Spiring IOC相关问题

1. IOC 容器只存放单例bean吗 (G96)
2. 结论：IOC 容器只存放单例bean

IOC容器初始化的时候，会将所有bean初始化在singletonObjects这个CurrentHashMap中， bean是单例的。

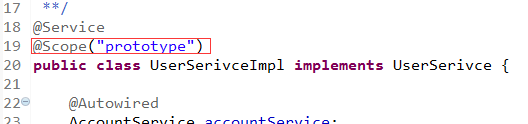
在获取bean的时候，首先会从singletonObjects去取值，通过debug，发现如果scope是单例，则可以获取到bean，如果scope是多例，则获取不到bean，需要 从一个叫mergedBeanDefinitions的CurrentHashMap中去获取bean的定义，然后再根据bean的scope去决定如何创建bean，如果scope=prototype，则每次都会创建一个新的实例。

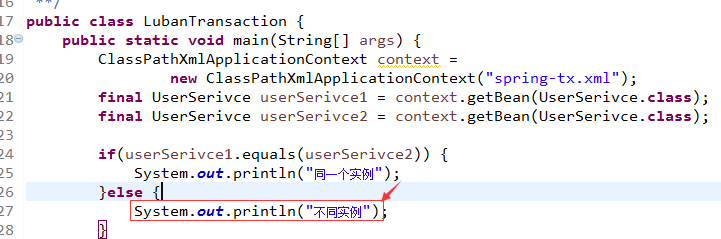
猜想：IOC在初始化时，只会将scope= singleton（单例）的对象进行实例化，而不会去实例化scope=prototype的对象（多例）；

