

jBPM4与Spring整合的2种方式Java认证考试 PDF转换可能丢失
图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/642/2021_2022_jBPM4_E4_B8_8ESp_c104_642255.htm 要知道如何将jBPM4与Spring整合，

可以先了解jBPM4的IOC容器，如果不了解的可以先

看ronghao的这篇文章<http://www.javaeye.com/topic/381607>，是介绍jBPM4的IOC容器的。下面我们介绍jBPM与Spring整合

的2种方式：第一种：手工将SessionFactory放入jBPM4中。

第1步：更改jbpm.spring.default.cfg.xml配置文件，将下面的部分注释掉 `lt.hibernate-configurationlt.cfg`

`resource="jbpm.hibernate.cfg.xml"`

`/lt./hibernate-configurationlt.hibernate-session-factory /gt.` 注释的部分是在jBPM4中创建了sessionFactory，而我们只需要一个sessionFactory.既然要将jBPM4与Spring的整合，那就希望

由Spring来统一管理sessionFactory和事务，在Spring的配置文件中构造一个sessionFactory. ProcessEngine是jBPM4的Facade模式，一切由它与外部交互，第2步：在Spring配置文件中写一个bean：

`gt. gt. gt.` 第3步：在ProcessFactoryBean中注入SessionFactory：

`public void setSessionFactory(SessionFactory sessionFactory) { this.sessionFactory = sessionFactory. }` 第4步：

在ProcessFactoryBean中创建一个SpringConfiguration，然后将sessionFactory放入 SpringConfiguration中，再

从SpringConfiguration得到processEngine，代码如下：

`public void afterPropertiesSet() { SpringConfiguration cfg = new SpringConfiguration(jbpmConfigurationLocation).`

`cfg.setApplicationContext(applicationContext).`

`cfg.setApplicationContext(applicationContext).`

cfg.setSessionFactory(sessionFactory). this.processEngine =
cfg.buildProcessEngine(). } 然后我们的工作就可以开展了，可
以从processEngine得到所有的service.比如： ProcessEngine
processEngine = (ProcessEngine) ctx.getBean("processEngine").
RepositoryService repositoryService =
processEngine.getRepositoryService()。 第2种：获得Hibernate
的SessionFactory. 第1步：与第一种方式的第1步一样。 第2步
：更改jbpm.tx.spring.cfg.xml配置文件：将 gt.改成 gt. 然后将 gt.
gt. 改成 gt. 这部分修改是将jBPM4创建的spring事务和hibernate
的session改成从当前的ThreadLocal中获得session和事务。 第3
步：在Spring配置文件中写bean， processEngine和template： gt.
gt. gt. gt. gt. gt. gt. processEngine直接使用factory-bean指
向jbpmConfiguration，也就是
org.jbpm.pvm.internal.cfg.SpringConfiguration，并
从SpringConfiguration的 buildProcessEngine中获得。
jbpmTemplate主要是控制事务，在processEngine外面创建事务
，这样使用的时候会先调用jbpmTemplate，再调
用processEngine，否则它会说事务没有启动。在jbpmTemplate
中注入processEngine： public void
setProcessEngine(ProcessEngine processEngine) {
this.processEngine = processEngine. } 这里我的理解是我们在上
面已经将事务改成从当前的Threadlocal中获得，所以jBPM4是
必须当前有事务，如果我没有创建事务的话，在使用
processEngine时就会说事务没有启动。所以我们就封装了一个
jbpmTemplate，如果我的理解有误还请大家指出。 第1种整
合方式的例子下载<http://www.family168.com/>，第2种整合方式

之后可以看我们的jBPM-Side里的控制台。java认证网，加入收藏 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