

sky

永久域名 <http://skyuck.javaeye.com>



skyuck

你让我觉得自己简直是个大坏蛋!

2010-04-26 通过网页

[>>更多闲聊](#)

浏览: 24022 次

性别:

来自: 郑州

我现在离线

[详细资料](#)

[留言簿](#)

搜索本博客

最近访客 [>>更多访客](#)



[akos](#)



[xinwei-16](#)



[zhangyaou](#)



[mrrightzhu](#)

博客分类

- [全部博客 \(103\)](#)
- [jboss \(3\)](#)
- [ruby on rails \(4\)](#)
- [ant \(1\)](#)
- [C# \(5\)](#)
- [心情 \(7\)](#)

2009-08-26

[简单Spring 总结](#) | [将JBPM4整合到web 工程---流程实例管理](#)

## JBPM4---简易的请假实例

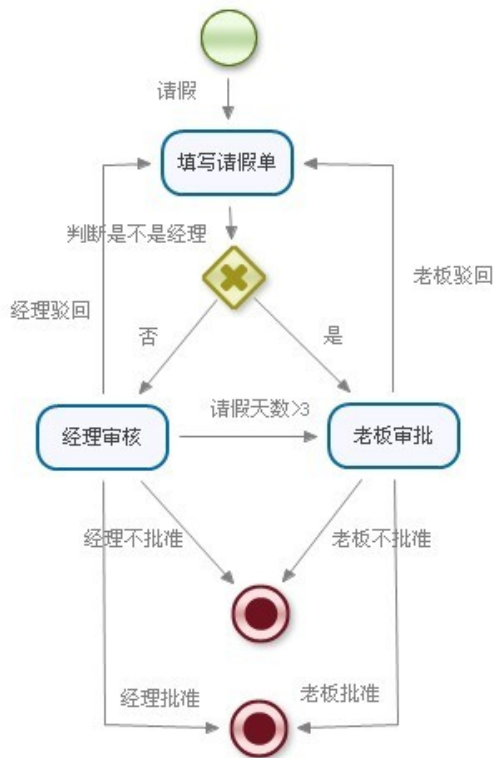
关键字: **jbpm4** 请假

1. 新建一个动态的web项目
2. 在lib目录下加入以下jar包（由于用到fileupload的上传，所以还要加入fileupload的依赖库）

引用

```
*JBPM4_HOME/jbpm.jar
*JBPM4_HOME/lib/*.jar
*jakarta-taglibs-standard-1.1.2_HOME/lib/jstl.jar
*jakarta-taglibs-standard-1.1.2_HOME/lib/standard.jar
*commons-fileupload-1.2.1_HOME/lib/commons-fileupload-1.2.1.jar
```

3. 绘制流程图（在resources目录下建立一个leave.jpdl.xml文件）



Java代码

```
1. <?xml version="1.0" encoding="UTF-8"?>
2.
3. <process name="leave" xmlns="http://jbpm.org/4.0/jpdl">
4.   <start g="201,14,48,48" name="开始">
```

[js \(4\)](#)

- [java \(34\)](#)
- [Hibernate \(14\)](#)
- [JBPM4 \(4\)](#)
- [spring \(4\)](#)
- [Struts 2.x \(5\)](#)
- [webservice \(3\)](#)

我的留言簿 >>更多留言

- 漂过
- by [jc\\_dreaming](#)
- 刚才误操作 并提交了 汗！ 语句错位了 本来是加了句话 粘错位置！ 谢谢您的回复 ...
- by [zilong8000](#)
- 谢谢您的回复 通过你的回复我 感觉自己在今后的学习又明朗了， 其实最近我就在学习线程， ...
- by [zilong8000](#)

其他分类

- [我的收藏 \(13\)](#)
- [我的论坛主题贴 \(5\)](#)
- [我的所有论坛贴 \(99\)](#)
- [我的精华良好贴 \(0\)](#)

最近加入圈子

- [JBPM @net](#)
- [EXT](#)
- [lucene爱好者](#)

存档

- [2010-06 \(2\)](#)
- [2010-05 \(3\)](#)
- [2010-04 \(3\)](#)
- [更多存档...](#)

最新评论

- [JBPM4---简易的请假实例](#)  
严重: Servlet.service() for servlet jsp thr ...  
-- by [YWF\\_110](#)
- [struts2处理.do后缀的请...](#)  
我也想用Struts2的拦截器解决，但是没想出来怎么做，希望你 你能给我详细的解决方案 ...

```
5.         <transition g="-42,-10" name="请假" to="填写请假单"/>
6.     </start>
7.     <task assignee="Kayzhan" g="178,87,92,52" name="填写请假单">
8.         <transition g="-97,2" name="判断是不是经理" to="是不是经理"/>
9.     </task>
10.    <decision expr="#{manager}" g="204,158,48,48" name="是不是经理">
11.        <transition g="-23,-11" name="否" to="经理审核"/>
12.        <transition g="14,-11" name="是" to="老板审批"/>
13.    </decision>
14.    <task assignee="Lingo" g="103,252,92,52" name="经理审核">
15.        <transition g="150,450:10,-21" name="经理批准" to="结束"/>
16.        <transition g="-22,-22" name="请假天数>3" to="老板审批"/>
17.        <transition g="-61,-1" name="经理不批准" to="终止"/>
18.        <transition g="149,114:-55,82" name="经理驳回" to="填写请假单"/>
19.    </task>
20.    <task assignee="ForgetDavi" g="278,251,92,52" name="老板审批">
21.        <transition g="326,450:-58,-24" name="老板批准" to="结束"/>
22.        <transition g="7,0" name="老板不批准" to="终止"/>
23.        <transition g="323,114:13,61" name="老板驳回" to="填写请假单"/>
24.    </task>
25.    <end g="219,429,48,48" name="结束" state="confirm"/>
26.    <end g="220,360,48,48" name="终止" state="dissent"/>
27. </process>
```

