Detect duplicated user

Definition

That step is used to define the proper rules to determine the potential conflicts between the identity for who is the request, and the Soffid existing identities. Whit that definition, Soffid will find the potential conflicts, and the end-user could select the best option to solve those (merge or create a new one).

Steps Tabs

Tasks details

- **Task name**: identified name for the task that will be created. For instance: Check duplicates for #{firstName} #{lastName}
- **Actor(s) expression**: write an expression to identify the actor depending on the requested role. One can use EL expressions based on role and application attributes. For instance: SOFFID_MANAGER/${primaryGroup}
- **Assignment script**: alternatively, write a Beanshell script to return the actor depending on the process variables. For instance: return primaryGroup.attributes{"owner"};
- **Weight threshold**: in the tab "User queries", you can define many different queries to search for similar users, and each query has a weight. If a user is found in one or more queries, the weight of each one of these queries are added. If the total weight is equal to or greater than the current threshold, the user is considered a user match.

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, all the identity attributes will be shown. You can choose the fields you want to show, 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.

User queries

This tab is only available when one of the below Step types is **Detect duplicated user**.

User queries allow you to customize a SCIM or Text query to detect duplicated users. You may define a weight for each query. If a user is found in one or more queries, the weight of each one of these queries are added. If the total weight is equal to or greater than the current weight threshold (defined on the Task details tab), the user is considered a user match.

