

Multiple Approval

We are going to design a Approval flow which would require more than one person's approval. The approval process would require each and every one person of the approval group to approve. When there's one of them rejects, the whole approval process would be rejected immediately. And, as expected, each of them would need to approve in order to get it approved in overall.

 If you are running on Joget DX, kindly refer to the [Process Enhancement Plugin](#).

For the process design, we would require 2 separate workflow processes.

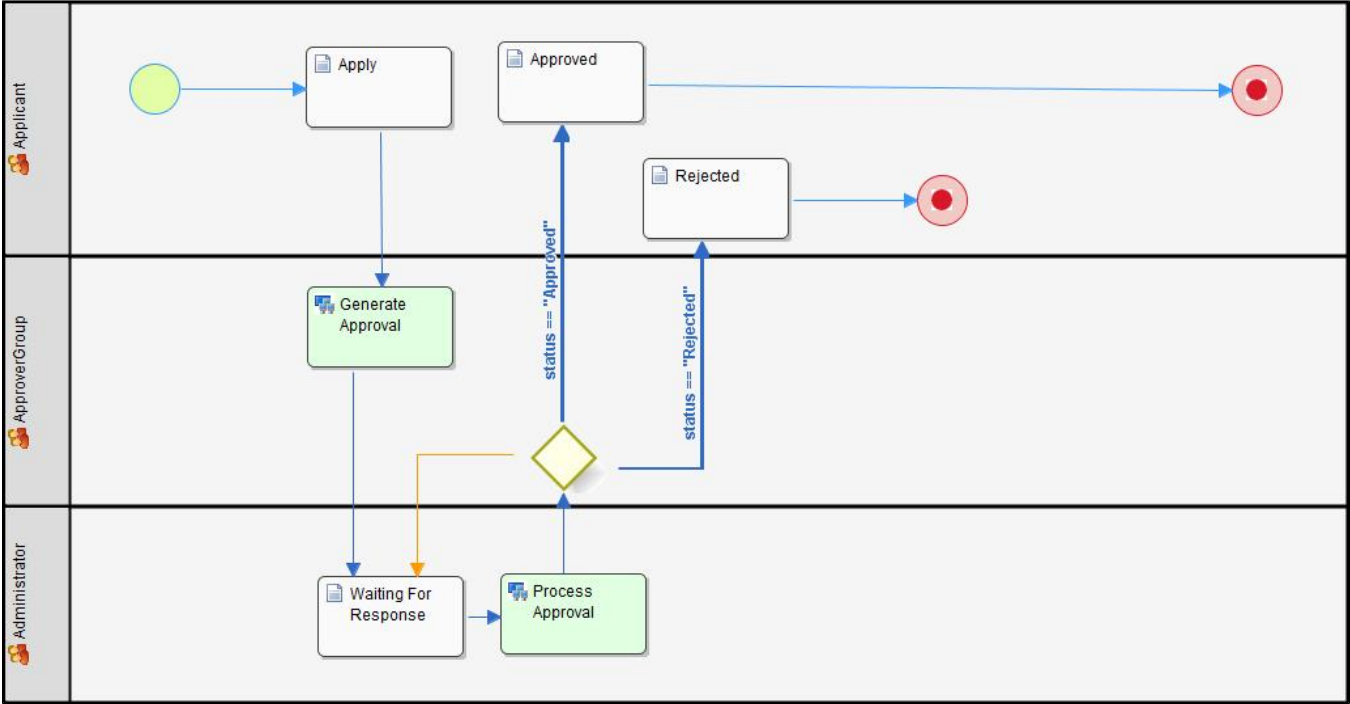


Figure 1: The main process - Apply Process

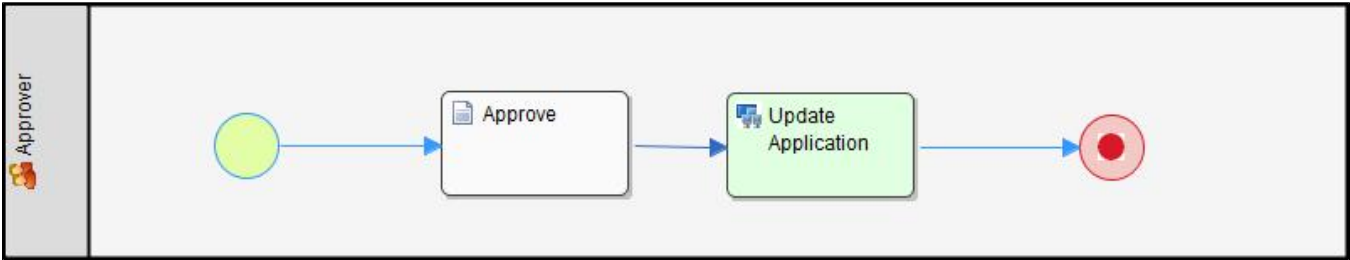


Figure 2: Individual Approval Process

Map Participants to Users

Map Activities to Forms

Map Tools to Plugins

Variable List

This is the list of participants defined in the Workflow design.

Applicant

ID : applicant

ApproverGroup

ID : approverGroup

Type	User
Value	✖ <u>clark</u> ✖ <u>cat</u> ✖ <u>admin</u>

Remove Mapping

Figure 3: Mapping of ApproverGroup

By design, one would need to kick start the "Apply" Process first and submits the "Apply" activity. Then, "Generate Approval" tool would spawn as many "Approval" Process as needed according to the number of users returned in "Approver Group" participant mapping.

Each of the "Approver" process instances would gather decision from its respectful Approver. Then, "Update Application" will execute to trigger "Waiting For Response" activity in the parent process ("Apply" Process) which will then execute the "Process Approval" tool to evaluate if it is enough to make a final decision and move forward.

Generate Approval

```

import org.joget.workflow.model.service.WorkflowManager;
import org.joget.apps.app.service.AppUtil;
import org.joget.apps.app.service.AppService;
import org.joget.workflow.model.WorkflowAssignment;
