

How do I get my plugins spring beans...



sthatcher 16 posts since

May 21, 2008

In developing a plugin I came into a situation where I needed to lookup a spring bean (without injection) but wasn't sure exactly how to within the clearspace framework (version 2.5.4). First off, the reason: I'm placing a link on the Actions tab of a community but I'm using a `<when>` tag in my plugin.xml to determine whether the link should be shown or not. The test is somewhat complex and is part of a Manager class. In order to support the `<when>` tag I created a class with a static function that tries to lookup the Manager (which is configured as a Spring Bean) via the `JiveContext.getJiveManager()` function but I get the following error:

```
22:42:24,187 [http-8080-2] ERROR com.jivesoftware - Error loading proxy 'TestManager' for class 'com.test.impl.TestManagerImpl, returning unproxied object
```

```
org.springframework.beans.factory.NoSuchBeanDefinitionException: No bean named 'TestManager' is defined
```

```
at
```

```
com.jivesoftware.community.lifecycle.spring.JiveDefaultListableBeanFactory.getBeanDefinition(JiveDefaultListableBeanFactory.java:183)
```

```
at
```

```
org.springframework.beans.factory.support.AbstractBeanFactory.getMergedLocalBeanDefinition(AbstractBeanFactory.java:149)
```

```
at
```

```
org.springframework.beans.factory.support.AbstractBeanFactory.doGetBean(AbstractBeanFactory.java:189)
```

```
at
```

```
org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:183)
```

```
at
```

```
org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:164)
```

```
at
```

```
org.springframework.context.support.AbstractApplicationContext.getBean(AbstractApplicationContext.java:100)
```

The spring.xml file is:

How do I get my plugins spring beans...

```
<?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.springframework.org/schema/beans/spring-beans-2.5.xsd"
  default-destroy-method="destroy">

  <bean id="testManager" class="com.test.impl.TestManagerImpl" parent="jiveManager">
    <property name="communityManager" ref="communityManagerImpl"></property>
  </bean>

</beans>
```

Attached is an example of plugin and that produces the above error and is the same error given for the plugin that I'm developing. Otherwise everything else is working with no other errors. I have at least determined that spring is indeed creating the class. Any suggestions??

Regards,

Sean

Attachments:

- [testplugin.zip](#) (7.8 K)

Tags: plugins, spring



[todd.west](#) 15,926 posts since

Jul 14, 2008 1. **Re: How do I get my plugins spring beans with JiveContext?** Dec 2, 2008 6:19 AM

How do I get my plugins spring beans...

Hi Sean,

I haven't been able to test this but what if you do something like:

```
TestManager testManager = (TestManager) JiveApplication.getContext().getSpringBean("testManager");
```

-Todd



[sthatcher](#) 16 posts since

May 21, 2008 2. **Re: How do I get my plugins spring beans with JiveContext?** Dec 2, 2008 8:10 AM

Todd,

Thanks for the reply; turns out the answer is within your reply but not the `getSpringBeans` function. I was using the `getEffectiveContext` and not `getContext`. Making the substitution, then the code works. Although I'm not sure what the fundamental difference is. The javadocs mention the spring context for the `getContext` function (should have been a clue for me); but the `getEffectiveContext` returns a context with the appropriate permissions for the "effective" user. So, yes my problem is fixed but could you give an example of when I would use `getEffectiveContext`?

Thanks,

Sean



[todd.west](#) 15,926 posts since

Jul 14, 2008 3. **Re: How do I get my plugins spring beans with JiveContext?** Dec 2, 2008 8:25 AM

Hi Sean,

How do I get my plugins spring beans...

The difference between the context and the effectiveContext depends on which kind of managers you are looking to retrieve. Using just straight context will get unproxied managers which means that effectively no permissions are enforced on these objects. Using the effectiveContext will return proxied managers which will ensure that the user executing this code has the appropriate permissions to do so.

If the user is a system admin they will get unproxied managers in either case. So basically, you would want to use the effectiveContext if you are looking to restrict access to managers based on permissions. Hopefully that makes sense. Thanks.

-Todd