#### 4.编写流程发布及控制的类

Java代码

```
1. package zzg;
2.
3. import java.io.IOException;
4. import java.lang.reflect.Method;
5. import java.util.ArrayList;
6. import java.util.HashMap;
7. import java.util.Iterator;
8. import java.util.LinkedHashMap;
9. import java.util.List;
10. import java.util.Map;
11. import java.util.Set;
12. import java.util.zip.ZipInputStream;
13.
14. import javax.servlet.ServletConfig;
15. import javax.servlet.ServletException;
16. import javax.servlet.http.HttpServlet;
17. import javax.servlet.http.HttpServletRequest;
18. import javax.servlet.http.HttpServletResponse;
19.
20. import org.apache.commons.fileupload.DiskFileUpload;
21. import org.apache.commons.fileupload.FileItem;
22. import org.jbpm.api.Configuration;
23. import org.jbpm.api.Execution;
24. import org.jbpm.api.ExecutionService;
25. import org.jbpm.api.ProcessDefinition;
26. import org.jbpm.api.ProcessDefinitionQuery;
27. import org.jbpm.api.ProcessEngine;
28. import org.jbpm.api.ProcessInstance;
29. import org.jbpm.api.RepositoryService;
30. import org.jbpm.api.TaskService;
31. import org.jbpm.api.task.Task;
32.
33. /**
34.  * Servlet implementation class LeaveServlet
35.  */
```

-- by [hunnuxiaobo](#)

■ [心底柔然的角落](#)

对了，你们公司的名字叫什么？

留言给我说一下。

-- by [xiaojing3517](#)

■ [对je有些失望了](#)

靠！你现在才有点失望，我早就

有了。可惜我的二百多篇文章。

-- by [TaoistWar](#)

■ [struts2处理.do后缀的请求...](#)

我有一个问题想请教你。我希望

一个这样

的url: <http://myserver/g...>

-- by [hunnuxiaobo](#)

评论排行榜

■ [JBPM4---简易的请假实例](#)

■ [将JBPM4整合到web工程](#)

■ [struts2处理.do后缀的请求](#)

■ [spring2.5 not found](#)

[org.springframework....](#)

■ [对je有些失望了](#)



