# flowable工作流使用说明

注意：流程模块中，凡是涉及到用户的，使用username做唯一Key，即用户名，并非id

注意：流程定义中第一个用户任务一定要是申请人节点，节点id必须为start

第一次启动项目会自动生成flw\_开头的表，可忽略表不存在的报错，后面再次启动就会正常

## 科普工作流的基本使用概念

工作流，首先需要定义一个流程模板，咱们这里用web版流程设计器编辑，得出的是一个流程定义的xml对象。咱们把它叫流程模板，他的key我们叫**流程定义key**，这个是代表一套流程模板的，同一个key能有多个版本的模板。每个版本的模板对象都有个id，我们叫**流程定义id。**通常，我们是根据key来获取最新版本的模板启动一个流程（当然，也可以根据定义id来开启一个流程，正常逻辑应该是取最新去，又指定的业务情景就使用id），这个被启动的流程，我们叫**流程实例**，同样，它也有个id，我们叫**流程实例id** 。

对于用户任务节点，同样的，有两种，一个是定义模板中取来的，里面的信息是事先定义好的，不变的，里面的id就是我们设计流程的时候输入的id。一个是流程实例中取来的，这个对象的id是自动生成的，一般就是uuid。

在我所定义的流程与业务的关联关系表flow\_my\_business中：

data\_id 业务数据id

process\_definition\_key 流程定义key

process\_definition\_id 流程定义id

process\_instance\_id 流程实例id

task\_id 流程实例中的待处理节点id，就是uuid那个

task\_name 这个节点的名字，咱们输入的中文的那个

task\_name\_id 这个节点的定义中的id，比如第一个任务节点我所强调的start

操作流程实例的时候，用的就是实例的id；操作流程模板，用的便是定义id。

## 1.整合（jeecg-boot）

### 1.1、添加模块

复制模块代码jeecg-boot-module-flowable

#### 1.1.1、主工程pom.xml `<modules>`中加入

|  |
| --- |
| <module>jeecg-boot-module-flowable</module> |

#### 1.1.2、jeecg-boot-module-system 工程中加入依赖

注：子模块中引用亦可

|  |
| --- |
| <dependency>  <groupId>org.jeecgframework.boot</groupId>  <artifactId>jeecg-boot-module-flowable</artifactId>  <version>3.0</version>  </dependency> |

#### 1.1.3、**yml**

|  |
| --- |
| # flowable config #  flowable:  process:  definition-cache-limit: -1  database-schema-update: true  activity-font-name: 宋体  label-font-name: 宋体  annotation-font-name: 宋体 |

### 1.2、执行**flowable** 数据库脚本

|  |
| --- |
| doc/flowable.sql |

### 1.3、实现必要接口 IFlowThirdService.java

注意：必须实现接口中定义的所有方法，并注入到spring容器

|  |
| --- |
| @Service  public class FlowThirdServiceImpl implements IFlowThirdService {  } |

### 1.4、前端整合

1、复制代码ant-design-vue-jeecg/src/views/flowable

2、安装 yarn add workflow-bpmn-modeler

3、文件main.js 引入 elementui 必须，因为流程设计器基于elementui

|  |
| --- |
| import ElementUI from 'element-ui';  import 'element-ui/lib/theme-chalk/index.css';  Vue.use(ElementUI); |

## 使用

### 2.1前端

#### 2.1.1添加配置流程设计器菜单

ant-design-vue-jeecg/src/views/flowable/modeler/modelerDesign.vue文件配置菜单或路由，进入该页面进行流程定义

设计流程时，切记设定好用户任务节点的审批人员，网关连线设定好条件，否则系统必然报错。

本流程模块设计，**第一个节点必须设置为发起人节点，节点id约定为start**，否则后果自负（该节点设置的候选人无效），本流程逻辑设计：当流程启动时会自动以申请人的身份完成第一个节点，方便实现退回到流程发起人而不终止流程。终止流程操作将为独立的一种操作（撤回，即销毁流程），正常流程应该顺利审批完成所有节点，驳回在此被设计为流程的一部分（原本activiti中，驳回到发起人便会销毁流程，不符合国情，所以有此设计）。

#### 2.1.2业务使用到的组件

ant-design-vue-jeecg/src/views/flowable/components文件夹下

FlowableMixin.js 按钮权限相关混入文件，案例示范显示隐藏权限逻辑，具体逻辑随实际业务修改

|  |  |  |
| --- | --- | --- |
| **文件** | **作用说明** | **权限建议** |
| ActApplyBtn.vue | 提交启动流程 | 数据草稿状态下可见 |
| ActCancelBtn.vue | 撤回销毁流程 | 数据提交后可见 |
| ActHandleBtn.vue | 处理流程-0通过、退回节点（1驳回、2退回、3重新提交） | 通过：处理人可见（第一个用户节点时文字信息改为提交）  退回：处理人可见（第一个用户节点不可见）  重新提交：处理人且第一个节点可见 |
| ActHistoricDetailBtn.vue | 审批处理历史 | 数据有实例id就可见 |
| HistoricDetail.vue | 审批处理历史页面组件 |  |

