# 流程图说明

## 流程图一

${day}的day表示请假天数

填写请假单task待办人：${user}

部门经理审批task待办人：${dm}

总经理审批task待办人：${pm}

人事归档task待办人：${hr}

以上变量需要在后续操作中提供

## 流程图二

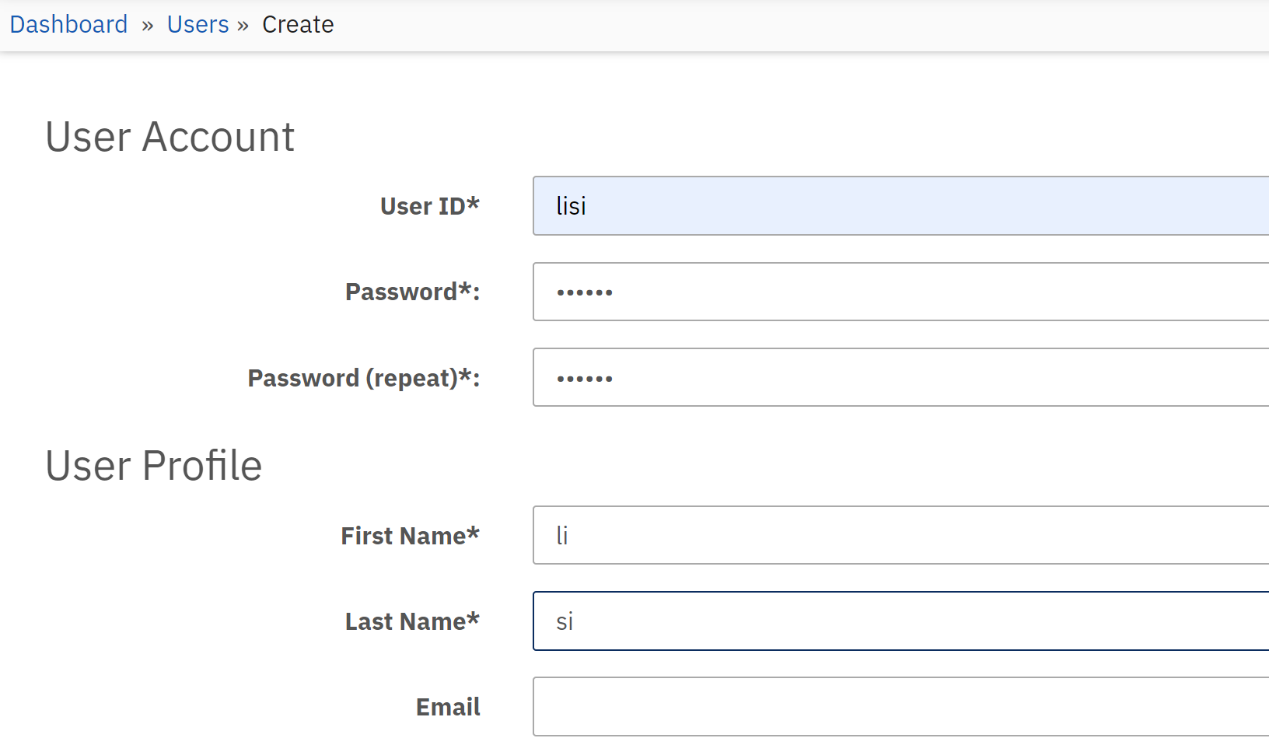
# admin登录并添加其它用户

地址：<http://localhost:8080/>

使用admin账号登录，admin账号可在项目application.yml中配置

默认密码：123456

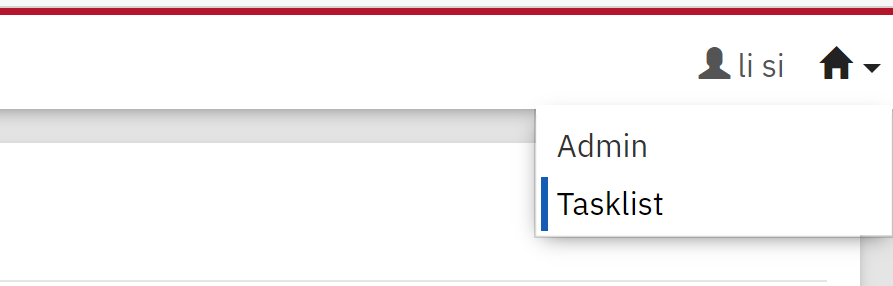
1）：选择admin，再选择createUser

2）：创建几个用户，后面需要用到

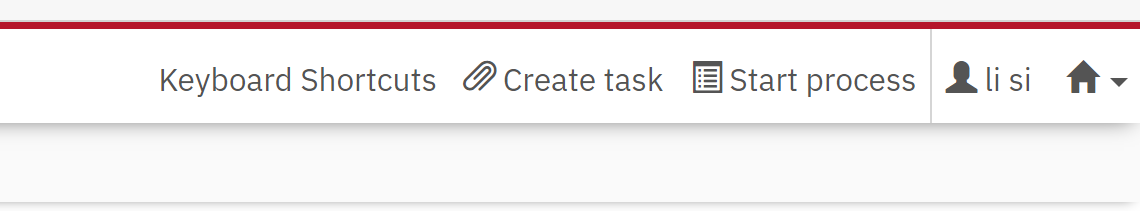
后面有用到的账号：lisi，zhangsan，wangwu，zhaoliu

# 登录lisi账号，启动一个请假流程实例

1）：在右上角有个房子图标，选择 tasklist

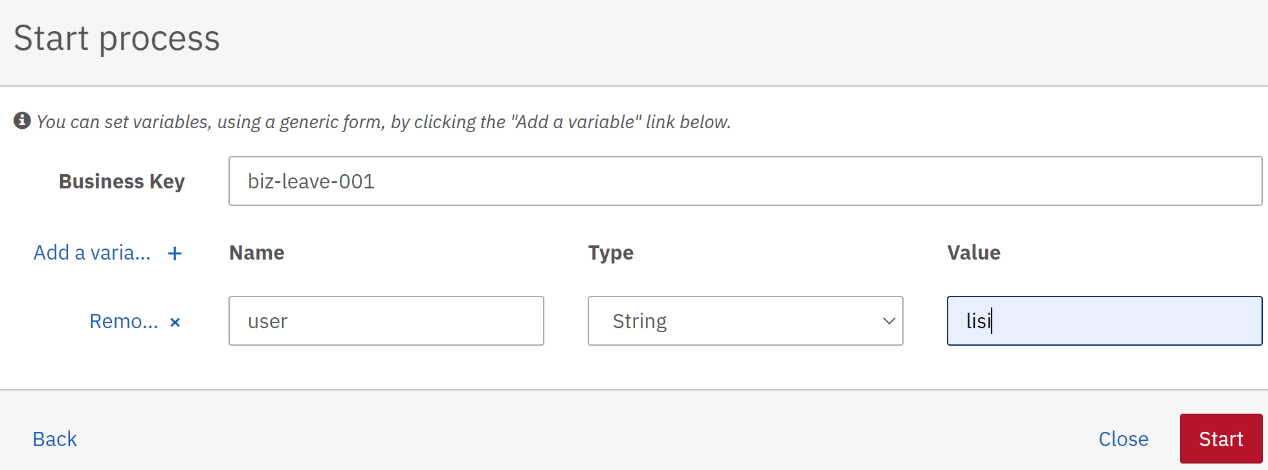


2）：然后选择右上角菜单 start process



3）：选择 Process\_leave 流程

4）：填写相关参数，启动流程，如下图：

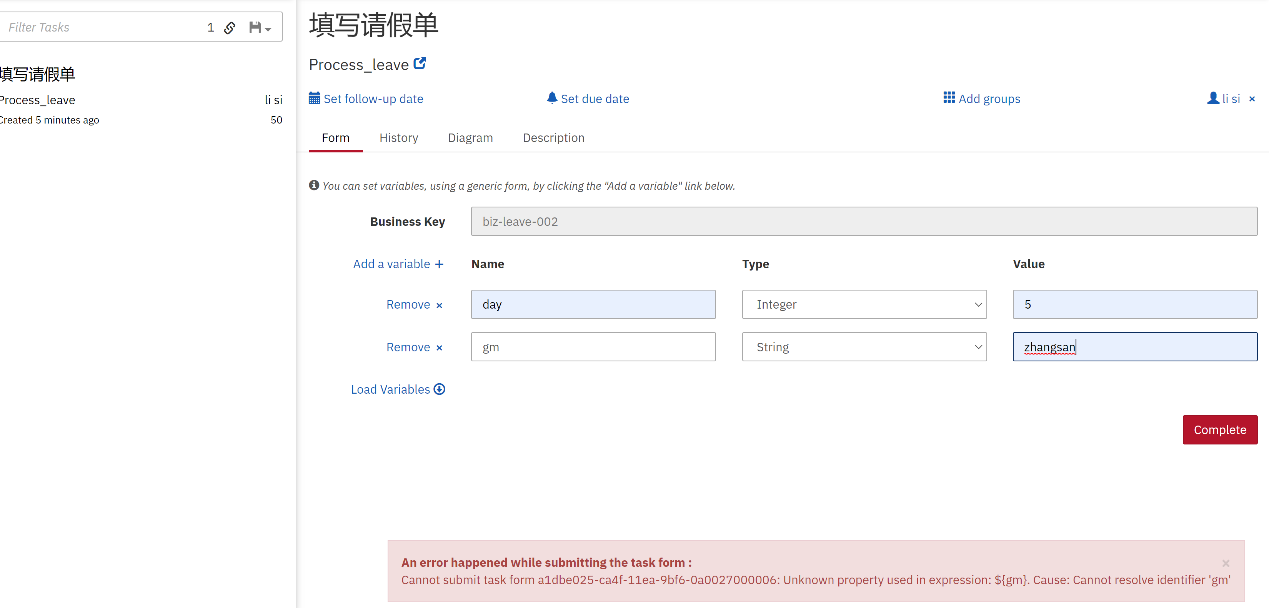


因为流程图定义了请假待办人表达式为：${user}，所以要提供user变量

上图中user是lisi这个账号

# 完成请假单任务

刷新任务列表，lisi就可以看到刚启动的任务了