RSS



Add to Google



订阅到 鲜果

[\[什么是RSS?\]](#)

```
36. public class LeaveServlet extends HttpServlet {
37.     private static final long serialVersionUID = 1L;
38.     private ProcessEngine processEngine = null;
39.     private RepositoryService repositoryService = null;
40.     private ExecutionService executionService = null;
41.     private ProcessInstance processInstance = null;
42.     private TaskService taskService = null;
43.     private Task task = null;
44.     private Execution execution = null;
45.     private String url = null;
46.     private List list = null;
47.     private List<Task> taskList = null;
48.
49.     /* 初始化ProcessEngine. */
50.     public void init(ServletConfig config) throws ServletException {
51.         super.init(config);
52.         // 获取流程引擎
53.         processEngine = new Configuration().buildProcessEngine();
54.         repositoryService = processEngine.getRepositoryService();
55.         executionService = processEngine.getExecutionService();
56.         taskService = processEngine.getTaskService();
57.     }
58.
59.     protected void doGet(HttpServletRequest request,
60.         HttpServletResponse response) throws ServletException, IOException {
61.         process(request, response);
62.     }
63.
64.     protected void doPost(HttpServletRequest request,
65.         HttpServletResponse response) throws ServletException, IOException {
66.         process(request, response);
67.     }
68.
69.     protected void process(HttpServletRequest request,
70.         HttpServletResponse response) throws ServletException, IOException {
71.         String zzg = request.getParameter("zzg");
72.
73.         if (zzg.equals("deploy")) {
74.             try {
75.                 deploy(request, response);
76.             } catch (Exception e) {
77.                 e.printStackTrace();
78.             }
79.             list = getLatestProcessDefinition(request, response);
80.             request.setAttribute("process", list);
81.             request.getRequestDispatcher("/index.jsp").forward(request,
82.                 response);
83.         } else if (zzg.equals("leave")) {
84.             url = leave(request, response);
85.             request.getRequestDispatcher(url).forward(request, response);
86.         } else if (zzg.equals("query")) {
87.             list = getProcessInstanceById(request, response);
88.             request.setAttribute("pi", list);
89.             request.getRequestDispatcher("/index.jsp").forward(request,
90.                 response);
91.         } else if (zzg.equals("apply")) {
92.             apply(request, response);
93.             list = getProcessInstanceById(request, response);
94.             request.setAttribute("pi", list);
95.             request.getRequestDispatcher("/index.jsp").forward(request,
96.                 response);
97.         } else if (zzg.equals("getTask")) {
98.             list = getTask(request, response);
99.             request.setAttribute("taskList", list);
100.            request.getRequestDispatcher("/check.jsp").forward(request,
```

```

101.         response);
102.     } else if (zzg.equals("view")) {
103.         Map<String, Object> map = view(request, response);
104.         request.setAttribute("map", map);
105.         request.getRequestDispatcher("/view.jsp")
106.             .forward(request, response);
107.     } else if (zzg.equals("confirm")) {
108.         confirm(request, response);
109.         list = getTask(request, response);
110.         request.setAttribute("taskList", list);
111.         request.getRequestDispatcher("/check.jsp").forward(request,
112.             response);
113.     } else if (zzg.equals("dissent")) {
114.         dissent(request, response);
115.         list = getTask(request, response);
116.         request.setAttribute("taskList", list);
117.         request.getRequestDispatcher("/check.jsp").forward(request,
118.             response);
119.     }
120.
121.     /*
122.     * try { Class clz = this.getClass(); Method method =
123.     * clz.getDeclaredMethod(action, HttpServletRequest.class,
124.     * HttpServletResponse.class); method.invoke(this, request, response); }
125.     * catch (Exception ex) { ex.printStackTrace(); }
126.     */
127. }
128.
129.     protected void deploy(HttpServletRequest request,
130.         HttpServletResponse response) throws Exception, IOException {
131.         String temp = getServletContext().getRealPath("/temp");
132.         String upload = getServletContext().getRealPath("/upload");
133.         DiskFileUpload diskFileUpload = new DiskFileUpload();
134.         diskFileUpload.setFileSizeMax(1 * 1024 * 1024);
135.         diskFileUpload.setSizeThreshold(4096);
136.         diskFileUpload.setRepositoryPath(temp);
137.         List fileItems = diskFileUpload.parseRequest(request);
138.         Iterator iter = fileItems.iterator();
139.         if (iter.hasNext()) {
140.             FileItem item = (FileItem) iter.next();
141.             if (!item.isFormField()) {
142.                 String name = item.getName();
143.                 long size = item.getSize();
144.                 System.out.println("name: " + name + "&size:" + size);
145.                 if (name != null && !name.equals("") && size > 0) {
146.                     // Deploying a process部署流程
147.                     repositoryService.createDeployment()
148.                         .addResourcesFromZipInputStream(
149.                             tem.getInputStream()))
150.                         .deploy();
151.                 }
152.             }
153.         }
154.     }
155.
156.     protected String leave(HttpServletRequest request,
157.         HttpServletResponse response) throws ServletException, IOException {
158.         // 根据流程定义的KEY值启动一个新的流程实例
159.         processInstance = executionService.startProcessInstanceByKey("leave");
160.         // 判断流程定义的活动是否是“填写请假单”的活动
161.         if (processInstance.isActive("填写请假单")) {
162.             url = "/apply.jsp";
163.         }

```

```

164.         return url;
165.     }
166.
167.     protected Map<String, Object> apply(HttpServletRequest request,
168.         HttpServletResponse response) throws ServletException, IOException {
169.
170.         Map<String, Object> variables = new HashMap<String, Object>();
171.         String name = request.getParameter("applyName");
172.         String time = request.getParameter("applyTime");
173.         String leaveDay = request.getParameter("leaveDay");
174.         String content = request.getParameter("content");
175.         String position = request.getParameter("position");
176.
177.         variables.put("name", name);
178.         variables.put("time", time);
179.         variables.put("leaveDay", leaveDay);
180.         variables.put("content", content);
181.         variables.put("position", position);
182.         if (position.trim().equals("经理")) {
183.             // "manager" 变量名与 leave.jpdl.xml 中 decision 里面的变量一致，同时设置的值也要
保持一致
184.             variables.put("manager", "是");
185.         } else {
186.             variables.put("manager", "否");
187.         }
188.         // 获取任务列表，因为我们在填写请假单的时候是分配给“zzg”的，见 leave.jpdl.xml
189.         taskList = taskService.findPersonalTasks("zzg");
190.         // 根据任务列表获取任务
191.         task = taskList.get(0);
192.         // 把设置好的变量放到任务服务里面并根据任务 ID 结束任务
193.         taskService.setVariables(task.getId(), variables);
194.         taskService.completeTask(task.getId());
195.         // 判断任务活动是否到达老板审批活动
196.         // execution = executionService.findExecutionById(task.getExecutionId());
197.         // System.out.println("isActive: "
198.         //     + execution.getProcessInstance().isActive("老板审批"));
199.         return variables;
200.     }
201.
202.     /**
203.      * 获取任务列表
204.      *
205.      * @param request
206.      * @param response
207.      * @return
208.      * @throws ServletException
209.      * @throws IOException
210.      */
211.     protected List<Task> getTask(HttpServletRequest request,
212.         HttpServletResponse response) throws ServletException, IOException {
213.         String name = request.getParameter("name");
214.         taskList = taskService.findPersonalTasks(name);
215.         return taskList;
216.     }
217.
218.     /**
219.      * 查看任务
220.      *
221.      * @param request
222.      * @param response
223.      * @return
224.      * @throws ServletException
225.      * @throws IOException
226.      */

