xMIS 快速开发手册

用系统公用组件、命名规则、系统字典等速查手册

# Java工具类

## com.bitc.jk.xmis.util.WebUtil Web相关工具

### 生成JSON字符串

String buildJsonStoreByList(List<?>)

String buildJsonStoreByList(List<?>, int)

适用于POJO的List，第二个参数为总记录数，不指定的情况下默认为List.getSie()

String buildStoreForJsonWithoutPageInfo(List, String)

适用于没有分页信息的JDBC查询结果（使用JdbcTemplate. queryForList获取的结果）

### 向response里写入信息

returnResponse(HttpServletResponse, String)

### 获取当前用户，不需要参数即可获取，从当前进程中获取

EmployeeObj getCurrentEmployee()

### 获取当前用户，从session中获取

EmployeeObj getCurrentEmployee(HttpSession)

### 通过request获取客户端IP地址

String getIpAddr(HttpServletRequest)

### 向JSP页面输出Request脚本

outputRequestParameterAsScript(HttpServletRequest, JspWriter, String)

*注：配合JS中的*[*Request*](#_Request请求参数)*对象一起使用*

### 生成StandardQuery对象

StandardQuery generateStandardQueryByInkling(HttpServletRequest)

注：参见通用模糊查询组件InklingSearchField

## com.bitc.jk.xmis.util.TemplateUtil 格式化字符串

### 格式化字符串

format(String, Map<String, Object>)

替换字符串中的变量，变量以{}开头结尾

format(String, Object[])

替换字符串中的占位变量，如：{0},{1}。必须从0开始不间断编号

## com.bitc.jk.xmis.util.CryptUtil 加密工具

### MD5 信息摘要

String CryptUtil.MD5(String)

## com.bitc.jk.xmis.util.DateUtil 日期时间工具

### 格式化日期

String formatDate(Date)

String formatDate(Date, String)

String formatDateTime(Date)

### 解析字符串为日期

java.util.Date parseDateTime(String)

java.util.Date parseDate(String, String)

### 获取当前日期、时间

java.sql.Date getCurrentDate()

java.sql.Date getCurrentDateTime()

## com.bitc.jk.xmis.web.SpringInit

### 获取Spring Bean

Object getBean(String)

## com.bitc.jk.xmis.util.JSONUtils JSON工具

## com.bitc.jk.xmis.util.UUIDUtils UUID工具

# Extjs Ajax Response

## Ext.Ajax.request的Response

类：com.bitc.jk.xmis.web.response.ExtjsAjaxResponse

属性：success、message、messageTitle、attributes

说明：success为此请求在业务上成功与否的标志；message与messageTitle配置提示信息；attributes中添加需要返回到前台的数据

## Ext.form.BasicForm.submit的Response

同Ext.Ajax.request的Response

## Ext.form.BasicForm.load的Response

类：com.bitc.jk.xmis.web.response.ExtjsFormLoadResponse

属性：data

说明：data为POJO，该对象的属性对应上FormPanel中的Field的name(ComboBox为hiddenName)

## LoginTimeout的Response

类：com.bitc.jk.xmis.web.response.LoginTimeoutResponse

属性：redirectUri

说明：在Session超时的情况下发起Ajax请求时，系统自动返回此响应，页面会自动跳转到登录页面，redirectUri指定重新登录后跳转到哪个页面。一般为登录前的页面。

