

ringSingleton的陷阱介绍 PDF转换可能丢失图片或格式，建议
阅读原文

https://www.100test.com/kao_ti2020/461/2021_2022_ringSingle_c104_461628.htm 这是一个真实的案例，我们在项目中使

用Spring和ACEGI，我之所以选择ACEGI，除了它对权限的良好控制外，我还看好它的SecurityContextHolder，通过代码

```
Authentication auth = SecurityContextHolder.getContext ( )
```

。getAuthentication ()；我可以很容易在系统任意一层得到用户的信息，而不用把用户信息在参数里传来传去，（这也是struts的缺点之一）但是我在每一次要得到用户信息的时候

都写上面的一段代码，未免有些麻烦，所以我在BaseService，

BaseDao里都提供了如下方法：/**/** * get current login user

```
info * @return UserInfo */ protected UserInfo getUserInfo ( ) ...
```

```
...{ return getUserContext ( )。getUserInfo ( ) ; }/**/** * get
```

```
current login user context * @return UserContext */ protected
```

```
UserContext getUserContext ( ) .....{ Authentication auth =
```

```
SecurityContextHolder.getContext ( )。getAuthentication ( )
```

```
； return ( UserContext ) auth.getPrincipal ( ) ; }这样在其他的
```

Service和Dao类里可以通过 super.getUserContext ()，

super.getUserInfo () 来得到用户的信息，这也为问题的产生

提供了温床。请看如下代码：public class SomeService extends

```
BaseService implements SomeInterface .....{ private UserInfo user
```

```
= super.getUserInfo ( ) ; public someMethod ( ) .....{ int
```

```
userID = this.user.getUserID ( ) ; String userName =
```

```
this.user.getUserName ( ) ; //bla bla do something user userID
```

```
and userNaem } }这段代码在单元测试的时候不会用任何问题
```

，但是在多用户测试的情况下，你会发现任何调用SomeService里someMethod（）方法的userID和userName都是同一个人，也就是第一个登陆的人的信息。Why？其根本原因是Spring的Bean在默认情况下是Singleton的，Bean SomeService的实例只会生成一份，也就是所SomeService实例的user对象只会被初始化一次，就是第一次登陆人的信息，以后不会变了。所以BaseService想为开发提供方便，确给开发带来了风险正确的用法应该是这样的

```
public class SomeService extends BaseService implements SomeInterface .....{ public someMethod（） .....{ int userID = super.getUserInfo（）.getUserID（）； String userName = super.getUserInfo（）.getUserName（）； //bla bla do something user userID and userNaem }}
```

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com