

工作流相关接口

地址: <http://localhost:8082/swagger-ui.html>

强调： 整个流程测试过程，必须用有效的token去进行接口测试。

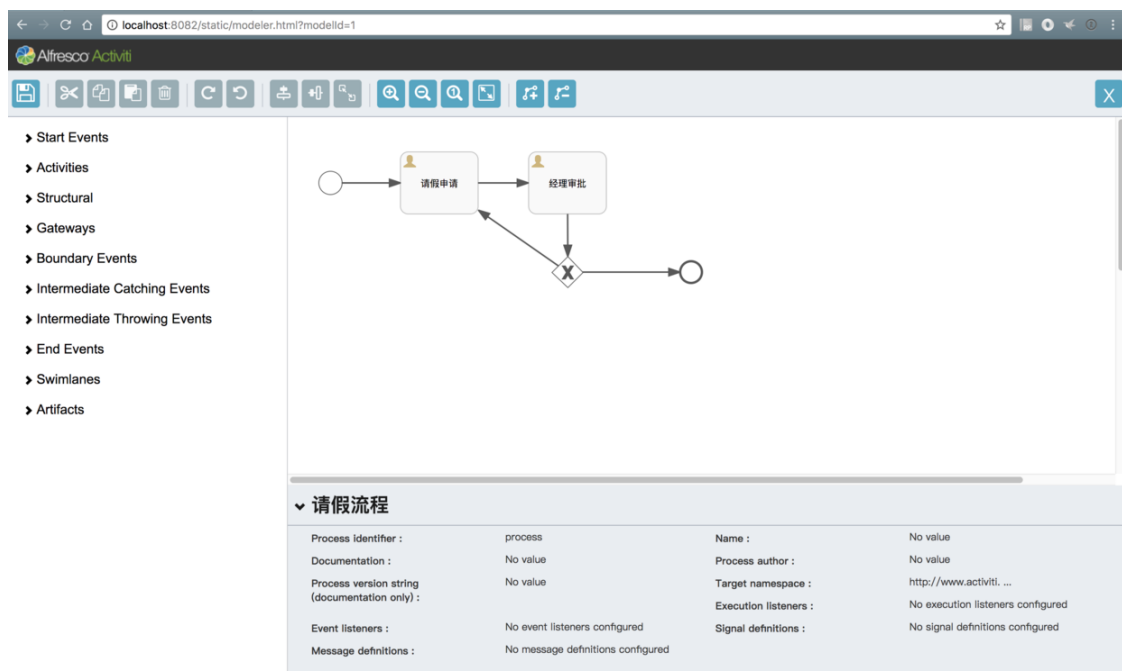
工作流

流程模板相关接口

流程模型模块：流程模型Model操作相关		Show/Hide	List Operations	Expand Operations
GET	/models			获取所有模型
POST	/models			新建一个空模型
POST	/models/uploadFile			上传一个已有模型
DELETE	/models/{id}			删除模型
POST	/models/{id}/deployment			发布模型为流程定义

流程设计

地址: <http://localhost:8082/static/modeler.html?modelId=1>



任务相关接口

工作流任务：工作流任务模块

Show/Hide | List Operations | Expand Operations

GET	/tasks/apply	获取个人发起任务
GET	/tasks/diagram	获取任务流程状态图
GET	/tasks/finish	获取个人历史任务
GET	/tasks/todo	获取待审批任务

Demo（请假流程）

首先创建一个流程模板

POST /models

新建一个空模型

Response Class (Status 200)

OK

Model | Example Value

```
{}
```

Response Content Type */*

Parameters

Parameter	Value	Description	Parameter Type	Data Type
name	<div>模板名称</div>	name	query	string
key	<div>模板核心key, 创建流程用</div>	key	query	string
description	<div>模板描述</div>	description	query	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201	Created		
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out!

