JSF+Hibernate+Spring

1 **、**<context:annotation-config />在spring中的配置和<context:component-scan base-package=""/>配置的区别：

解答：<context:annotation-config />仅能够在已经在已经注册过的bean上面起作用。对于没有在spring容器中注册的bean，它并不能执行任何操作,也就是说如果你并没有spring容器中注册过bean（spring配置文件中配置bean就是注册）。

那么上述的那些注解并不会在你未注册过的bean中起作用。到这里你可能会说，不是还有@component,@service,@Repository,@Controller注解吗，我用了这些注解，并不需要在配置文件中配置bean了，用起来相当方便了，对，那是因为使用了这个注解

<context:component-scan base-package="com.somnus"/>

我们应当感谢它，该配置项自动将带有@component,@service,@Repository等注解的对象注册到spring容器中

另外该配置项其实也包含了自动注入上述processor的功能，因此当使用<context:component-scan/>后，即可将<context:annotation-config/>省去。

那如果我另外在配置文件注册了bean，但是bean中又使用了诸如@Autowired注解，它会起作用吗？答案是肯定的，<context:component-scan> 配置所产生的的处理那些注解的处理器工具，会处理所有绑定到容器上面的bean，不管是通过xml手动注册的还是通过scanning扫描注册的。

那么，如果我们既配置了<context:annotation-config/>，又配置了<context:component-scan/>，它们都具有处理在容器中注册的bean里面的注解的功能。会不会出现重复注入的情况呢？答案是否定的，两者同时存在的时候，前者会被忽略。也就是那些@Autowired，@Autowired等注入注解只会被注入一次

2、如何在jsf中，是的值改变的时候，立即触发：

|  |
| --- |
| <h:selectOneMenu value=*"#{registerBean.sex}"* valueChangeListener=*"#{registerBean.change}"* immediate=*"true"*onchange=*"submit()"*>  <f:selectItem itemLabel=*"男"* itemValue=*"man"*/>  <f:selectItem itemLabel=*"女"* itemValue=*"woman"*/>  </h:selectOneMenu> |

3、