注：此过程系统会自动完成，Action中不需要再判断超时

## TreeLoader的Response

类：com.bitc.jk.xmis.web.response.TreeLoaderResponse

属性：nodes，类型为com.bitc.jk.xmis.web.response.TreeNode

# JavaScript

## 全局变量

### 当前用户

变量名：\_\_emp

类型：Object

变量属性：同Java类EmployeeObj

文件：dynamic.jsp

### 数据库表名列名

变量名：DBSchema

类型：Object

变量属性：同Java类DBSchema

文件：resources/js/DBSchema.jsp

### 应用根路径

变量名：\_\_cxtPath

类型：String

文件：dynamic.jsp

### 应用完整路径

变量名：\_\_fullPath

类型：String

文件：dynamic.jsp

## JS工具类

## Extjs系统公用组件

### 关注

类名：com.bitc.xmis.workflow.Attention

继承自：Ext.util.Observable

用法：详见API DOC

说明：用户可以关注任何系统中的对象，比如合同、项目等。需要添加关注功能的地方可以引用此文件，创建此类的对象

文件：resources/js/Attention.js

### 关注列表

类名：com.bitc.xmis.workflow.AttentionList

继承自：Ext.list.ListView

用法：详见API DOC

说明：用户的关注列表，放在首页中

文件：resources/js/Attention.js

### 草稿

类名：com.bitc.xmis.sys.Draft

继承自：Ext.util.Observable

用法：详见API DOC

说明：用户可以关注任何系统中的对象，比如合同、项目等。需要添加关注功能的地方可以引用此文件，创建此类的对象

文件：resources/js/Draft.js

### 草稿箱/草稿列表

类名：com.bitc.xmis.sys.DraftList

继承自：Ext.list.ListView

用法：详见API DOC

说明：用户的关注列表，放在首页中

文件：resources/js/Draft.js

### 组织机构树

类名：com.bitc.xmis.OrgTreePanel

继承自：Ext.tree.TreePanel

用法：详见API DOC

说明：系统组织机构树，可以提供多种显示模式

文件：organization/OrgTreePanel.js

### Request请求参数

类名：com.bitc.xmis.Request

用法：详见API DOC

### 通用模糊查询组件

类名：Ext.ux.InklingSearchField

继承自：Ext.ux.form.SearchField

用法：详见API DOC

说明：需要配合后台代码。参见StandardQuery相关内容

## Extjs自定义组件

### 表单选择组件

类名：Ext.ux.form.ChooseField

用法：详见API DOC

说明：用户填写表单的时候可以打开一个Window选择数据，此组件可以自定义选择的数据来源、选择操作、清除数据操作等等

文件：resources/js/ChooseField.js

