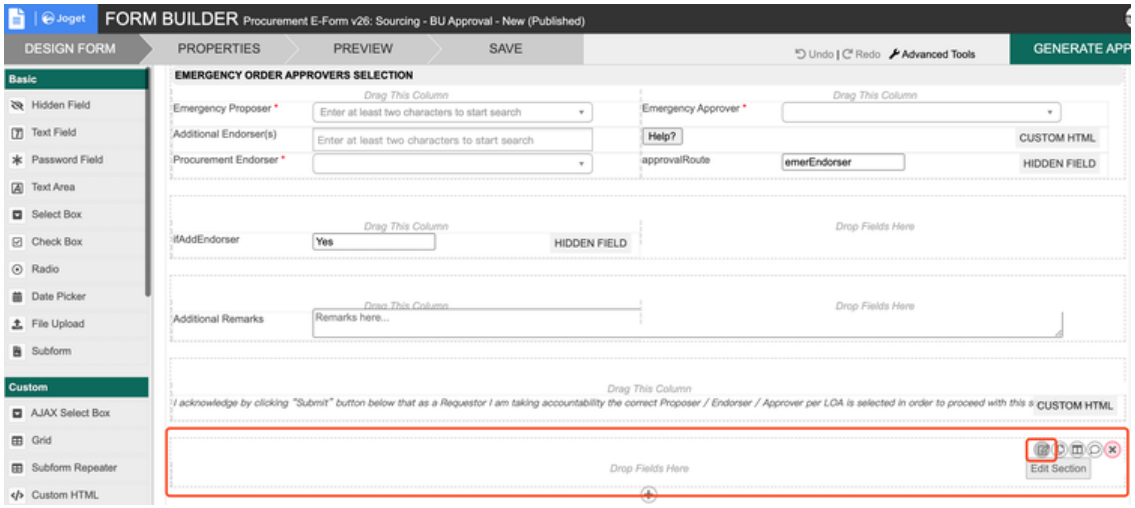
流程启动表单开启保存为草稿功能

背景介绍

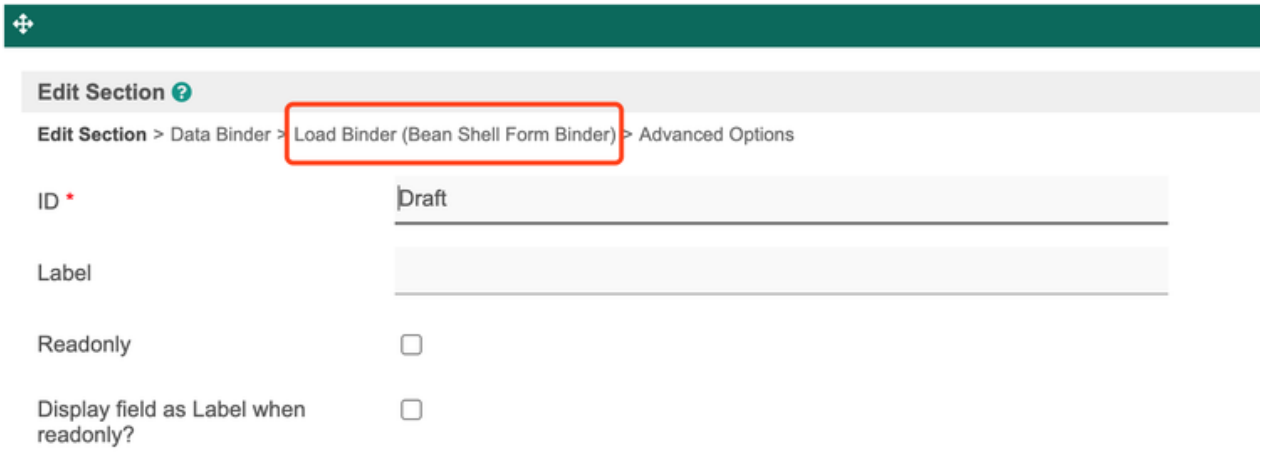
在默认的流程设计中，流程启动表单是没有保存为草稿的功能。 但是某些情况下，需要启动流程表单也保存为草稿。