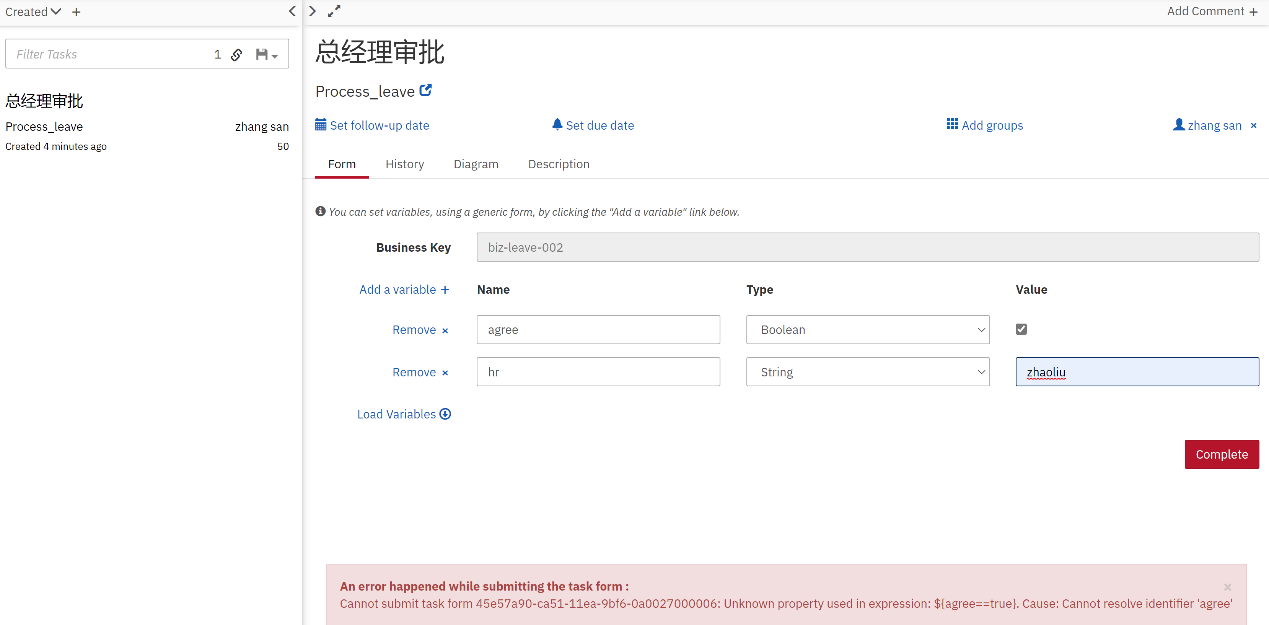
点击任务，填写相关变量，complete任务

上图所示：请假天数day是5天，大于3天，根据流程图定义，所以要走到总经理节点，需要提供gm变量参数

# 总经理办理任务

根据上一步提交的参数，需要zhangsan登录

如下图：

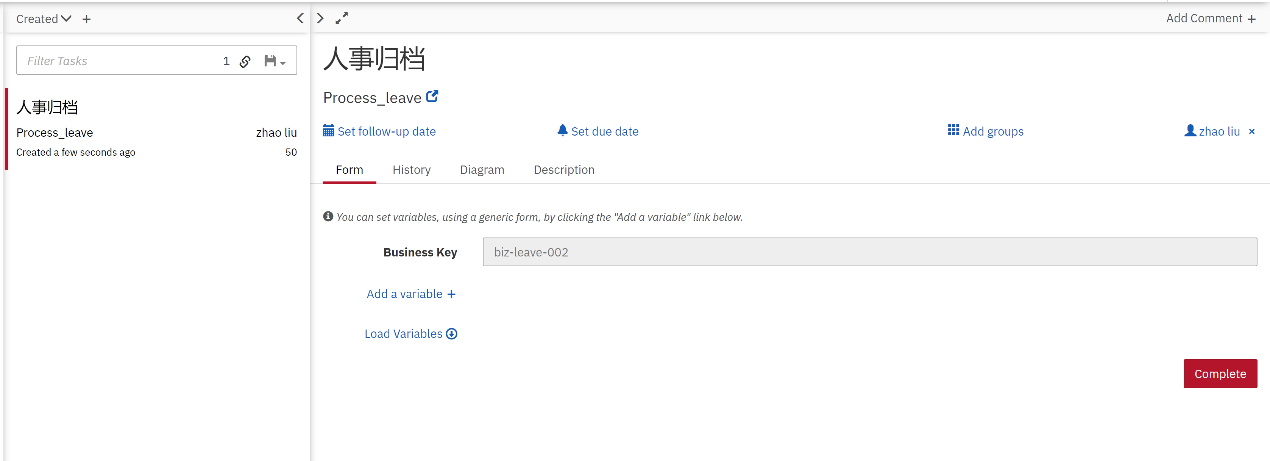


总经理需审批，需要提供变量 agree，表示是否同意，上图是agree=true，所以流程还要提供hr变量，流程会走到人事归档节点。

# 人事办理任务

根据流程图定义，总经理审批同意，并且提供变量hr是zhaoliu，所以zhaoliu登录

如下图：



直接点击Complete，即可完成任务，整个流程就结束了，tasklist也会清空此流程。

# ID生成器

Camunda默认是用uuid生成ID的，我们可以实现sdk中的接口IdGenerator自定义ID生成器：

public interface IdGenerator **{**

String getNextId**();**

**}**

在配置文件applicaton.yml中配置实现的生成器对象名称即可：

**camunda**:

**bpm**:

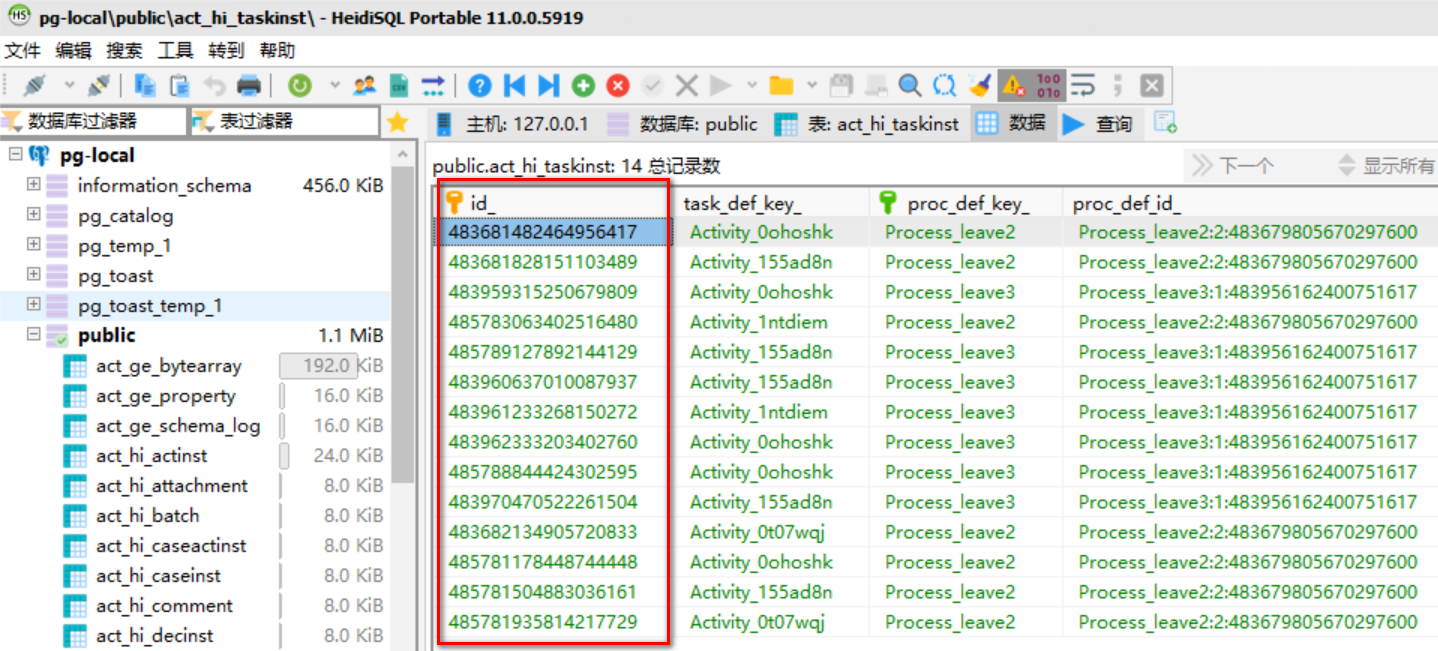
**id-generator**: snowFlakeIdGenerator

本例中使用的是雪花算法。

具体实现类可查看附件：





数据显示效果：

# 测试项目仓库

http://aic-svn.epfootwear.com/scm/git/bpm