Parameters

Parameter	Value	Description	Parameter Type	Data Type
name	请假流程v3	name	query	string
key	leaveV3	key	query	string
description	请假流程v3	description	query	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201	Created		
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out! Hide Response

Curl

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' 'http://localhost:8082/model
```

Request URL

```
http://localhost:8082/models?name=%E8%AF%B7%E5%81%87%E6%B5%81%E7%A8%B3&key=leaveV3&description=%E8%AF%B7%E5%81%87%E6%
```

Request Headers

```
{
  "Accept": "*/*"
}
```

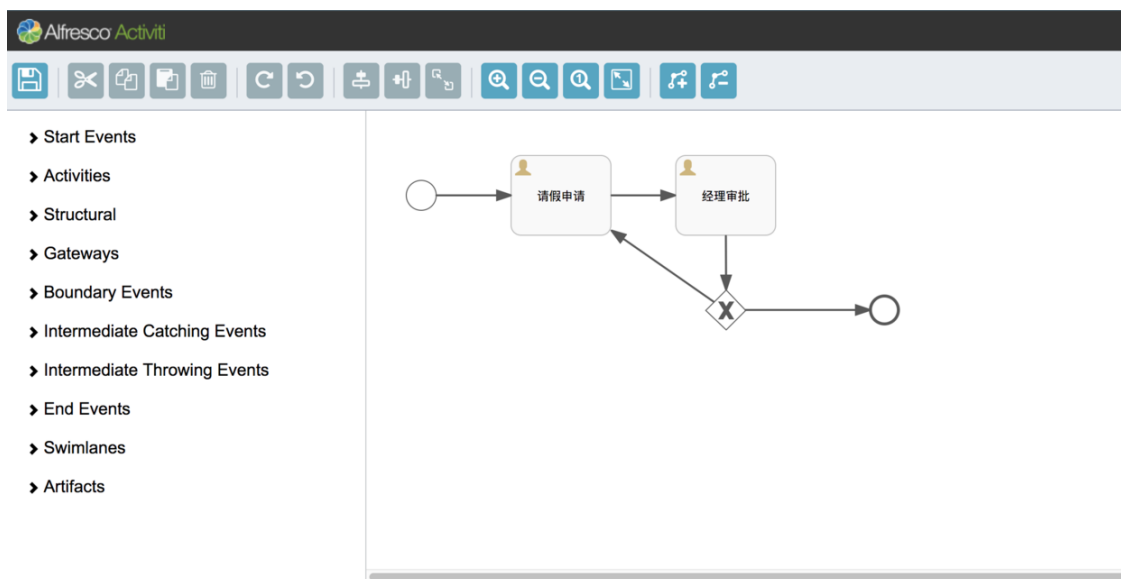
Response Body

```
{
  "modelId": "57501",
  "reason": "操作成功",
  "status": true
}
```

设计对应的流程（具体学习下activiti）

流程访问地址：<http://localhost:8082/static/modeler.html?modelId={此处对应上面的modelId}>

系统已创建请假流程，地址：<http://localhost:8082/static/modeler.html?modelId=1>



相关接口

请假任务模块：请假任务模块

Show/Hide | List Operations | Expand Operations

POST /leave/apply

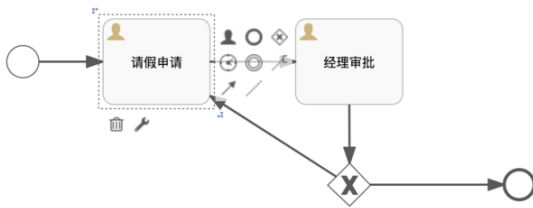
发起请假流程

POST /leave/approve

审批请假流程

流程几要素解析

- 流程发起人信息写入



请假申请

Multi-instance type :	None	Cardinality (Multi-instance) :	No value
Collection (Multi-instance) :	No value	Element variable (Multi-instance) :	No value
Completion condition (Multi-instance) :	No value	Is for compensation :	<input type="checkbox"/>
Form key :	leave-start.form	Assignments :	Assignee \${employee}
Priority :	No value	Due date :	No value
Task listeners :	No task listeners configured	Form properties :	3 form properties

在启动流程示例的时候，注意map传值，具体了解activities