### 自定义格式的NumberField

类名：Ext.ux.form.CustomNumberField

继承自：Ext.form.NumberField

xtype: custnumberfield

用法：详见API DOC

说明：用于输入格式化的数字，如金额、百分比数等

文件：resources/js/CustomNumberField.js

### 货币金额Field

类名：Ext.ux.form.MoneyField

继承自：Ext.ux.form.CustomNumberField

xtype: moneyfield

用法：详见API DOC

说明：用于输入金额的文本域

文件：resources/js/CustomNumberField.js

### 百分数Field

类名：Ext.ux.form.PercentField

继承自：Ext.ux.form.CustomNumberField

xtype: percentfield

用法：详见API DOC

说明：用于输入百分数的文本域

文件：resources/js/CustomNumberField.js

### 模糊查询组件：

类名：Ext.ux.InklingSearchField

继承自：Ext.ux.form.SearchField

用法：详见API DOC，JAVA中StandardQuery部分内容

文件：resources/js/InklingSearchField.js

## Extjs扩展(plugins)

### Grid每页显示条数扩展

类名：Ext.ux.PagelimitCombo

用于：Ext.PagingToolbar

文件：resources/js/ext-extra.js

### Grid行编辑扩展

类名：Ext.ux.GridRowEditForm

用于：Ext.grid.GridPanel

文件：resources/js/GridRowEditForm.js

### Grid Check列扩展

类名：Ext.ux.grid.CheckColumn

继承自：Ext.util.Observable

用于：Ext.grid.GridPanel

说明：加入到plugins中的同时,还应该加入到ColumnModel中

## Extjs功能改进

### 对Form Field添加Tooltip支持

为Tooltip配置qtip属性，支持String及Tooltip类型

### 对DisplayField添加click支持

为DisplayField配置clickAction属性，类型为Function

配置scope指定函数执行域

### 对日期选择控件DatePicker加入月份单击选择功能 （ext-extra.js）

### 屏蔽回退键BackSpace（ext-extra.js）

### 解决BasicForm.setValues时不能设置RadioGroup值的BUG。注意，RadioGroup必须设置name属性（ext-extra.js）

# 系统字典

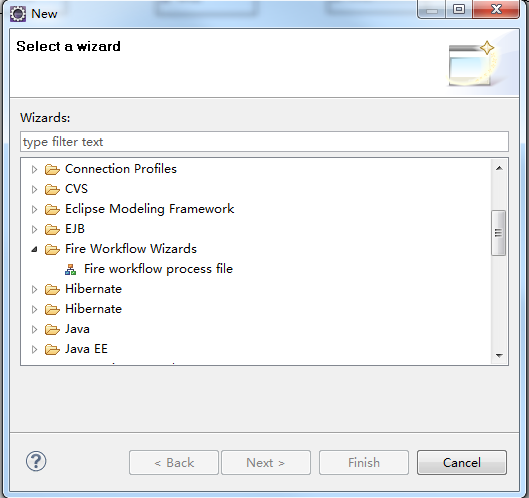
# 流程开发

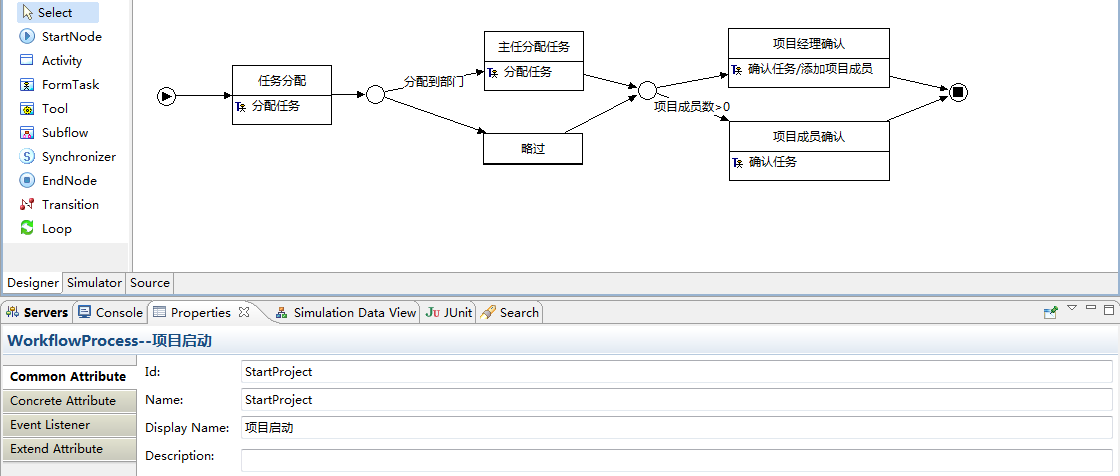
系统使用开源的工作流引擎FireWorkflow。其使用方法请参阅帮助文档。

## 流程开发步骤

### 画流程图

在Eclipse中安装FireWorkflow的插件后可以在新建文件对话框中看到“Fire workflow process file”