import org.joget.workflow.util.WorkflowUtil;
import java.util.Collection;
import java.util.HashMap;
import java.util.Map;
import org.joget.workflow.model.WorkflowProcess;
import org.joget.workflow.model.WorkflowProcessResult;

//constant value
String processDefKey = "approve";
String rowCountVariableName = "approvalCount";
String approverGroupParticipantId = "approverGroup";
String approvalIdsVariableName = "approvalIds";

//utility bean
WorkflowManager workflowManager = (WorkflowManager) AppUtil.getApplicationContext().getBean("workflowManager");
AppService appService = (AppService) AppUtil.getApplicationContext().getBean("appService");

//get processDefId
WorkflowProcess processDef = appService.getWorkflowProcessForApp(appDef.getId(), appDef.getVersion().toString(),
processDefKey);
String processDefId = processDef.getId();

//get foreign key
String processId = workflowAssignment.getProcessId();

int rowCount = 0;
Collection userList = null;
userList = WorkflowUtil.getAssignmentUsers(appDef.getAppId(), workflowAssignment.getProcessDefId(),
workflowAssignment.getProcessId(), workflowAssignment.getProcessVersion(), workflowAssignment.getActivityId(),
"", approverGroupParticipantId);

String approvalInstanceIds = "";
for(String user : userList){
    System.out.println(user);
    Map variables = new HashMap();
    variables.put("apply_id", processId);
    variables.put("approver", user);
    WorkflowProcessResult result = workflowManager.processStart(processDefId, null, variables, "admin",
null, false);
    approvalInstanceIds += result.getProcess().getInstanceId() + ",";
    rowCount++;
}

//set row count to workflow variable
workflowManager.processVariable(processId, rowCountVariableName, Integer.toString(rowCount));

//keep the list of approval instances
workflowManager.processVariable(processId, approvalIdsVariableName, approvalInstanceIds);