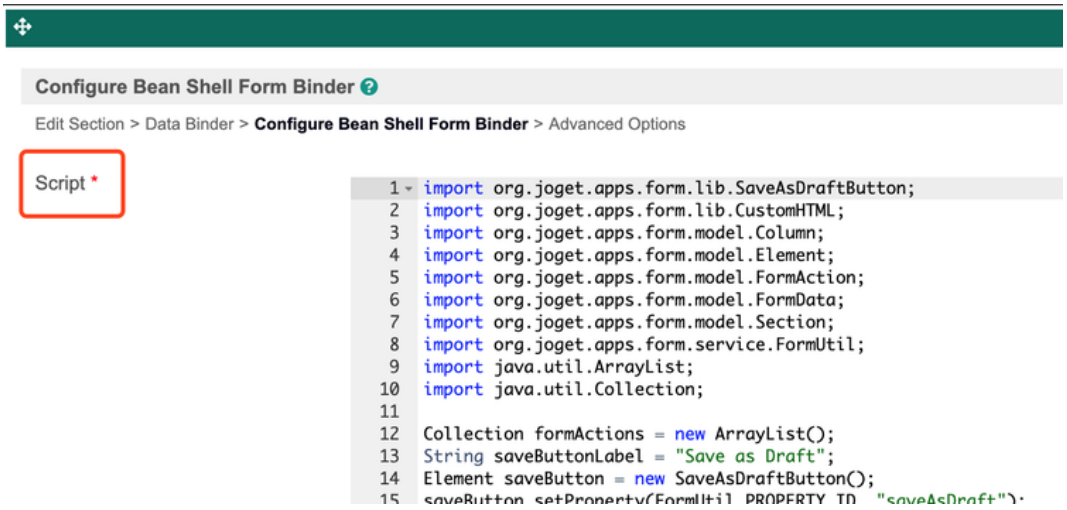
配置方式

1. 打开启动流程表单设计，并在表单最下方增加一个空的Section



2. 复制下方代码到分区的Load Binder中




Configure Bean Shell Form Binder
Edit Section > Data Binder > Configure Bean Shell Form Binder > Advanced Options
Script *
1 import org.joget.apps.form.lib.SaveAsDraftButton;
2 import org.joget.apps.form.lib.CustomHTML;
3 import org.joget.apps.form.model.Column;
4 import org.joget.apps.form.model.Element;
5 import org.joget.apps.form.model.FormAction;
6 import org.joget.apps.form.model.FormData;
7 import org.joget.apps.form.model.Section;
8 import org.joget.apps.form.service.FormUtil;
9 import java.util.ArrayList;
10 import java.util.Collection;
11
12 Collection formActions = new ArrayList();
13 String saveButtonLabel = "Save as Draft";
14 Element saveButton = new SaveAsDraftButton();
15 saveButton.setProperty(FormUtil.PROPERTY_ID, "saveAsDraft");

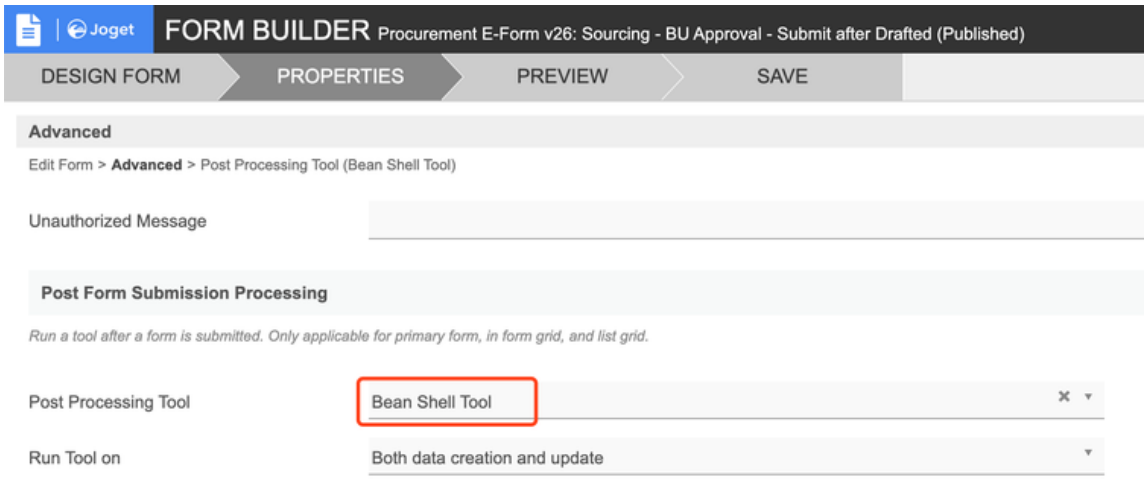
```
//
import org.joget.apps.form.lib.SaveAsDraftButton;
import org.joget.apps.form.lib.CustomHTML;
import org.joget.apps.form.model.Column;
import org.joget.apps.form.model.Element;
import org.joget.apps.form.model.FormAction;
import org.joget.apps.form.model.FormData;
import org.joget.apps.form.model.Section;
import org.joget.apps.form.service.FormUtil;
import java.util.ArrayList;
import java.util.Collection;

Collection formActions = new ArrayList();
String saveButtonLabel = "Save as Draft";
Element saveButton = new SaveAsDraftButton();
saveButton.setProperty(FormUtil.PROPERTY_ID, "saveAsDraft");
saveButton.setProperty("label", saveButtonLabel);
formActions.add(saveButton);
Section section = element;
ArrayList columns = (ArrayList) section.getChildren();
Column column = columns.get(0);
column.setProperty("horizontal", "true");
column.setChildren(formActions);
//add a custom html to fix the layout issue
Element html = new CustomHTML();
String script = "<script>$(document).ready(function(){";
script +=
"$(\"#"+section.getPropertyString("id")+"\").find(\".form-cell\").prependT
o(\"#section-actions .form-column\");";
script += "$(\"#"+section.getPropertyString("id")+"\").remove();";

//check whether it is save as draft and redirect to refresh after
submission
FormData fd = formData;
if (fd.getRequestParameter("saveAsDraft") != null) {
    script += "window.location.href=\'refresh\'";
}


script += "});</script>";
html.setProperty("id", "button_layout_fixes");
html.setProperty("label", "");
html.setProperty("value", script);

formActions.add(html);
return null;
```