# Service Api单元测试

## TaskService

### 根据业务key获取流程执行中的任务

@Test

public void testQueryTaskListByBizKey**()** **{**

String bizKey **=** "biz-leave-20200724002"**;**

List**<**Task**>** list **=** **this.**taskService**.**createTaskQuery**()**

**.**processInstanceBusinessKey**(**bizKey**)**

**.**list**();**

Assertions**.**assertFalse**(**CollectionUtils**.**isEmpty**(**list**),** "任务为空"**);**

list**.**forEach**(**t **->** **{**

System**.**out**.**println**(**"任务ID: " **+** t**.**getId**());**

System**.**out**.**println**(**"任务名称: " **+** t**.**getName**());**

System**.**out**.**println**(**"流程实例ID: " **+** t**.**getProcessInstanceId**());**

System**.**out**.**println**(**"代理人: " **+** t**.**getAssignee**());**

System**.**out**.**println**(**"执行ID: " **+** t**.**getExecutionId**());**

System**.**out**.**println**(**"----------------------------------------------"**);**

**});**

**}**

测试结果：

任务ID: 483970470522261504

任务名称: 部门经理审批

流程实例ID: 483962333199208448

代理人: zhangsan

执行ID: 483962333199208448

----------------------------------------------

任务ID: 485789127892144129

任务名称: 部门经理审批

流程实例ID: 483962333199208448

代理人: null

执行ID: 485788844407525376

### 完成任务并设置流程变量

@Test

public void testCompleteTask**()** **{**

//添加流程变量

Map**<**String**,** Object**>** variables **=** **new** HashMap**<>();**

variables**.**put**(**"dmList"**,** "zhangsan,wangwu"**);**

**this.**taskService**.**complete**(**taskId**,** variables**);**

**}**

### 根据代理人或候选人查询任务

@Test

public void testQueryTaskByAssigneeOrCandidateUser**()** **{**

String user **=** "zhangsan"**;**

List**<**Task**>** list **=** **this.**taskService**.**createTaskQuery**()**

**.**taskCandidateUser**(**user**) //候选人**

//.taskAssignee(user) //代理人

**.**list**();**

Assertions**.**assertFalse**(**CollectionUtils**.**isEmpty**(**list**),** "任务为空"**);**

list**.**forEach**(**t **->** **{**

System**.**out**.**println**(**"任务ID: " **+** t**.**getId**());**

System**.**out**.**println**(**"任务名称: " **+** t**.**getName**());**

System**.**out**.**println**(**"流程实例ID: " **+** t**.**getProcessInstanceId**());**

System**.**out**.**println**(**"代理人: " **+** t**.**getAssignee**());**

System**.**out**.**println**(**"执行ID: " **+** t**.**getExecutionId**());**

System**.**out**.**println**(**"----------------------------------------------"**);**