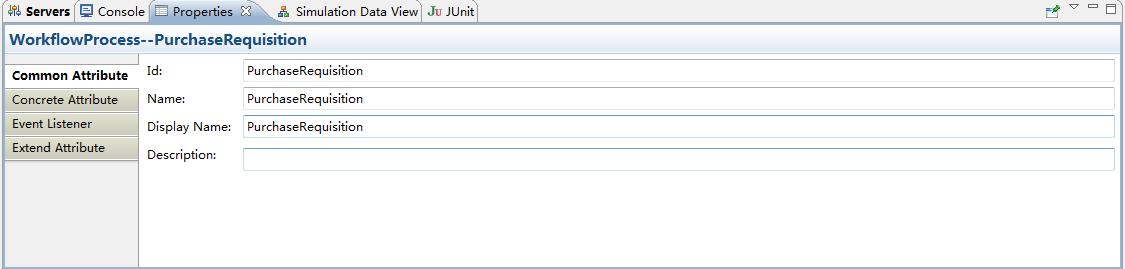




参照FireWorkflow的帮助文档，完成流程图。

### 设置流程属性

点击画布空白处，在Properties视窗中显示：



Id属性是只读的，无法编辑。但可以通过编辑Name属性来更改。其它元素的Id属性也都无法编辑，都是通过编辑Name属性来更改。

该流程Id属性非常重要，必须在系统中所有流程文件中是唯一的，启动流程时将作为唯一标识。

建议在com.bitc.jk.xmis.workflow.util.WorkflowConstants常量类中定义常量，方便其它地方引用此Id

static String WF\_STARTPROJECT = "StartProject";

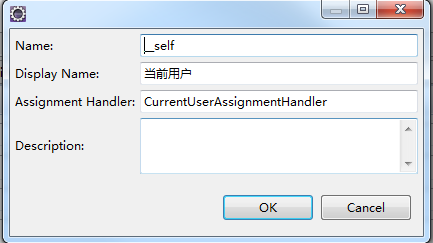
### 为节点定义属性

选中节点或路径，在Eclipse中的Properties视窗里能查看与编辑节点属性。要求为所有Activity及Task设置唯一的ID，设置易于理解的DisplayName。

### 为FormTask添加任务人Performer

选中FormTask，在“Concrete Attribute”中设置其Performer属性。

点击后面的按钮弹出窗口：



有三个必须填写的属性：Name、Display Name、Assignment Handler

Display Name填写容易理解的中文

Name需要依据AssignmentHandler来填写，不同的AssignmentHandler填写方法不一样。

AssignmentHandler可以扩展，目前已经扩展的选择有：

* CurrentUserAssignmentHandler

CurrentUserAssignmentHandler

当前用户为任务人，一般为流程的第一个节点。

Name可以随意填写，无作用

通常在启动流程后，调用

com.bitc.jk.xmis.workflow.util.WorkflowUtil.completeFirstTask(IWorkflowSession, IProcessInstance, String)

来完成第一个节点的任务

* PreviousStepSpecifyAssignmentHandler

PreviousStepSpecifyAssignmentHandler

表示由上一步指定该步骤的接收人。

Name填写在本流程定义中唯一的一个变量名（比如\_\_manager）

在上一步骤完成任务前必须调用方法设置该变量的值来指定接收人，方法有两个：

com.bitc.jk.xmis.workflow.util.WorkflowUtil.specifyActor(IWorkflowSession, String, String)

该

* + 参数1：当前的WorkflowSession，由RuntimeContext获取；
  + 参数2：任务人ID
  + 参数3：指定的变量名（比如\_\_manager）

com.bitc.jk.xmis.workflow.util.WorkflowUtil.specifyActors(IWorkflowSession, List<String>, String)

* + 参数2：任务人ID的List，用于指定多个接收人
* ExpressionAssignmentHandler

ExpressionAssignmentHandler

使用表达式来计算出接收人。可以说是功能最强大的AssignmentHandler。使用IKExpression来解析表达式，支持自定义函数。参看表达式一节

Name即为表达式，如取当前部门的主任与副主任，表达式为：

$join($post(taskDeptId,11),$post(taskDeptId,12))

表达式中可以使用流程变量，比如这里的“taskDeptId”

注意表达式返回的类型必须为List<String>

### 为FormTask设置Form URL

配置EditForm/ViewForm/ListForm的URI，用户点击任务时将打开此URL地址，例：

/projectMain.do?method=getProjectInfo&projectID={projectID}

URI中可以使用流程实例变量，变量名以大括号括起来。



### 编写启动流程的代码

在业务代码中（Service层）嵌入流程控制语句，这样能确保流程数据与业务数据操作在同一个事务中。

在Service中引入

import org.fireflow.engine.RuntimeContext;

......