证实：找到singletonObjects.put方法，debug看一下singletonObjects.put的前提条件是什么

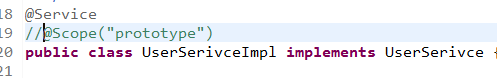
2、场景测试：

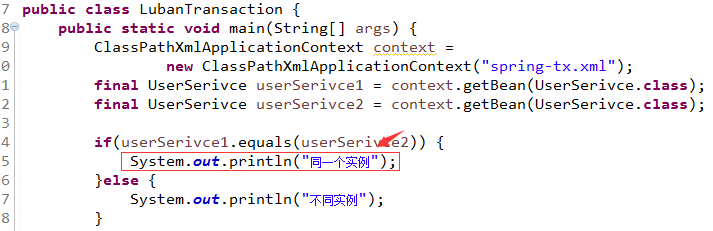
作用域为多例，获取不同实例





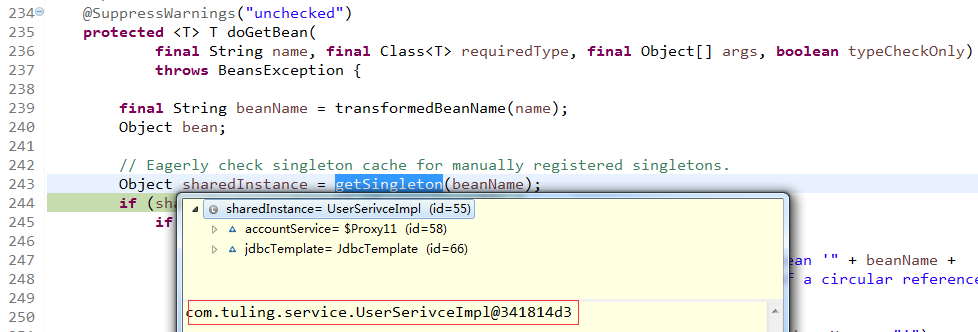
作用域为单例，获取相同实例



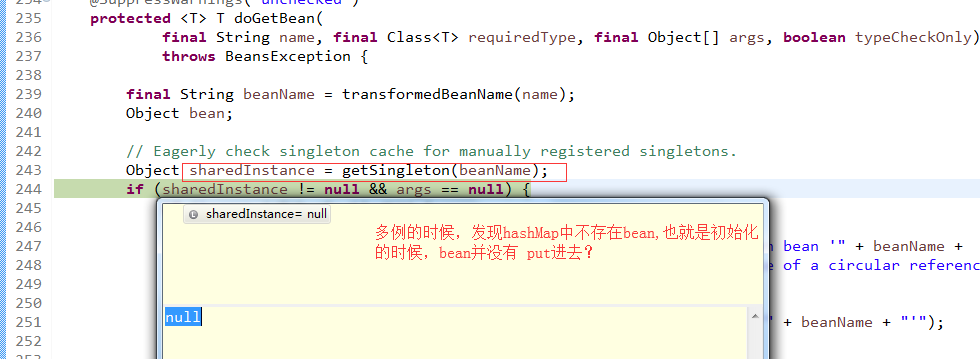


3、源码分析：AbstractBeanFactory#doGetBean

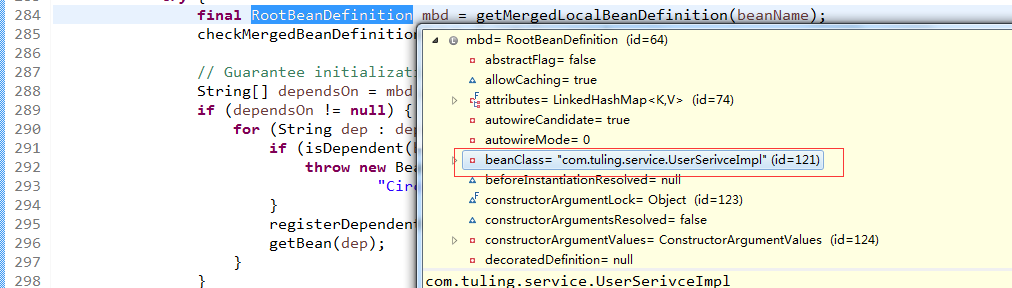
单例的场景，直接从singletonObjects 这个Map中获取bean



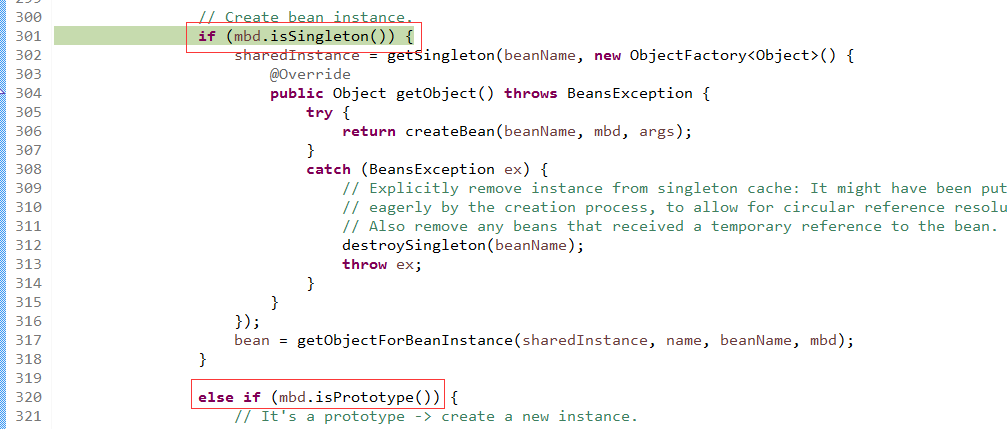
多例的场景，发现从singletonObjects中拿不到值



接下来，发现它是从一个叫 mergedBeanDefinitions 的HashMap中获取了RootBeanDefinition，里面包含了bean的一些基础信息。



最后根据 bean的scope属性,来做处理，如果作用域是单例，则直接从容器中获取，如果作用域是多例，则创建一个实例，当然，作用域还有其它，自己可以一一去验证



Bean作用域：

scope="singleton"

scope="prototype"

scope="request"

scope="session"

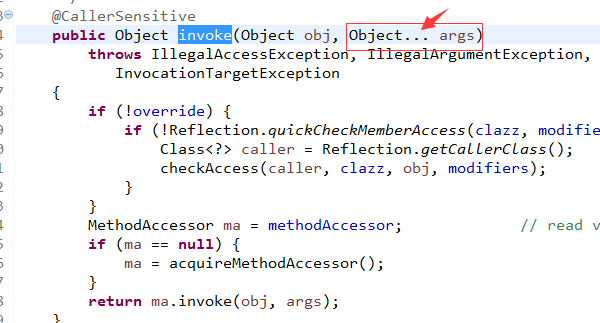
scope="global session"

demo案例：**[Spring IOC容器中Bean的实例化过程](https://blog.csdn.net/yang1464657625/article/details/54669157)**

