Configuring LDAP Users on Cisco Meeting Server via API

Contents

Introduction

This document describes configuration of LDAP (Lightweight Directory Access Protocol) on Cisco Meeting Server via API (Application Programming Interface).

Prerequisites

PostMan App

Cisco Meeting Server (CMS)

Microsoft Active Directory

Requirements

There are no specific requirements for this document.

Components Used

Cisco Meeting Server

Microsoft Active Directory

Background Information

High level configuration flow to sync LDAP via API.

Step 1. Configure /ldapServers parameter thru API as described below

1. LDAP server's address/port information
2. Username and password for accessing the server
3. Secure of non secure ldap.

Step 2 : Configure /ldapMappings parameter through API as described below

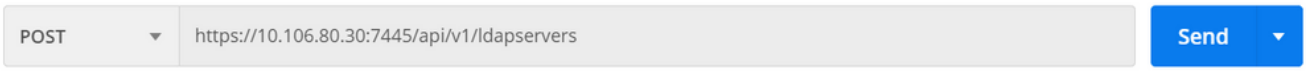
1. LDAP user properties objects to cms corresponding user objects
2. Example cms user jid will map to $sAMAccountName$@domain.com on cms and etc.

Step 3: Configure /ldapSources parameters thru API as described below which to tie ldapServers and ldapMappings object.

Configure

Step 1. Configure /ldapServers

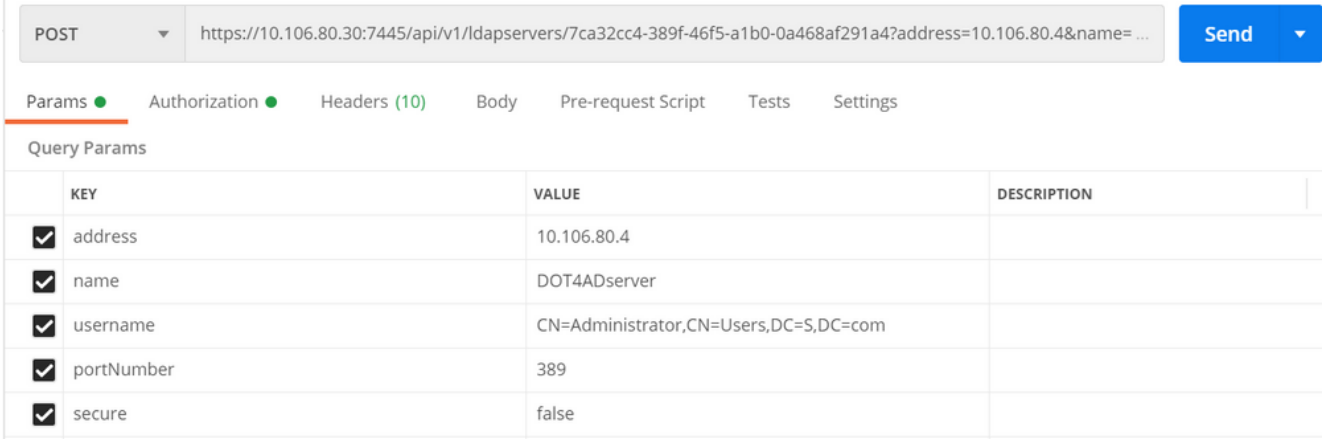
1. Send a POST for /ldapServers , which would create a ldapServer ID. Use unique /ldapServers ID for further configuration.



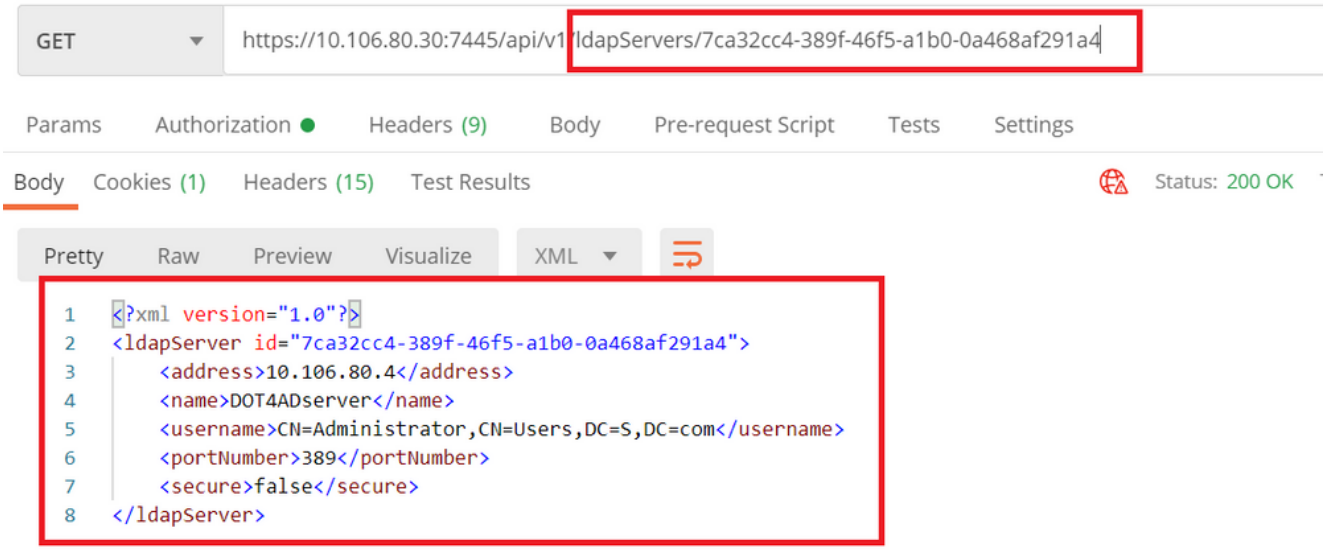
2. Response to POST would return in similar format <ldapServer id="7ca32cc4-389f-46f5-a1b0-0a468af291a4">
3. Capture below information to update LDAP Server ID per the CMS API Reference Guide