Examples

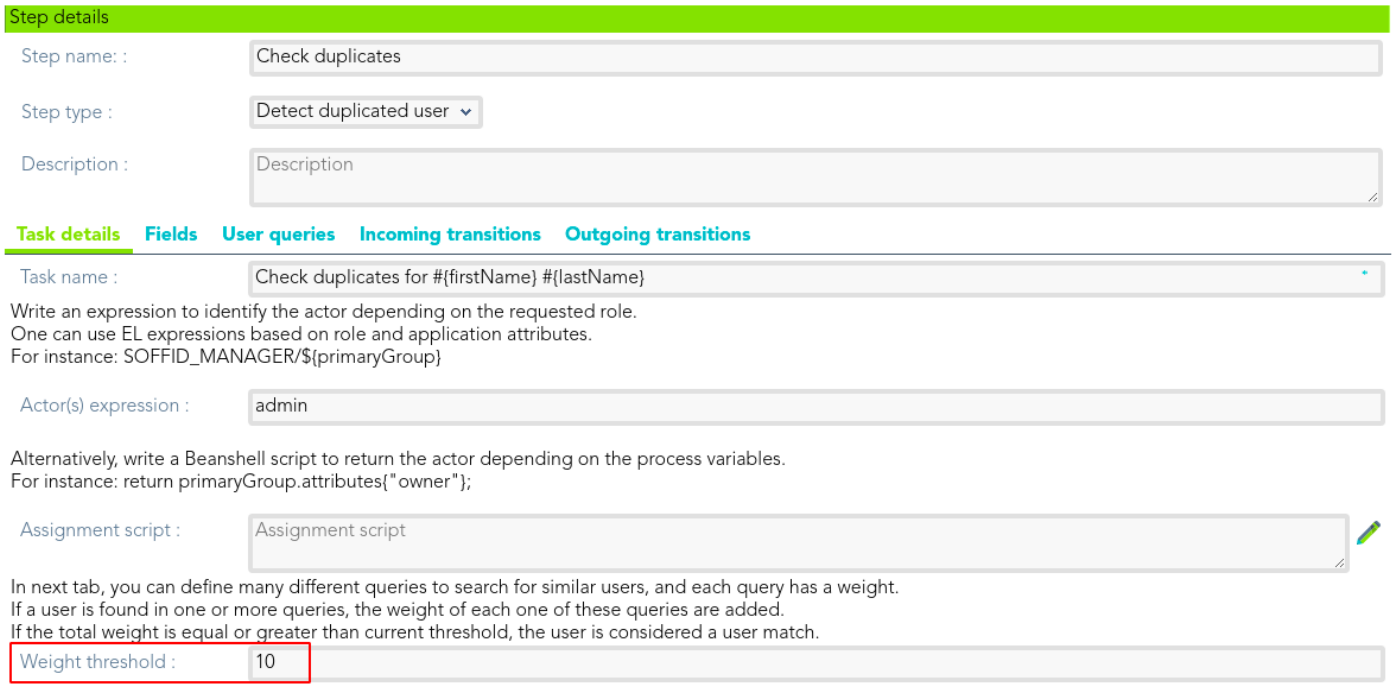
Text Query

```
${lastName}
```

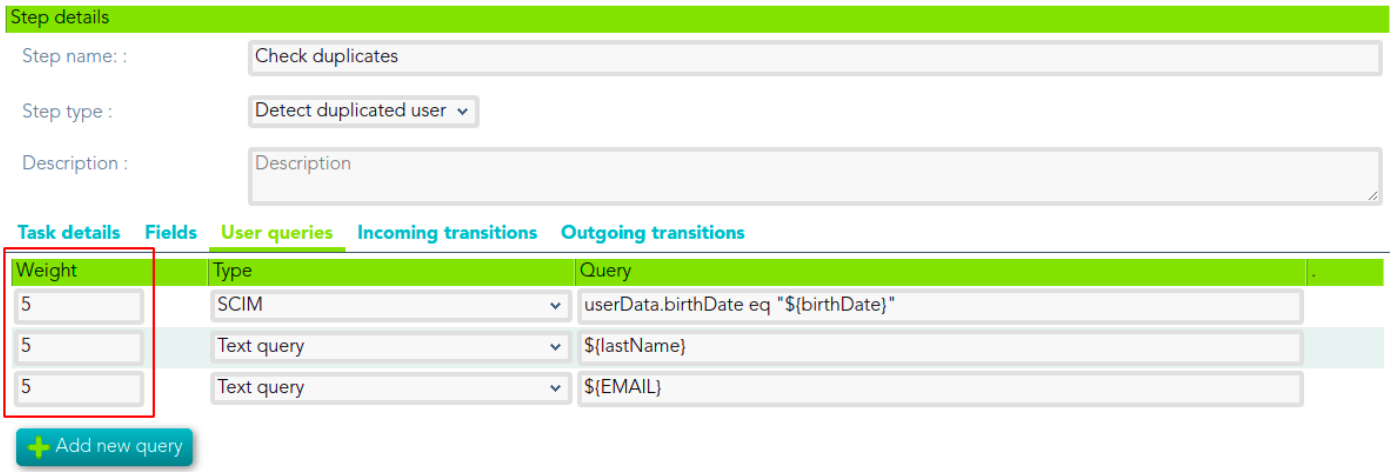
SCIM Query

```
attributes.birthDate eq "${birthDate}"
```

Define the weight threshold on the Task detail tab



Define the weight for each query on the User query tab: A user is considered duplicated when at least two queries are true.



Incoming transitions

The Incoming transitions tab displays the previous steps where the flow comes from. When you create a process from a template or from scratch default incoming transitions are defined. It is allowed to customize the default setup, add new transitions, or delete transitions.

- **From**: the previous step, where the flow comes. Allows you to select where the workflow comes from.
- **Incoming transition**: brief name to identify the transition. That is the name of the action the form will show to the final user.
- **To**: current step.
- **Action**: allows creating a custom script to perform specific actions.

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

Example

The incoming script action is the same outgoing script action of the previous step.

```
selector = executionContext.getVariable("userSelector");
user = serviceLocator.getUserService().findUserByUserName(selector);
executionContext.setVariable("testName", user.firstName);
executionContext.setVariable("testOperation", "CHECK");
```

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

Add comments to the task:

```
executionContext.getToken().addComment("Automatic comments.....");
```

Revision #13
Created 4 June 2021 06:41:58
Updated 24 November 2022 12:05:45