService.add();
48 }
```

场景六：@Autowired 注解和@Qualifier 注解联合

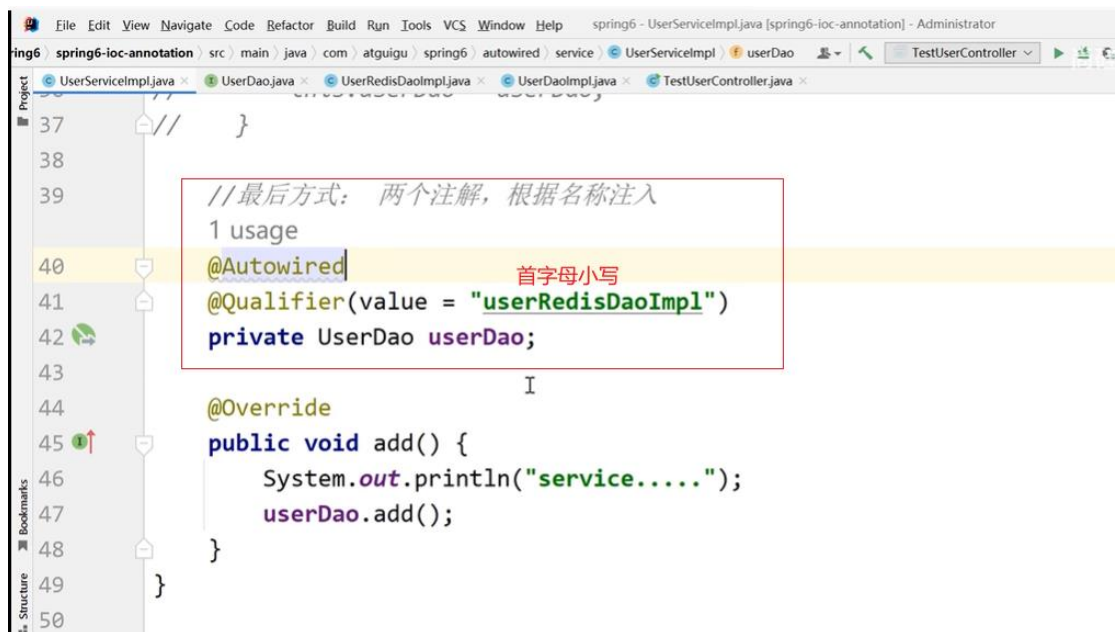
根据名称注入：名称类的首字母小写



```
import org.springframework.stereotype.Repository;

@Repository
public class UserRedisDaoImpl implements UserDao {

    @Override
    public void add() {
        System.out.println("dao redis.....");
    }
}
```



```
//最后方式： 两个注解，根据名称注入
1 usage
@Autowired //首字母小写
@Qualifier(value = "userRedisDaoImpl")
private UserDao userDao;

@Override
public void add() {
    System.out.println("service.....");
    userDao.add();
}
```