```

```

227.         protected Map<String, Object> view(HttpServletRequest request,
228.             HttpServletResponse response) throws ServletException, IOException {
229.             String taskId = request.getParameter("id");
230.             Set<String> set = taskService.getVariableNames(taskId);
231.             Map<String, Object> map = taskService.getVariables(taskId, set);
232.
233.             return map;
234.         }
235.
236.         protected void confirm(HttpServletRequest request,
237.             HttpServletResponse response) throws ServletException, IOException {
238.             String taskId = request.getParameter("id");
239.             task = taskService.getTask(taskId);
240.             execution = executionService.findExecutionById(task.getExecutionId());
241.             if (execution.getProcessInstance().isActive("老板审批")) {
242.                 taskService.completeTask(taskId, "老板批准");
243.             } else if (execution.getProcessInstance().isActive("经理审核")) {
244.                 String variable = (String) taskService.getVariable(taskId,
245.                     "leaveDay");
246.                 if (Integer.valueOf(variable) > 3) {
247.                     taskService.completeTask(taskId, "请假天数>3");
248.                 } else {
249.                     taskService.completeTask(taskId, "经理批准");
250.                 }
251.             }
252.         }
253.
254.         protected void dissent(HttpServletRequest request,
255.             HttpServletResponse response) throws ServletException, IOException {
256.             String taskId = request.getParameter("id");
257.             task = taskService.getTask(taskId);
258.             execution = executionService.findExecutionById(task.getExecutionId());
259.             if (execution.getProcessInstance().isActive("老板审批")) {
260.                 taskService.completeTask(taskId, "老板不批准");
261.             } else if (execution.getProcessInstance().isActive("经理审核")) {
262.                 taskService.completeTask(taskId, "经理不批准");
263.             }
264.         }
265.
266.         /**
267.          * 获取最新版本的流程定义
268.          * @param request
269.          * @param response
270.          * @return
271.          * @throws ServletException
272.          * @throws IOException
273.          */
274.         protected List<ProcessDefinition> getLatestProcessDefinition(
275.             HttpServletRequest request, HttpServletResponse response)
276.             throws ServletException, IOException {
277.             List<ProcessDefinition> processDefinitions = repositoryService
278.                 .createProcessDefinitionQuery().orderAsc(
279.                     ProcessDefinitionQuery.PROPERTY_NAME).list();
280.
281.             Map<String, ProcessDefinition> map = new LinkedHashMap<String, ProcessDefinition>();
282.
283.             for (ProcessDefinition pd : processDefinitions) {
284.                 String key = pd.getKey();
285.                 ProcessDefinition definition = map.get(key);
286.                 if ((definition == null)
287.                     || (definition.getVersion() < pd.getVersion())) {
288.                     map.put(key, pd);
289.                 }
290.             }
291.         }

```

```

289.         return new ArrayList(map.values());
290.     }
291.
292.     /**
293.      * 根据流程定义ID获取流程实例
294.      * @param request
295.      * @param response
296.      * @return
297.      * @throws ServletException
298.      * @throws IOException
299.      */
300.     protected List<ProcessInstance> getProcessInstanceById(
301.         HttpServletRequest request, HttpServletResponse response)
302.         throws ServletException, IOException {
303.         String pdId = request.getParameter("id");
304.         return executionService.createProcessInstanceQuery()
305.             .processDefinitionId(pdId).list();
306.     }
307.
308. }

```

5. 在webapp目录下创建一个index.jsp页面，用来发布流程、显示流程定义及流程实例

Java代码

```

1.  <%@page contentType="text/html; charset=UTF-8" %>
2.  <%@page import="org.jbpm.api.*" %>
3.  <%@page import="java.util.*" %>
4.  <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
5.
6.  <html>
7.      <head>
8.          <title>jBPM4流程发布</title>
9.      </head>
10.     <body>
11.         <form action="leave?zzg=deploy" method="POST" enctype="multipart/form-data">
12.             <input type="file" name="upload">
13.             <input type="submit" value="发布">
14.         </form>
15.         <ul>
16.             <li><a href="leave?zzg=leave">请假</a></li>
17.             <li><a href="leave?zzg=getTask&name=Lingo" target="_blank">经理的任务列
18. 表</a></li>
19.             <li><a href="leave?zzg=getTask&name=ForgetDavi" target="_blank">老板的任
20. 务列表</a></li>
21.         </ul>
22.         <table width=480 border=1>
23.             <tr><td colspan=5>流程定义</td></tr>
24.             <tr>
25.                 <td>ID</td>
26.                 <td>Key</td>
27.                 <td>名称</td>
28.                 <td>版本</td>
29.                 <td>操作</td>
30.             </tr>
31.             <c:forEach var="process" items="${process}">
32.                 <tr>
33.                     <td>${process.id}</td>
34.                     <td>${process.key}</td>
35.                     <td>${process.name}</td>
36.                     <td>${process.version}</td>
37.                     <td><a href="leave?zzg=query&id=${process.id }">查看</a></td>

```

```

38.         </table>
39.         <br>
40.         <br>
41.         <br>
42.         <table width=480 border=1>
43.             <tr><td colspan="5">流程实例</td></tr>
44.             <tr>
45.                 <td>ID</td>
46.                 <td>Key</td>
47.                 <td>状态</td>
48.             </tr>
49.             <c:forEach var="pi" items="${pi}">
50.                 <tr>
51.                     <td>${pi.id}</td>
52.                     <td>${pi.key}</td>
53.                     <td>${pi.state}</td>
54.                 </tr>
55.             </c:forEach>
56.         </table>
57.     </body>
58. </html>