```

[Update Application](#)

```

import java.util.*;
import org.joget.apps.app.service.*;
import org.joget.workflow.model.*;
import org.joget.workflow.model.service.*;

//constant value
String processInstanceId = "#variable.apply_id#";
String activityDefId = "waitingForResponse";

//utility bean
WorkflowManager workflowManager = (WorkflowManager) AppUtil.getApplicationContext().getBean("workflowManager");
WorkflowUserManager workflowUserManager = (WorkflowUserManager) AppUtil.getApplicationContext().getBean
("workflowUserManager");

//get current user username and temporary set current user to roleAnonymous to get the assignment
String username = workflowUserManager.getCurrentUsername();
workflowUserManager.setCurrentThreadUser("roleAnonymous");

//get assignment
Collection assignments = workflowManager.getAssignmentList(null, null, processInstanceId, activityDefId, null,
null, null, 1);

if (assignments != null && !assignments.isEmpty()) {
    WorkflowAssignment ass = (WorkflowAssignment) assignments.iterator().next();
    String actId = ass.getActivityId();
    //accept and complete assignment
    workflowManager.assignmentAccept(actId);
    workflowManager.assignmentComplete(actId, null);
}

//set the current user back to original
workflowUserManager.setCurrentThreadUser(username);

```

```

import org.joget.apps.app.service.*;
import org.joget.apps.form.dao.*;
import org.joget.apps.form.model.*;
import org.joget.apps.form.service.*;
import org.joget.workflow.model.*;
import org.joget.workflow.model.service.*;

//constant value
String foreignKey = "customProperties.apply_id";
String formDefId = "approveForm";
String tableName = "multiApproval_approvals";
String rowCountVariableName = "approvalCount";
String statusVariableName = "status";
int approvalCount = Integer.parseInt("#variable.approvalCount#");
String approvalIds = "#variable.approvalIds#";

//utility bean
FormDataDao formDataDao = (FormDataDao) AppUtil.getApplicationContext().getBean("formDataDao");
WorkflowManager workflowManager = (WorkflowManager) AppUtil.getApplicationContext().getBean("workflowManager");
AppService appService = (AppService) AppUtil.getApplicationContext().getBean("appService");

//get foreign key
String processId = workflowAssignment.getProcessId();

//build condition
String condition = " WHERE " + foreignKey + " = ?";
Object[] paramsArray = new Object[]{processId};

//get approval data
FormRowSet rows = new FormRowSet();
rows = formDataDao.find(formDefId, tableName, condition, paramsArray, "dateCreated", false, null, null);

int rowCount = 0;
String status = "";
String recordId = "";
for (FormRow r : rows) {
    recordId = r.getId();
    String recordStatus = r.get("status");

    if(recordStatus.equalsIgnoreCase("Rejected")){
        status = "Rejected";
        break;
    }
    rowCount++;
}

if(status.equalsIgnoreCase("Rejected")){
    workflowManager.processVariable(processId, statusVariableName, "Rejected");
    //terminate any remaining approval instances
    String[] approvalIdsSplit = approvalIds.split(",");
    for(String approvalId : approvalIdsSplit){
        if(!approvalId.equalsIgnoreCase("") && !approvalId.equalsIgnoreCase(recordId)){
            try{
                workflowManager.processAbort(approvalId);
            }catch(Exception e){
            }
        }
    }
}
}else if(rowCount >= approvalCount){
    workflowManager.processVariable(processId, statusVariableName, "Approved");
}
}

```

Filter by App Search Process Id

☐	Process ID	Start Time	Process name	Requester	Process Version	State
☐	758_multiApproval_approve	Thu May 22 20:46:58 SGT	Approve	admin	5	open.running
☐	757_multiApproval_approve	Thu May 22 20:46:58 SGT	Approve	admin	5	open.running
☐	756_multiApproval_approve	Thu May 22 20:46:58 SGT	Approve	admin	5	open.running
☐	755_multiApproval_apply	Thu May 22 20:46:44 SGT	Apply	admin	5	open.running

Figure 4: View of instances after submitting the "Apply" activity with all the "Approve" process spawned

Multi Approval

Menu
Home
Apply
Inbox

Apply - Rejected

Approved

Application Details

NameAdmin Admin
ReasonTest Reason

Approval Trail

	Name	Status	Remark
1	Admin Admin	Approved	Test Approve 1
2	Cat	Approved	Test Approve 2
3	Clark	Rejected	Reject 1

Save As Draft
Complete

Print

Powered by Joget

Figure 5: Result of Apply process

KB:Download the App

Related Elements

- Designing a Process
- Bean Shell for Process Tool