



RSA IMG Connector Datasheet for SCIM

This data sheet provides the configuration information required to create a new SCIM connector.

Supported Software

RSA IMG Version: 6.8.1 and above

End point Application: SCIM Based Server (e.g. Unbound ID SCIM server)

Prerequisites

Unbound ID SCIM server set-up

1. Download Unbound Id Reference Server Implementation from <https://www.unboundid.com/labs/projects/system-for-cross-domain-identity-management/docs/scim-sdk-docs>
2. Run the server as per the guidelines given in its readme.txt
3. Once server is installed successfully, it will be accessible at URL: <http://localhost:8080>

Users and Groups can be viewed on <http://localhost:8080/Users> and <http://localhost:8080/Groups> respectively

Configuration

General:

Field Name	Value
Name	SCIM Connector
Description	SCIM Connector

Server	AFX Server
Connector Template	<i>SCIM</i>
State	Active
Export As Template	Yes

Settings:

Connection Parameters	Description
InstanceURL	Base URL of the SCIM instance
SCIM version	SCIM API version. For example, v1
Admin UserName	User Name of User having administrative privileges
Admin password	Password of User having administrative privileges

Capabilities

Commands

Following commands are supported by RSA IMG SCIM Connector:

1. CreateAccount
2. DeleteAccount
3. ResetPassword
4. EnableAccount
5. DisableAccount

6. AddAccountToGroup
7. RemoveAccountFromGroup
8. UpdateAccount
9. CreateGroup
10. DeleteGroup
11. AddGroupToGroup
12. RemoveGroupFromGroup

Command Input Parameters

Go to 'Commands' tab

CreateAccount

- Map Account to \${User.Name}
- Map Password to \${AccountTemplate.Password}
- Map Owner to \${AccountTemplate.Owner}
- Map Person Party Id to \${User.Uniqueld}
- Map Description to \${AccountTemplate.Description}
- Map Email to \${User.Email}
- Map Fax to \${AccountTemplate.Fax}

DeleteAccount

- Map UserName to \${Account.Name} OR Map UserId to \${Account.ExternalId}

DisableAccount

- Map UserName to \${Account.Name} OR Map UserId to \${Account.ExternalId}

EnableAccount

- Map UserName to \${Account.Name} OR Map UserId to \${Account.ExternalId}

ResetPassword

- Map UserName to \${Account.Name} OR Map UserId to \${Account.ExternalId}
- Map New password to \${AccountPassword}

UpdateAccount

- Map UserName to \${Account.Name} OR Map UserId to \${Account.ExternalId}
- Map FormatedName to the full name, including all middle names, titles, and suffixes appropriately formatted for display

- Map GivenName to \${User.First_Name}
- Map FamilyName to \${User.Last_Name}
- Map UserType to \${User.User_Type} to identify the organization to user relationship.
- Map Email to \${User.Email_Address}

CreateGroup

- Map GroupName to \${Group.Name}

DeleteGroup

- Map GroupName to \${Group.Name} OR Map GroupId to \${Group.ExternalId}

AddAccountToGroup

- Map UserName to \${Account.Name} OR Map UserId to \${Account.ExternalId}
- Map GroupName to \${Group.Name} OR Map GroupId to \${Group.ExternalId}

RemoveAccountFromGroup

- Map UserName to \${Account.Name} OR
- Map UserId to \${Account.ExternalId}
- Map GroupName to \${Group.Name} OR Map GroupId to \${Group.ExternalId}

AddGroupToGroup

- Map ParentGroupName to \${Group.Name} OR
- Map GroupId to \${Group.ExternalId}
- Map GroupName to \${Group.Name} OR Map GroupId to \${Group.ExternalId}

RemoveGroupFromGroup

- Map ParentGroupName to \${Group.Name} OR
- Map GroupId to \${Group.ExternalId}
- Map GroupName to \${Group.Name} OR
- Map GroupId to \${Group.ExternalId}

Limitations

1. DeleteAccount / ResetPassword / AddAccountToGroup /RemoveAccountFromGroup:
 - For all these commands, UserId (Service Provider's side) is required to perform the operation.
 - As providing UserId for these operations is not preferable, the workaround is to provide UserName and using it, UserId of that User can be fetched (only if the implementer supports "filter"). But, if the implementer doesn't support it, then UserId has to be provided for these commands to work.
 - Since either UserID or UserName can be used at a time, you need to edit the attributes before creating the connector.

2. DeleteGroup / AddGroupToGroup / RemoveGroupFromGroup:

- For all these commands, GroupId (Service Provider's side) is required to perform the operation.
- As providing GroupId for these operations is not preferable, the workaround is to provide GroupName and using it, GroupId of that Group can be fetched (only if the implementer supports "filter"). But, if the implementer doesn't support it, then GroupId has to be provided for these commands to work.
- Since either GroupId or GroupName can be used at a time, you need to edit the attributes before creating the connector.

3. EnableAccount / DisableAccount:

- According to the SCIM specification, "Active" attribute is a Boolean value indicating the User's administrative status.
- The definitive meaning of this attribute is determined by the SCIM implementer. Though a value of true means User is, for example, able to login; while a value of false implies that the user's account has been suspended. So, meaning of 'Active' attribute may vary according to the implementers. Thus, these commands are possible for those implementers who define "Active" as account status representing attribute.
- Since only one can be used at a time, please edit the attributes before creating the connector. If "Active" attribute of SCIM refers the status of account only then the EnableAccount/ DisableAccount commands are supported otherwise not. E.g. UnboundID doesn't support these commands.

Copyright © 2015 EMC Corporation. All Rights Reserved. Published in the USA.

Trademarks

RSA, the RSA Logo, and EMC are either registered trademarks or trademarks of EMC Corporation in the United States and/or other countries. All other trademarks used herein are the property of their respective owners. For a list of EMC trademarks, go to www.rsa.com/legal/trademarks_list.pdf.