```

6. 在webapp目录创建一个apply.jsp页面，用来填写请假单

Java代码

```

1. <%@page contentType="text/html; charset=UTF-8" %>
2.
3. <html>
4.     <head>
5.         <title>填写申请单</title>
6.     </head>
7.     <body>
8.         <form action="leave?zzg=apply" method="POST">
9.             <table width="300" border=1>
10.                 <tr>
11.                     <td><label>申请人</label></td>
12.                     <td><input type="text" value="zzg" name="applyName"></td>
13.                 </tr>
14.                 <tr>
15.                     <td><label>职位</label></td>
16.                     <td><input type="text" value="经理" name="position"></td>
17.                 </tr>
18.                 <tr>
19.                     <td><label>申请时间</label></td>
20.                     <td><input type="text" value="2009-03-09" name="applyTime"></td>
21.                 </tr>
22.                 <tr>
23.                     <td><label>请假天数</label></td>
24.                     <td><input type="text" value="3" name="leaveDay"></td>
25.                 </tr>
26.                 <tr>
27.                     <td><label>请假原因</label></td>
28.                     <td><textarea name="content" rows=3 cols=200>病假</textarea></td>
29.                 </tr>
30.                 <tr>
31.                     <td><input type="submit" value="申请"></td>
32.                     <td><input type="reset" value="取消"></td>
33.                 </tr>
34.             </table>
35.         </form>
36.     </body>
37. </html>

```



7.在webapp目录下创建一个check.jsp文件，用来显示任务列表并添加三个操作

Java代码

```
1. <%@page contentType="text/html;charset=UTF-8" %>
2. <%@page import="org.jbpm.api.*" %>
3. <%@page import="java.util.*" %>
4. <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
5.
6. <html>
7.     <head>
8.         <title>请假审批</title>
9.     </head>
10.    <body>
11.        <table width=600 border=1>
12.            <tr><td colspan="8">任务列表</td></tr>
13.            <tr>
14.                <td>ID</td>
15.                <td>名称</td>
16.                <td>分配人</td>
17.                <td>创建时间</td>
18.                <td>持续时间</td>
19.                <td>优先级</td>
20.                <td>描述</td>
21.                <td>操作</td>
22.            </tr>
23.            <c:forEach var="task" items="${taskList}">
24.                <tr>
25.                    <td>${task.id}</td>
26.                    <td>${task.name}</td>
27.                    <td>${task.assignee}</td>
28.                    <td>${task.createTime}</td>
29.                    <td>${task.duedate}</td>
30.                    <td>${task.priority}</td>
31.                    <td>${task.description}</td>
32.                    <td>
33.                        <a href="leave?zsg=view&id=${task.id }" target="_blank"
34.                            >审核</a>
35.                        <a href="leave?zsg=confirm&id=${task.id }&name=${task.
36.                            assignee}">批准</a>
37.                        <a href="leave?zsg=dissent&id=${task.id }&name=${task.
38.                            assignee}">不批准</a>
39.                    </td>
40.                </tr>
41.            </c:forEach>
42.        </table>
43.    </body>
44. </html>
```

8.在webapp目录下创建一个view.jsp文件，用来显示请假内容

Java代码

```
1. <%@page contentType="text/html;charset=UTF-8" %>
2. <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
3.
4. <html>
5.     <head>
6.         <title>审核</title>
7.     </head>
8.     <body>
9.         <table width="300" border=1>
10.            <tr>
11.                <td><label>申请人</label></td>
```

```
12.         <td>${map.name }</td>
13.     </tr>
14.     <tr>
15.         <td><label>职位</label></td>
16.         <td>${map.position }</td>
17.     </tr>
18.     <tr>
19.         <td><label>申请时间</label></td>
20.         <td>${map.time }</td>
21.     </tr>
22.     <tr>
23.         <td><label>请假天数</label></td>
24.         <td>${map.leaveDay }</td>
25.     </tr>
26.     <tr>
27.         <td><label>请假内容</label></td>
28.         <td>${map.content }</td>
29.     </tr>
30. </table>
31. </body>
32. </html>
```

#### 8. 修改web.xml

##### Java代码

```
1. <?xml version="1.0" encoding="UTF-8"?>
2. <web-app id="WebApp_ID" version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://j
ava.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
3.     <display-name>LeaveJbpm</display-name>
4.     <servlet>
5.         <description>
6.         </description>
7.         <display-name>LeaveServlet</display-name>
8.         <servlet-name>LeaveServlet</servlet-name>
9.         <servlet-class>
10.         zzg.LeaveServlet</servlet-class>
11.     </servlet>
12.     <servlet-mapping>
13.         <servlet-name>LeaveServlet</servlet-name>
14.         <url-pattern>/leave</url-pattern>
15.     </servlet-mapping>
16.     <welcome-file-list>
17.         <welcome-file>index.jsp</welcome-file>
18.     </welcome-file-list>
19. </web-app>
```

9. 把leave.jpdl.xml文件压缩成zip文件发布即可。

[查看图片附件](#)