**以上按钮流程相关props：**

dataId：业务数据的主Id

Variables：流程变量，比如网关节点需要的判断参数

关于各按钮在业务页面显示的权限由各自页面实际情况进行控制，下面进行详述

#### 2.1.3业务案例

ant-design-vue-jeecg/src/views/flowable/test\_demo/TestDemoList.vue 页面配置进菜单查看

### 2.2后端

业务层Service必须继承FlowCallBackServiceI接口并实现其方法，实现的方法体可不写代码，根据具体业务取舍，例：

|  |
| --- |
| @Service("testDemoService")  public class TestDemoServiceImpl extends ServiceImpl<TestDemoMapper, TestDemo> implements ITestDemoService, FlowCallBackServiceI {  @Autowired  FlowCommonService flowCommonService;  .......  } |

所实现的方法afterFlowHandle将在流程处理过程中被回调，用以增强处理业务层业务逻辑

#### 2.2.1业务数据新增（草稿）

在业务数据新增时，必须初始化业务与流程的关联，调用FlowCommonService.initActBusiness方法即可，注意方法注释描述

|  |
| --- |
| /\*\*  \* 初始生成业务与流程的关联信息<br/>  \* 当业务模块新增一条数据后调用，此时业务数据关联一个流程定义，以备后续流程使用  \* @return 是否成功  \* @param title 必填。流程业务简要描述。例：2021年11月26日xxxxx申请  \* @param dataId 必填。业务数据Id，如果是一对多业务关系，传入主表的数据Id  \* @param serviceImplName 必填。业务service注入spring容器的名称。  \* 例如：@Service("demoService")则传入 demoService  \* @param processDefinitionKey 必填。流程定义Key，传入此值，未来启动的会是该类流程的最新一个版本  \* @param processDefinitionId 选填。流程定义Id，传入此值，未来启动的为指定版本的流程  \*/  public boolean initActBusiness(String title,String dataId, String serviceImplName, String processDefinitionKey, String processDefinitionId){  }  flowCommonService.initActBusiness("流程标题",dataId,"testDemoService","test-demo",null); |

删除业务数据时调用flowCommonService.delActBusiness(id.toString());

|  |
| --- |
| @Override  public boolean removeById(Serializable id) {  /\*\*删除数据，移除流程关联信息\*\*/  flowCommonService.delActBusiness(id.toString());  return super.removeById(id);  } |

其他参见源码 TestDemoServiceImpl

### 2.3流程审批操作的开发

**后端：**

参见源码 TestDemoServiceImpl

实现FlowCallBackServiceI中的业务方法以及数据列表接口的一些流程相关过滤即可，主要操作在前端

**前端：**

参见源码TestDemoList.vue对流程的使用，案例中未对按钮的显示权限进行处理

流程开发主要工作在前端对这些操作按钮的显示权限上的定义以及表单的编辑权限定义，需要根据实际业务进行处理。一般来讲，有以下建议：

1. 根据业务列表接口返回的数据todoUsers doneUsers 参数与当前登录人的username进行对比控制按钮显示与否以及列表数据的过滤（数据过滤由后端处理）

参数为username字符串数组json字符串

todoUsers :当前节点可以审批的人，相当于他们的待办

doneUsers ：审批处理过的人，相当于已办

1. 根据actStatus 流程状态以及 taskNameId 当前待处理的节点Id来控制数据表单的可编辑状态、输入项的增减变化以及按钮的显隐

### 2.4流程状态的解释

草稿：业务数据未与流程相关联或已关联定义信息但是未启动流程

启动：业务数据与流程已关联并通过了第一个节点（申请人节点start）

撤回：启动的流程被销毁，业务数据与流程定义信息存在关联，可直接启动（亦可重新关联 别的流程再启动）

驳回：驳回状态的流程并未结束，流程实例依然存在，被驳回到的节点需要重新审批，业务 按钮中，退回与驳回实际上都是驳回，当驳回到申请人节点start时，请注意业务 开发中表单编辑权限需要打开，此时再度提交流程实际上是审批操作，注意页面上 的文字描述

审批中：流程正常流转中

审批通过：流程全部审批完成自动结束了

审批异常：预定义，暂未用到

## 关于会签

**设计案例**  


**使用**

|  |
| --- |
| <act-apply-btn @success="loadData" :data-id="record.id"  :variables="{ assigneeList:[]}"></act-apply-btn> |

见demo代码，会签节点处理必须传入 assigneeList 数组变量到后台，不可为 null 否则会报错。

assigneeList 即为 设计中的集合名，需保持一致，执行方式需选 并行

assigneeList 为空数组时，后台处理逻辑为取节点中配置的候选人员目标；如果assigneeList 不为空，比如[‘admin’,’jeecg’]，备选目标则为此两人；如果后台有对此节点自定义flowCandidateUsernamesOfTask实现，则取后台设置的用户对象。