Incoming triggers examples

Load triggers

When you are configuring an agent, depending on the connector type, it will be able to define BeanShell scripts that will be triggered when data is loaded into Soffid, **incoming triggers**.

Triggers can be used to validate or perform a specific action just before performing an operation or just after performing an operation into Soffid objects.

The trigger result will be a boolean value, true to continue or false to stop.

Use case examples

Example 1

A user can have more than one account on a target system. We want to reconcile only those accounts which have the same name on Soffid and on target system.

```
name = newObject{"accountName"};
uaList = serviceLocator.getAccountService().findUsersAccounts(name,"agentName");

for(userAccount: uaList) {
  if (userAccount.name.equals(name)) {
    return true;
  } else  {
    return false;
  }
}

return false;
```

Example 2

Update only users who belong to Soffid company.

```
name = newObject{"accountName"};
user = serviceLocator.getUserService().findUserByUserName(name);

if (user != null) {
  attributes = serviceLocator.getUserService().findUserAttributes(name);
  company = attributes.get("company");
  if (company != null && company.equals("Soffid")) {
  return true;
}
return false;
```

Example 3

Discard to create some accounts (PreInsert account).

```
log.info("********** Pre Insert Account");
cuentas = new java.util.HashMap();
cuentas.put("admin",null);
account = newObject{"accountName"};

if (cuentas.containsKey(account)) {
  log.info("TRIGGER ACCOUNT PREINSERT - Discarded to create the account " + account);
  return false;
}

log.info("TRIGGER ACCOUNT PREINSERT - Correct account " + account);
return true;
```

Example 4

If does not exist a mail domain, it wil be created.

```
mailDomain = newObject{"mailDomain"};
if (mailDomain != void && mailDomain != null) {
  existMD = serviceLocator.getMailListsService().findMailDomainByName(mailDomain);
  if (existMD == null) {
    newMailDomain = new com.soffid.iam.api.MailDomain();
    newMailDomain.setName(mailDomain);
    newMailDomain.setDescription(mailDomain);
```

```
    serviceLocator.getMailListsService().create(newMailDomain);
  }
}
return true;
```

Example 5

Avoid deleting users (PreDelete user).

```
return false;
```

Example 6

If throw the exception, the return will be is false. True in other cases.

```
if (....)
  throw new Exception("Exception message....");
return true;
```

Example 7

The new group on the target system will have the same id that the old group.

```
newObject{"idGroup"} = oldObject{"idGroup"};
.....
```

Example 8

Get the attribute company option 1:

```
company = source{"attributes"}{"company"};
```

Get the attribute company option 2

```
userName = source{"userName"};
attributes = serviceLocator.getUserService().findUserAttributes(userName);
company = attributes.get("company");
```

Example 9

Update the company attribute:

```
userName = newObject{"userName"};
// Check if the user exists
user = serviceLocator.getUserService().findUserByUserName(userName);
if (user == null) {
return false;
}
log.info("****************** USER Object: " + user);

attributes = serviceLocator.getUserService().findUserAttributes(userName);
if (attributes == null) {
attributes = new HashMap();
}

attributes.put("company", "<COMPANY_NAME>");
serviceLocator.getUserService().updateUserAttributes(userName, attributes);

return true;
```

Example 10

Check user type

```
userName = newObject{"userName"};
user = serviceLocator.getUserService().findUserByUserName(userName);

if (user.userType.equals("I")){
  .....
  //TODO
} else {
    .....
  //TODO
}
return true;
```

Revision #16
Created 10 August 2021 09:32:38
Updated 29 April 2024 14:21:41