```

    @PostMapping("/apply")
    @ApiOperation(value = "发起请假流程")
    public ObjectRestResponse applyLeave(String startDate, String endDate, String reason) {
        ExecutionEntity pi = null;
        try {
            Map<String, Object> map = new HashMap<>();
            // 创建流程发起人
            map.put("employee", BaseContextHandler.getUsername());
            // 与入流程相关数据
            map.put("reason", reason);
            map.put("startDate", new DateTime(startDate).toDate());
            map.put("endDate", new DateTime(endDate).toDate());
            // 请假流程key
            String processName = "process";
            // 流程启动
            identityService.setAuthenticatedUserId(BaseContextHandler.getUsername());
            // 启动一个流程实例
            pi = (ExecutionEntity) runtimeService.startProcessInstanceByKey(processName, map);
            String taskId = pi.getTasks().get(0).getId();
            // 根据实例把流程往下推
            taskService.complete(taskId, map);
        } finally {
            identityService.setAuthenticatedUserId(null);
        }
        if(pi!=null) {
            return new ObjectRestResponse().data(pi.getProcessInstanceId());
        }else{
            return new ObjectRestResponse().data("");
        }
    }
}

```

● 流程审批人设置

说明：流程审批人岗位和组织架构中的部门岗位对应，必须一致。

配置流程岗位

初始化

组织架构

输入关键字进行过滤

超级租户部门

普通租户

组织相关

用户管理

岗位管理

岗位名称

搜索

添加

关联用户

关联角色

数据授权

职位	编码	类型	操作
部门经理	12	flow	<div>编辑</div> <div>删除</div>

Total 1

20/page

<1>Go to1

编辑



* 职位

部门经理

* 编码

12

类型

流程岗位

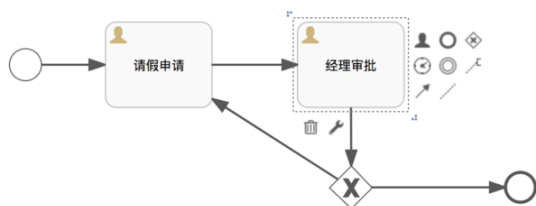
流程岗位

权限岗位

取消

确定

设置流程审批岗位



▼ 经理审批

Multi-instance type :	None	Cardinality (Multi-instance) :	No value
Collection (Multi-instance) :	No value	Element variable (Multi-instance) :	No value
Completion condition (Multi-instance) :	No value	Is for compensation :	<input type="checkbox"/>
Form key :	No value	Assignments :	1 Candidate groups
Priority :	No value	Due date :	No value
Task listeners :	No task listeners configured	Form properties :	No form properties selected

Assignment

Assignee

Candidate users

- +

Candidate groups

- +

Cancel

Save

- 流程审批

```

@PostMapping("/approve")
@ApiOperation(value = "审批请假流程")
public ObjectRestResponse applyLeave(String processInstanceId, boolean agree) {
    Task task = taskService.createTaskQuery().processInstanceId(processInstanceId).singleResult();
    Map<String, Object> map = new HashMap<>();
    map.put("agree", agree);
    // 获取当前人拥有的流程岗位
    List<String> userFlowPositions = userFeign.getUserFlowPositions(BaseContextHandler.getUserID());
    // 获取当前流程具有审批的岗位
    List<IdentityLink> identityLinksForTask = taskService.getIdentityLinksForTask(task.getId());
    boolean flag = false;
    // 判断当前人是否需要相关审批岗位
    for (IdentityLink link : identityLinksForTask) {
        long count = userFlowPositions.stream().filter(new Predicate<String>() {
            @Override
            public boolean test(String s) { return s.equals(link.getGroupId()); }
        }).count();
        if (count > 0) {
            flag = true;
            break;
        }
    }
    // 当前人具有相关审批权限,则可以审批请假流程
    if (flag) {
        task.setAssignee(BaseContextHandler.getUsername());
        taskService.saveTask(task);
        taskService.complete(task.getId(), map);
        return new ObjectRestResponse().data(true);
    }
    return new ObjectRestResponse().data(false);
}

```