[简单Spring 总结](#) | [将JBPM4整合到web 工程---流程实例管理](#)

15:46 | [浏览 \(3822\)](#) | [评论 \(7\)](#) | 分类: [JBPM4](#) | [相关推荐](#)

#### 评论

7 楼 [YWF\\_110](#) 2010-04-30 [引用](#)

严重: Servlet.service() for servlet jsp threw exception

org.jbpm.api.JbpmException:

error: decision 'exclusive1' must have one of: expr attribute, handler attribute, handler element or condition expressions

[line=16 column=50 ]

```

at org.jbpm.pvm.internal.xml.ProblemList.getJbpmException(ProblemList.java:175)
at org.jbpm.pvm.internal.xml.ProblemList.getJbpmException(ProblemList.java:141)
at org.jbpm.pvm.internal.repository.DeployerManager.deploy(DeployerManager.java:50)
at org.jbpm.pvm.internal.repository.RepositorySessionImpl.deploy(RepositorySessionImpl.java:61)
at org.jbpm.pvm.internal.cmd.DeployCmd.execute(DeployCmd.java:47)
at org.jbpm.pvm.internal.cmd.DeployCmd.execute(DeployCmd.java:33)
at org.jbpm.pvm.internal.svc.DefaultCommandService.execute(DefaultCommandService.java:42)
at org.jbpm.pvm.internal.tx.StandardTransactionInterceptor.execute(StandardTransactionInterceptor.java:54)
at org.jbpm.pvm.internal.svc.EnvironmentInterceptor.executeInNewEnvironment(EnvironmentInterceptor.java:53)
at org.jbpm.pvm.internal.svc.EnvironmentInterceptor.execute(EnvironmentInterceptor.java:40)
at org.jbpm.pvm.internal.svc.RetryInterceptor.execute(RetryInterceptor.java:55)
at org.jbpm.pvm.internal.svc.SkipInterceptor.execute(SkipInterceptor.java:43)
at org.jbpm.pvm.internal.repository.DeploymentImpl.deploy(DeploymentImpl.java:91)
at org.apache.jsp.index_jsp._jspService(index_jsp.java:71)
at org.apache.jasper.runtime.HttpJspBase.service(HttpJspBase.java:70)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:717)
at org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:374)
at org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:342)
at org.apache.jasper.servlet.JspServlet.service(JspServlet.java:267)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:717)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:290)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)
at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:233)
at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:191)
at org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:128)
at org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:102)
at org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:109)
at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:293)
at org.apache.coyote.http11.Http11Processor.process(Http11Processor.java:849)
at org.apache.coyote.http11.Http11Protocol$Http11ConnectionHandler.process(Http11Protocol.java:583)
at org.apache.tomcat.util.net.JIoEndpoint$Worker.run(JIoEndpoint.java:454)
at java.lang.Thread.run(Thread.java:619)

```

这是什么错

6 楼 [xif](#) 2010-03-09 [引用](#)

蛮好

5 楼 [Wallian\\_hua](#) 2009-12-17 [引用](#)

alymail 写道

```

org.jbpm.api.JbpmException: no org.jbpm.pvm.internal.id.DbidGenerator in current environment
at org.jbpm.pvm.internal.env.EnvironmentImpl.getFromCurrent(EnvironmentImpl.java:204)
at org.jbpm.pvm.internal.env.EnvironmentImpl.getFromCurrent(EnvironmentImpl.java:190)
at org.jbpm.pvm.internal.hibernate.DbSessionImpl.newTask(DbSessionImpl.java:368)
at org.jbpm.pvm.internal.hibernate.DbSessionImpl.createTask(DbSessionImpl.java:361)
at org.jbpm.jpdl.internal.activity.TaskActivity.execute(TaskActivity.java:62)
at org.jbpm.jpdl.internal.activity.TaskActivity.execute(TaskActivity.java:57)
at org.jbpm.pvm.internal.model.op.ExecuteActivity.perform(ExecuteActivity.java:60)
at org.jbpm.pvm.internal.model.ExecutionImpl.performAtomicOperationSync(ExecutionImpl.java:655)
at org.jbpm.pvm.internal.model.ExecutionImpl.performAtomicOperation(ExecutionImpl.java:615)
at org.jbpm.pvm.internal.model.ExecutionImpl.start(ExecutionImpl.java:215)
at org.jbpm.pvm.internal.cmd.StartProcessInstanceInLatestCmd.execute(StartProcessInstanceInLatestCmd.java:65)
at org.jbpm.pvm.internal.cmd.StartProcessInstanceInLatestCmd.execute(StartProcessInstanceInLatestCmd.java:38)
at org.jbpm.pvm.internal.svc.DefaultCommandService.execute(DefaultCommandService.java:42)
at org.jbpm.pvm.internal.spring.CommandTransactionCallback.doInTransaction(CommandTransactionCallback.java:50)
at org.springframework.transaction.support.TransactionTemplate.execute(TransactionTemplate.java:128)
at org.jbpm.pvm.internal.tx.SpringTransactionInterceptor.execute(SpringTransactionInterceptor.java:77)
at org.jbpm.pvm.internal.svc.EnvironmentInterceptor.executeInNewEnvironment(EnvironmentInterceptor.java:53)
at org.jbpm.pvm.internal.svc.EnvironmentInterceptor.execute(EnvironmentInterceptor.java:40)
at org.jbpm.pvm.internal.svc.RetryInterceptor.execute(RetryInterceptor.java:55)
at org.jbpm.pvm.internal.svc.SkipInterceptor.execute(SkipInterceptor.java:43)
at org.jbpm.pvm.internal.svc.ExecutionServiceImpl.startProcessInstanceByKey(ExecutionServiceImpl.java:66)
at com.test.impl.MyServiceImpl.startProcessInstanceByKey(MyServiceImpl.java:55)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

