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. In that case, the request will be launched automatically when the end users request to use a privileged account to connect to a protected resource.

Administrator users can define on XACML Policy Management page the rules to request the use of some privileged accounts.

Steps Tabs

Task details

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

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 fields defined on the attributes tab will be shown. 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

```
account = (inputFields.get("account")!=null) ? inputFields.get("account").value : null;
systemName = (inputFields.get("systemName")!=null) ? inputFields.get("systemName").value : null;
.....
```

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

```
accounts = serviceLocator.getAccountService().findAccountByJsonQuery("name eq \"" +
executionContext.getVariable("account") + "\"");
if (!accounts.isEmpty() {
for (account:accounts) {
owners = serviceLocator.getAccountService().getAccountUsers(account);
// TO-DO
}
}
```

Revision #5
Created 29 June 2022 12:35:24
Updated 18 June 2024 13:38:01