private RuntimeContext workflowContext;

public RuntimeContext getWorkflowContext() {

return workflowContext;

}

public void setWorkflowContext(RuntimeContext aWorkflowContext) {

this.workflowContext = aWorkflowContext;

}

当然，在Spring配置中必须为Service注入这个属性，比如：

<bean id="deviceServiceImpl" class="com.bitc.jk.xmis.service.impl.DeviceServiceImpl">

<property name="deviceDAO">

<ref local="deviceDAOImpl" />

</property>

<property name="projectDAO">

<ref bean="projectDAOImpl" />

</property>

<property name="workflowContext">

<ref bean="fireRuntimeContext"/>

</property>

</bean>

在Service的方法里启动流程：

IWorkflowSession workflowSession =workflowContext.getWorkflowSession();

SimpleEmployee emp = WebUtil.getCurrentEmployee();

IProcessInstance processInstance = WorkflowUtil.startProcess(workflowSession, workflowConstants. WF\_PURCHASE\_REQUISITION, emp.getUserID() + "");

流程启动后，流程将自动运行至第一个FormTask任务。很多时候第一个任务就是当前用户的任务，这时候可以完成第一个任务：

//添加流程中要使用到的流程变量

WorkflowUtil.addProcessInstanceVariable(processInstance, "requestId",

requisition.getId());

WorkflowUtil.addProcessInstanceVariable(processInstance, "deptId",

requisition.getDept().getDepartmentID());

//完成第一个任务

WorkflowUtil.completeFirstTask(workflowSession, processInstance, String.valueOf(emp.getUserID()), "");

## 流程自动创建任务待办

model为

“wf-task”

appId为

workItemId

script为

"showTaskInfo"

url为

“/workflow.do?method=claim&workItemId=”+workItemId

该url指向WorkflowAtion.claim。签收后url改为

"/workflow.do?method=openForm&workItemId=" + workItemId

该url指向WorkflowAction.openForm，此方法将转发请求到workItem对应的Form URI

script改为

"doTask"

# 表达式计算（IKExpression）

IKExpression自带函数

|  |  |  |  |
| --- | --- | --- | --- |
| 函数名 | 参数 | 返回值类型 | 说明 |
| $CONTAINS | String  String | Boolean | 返回指定部门、指定职务的人员ID的List集合 |
| $STARTSWITH | java.lang.String  java.lang.String | Boolean | 是否以指定字符串开头 |
| $ENDSWITH | java.lang.String  java.lang.String | Boolean | 是否以指定字符串结尾 |
| $CALCDATE | java.util.Date date  int years 年增加数  int months 月增加数  int days 日增加数  int hours 时增加数  int minutes 分增加数  int seconds 秒增加数 | java.util.Date | 返回计算后的日期 |
| $SYSDATE | 无 | java.util.Date | 返回当前日期 |
| $DAYEQUALS | java.util.Date  java.util.Date | Boolean | 两日期是否为同一天  只比较年月日 |

系统自定义函数

|  |  |  |  |
| --- | --- | --- | --- |
| 函数名 | 参数 | 返回值类型 | 说明 |
| $post | int deptId:部门ID  int postId:职务ID | List<String> | 返回指定部门、指定职务的人员ID的List集合 |
| $role | int roleId:角色ID | List<String> | 返回指定角色ID的所有人员ID |
| $deptRole | int roleId:角色ID  int deptId:部门 | List<String> | 返回指定部门、指定角色的所有人员ID |
| $ids | String ids:以逗号隔开的id | List<String> | 把以逗号隔开的字符串转化为List集合 |
| $join | List<?> list1:集合1  List<?> list2:集合2 | List<?> | 把两个集合合二为一 |
| $substringBefore | String string:要截取的字符串  String split：截取标记字符串 | String | 在给定的字符串中截取split之前的字符串 |
| $substring | String str:要截取的字符串  int start:开始Index  int end:结束Index | String | 截取指定开始与结束索引之间的字符串 |