**Activiti流程开发规范文档**

# 一 创建流程实例

1. 步骤
   1. 根据申请人获取下个节点候选人列表
   2. 将表单基本信息插入BaseEntity表,同时获取OA\_ID值
   3. 插入业务表数据
   4. 启动流程实例
   5. 设置任务的候选人
   6. 完成
2. 必须参数
   1. 业务表ID
   2. 流程实例ID
   3. Vm表单key

# 二 任务列表

1. 待认领的任务列表
   1. 任务基本信息
   2. 查看任务明细
   3. 认领
2. 待办理的任务列表
   1. 快速办理
   2. 打开任务明细办理
   3. 跟踪办理明细
   4. 驳回
3. 已办理的任务列表
   1. 查询办理明细

# 三 我发起的流程实例

1. 运行中的流程
   1. 邮件催办
   2. 跟踪流程
2. 已结束的流程
   1. 查询审批明细

# 四 代码规范

1. 所有业务接口必须继承WfBaseService
2. 所有业务实现必须继承 WfBaseServiceImpl
3. 具体的业务实现 必须重新startFlow方法,该方法用于根据具体的流程启动一个流程实例
4. 所有业务对象的查询 必须同时查询出 BaseEntity的数据,可通过联合业务表+OA\_BASE表查询,如下

|  |
| --- |
| <resultMap type=*"authorityApply"* id=*"AuthorityApplyMapper"*>  <result property=*"authorityId"* column=*"ID"*/>  <result property=*"processInstanceId"* column=*"PROCESS\_INSTANCE\_ID"*/>  <result property=*"userERP"* column=*"USER\_ID"*/>  <result property=*"formName"* column=*"FORM\_NAME"*/>  <result property=*"authorityType"* column=*"ATYPE"*/>  <result property=*"userName"* column=*"ANAME"*/>  <result property=*"userDep"* column=*"ADEP"*/>  <result property=*"authorityPost"* column=*"APOST"*/>  <result property=*"roleId"* column=*"ROLEID"*/>  <result property=*"roleName"* column=*"ROLENAME"*/>  <result property=*"authorityCause"* column=*"ACAUSE"*/>  <result property=*"authorityApplyName"* column=*"AAPPLYNAME"*/>  <result property=*"authorityApplyERP"* column=*"AAPPLYERP"*/>  <result property=*"taskStartTime"* column=*"APPLY\_TIME"*/>  <result property=*"taskEndTime"* column=*"END\_TIME"*/>  <result property=*"reason"* column=*"REASON"*/>  <result property=*"taskFlag"* column=*"TASK\_FLAG"*/>  <result property=*"userName"* column=*"USER\_NAME"*/>  <!-BaseEntity的数据-->  <result property=*"oAId"* column=*"OA\_ID"*/>  <result property=*"processInstanceId"* column=*"PROCESS\_INSTANCE\_ID"*/>  <result property=*"userERP"* column=*"USER\_ERP"*/>  <result property=*"userName"* column=*"USER\_NAME"*/>  <result property=*"userDep"* column=*"USER\_DEP"*/>  <result property=*"formName"* column=*"FORM\_NAME"*/>  <result property=*"formType"* column=*"FORM\_TYPE"*/>  <result property=*"taskFlag"* column=*"TASK\_FLAG"*/>  <result property=*"taskStartTime"* column=*"TASK\_START\_TIME"*/>  <result property=*"taskEndTime"* column=*"TASK\_END\_TIME"*/>  <result property=*"reason"* column=*"REASON"*/>    </resultMap>  <!-- 查询一个流程 -->  <select id=*"getApplyById"* parameterType=*"authorityApply"* resultMap=*"AuthorityApplyMapper"*>  SELECT <include refid=*"authorityApplyAll"*/>,<include refid=*"com.jd.uim.data.hander.db.api.workflow.DbHandlerOABase.OABaseAll\_Include"*/>  FROM AUTHORITY\_APPLY A,OA\_BASE B  WHERE A.OA\_ID=B.OA\_ID  <include refid=*"searchAllWhere"*/>  </select> |

1. 所有业务POJO必须继承BaseEntity
2. 办理任务,认领任务,跟踪任务,查询任务明细,催办等功能都是统一个入口，所以必须规范好公共参数，可使用冗余数据到业务表，每个表单隐藏必须的表单数据(同时要求隐藏字段的ID必须一致)

# 五 待验证问题

1. 部署流程时定义每个节点属性(如下)—>重点

节点一

节点二

属性:

key1=M1

key2=最高级别

1:根据属性可实现动态办理人

2:根据属性可实现找不到办理人时是否跳过节点

3：…………..

1. 如果使用流程引擎自带的查询如何实现分页查询所有流程的任务信息,目前的查询都是根据流程key+(办理人/候选人)查询，可考虑使用原生态SQL查询
2. 获取流程图，在流程跟踪的时候需要显示流程图信息,需要验证
3. 流程办理是如何属性办理意见(同意和驳回都有可能输入意见),而且在跟踪的时候需要将每个人的办理意见显示到每个办理人办理的节点上, 需要验证
4. 其他