Soffid Objects

You can consult the list of Soffid attributes:

1. User Object
2. Account Object
3. Group Object
4. Role Object
5. Grant Object
6. Maillist Object
7. Membership Object
8. dispatcherService
9. Authoritative change object

User object

A user objects are maps that hold the information belonging to a single user account.

Attribute	Type	Description
id	Long	user id
accountId	Long	account id
accountName	String	account name
system	String	managed system (agent) name
accountDescription	String	account description
active	Boolean	true if user is active
accountDisabled	Boolean	true if account is diabled

Attribute	Type	Description
mailAlias	String	blank separated mails
userName	String	user name
primaryGroup	String	user's primary group name
comments	String	user's comments
createdOn	Date	user creation date
modifiedOn	Date	user last modification date
mailDomain	Date	user mail domain (email right side of @)
fullName	String	user full name
shortName	String	user mail name (email left side of @)
firstName	String	user first name
lastName	String	user last name
lastName2	String	user second last name (when applicable)
mailServer	String	mail server host name
homeServer	String	home drive server host name
profileServer	String	roaming profile server host name
phone	String	user's phone number
userType	String	user type
createdBy	String	user name creator of this user
modifiedBy	String	user name modifier of this user
secondaryGroups	List<Map<String,Object>>	list of groups the user belongs to, including primary group The attributes of the inner map are described later

Attribute	Type	Description
attributes	Map<String,String>	additional user attributes
grantedRoles	List<Map<String,Object>>	list of grants directly granted to the user
allGrantedRoles	List<Map<String,Object>>	list of grants directly on indirectly granted to the user
granted	List<String>	list of role names and group names directly granted to the user
allGranted	List<String>	list of role names and group names directly or indirectly granted to the user

Account object

An account object holds the information belonging to an account.

Attribute	Type	Description
accountDescription	String	account description
accountDisabled	Boolean	true if account is diabled
accountId	Long	account id
accountName	String	account name
allGranted	List<String>	list of role names directly or indirectly granted to the user
allGrantedRoles	List<Map<String,Object>>	list of grants directly on indirectly granted to the user
attributes	Map<String,String>	additional account attributes
granted	List<String>	list of role names directly granted to the user
grantedRoles	List<Map<String,Object>>	list of grants directly granted to the user
lastLogin	Calendar	lastLogin
lastPasswordUpdate	Calendar	lastPasswordUpdate
lastUpdate	Calendar	lastUpdate
passwordExpiration	Calendar	passwordExpiration

Attribute	Type	Description
passwordPolicy	String	password policy
system	String	managed system (agent) name
type	AccountType	"U"=user, "S"=shared, "P"=privileged, "I"=ignored

Group object

An group object holds the information belonging to a group.

Attribute	Type	Description
groupId	Long	group id
name	String	group name
description	String	group description
parent	String	parent group name
server	String	home server host name
disabled	boolean	true if the group is disabled
accountingGroup	String	group accounting information
type	String	group type
driveLetter	String	home server letter to connect to
users	List<Map<String,Object>>	list of users belonging to this group
userNames	List<String>	list of user names belonging to this group
allUsers	List<Map<String,Object>>	list of users directly or indirectly belonging to this group
allUserNames	List<String>	list of user names either directly or indirectly grantee of this role
grantedRoles	List<Map<String,Object>>	list of roles granted to this group

Attribute	Type	Description
grantedRoleNames	List<String>	list of role names granted to this group

Role object

An role object holds the information belonging to a role.

Attribute	Type	Description
roleId	Long	role id
system	String	managed system (agent) name
name	String	role name
application	String	application system name
category	String	role category
passwordProtected	boolean	true if role should be password protected (where applicable)
description	String	Role description
wfmanaged	boolean	true if role should be displayed in self service requests
domain	String	custom domain for this role: Use com.soffid.iam.api.DomainType constants or configured custom domain
ownedRoles	List<Map<String,Object>>	list of roles granted to this one
ownerRoles	List<Map<String,Object>>	list of roles grantee of this one
ownerGroups	List<Map<String,Object>>	list of groups grantee of this role
grantedAccountNames	List<String>	list of account names directly grantee of this role
grantedAccounts	List<Map<String,Object>>	list of users directly grantee of this role
allGrantedAccountNames	List<String>	list of account names either directly or indirectly grantee of this role
allGrantedAccounts	List<Map<String,Object>>	list of users either directly or indirectly grantee of this role
attributes	Map<String,Object>	role's custom attributes

Grant object

Grant, grantedRole & allGrantedRoles

The objects grant, grantedRole and allGrantedRoles are used to assing roles to accounts and roles.