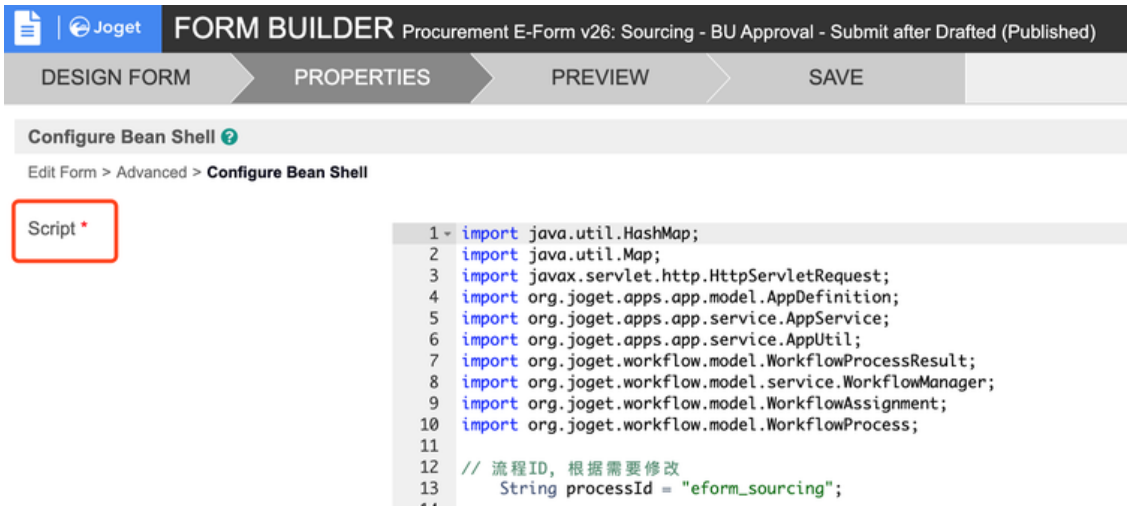
3. 保存表单，从前端启动流程即可看到保存为草稿按钮

非流程启动表单发起流程

1. 进入普通表单设计器，增加表单后续处理工具

> 在上方的保存为草稿表单中，由于设置了该表单为启动流程表单，所以【不能】作为发起流程表单。
>
> 如果需要【草稿表单】发起流程，则需要复制【发起流程表单】，新建【普通表单】发起流程





2. 使用以下代码，填入代码块中（注意其中的变量设置）

```
//
import java.util.HashMap;
import java.util.Map;
import javax.servlet.http.HttpServletRequest;
import org.joget.apps.app.model.AppDefinition;
import org.joget.apps.app.service.AppService;
import org.joget.apps.app.service.AppUtil;
import org.joget.workflow.model.WorkflowProcessResult;
import org.joget.workflow.model.service.WorkflowManager;
import org.joget.workflow.model.WorkflowAssignment;
import org.joget.workflow.model.WorkflowProcess;

 // ID,
    String processId = "eform_sourcing";

    String status = "#form.geg_sourcing.processStatus#";
    String requester = "#form.geg_sourcing.requestUser#";
    String endorser = "#form.geg_sourcing.endorser#";
    String approver = "#form.geg_sourcing.approver#";
    String refNo = "#form.geg_sourcing.sourcingID#";

    public Object execute(WorkflowAssignment assignment, AppDefinition
appDef, HttpServletRequest request) {
        AppService appService = (AppService)
AppUtil.getApplicationContext().getBean("appService");
        WorkflowManager workflowManager = (WorkflowManager)
AppUtil.getApplicationContext().getBean("workflowManager");
        //get process
        WorkflowProcess process =
appService.getWorkflowProcessForApp(appDef.getAppId(),
appDef.getVersion().toString(), processId);

        Map variables = new HashMap();
        variables.put("status",status);
        variables.put("requester",requester);
        variables.put("Endorser",endorser);
        variables.put("Approver",approver);
        variables.put("refNo",refNo);

        //start process
        WorkflowProcessResult result =
workflowManager.processStart(process.getId(), null, variables, null,
"#form.geg_sourcing.id#", false);
        System.out.println(result.getProcess().getId());
        return null;
    }

//call execute method with injected variable
   return execute(workflowAssignment, appDef, request);
```