下面我们来看看Spring中的单例实现，当我们试图从Spring容器中取得某个类的实例时，默认情况下，Spring会才用单例模式进行创建。

|  |
| --- |
| <bean id="date" class="java.util.Date"/> |
| <bean id="date" class="java.util.Date" scope="singleton"/> (仅为Spring2.0支持) |
| <bean id="date" class="java.util.Date" singleton="true"/> |

以上三种创建对象的方式是完全相同的，容器都会向客户返回Date类的单例引用。那么如果我不想使用默认的单例模式，每次请求我都希望获得一个新的对象怎么办呢？很简单，将scope属性值设置为prototype(原型)就可以了

|  |
| --- |
| <bean id="date" class="java.util.Date" scope="prototype"/> |

通过以上配置信息，Spring就会每次给客户端返回一个新的对象实例。  
那么Spring对单例的底层实现，到底是饿汉式单例还是懒汉式单例呢？

Spring框架对单例的支持是采用单例注册表的方式进行实现的，源码如下：

1. **public** **abstract** **class** AbstractBeanFactory **implements** ConfigurableBeanFactory{
2. /\*\*
3. \* 充当了Bean实例的缓存，实现方式和单例注册表相同
4. \*/
5. **private** **final** Map singletonCache=**new** HashMap();
6. **public** Object getBean(String name)**throws** BeansException{
7. **return** getBean(name,**null**,**null**);
8. }
9. ...
10. **public** Object getBean(String name,Class requiredType,Object[] args)**throws** BeansException{
11. //对传入的Bean name稍做处理，防止传入的Bean name名有非法字符(或则做转码)
12. String beanName=transformedBeanName(name);
13. Object bean=**null**;
14. //手工检测单例注册表
15. Object sharedInstance=**null**;
16. //使用了代码锁定同步块，原理和同步方法相似，但是这种写法效率更高
17. **synchronized**(**this**.singletonCache){
18. sharedInstance=**this**.singletonCache.get(beanName);
19. }
20. **if**(sharedInstance!=**null**){
21. ...
22. //返回合适的缓存Bean实例
23. bean=getObjectForSharedInstance(name,sharedInstance);
24. }**else**{
25. ...
26. //取得Bean的定义
27. RootBeanDefinition mergedBeanDefinition=getMergedBeanDefinition(beanName,**false**);
28. ...
29. //根据Bean定义判断，此判断依据通常来自于组件配置文件的单例属性开关
30. //<bean id="date" class="java.util.Date" scope="singleton"/>
31. //如果是单例，做如下处理
32. **if**(mergedBeanDefinition.isSingleton()){
33. **synchronized**(**this**.singletonCache){
34. //再次检测单例注册表
35. sharedInstance=**this**.singletonCache.get(beanName);
36. **if**(sharedInstance==**null**){
37. ...
38. **try** {
39. //真正创建Bean实例
40. sharedInstance=createBean(beanName,mergedBeanDefinition,args);
41. //向单例注册表注册Bean实例
42. addSingleton(beanName,sharedInstance);
43. }**catch** (Exception ex) {
44. ...
45. }**finally**{
46. ...
47. }
48. }
49. }
50. bean=getObjectForSharedInstance(name,sharedInstance);
51. }
52. //如果是非单例，即prototpye，每次都要新创建一个Bean实例
53. //<bean id="date" class="java.util.Date" scope="prototype"/>
54. **else**{
55. bean=createBean(beanName,mergedBeanDefinition,args);
56. }
57. }
58. ...
59. **return** bean;
60. }
61. }

**刚才的源码中，大家真正要记住的是Spring对bean实例的创建是采用单例注册表的方式进行实现的，而这个注册表的缓存是HashMap对象，如果配置文件中的配置信息不要求使用单例，Spring会采用新建实例的方式返回对象实例.**

**后面一段是判定是否为socpe为singleton，是的话执行单例模式的过程，只创建一个新的对象，否的话则可以创建多个对象。**