Attribute	Type	Description
domainValue	String	grant value (if any)
grantedRole	String	granted role name
grantedRoleId	Long	granted role id
grantedRoleObject	role object	granted role
grantedRoleSystem	String	granted role managed system (agent) name
id	Long	grant id
ownerAccount	String	grantee account name
ownerAccountObject	account object	grantee account
ownerGroup	String	grantee group name
ownerRoleId	String	grantee role id
ownerRoleName	String	grantee role name
ownerSystem	String	grantee account or role managed system name
ownerUser	String	grantee user name

> By default, use the **grantedRole** object

> The **grant** object adds the roles and the groups, use it only in this case

Examples

Grant

Example to map a grant object (assign a role to an account):

System attribute	Direction	Soffid attribute

role_name	=>	grantedRole
account_name	=>	ownerAccount

GrantedRole

Example to map a grantedRole object (assign a role as a child of another role):

System attribute	Direction	Soffid attribute
role_name	=>	grantedRole
parent_role_name	=>	ownerRoleName

AllGrantedRoles

Example to map a allGrantedRoles object in a holderGroup (assign a role to an account in a specific group):

System attribute	Direction	Soffid attribute
role_name	=>	grantedRole
parent_role_name	=>	ownerRoleName
group_code	=>	domainValue
group_code	=>	holderGroup
userName	=>	ownerUser

Maillist object

Attribute	Type	Description
id	Long	internal mail list id
name	String	mail list name (the initial part, before the @ sign)
domain	String	mail list domain (the remaining part after the @ sign)
system	String	managed system (agent) name

Attribute	Type	Description
description	String	mail list description
users	String array	user names that are bound to this mail list
groups	String array	group names thta are subscribed to this mai list
roles	String array	role names that grant access to this mail list
lists	String array	Nested mail lists
explodedUsers	String array	Names of the users that should be subscribed to this mail list, including the users that should be subscribed due to group or role membership
explodedUserAddresses	String array	Mail addresses of any exploded User

Membership object

A membership object contains the user account information as well as the group the user belongs to.

Attribute	Type	Description
userName	String	User name
user	Map<String,Object>	user object
groupName	String	Group name
group	Map<String,Object>	group object
attributes	Map<String,Object>	Membership custom attributes

dispatcherService

dispatcherService is an object available from agents' attribute translation rules.

This object contains four methods:

method name	parameters	result type	comments

soffidToSystem	ExtensibleObject soffidObject	ExtensibleObject	Uses attribute translation tables to transform a soffid object to a target system object. Mind to fill-in objectType property to use the proper object mapping
systemToSoffid	ExtensibleObject systemObject	ExtensibleObject	Uses attribute translation tables to transform a target system object to a Soffid object. Mind to fill-in objectType property to use the proper object mapping
search	ExtensibleObject exampleObject	ExtensibleObject	Uses the exampleObject to perform a query by example on the target system. If the object exists on the target system, it is returned. Mind to fill-in objectType property with the desired system object type
invoke	String verb String action Map parameters	List of Map	This method allows arbitrary executions on the target system, but it semantics can change depending on the connector used. For instance, it can be used to perform a GET on the target system in REST connector, can issue an LDAP query on ActiveDirectory connector, can execute a SELECT sentence on a SQL connector, or can execute an operating system command in Shell connector. The results are returned as a list of objects (map).

Examples

Snippet to query the sys_id attribute for a grant owner

```
System.out.println("Searching id for "+ownerRoleName);
com.soffid.iam.sync.intf.ExtensibleObject eo = new com.soffid.iam.sync.intf.ExtensibleObject();
```

```
eo.setObjectType("ROLE");
eo{"name"} = ownerRoleName;
eo = dispatcherService.search(eo);
System.out.println("FOUND "+eo{"sys_id"});
return eo{"sys_id"};
```

Snippet that performs a REST query to get group to role assignments in ServiceNow

```
list = dispatcherService.invoke ("GET",
  "https://arxusdev.service-
now.com/api/now/table/sys_group_has_role?sysparm_exclude_reference_link=true&amp;sysparm_display_value
=all&amp;sysparm_fields=role%2Cgroup&amp;sysparm_query=group="+sys_id,
  null).
  get(0).get("result")

r = new java.util.LinkedList();
for ( d: list)
{
  grant = new java.util.HashMap();
  grant{"grantedRole"} = d.get("role").get("display_value");
  grant{"grantedRoleSystem"} = "ServiceNow";
  grant{"ownerRoleName"} = name;
  grant{"ownerSystem"} = "ServiceNow";
  r.add  (grant);
}
return r;
```