Parameters	Type/Value	Description/Notes
address *	String	The address of the LDAP server to connect to.
portNumber *	Number	The TCP or TLS port number to connect to on the remote LDAP server.
username	String	The username to use when retrieving information from the LDAP server.
password	String	The password of the account associated with username.
secure *	true|false	Whether to make a secure connection to the LDAP server. If “true” then TLS will be used; if “false”, TCP will be used.

4. Sample POST Method with Parameters



5. Perform a GET to verify configured parameters



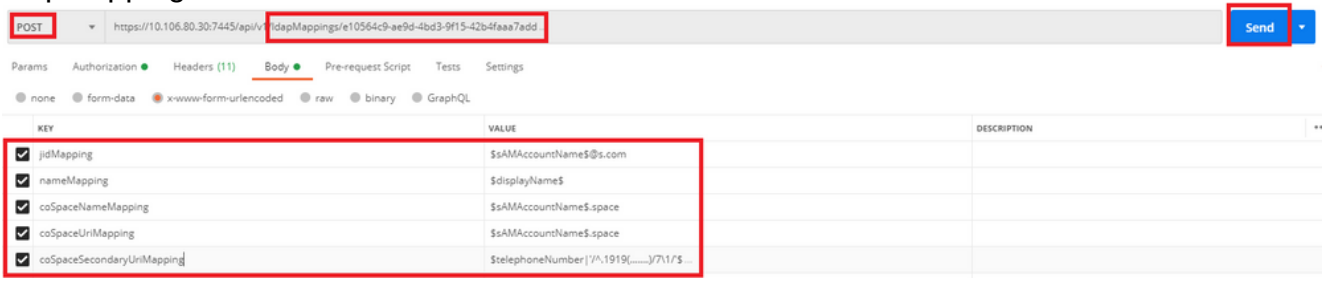
Step 2, Configure /ldapMappings

1. Send a POST for /ldapMappings to create a /ldapMappings ID. Use /ldapMappings ID and configure below parameters.

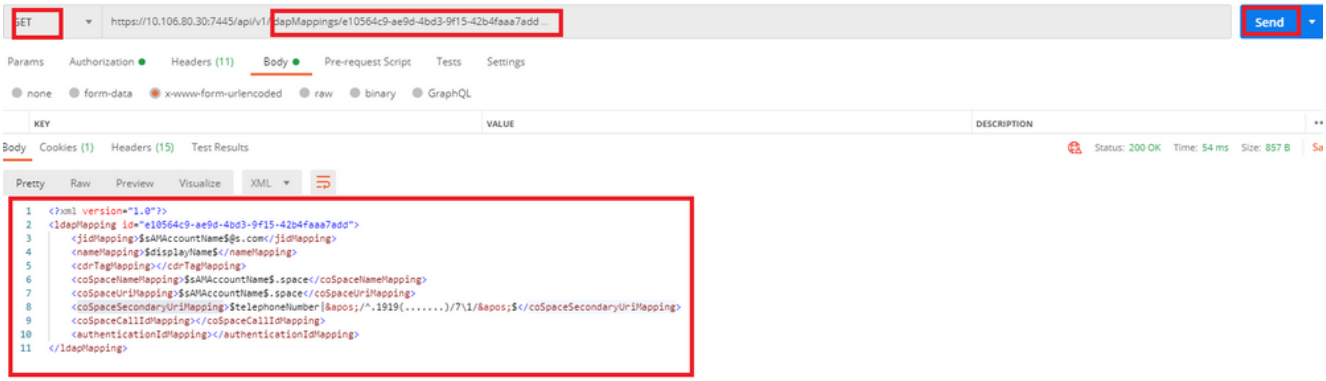
2. Capture below information to update LDAP Mapping ID per the [CMS API Reference Guide]