**});**

**}**

### 拾取任务

如果任务是指定了多个候选人，则需要先拾取任务，才能接着完成任务

拾去任务也是只要有一个人拾取任务，其他候选人就不能拾取了。

@Test

public void testClaimTask**()** **{**

//任意用户都可拾取任务，即在非候选人中都可拾取，

//这种任务权限需要应用自行控制

String user **=** "zhangsan"**;**

**this.**taskService**.**claim**(**taskId**,** user**);**

**}**

### 设置代理人

设置代理人，也可以取消代理人，对于拾取的任务可以通过这种方式取消任务拾取。

@Test

public void testSetAssignee**()** **{**

//值为 null 表示放弃此任务,候选人又可看到此任务

String user **=** **null;**

**this.**taskService**.**setAssignee**(**taskId**,** user**);**

**}**

### 根据任务ID查询

根据任务ID查询详细信息

@Test

public void testGetTaskById**()** **{**

Task task **=** **this.**taskService**.**createTaskQuery**()**

**.**taskId**(**taskId**)**

**.**singleResult**();**

Assertions**.**assertNotNull**(**task**,** "任务不存在"**);**

System**.**out**.**println**(**"任务ID: " **+** task**.**getId**());**

System**.**out**.**println**(**"任务名称: " **+** task**.**getName**());**

System**.**out**.**println**(**"流程实例ID: " **+** task**.**getProcessInstanceId**());**

System**.**out**.**println**(**"代理人: " **+** task**.**getAssignee**());**

System**.**out**.**println**(**"执行ID: " **+** task**.**getExecutionId**());**

System**.**out**.**println**(**"----------------------------------------------"**);**

**}**

## HistoryService

### 查询某个流程实例执行的Task任务实例记录

public void testQueryHistoryTask**()** **{**

String assignee **=** "zhangsan"**;**

String processIntanceId **=** "483681482443984896"**;**

List**<**HistoricTaskInstance**>** list **=** historyService**.**createHistoricTaskInstanceQuery**()**

**.**processInstanceId**(**processIntanceId**)**

//.taskAssignee(assignee)

**.**orderByHistoricActivityInstanceStartTime**().**asc**()**

**.**list**();**

**if** **(**list **==** **null** **||** list**.**isEmpty**())** **{**

System**.**err**.**println**(**"任务为空"**);**

**return;**

**}**

list**.**forEach**(**t **->** **{**

System**.**out**.**println**(**"任务ID: " **+** t**.**getId**());**

System**.**out**.**println**(**"任务名称: " **+** t**.**getName**());**

System**.**out**.**println**(**"流程实例ID: " **+** t**.**getProcessInstanceId**());**

System**.**out**.**println**(**"开始时间: " **+** t**.**getStartTime**());**

System**.**out**.**println**(**"结束时间: " **+** t**.**getEndTime**());**

System**.**out**.**println**(**"持续时长【毫秒】: " **+** t**.**getDurationInMillis**());**

System**.**out**.**println**(**"--------------------------------------------------------"**);**

**});**

**}**

查询流程的执行Activity活动记录

public void testQueryHistoryActivity**()** **{**

String processInstanceId **=** "483681482443984896"**;**

List**<**HistoricActivityInstance**>** list **=** historyService**.**createHistoricActivityInstanceQuery**()**//创建历史活动实例的查询

**.**processInstanceId**(**processInstanceId**)**//

**.**orderByHistoricActivityInstanceStartTime**().**asc**()**//

**.**list**();**

Assertions**.**assertFalse**(**CollectionUtils**.**isEmpty**(**list**),** "活动为空"**);**

**for** **(**HistoricActivityInstance hai **:** list**)** **{**

System**.**out**.**println**(**"活动ID: " **+** hai**.**getId**());**

System**.**out**.**println**(**"活动名称: " **+** hai**.**getActivityName**());**

System**.**out**.**println**(**"实例定义ID: " **+** hai**.**getProcessDefinitionId**());**

System**.**out**.**println**(**"活动类型: " **+** hai**.**getActivityType**());**

System**.**out**.**println**(**"开始时间: " **+** hai**.**getStartTime**());**

System**.**out**.**println**(**"结束时间: " **+** hai**.**getEndTime**());**

System**.**out**.**println**(**"持续时长【毫秒】: " **+** hai**.**getDurationInMillis**());**

System**.**out**.**println**(**"----------------------------------------------------------"**);**