Snippet of invoke usage on a relational database

```
// Table ITREPRT
role = source{"granted"}.size() == 0 ? "" : source{"granted"}.get(0);
System.out.println ("********** ROLE "+role);
args = new java.util.HashMap();
args.put("user", source{"accountName"}.toUpperCase());
if (role.equals ("Receptores PR") || role.equals("Jefes_Personal")) {
  r = dispatcherService.invoke("select", "* from ITREPRT where IDUSER=:user", args);
  if (r.size() == 0) {
    dispatcherService.invoke("insert", "into ITREPRT(IDUSER,NOMECO) values (:user, 1)", args);
  }
```

```
} else {
  dispatcherService.invoke("delete", "from ITREPRT where IDUSER=:user", args);
}
// TABLE MRGEUCT
cc = source{"attributes"}{"dominio"};
if ( source{"userType"} .equals ("T")) {
  cc = source{"userName"}.substring(1);
}
while (cc != null && cc.startsWith("0")) cc = cc.substring(1);
System.out.println ("********** COST CENTER "+cc);
if (cc != null && ! cc.trim().isEmpty())
{
  args = new java.util.HashMap();
  args.put("user", source{"accountName"}.toUpperCase());
  args.put("cc", cc);
  r = dispatcherService.invoke("SELECT", "* from MRGEUCT where IDUSER=:user and MOARPR=:cc", args);
  if (r.size() == 0) {
    dispatcherService.invoke("INSERT", "into MRGEUCT(MOARPR,CENTRA, IDUSER, NOTIFI ) "+
                        "values ('II', :cc, :user, 'S')", args);
    dispatcherService.invoke("INSERT", "into MRGEUCT(MOARPR,CENTRA, IDUSER, NOTIFI ) "+
                        "values ('BM', :cc, :user, 'S')", args);
    dispatcherService.invoke("DELETE", "FROM MRGEUCT WHERE CENTRA!=:cc AND IDUSER=:user", args);
  }
}
return true;
```

Snippet of invoke usage on a Active Directory I

```
hashMap = new java.util.HashMap();
list = serviceLocator.getDispatcherService().invoke("AD soffid.pat",
                                        "select",
                                        "(&(objectClass=user))",
                                        hashMap);
out.println("** list.size -- " + list.size());
```

Snippet of invoke usage on a Active Directory II

```
ACC = source{"accountName"};
la = dispatcherService.invoke("AD soffid.pat", "(&(objectClass=user)(sAMAccountName=userName))", new
java.util.HashMap());
```

Authoritative change object

A user objects are maps that hold the information belonging to a single user account

Attribute	Type	Description
id	Long	user id
accountId	Long	account id
accountName	String	account name
system	String	managed system (agent) name
accountDescription	String	account description
active	Boolean	true if user is active
accountDisabled	Boolean	true if account is diabled
mailAlias	String	blank separated mails
userName	String	user name
primaryGroup	String	user's primary group name
comments	String	user's comments
createdOn	Date	user creation date
modifiedOn	Date	user last modification date
mailDomain	Date	user mail domain (email right side of @)
fullName	String	user full name
shortName	String	user mail name (email left side of @)
firstName	String	user first name
lastName	String	user last name
lastName2	String	user second last name (when applicable)
mailServer	String	mail server host name

Attribute	Type	Description
homeServer	String	home drive server host name
profileServer	String	roaming profile server host name
phone	String	user's phone number
userType	String	user type
createdBy	String	user name creator of this user
modifiedBy	String	user name modifier of this user
secondaryGroups	List<Map<String,Object>>	list of groups the user belongs to, including primary group The attributes of the inner map are described in the link
secondariGroups2	List<Map<String,Object>>	list of user memberships, excluding primary group The attributes of the inner map are described link
attributes	Map<String,String>	additional user attributes
grantedRoles	List<Map<String,Object>>	list of grants directly granted to the user
allGrantedRoles	List<Map<String,Object>>	list of grants directly on indirectly granted to the user
granted	List<String>	list of role names and group names directly granted to the user
allGranted	List<String>	list of role names and group names directly or indirectly granted to the user

Revision #19
Created 14 April 2021 13:50:40
Updated 6 February 2026 17:58:22 by Sion Vives