二、原型request的放不？(G96)

回复：这个应该和IOC没有关系，request对象只是一个参数。

场景分析：SpringMVC在接受一个http请求后，会根据URl去匹配具体的bean（可以这样理解：bean=Map.get(url)，这个map是在容器初始化的时候创建的），然后通过反射获取class实例，最终执行method.invoke方法的时候，会将request对象的值封装到args数组中，当然args可能还包含response对象、注解参数、非注解参数的值。



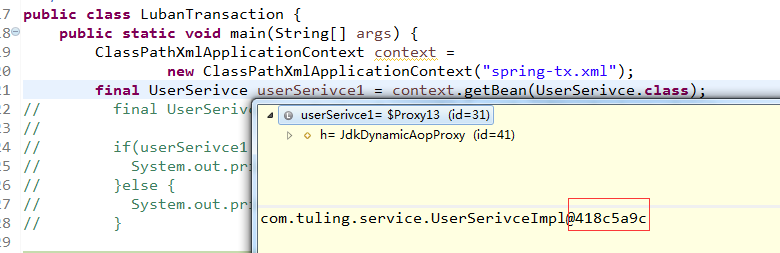
1. Spring里的bean的循环引用是怎么解决的？(G96)

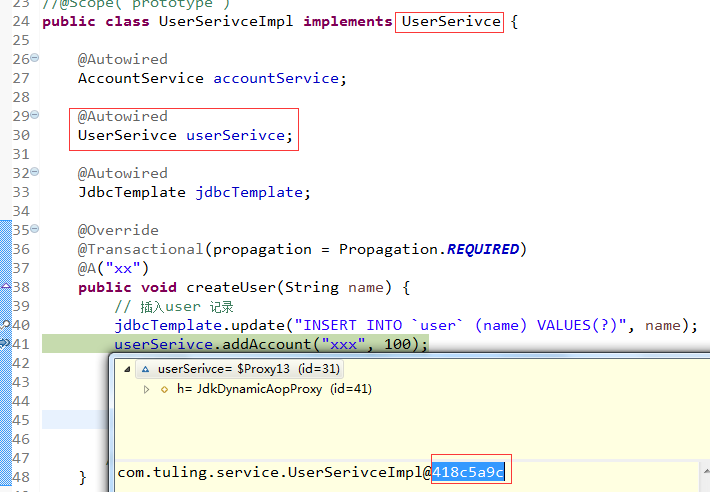
场景一（引用本身，A引用A）

1、结论：在启动spring IoC时，容器自动装载了一个AutowiredAnnotationBeanPostProcessor后置处理器，当容器扫描到@Autowied、@Resource或@Inject时，就会在IoC容器自动查找需要的bean，并装配给该对象的属性。在使用@Autowired时，首先在容器中查询对应类型的bean,如果查询结果刚好为一个，就将该bean装配给@Autowired指定的数据，如果查询的结果不止一个，那么@Autowired会根据名称来查找。如果查询的结果为空，那么会抛出异常。解决方法时，使用required=false

2、场景测试(循环引用)

从容器中获取bean





注：如果将userService.addAccount 改成 this. addAccount

this.addAccount只会被看作一个内部方法，不会被TransactionInterceptor#invoke拦截，不会被事物代理。

场景二（互相引用 A引用B，B引用A）





参考资料

[Spring IOC 容器源码分析 - 循环依赖的解决办法](https://www.coolblog.xyz/2018/06/08/Spring-IOC-%E5%AE%B9%E5%99%A8%E6%BA%90%E7%A0%81%E5%88%86%E6%9E%90-%E5%BE%AA%E7%8E%AF%E4%BE%9D%E8%B5%96%E7%9A%84%E8%A7%A3%E5%86%B3%E5%8A%9E%E6%B3%95/)

1. Spring注解是什么时候加载的？(C51)

回复：后置处理器AutowiredAnnotationBeanPostProcessor，同上

五、为什么会死循环，动态代理debug的时候除了代理对象调用的方法外，如果断点不结束，调用方法前后的代码都会一直执行，而且返回的对象是NULL ？ (A84)

回复：没看明白，动态代理要看具体什么场景，如：事物的动态代理、Aop的动态代理等等。而动态代理又有jdk的动态代理、cglib的动态代理。