```

```

at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
at java.lang.reflect.Method.invoke(Method.java:597)
at org.springframework.aop.support.AopUtils.invokeJoinpointUsingReflection(AopUtils.java:307)
at org.springframework.aop.framework.ReflectiveMethodInvocation.invokeJoinpoint(ReflectiveMethodInvocation.java:182)
at org.springframework.aop.framework.ReflectiveMethodInvocation.proceed(ReflectiveMethodInvocation.java:149)
at org.springframework.transaction.interceptor.TransactionInterceptor.invoke(TransactionInterceptor.java:106)
at org.springframework.aop.framework.ReflectiveMethodInvocation.proceed(ReflectiveMethodInvocation.java:171)
at org.springframework.aop.framework.JdkDynamicAopProxy.invoke(JdkDynamicAopProxy.java:204)
at $Proxy2.startProcessInstanceByKey(Unknown Source)
at zzg.LeaveServlet.leave(LeaveServlet.java:155)
at zzg.LeaveServlet.process(LeaveServlet.java:84)
at zzg.LeaveServlet.doGet(LeaveServlet.java:57)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:617)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:717)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:290)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)
at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:233)
at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:191)
at org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:128)
at org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:102)
at org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:109)
at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:286)
at org.apache.coyote.http11.Http11Processor.process(Http11Processor.java:845)
at org.apache.coyote.http11.Http11Protocol$Http11ConnectionHandler.process(Http11Protocol.java:583)
at org.apache.tomcat.util.net.JIoEndpoint$Worker.run(JIoEndpoint.java:447)
at java.lang.Thread.run(Thread.java:619)

```

这个是什么错误啊？我是startProcessInstanceByKey就报错了。

你使用的类库版本不对。

在回复一下

Caused by: com.mysql.jdbc.exceptions.MySQLIntegrityConstraintViolationException: Cannot delete or update a parent row: a foreign key constraint fails

这个错误时由于你使用的  
MySQL 方言不对

使用  
org.hibernate.dialect.MySQLInnoDBDialect  
这个方言就没有问题了。

4 楼 [alymail](#) 2009-12-15 引用

```

org.jbpm.api.JbpmException: no org.jbpm.pvm.internal.id.DbidGenerator in current environment
at org.jbpm.pvm.internal.env.EnvironmentImpl.getFromCurrent(EnvironmentImpl.java:204)
at org.jbpm.pvm.internal.env.EnvironmentImpl.getFromCurrent(EnvironmentImpl.java:190)
at org.jbpm.pvm.internal.hibernate.DbSessionImpl.newTask(DbSessionImpl.java:368)
at org.jbpm.pvm.internal.hibernate.DbSessionImpl.createTask(DbSessionImpl.java:361)
at org.jbpm.jpdl.internal.activity.TaskActivity.execute(TaskActivity.java:62)
at org.jbpm.jpdl.internal.activity.TaskActivity.execute(TaskActivity.java:57)
at org.jbpm.pvm.internal.model.op.ExecuteActivity.perform(ExecuteActivity.java:60)
at org.jbpm.pvm.internal.model.ExecutionImpl.performAtomicOperationSync(ExecutionImpl.java:655)
at org.jbpm.pvm.internal.model.ExecutionImpl.performAtomicOperation(ExecutionImpl.java:615)
at org.jbpm.pvm.internal.model.ExecutionImpl.start(ExecutionImpl.java:215)
at org.jbpm.pvm.internal.cmd.StartProcessInstanceInLatestCmd.execute(StartProcessInstanceInLatestCmd.java:65)
at org.jbpm.pvm.internal.cmd.StartProcessInstanceInLatestCmd.execute(StartProcessInstanceInLatestCmd.java:38)
at org.jbpm.pvm.internal.svc.DefaultCommandService.execute(DefaultCommandService.java:42)