**}**

**}**

结果：

活动ID: StartEvent\_1:483681482443984897

活动名称: 开始

实例定义ID: Process\_leave2:2:483679805670297600

活动类型: startEvent

开始时间: Thu Jul 23 14:18:37 CST 2020

结束时间: Thu Jul 23 14:18:37 CST 2020

持续时长【毫秒】: 1

----------------------------------------------------------

活动ID: Activity\_0ohoshk:483681482464956416

活动名称: 填写请假

实例定义ID: Process\_leave2:2:483679805670297600

活动类型: userTask

开始时间: Thu Jul 23 14:18:37 CST 2020

结束时间: Thu Jul 23 14:19:59 CST 2020

持续时长【毫秒】: 82417

----------------------------------------------------------

活动ID: Activity\_155ad8n:483681828151103488

活动名称: 部门经理审批

实例定义ID: Process\_leave2:2:483679805670297600

活动类型: userTask

开始时间: Thu Jul 23 14:19:59 CST 2020

结束时间: Thu Jul 23 14:21:12 CST 2020

持续时长【毫秒】: 73134

----------------------------------------------------------

活动ID: Activity\_0t07wqj:483682134905720832

活动名称: 总经理审批

实例定义ID: Process\_leave2:2:483679805670297600

活动类型: userTask

开始时间: Thu Jul 23 14:21:12 CST 2020

结束时间: Wed Jul 29 09:22:03 CST 2020

持续时长【毫秒】: 500450977

----------------------------------------------------------

活动ID: Activity\_0ohoshk:485781178444550144

活动名称: 填写请假

实例定义ID: Process\_leave2:2:483679805670297600

活动类型: userTask

开始时间: Wed Jul 29 09:22:03 CST 2020

结束时间: Wed Jul 29 09:23:21 CST 2020

持续时长【毫秒】: 77828

----------------------------------------------------------

活动ID: Activity\_155ad8n:485781504883036160

活动名称: 部门经理审批

实例定义ID: Process\_leave2:2:483679805670297600

活动类型: userTask

开始时间: Wed Jul 29 09:23:21 CST 2020

结束时间: Wed Jul 29 09:25:04 CST 2020

持续时长【毫秒】: 102737

----------------------------------------------------------

活动ID: Activity\_0t07wqj:485781935814217728

活动名称: 总经理审批

实例定义ID: Process\_leave2:2:483679805670297600

活动类型: userTask

开始时间: Wed Jul 29 09:25:04 CST 2020

结束时间: Wed Jul 29 09:29:33 CST 2020

持续时长【毫秒】: 268837

----------------------------------------------------------

活动ID: Activity\_1ntdiem:485783063398322176

活动名称: 人事归档

实例定义ID: Process\_leave2:2:483679805670297600

活动类型: userTask

开始时间: Wed Jul 29 09:29:33 CST 2020

结束时间: Wed Jul 29 09:30:02 CST 2020

持续时长【毫秒】: 29426

----------------------------------------------------------

活动ID: Event\_1254jkp:485783186824105984

活动名称: 结束

实例定义ID: Process\_leave2:2:483679805670297600

活动类型: noneEndEvent

开始时间: Wed Jul 29 09:30:02 CST 2020

结束时间: Wed Jul 29 09:30:02 CST 2020

持续时长【毫秒】: 0

根据业务KEY查询历史流程实例

@Test

public void testQueryHistoryProcessInstance**()** **{**

String bizKey **=** "biz-leave-20200724002"**;**

List**<**HistoricProcessInstance**>** list **=** historyService**.**createHistoricProcessInstanceQuery**()**//创建历史流程实例查询

**.**processInstanceBusinessKey**(**bizKey**)**

**.**orderByProcessInstanceStartTime**().**asc**()**

**.**list**();**

**if** **(**list **==** **null** **||** list**.**isEmpty**())** **{**

System**.**err**.**println**(**"历史流程实例为空"**);**

**return;**

**}**

**for** **(**HistoricProcessInstance hpi **:** list**)** **{**

System**.**out**.**println**(**"流程实例ID: " **+** hpi**.**getId**());**

System**.**out**.**println**(**"实例定义ID: " **+** hpi**.**getProcessDefinitionId**());**

System**.**out**.**println**(**"开始时间: " **+** hpi**.**getStartTime**());**

System**.**out**.**println**(**"结束时间: " **+** hpi**.**getEndTime**());**

System**.**out**.**println**(**"持续时长【毫秒】: " **+** hpi**.**getDurationInMillis**());**

System**.**out**.**println**(**"-------------------------------------------------------------"**);**

**}**

**}**

结果：

流程实例ID: 483962333199208448

实例定义ID: Process\_leave3:1:483956162400751617

开始时间: Fri Jul 24 08:54:37 CST 2020

结束时间: null

持续时长【毫秒】: null

## RepositoryService、RuntimeService

### 获取当前任务节点Activity信息

@Test

public void testGetActivity**()** **{**

// 1,根据任务ID查询任务实例

String taskId **=** "483682134905720833"**;**

Task task **=** **this.**taskService**.**createTaskQuery**().**taskId**(**taskId**).**singleResult**();**

