Start

Definition

That is the first step of the workflow. At that step, you could define the fields you want to show when the end users will go to make a request.

Steps Tabs

Task details

In this tab you could configure next parameters:

- **Task name**: identified name for the task that will be created when the workflow is requested.
- **Permission request screen type**: this allows you to select how the permissions will be displayed on the screen. There are two available options:
  - **List of permissions**: this option needs to configure a user selector on the fields tab. When end-users request a process, first of all, they will select the user and the permissions, and then the permissions, the list of available permissions depends on the selected user.
  - **Self service request**: if you select the self-service request, it will not be mandatory to configure the user selector on the fields tab. That option can be configured to request permission for your own user, or to third users configuring the user selector. When end-users request a process, the available permissions will be displayed to select from the information system for the roles defined. When you select one or more roles, those will be added to the shopping cart to make the request.
- **Role selection filter**: this allows you to define a Script that returns the available roles to select. At the script window, you could find information about the available context variables.
- **Application selection filter**: this allows you to define a Script that returns the available applications to select. At the script window, you could find information about the available context variables.

Fields

In this tab, you could choose what fields the process form will show to the end users. You can choose these fields from all identity attributes, and from the attributes defined for the workflow on the Attributes Tab.

By default, only the Permissions field will be shown. That field is defined on the attributes tab. You can choose the fields you want to show when the end-users, add new fields, and delete the fields that do not need to generate a task. Also, you can sort the fields, you only need to drag and drop on the Order column.

For each field, you may indicate if it is a readOnly field, and you may add a Validation script and Visibility script. The validation script allows you to define rules, the field has to comply with these rules. The visibility script allows you to define the rules to show or hide a field.

Validation examples

```
if (value == null || value.equals(""))
  throw new Exception("The userName is mandatory");
else
  return true;
```

It is also allowed in the following manner:

```
if (value == null || value.equals(""))
  return ("The userName is mandatory");
else
  return true;
```

Validate that a certain field is not repeated:

```
userList = serviceLocator.getUserService().findUserByJsonQuery("attributes.field_XX eq \"" + value +"\"");
if (!userList.isEmpty() {
  return "the field field_XX is associated to another user";
}
return true;
```

Visibility example

```
user = serviceLocator.getUserService().getCurrentUser();
if ("admin".equals(user.userName))
```

```
return false;
```

Triggers

On the trigger tab, you could define different triggers using custom scripts. Those triggers will be launched with the events you will define.

- **onLoad**: you can use that trigger to perform some actions before the execution of the step.
- **on PrepareTransition**: you can use that trigger to perform some actions after the execution of the step and before starting a transition to another step.
- **onChange**: you can use that trigger to perform some actions when the value of the attribute is changed. You could choose the field from a list.

Example

1. Calculate the email when firstName or lastName changes and depending on the userType:

```
firstName   = (inputFields.get("firstName")!=null) ? inputFields.get("firstName").value : null;
lastName    = (inputFields.get("lastName")!=null) ? inputFields.get("lastName").value : null;
userType    = (inputFields.get("userType")!=null) ? inputFields.get("userType").value : null;

if (firstName!=null && !firstName.trim().isEmpty() &&
    lastName!=null && !lastName.trim().isEmpty() &&
    userType!=null && !userType.trim().isEmpty()) {

  emailAddress = firstName + "." + lastName;
   if ("E".equals(userType)) {
   emailAddress = emailAddress + ".ext@soffid.com";
   } else {
     emailAddress = emailAddress + "@soffid.com";
   }
  inputFields.get("emailAddress").value = emailAddress;

}
```

> You can find more information about StandardUserWindow.java on Github.

2. Load the user data into the form.

```
user = serviceLocator.getUserService().getCurrentUser();
task.getVariables().put("action", "M");
task.getVariables().put("userSelector", user.userName);
workflowWindow.fetchUserAttributes()
```

Incoming transitions

This process type does not have task details for the start step.

Outgoing transitions

The Outcoming transition tab displays the next steps where the flow can go from the current step. When you create a process from a template or from scratch default outcoming transitions are defined. It is allowed to customize the default setup, add new transitions, or delete transitions.

- **From**: current step.
- **Incoming transition**: name of the transition.
- **To**: the next step, where the flow goes.
- **Action**: allows creating a custom script to perform specific actions.

When you create an outcoming transition, Soffid creates the proper incoming transition.

Example

Validation of mandatory fields:

```
a = executionContext.getVariable("firstName");
if (a==null || "".equals(a.trim()))
  throw new Exception("First name is mandatory");

a = executionContext.getVariable("lastName");
if (a==null || "".equals(a.trim()))
  throw new Exception("Last name is mandatory");

a = executionContext.getVariable("primaryGroup");
if (a==null || "".equals(a.trim()))
  throw new Exception("Primery group is mandatory");

return true;
```

To request the process is only allowed for Internal users:

```
userSelector = executionContext.getVariable("userSelector");
user = serviceLocator.getUserService().findUserByUserName(userSelector);
if (user.userType.equals("I") || user.userType.equals("S")) {
throw new Exception ("To request the process is only allowed for Internal users");
}
```

Revision #17
Created 15 June 2021 13:33:52 by pgarcia@soffid.com
Updated 30 October 2023 10:00:34 by pgarcia@soffid.com