```

```

at org.jboss.pvm.internal.spring.CommandTransactionCallback.doInTransaction(CommandTransactionCallback.java:50)
at org.springframework.transaction.support.TransactionTemplate.execute(TransactionTemplate.java:128)
at org.jboss.pvm.internal.tx.SpringTransactionInterceptor.execute(SpringTransactionInterceptor.java:77)
at org.jboss.pvm.internal.svc.EnvironmentInterceptor.executeInNewEnvironment(EnvironmentInterceptor.java:53)
at org.jboss.pvm.internal.svc.EnvironmentInterceptor.execute(EnvironmentInterceptor.java:40)
at org.jboss.pvm.internal.svc.RetryInterceptor.execute(RetryInterceptor.java:55)
at org.jboss.pvm.internal.svc.SkipInterceptor.execute(SkipInterceptor.java:43)
at org.jboss.pvm.internal.svc.ExecutionServiceImpl.startProcessInstanceByKey(ExecutionServiceImpl.java:66)
at com.test.impl.MyServiceImpl.startProcessInstanceByKey(MyServiceImpl.java:55)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
at java.lang.reflect.Method.invoke(Method.java:597)
at org.springframework.aop.support.AopUtils.invokeJoinpointUsingReflection(AopUtils.java:307)
at org.springframework.aop.framework.ReflectiveMethodInvocation.invokeJoinpoint(ReflectiveMethodInvocation.java:182)
at org.springframework.aop.framework.ReflectiveMethodInvocation.proceed(ReflectiveMethodInvocation.java:149)
at org.springframework.transaction.interceptor.TransactionInterceptor.invoke(TransactionInterceptor.java:106)
at org.springframework.aop.framework.ReflectiveMethodInvocation.proceed(ReflectiveMethodInvocation.java:171)
at org.springframework.aop.framework.JdkDynamicAopProxy.invoke(JdkDynamicAopProxy.java:204)
at $Proxy2.startProcessInstanceByKey(Unknown Source)
at zzg.LeaveServlet.leave(LeaveServlet.java:155)
at zzg.LeaveServlet.process(LeaveServlet.java:84)
at zzg.LeaveServlet.doGet(LeaveServlet.java:57)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:617)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:717)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:290)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)
at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:233)
at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:191)
at org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:128)
at org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:102)
at org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:109)
at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:286)
at org.apache.coyote.http11.Http11Processor.process(Http11Processor.java:845)
at org.apache.coyote.http11.Http11Protocol$Http11ConnectionHandler.process(Http11Protocol.java:583)
at org.apache.tomcat.util.net.JIoEndpoint$Worker.run(JIoEndpoint.java:447)
at java.lang.Thread.run(Thread.java:619)

```

这个是什么错误啊？我是startProcessInstanceByKey就报错了。

3 楼 [Wallian\\_hua](#) 2009-09-23 [引用](#)

我也遇见同样的问题，当我跳入下一个节点的时候，就会报错了。  
说是execution 和 INSTANCE\_引用了。。导致不能掉转到下个节点。

2 楼 [skyuck](#) 2009-09-08 [引用](#)

xiuying 写道

```
taskService.completeTask(taskId, "经理批准");
```

经理审批的时候报数据库外键引起错误：

Caused by: com.mysql.jdbc.exceptions.MySQLIntegrityConstraintViolationException: Cannot delete or update a parent row: a foreign key constraint fails

```

at com.mysql.jdbc.SQLError.createSQLException(SQLError.java:931)
at com.mysql.jdbc.MysqlIO.checkErrorPacket(MysqlIO.java:2985)
at com.mysql.jdbc.MysqlIO.sendCommand(MysqlIO.java:1631)
at com.mysql.jdbc.MysqlIO.sqlQueryDirect(MysqlIO.java:1723)
at com.mysql.jdbc.Connection.execSQL(Connection.java:3283)
at com.mysql.jdbc.PreparedStatement.executeInternal(PreparedStatement.java:1332)
at com.mysql.jdbc.PreparedStatement.executeUpdate(PreparedStatement.java:1604)
at com.mysql.jdbc.PreparedStatement.executeUpdate(PreparedStatement.java:1519)
at com.mysql.jdbc.PreparedStatement.executeUpdate(PreparedStatement.java:1504)
at org.hibernate.persister.entity.AbstractEntityPersister.delete(AbstractEntityPersister.java:2551)
... 32 more

```

遇到过吗?

没有遇到!

1 楼 [xiuying](#) 2009-09-08 [引用](#)

```
taskService.completeTask(taskId, "经理批准");
```

经理审批的时候报数据库外键引起错误:

Caused by: com.mysql.jdbc.exceptions.MySQLIntegrityConstraintViolationException: Cannot delete or update a parent row: a foreign key constraint fails

at com.mysql.jdbc.SQLException.createSQLException(SQLError.java:931)

at com.mysql.jdbc.MysqlIO.checkErrorPacket(MysqlIO.java:2985)

at com.mysql.jdbc.MysqlIO.sendCommand(MysqlIO.java:1631)

at com.mysql.jdbc.MysqlIO.sqlQueryDirect(MysqlIO.java:1723)

at com.mysql.jdbc.Connection.execSQL(Connection.java:3283)

at com.mysql.jdbc.PreparedStatement.executeInternal(PreparedStatement.java:1332)

at com.mysql.jdbc.PreparedStatement.executeUpdate(PreparedStatement.java:1604)

at com.mysql.jdbc.PreparedStatement.executeUpdate(PreparedStatement.java:1519)

at com.mysql.jdbc.PreparedStatement.executeUpdate(PreparedStatement.java:1504)

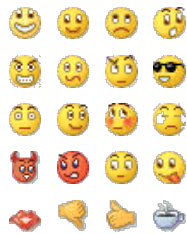
at org.hibernate.persister.entity.AbstractEntityPersister.delete(AbstractEntityPersister.java:2551)

... 32 more

遇到过吗?

发表评论

表情图标



字体颜色:

字体大小:

对齐:

提示: 选择您需要装饰的文字, 按上列按钮即可添加上相应的标签

您还没有登录, 请[登录](#)后发表评论(快捷键 Alt+S / Ctrl+Enter)