// 2,取出流程定义ID

String processDefinitionId **=** task**.**getProcessDefinitionId**();**

// 3,取出流程实例ID

String processInstanceId **=** task**.**getProcessInstanceId**();**

// 4,根据流程实例ID查询流程实例

ProcessInstance processInstance **=** **this.**runtimeService**.**createProcessInstanceQuery**()**

**.**processInstanceId**(**processInstanceId**).**singleResult**();**

// 5,根据流程定义ID查询流程定义的XML信息

ProcessDefinitionEntity processDefinition **=** **(**ProcessDefinitionEntity**)** **this.**repositoryService

**.**getProcessDefinition**(**processDefinitionId**);**

// 6,从流程实例对象里面取出当前活动节点ID

List**<**ExecutionEntity**>** executions **=** **this.**runtimeService**.**createExecutionQuery**()**

**.**processInstanceId**(**processInstanceId**)**

**.**list**()**

**.**stream**()**

**.**map**(**e **->** **(**ExecutionEntity**)** e**)**

**.**collect**(**toList**());**

// 7,使用活动ID取出xml和当前活动ID相关节点数据

**if** **(**executions**.**isEmpty**())** **{**

System**.**err**.**println**(**"流程execution为空"**);**

**return;**

**}**

executions**.**forEach**(**e **->** **{**

ActivityImpl activityImpl **=** processDefinition**.**findActivity**(**e**.**getActivityId**());**

System**.**out**.**println**(**"X: " **+** activityImpl**.**getX**());**

System**.**out**.**println**(**"Y: " **+** activityImpl**.**getY**());**

System**.**out**.**println**(**"With: " **+** activityImpl**.**getWidth**());**

System**.**out**.**println**(**"Height: " **+** activityImpl**.**getHeight**());**

System**.**out**.**println**(**"--------------------------------------------------------"**);**

**});**

**}**

### 获取当前任务活动节点的连接线

public void testGetOutGoingTransitions**()** **{**

// 1,根据任务ID查询任务实例

String taskId **=** "483682134905720833"**;**

Task task **=** **this.**taskService**.**createTaskQuery**().**taskId**(**taskId**).**singleResult**();**

System**.**out**.**println**(**"任务名称：" **+** task**.**getName**());**

// 2,根据流程定义ID查询流程定义的XML信息

ProcessDefinitionEntity processDefinition **=** **(**ProcessDefinitionEntity**)** **this.**repositoryService

**.**getProcessDefinition**(**task**.**getProcessDefinitionId**());**

// 3,获取流程执行实例

ExecutionEntity execution **=** **(**ExecutionEntity**)** **this.**runtimeService**.**createExecutionQuery**()**

**.**executionId**(**task**.**getExecutionId**())**

**.**singleResult**();**

// 4,获取活动节点

ActivityImpl activity **=** processDefinition**.**findActivity**(**execution**.**getActivityId**());**

System**.**out**.**println**(**"活动名称：" **+** activity**.**getName**());**

// 5,取出连线信息

List**<**PvmTransition**>** pvmTransitions **=** activity**.**getOutgoingTransitions**();**

**if** **(**pvmTransitions **==** **null** **||** pvmTransitions**.**isEmpty**())** **{**

System**.**err**.**println**(**"连线信息为空"**);**

**return;**

**}**

**for** **(**PvmTransition pvm **:** pvmTransitions**)** **{**

Properties props **=** pvm**.**getProperties**();**

props**.**toMap**().**forEach**((**k**,** v**)** **->** System**.**out**.**println**(**k **+** " : " **+** v**));**

System**.**out**.**println**(**"------------------------------------------------"**);**

**}**

**}**

测试结果：

------------------------------------------------

任务名称：总经理审批

活动名称：总经理审批

name : 驳回

condition : org.camunda.bpm.engine.impl.el.UelExpressionCondition@36d7a68a

documentation : null

conditionText : ${option=="N"}

------------------------------------------------

name : 同意

condition : org.camunda.bpm.engine.impl.el.UelExpressionCondition@42dd7d82

documentation : null

conditionText : ${option=="Y"}

# 前端流程图显示

说明：

1跳转页面

com.example.TestController#showDiagram

2获取流程图与流程实例流转的节点

com.example.TestController#getBpmnByProcessInstanceId

3 html页面

templates/diagram.html

页面中增加了高亮相关的样式

定义了svg标签，用来修改箭头图标样式