Parameters	Type/Value	Description/Notes
jidMapping	String	The template for generating user JIDs from the associated LDAP server's entries, for instance $sAMAccountName$@example.com.
nameMapping	String	The template for generating user names from the associated LDAP server's entries; for instance "cn" to use the common name.
cdrTagMapping	String	The template for generating a users' cdrTag value. Can be set either to a fixed value or be constructed from other LDAP fields for that user. The user's cdrTag is used in callLegStart CDRs. See the Cisco Meeting Server CDR Reference for details.
authenticationIdMapping	String	The template for generating authentication IDs from the associated LDAP server" s entries, for instance " $userPrincipalName$" .
coSpaceUriMapping	String	If these parameters are supplied, they ensure that each user account generated by this LDAP mapping has an associated personal coSpace. The user is automatically added as a member of the coSpace, with permissions defined above In order for that coSpace to be set up as required, these parameters provide the template for setting the coSpaces' URI, displayed name and configured Call ID. For example, setting coSpaceNameMapping to "cn personal coSpace" ensures that each user's coSpace is labelled with their name followed by "personal coSpace". Note that the generated coSpace will have its own cdrTag – and it will be the same as the user's cdrTag and cannot be changed other than by changing the cdrTagMapping above and re-syncing. (The coSpace's cdrTag is used in the callStart CDR. See the Cisco Meeting Server CDR Reference for details.) Note that the normal uniqueness rules apply to the URI and Call IDs of coSpaces set up in this way: it is not valid to have the same URI or Call ID for more than one coSpace set up by a given LDAP mapping, nor is it valid for such a coSpace URI or Call ID to be the same as one currently in use elsewhere on the Meeting Server.
coSpaceSecondaryUriMapping	String	
coSpaceNameMapping	String	
coSpaceCallIdMapping	String	

3. Configure below parameters for ldapMappings

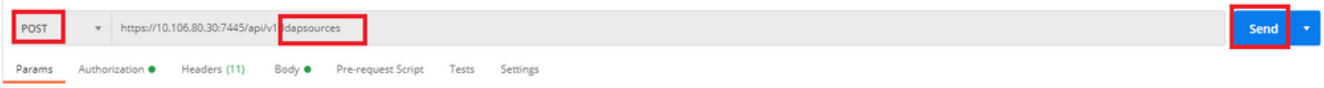


4. Perform a GET to verify configured parameters.



Step 3. Configure /ldapsources

1. Send a POST for /ldapsources to create a /ldapsources ID. Use /ldapsources ID and configure below parameters.

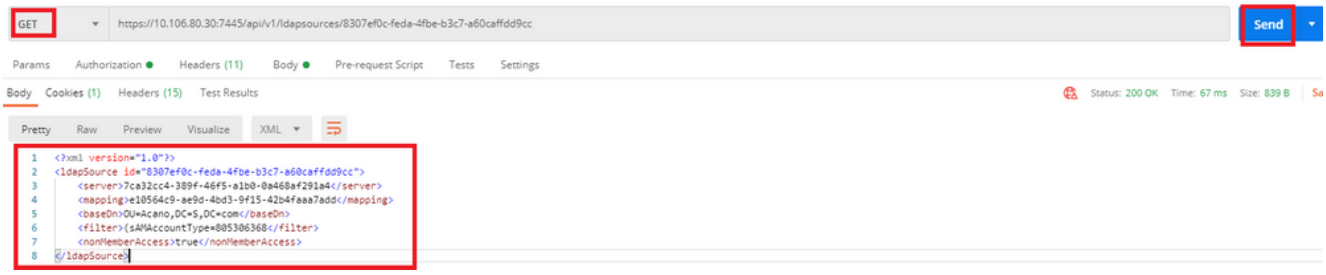


2. Capture below information to update LDAP Mapping ID per the CMS API Reference Guide

Parameters	Type/Value	Description/Notes
server *	ID	The ID of a previously-configured LDAP server (see above)
mapping *	ID	The ID of a previously-configured LDAP mapping (see above)
baseDn *	String	The distinguished name of the node in the LDAP server's tree from which users should be imported, for instance "cn=Use-ers,dc=<companyname>,dc=com"
filter	String	An LDAP filter string that records must satisfy in order to be imported as users, for instance "(objectClass=person)"
tenant	ID	If supplied, the ID for the tenant to which the LDAP source should be asso-ciated. Users imported with this LDAP source will be associated with that tenant
userProfile	ID	If supplied, this is the ID of the user profile to associate with users imported via this LDAP source. This parameter is present from version 2.0 onwards.
nonMemberAccess	**true**| false	This parameter pre-configures newly created spaces to allow or disallow non-member access. Spaces existing before the LDAP sync are not affected. true - no passcode is required to access the space and non-members are able to access the created spaces. This is the default setting and matches behavior before this parameter was introduced in version 2.0. false - ensures the member must configure non-member access and set a passcode as part of the LDAP sync. This setting allows a company to enforce passcode protection for non-member access to all user spaces. For more information, see Section 1.2.

3. Configure below parameters for ldapSources

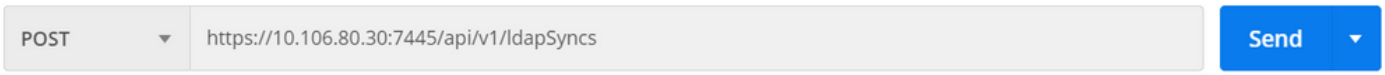
4. Perform a GET to verify configured parameters.



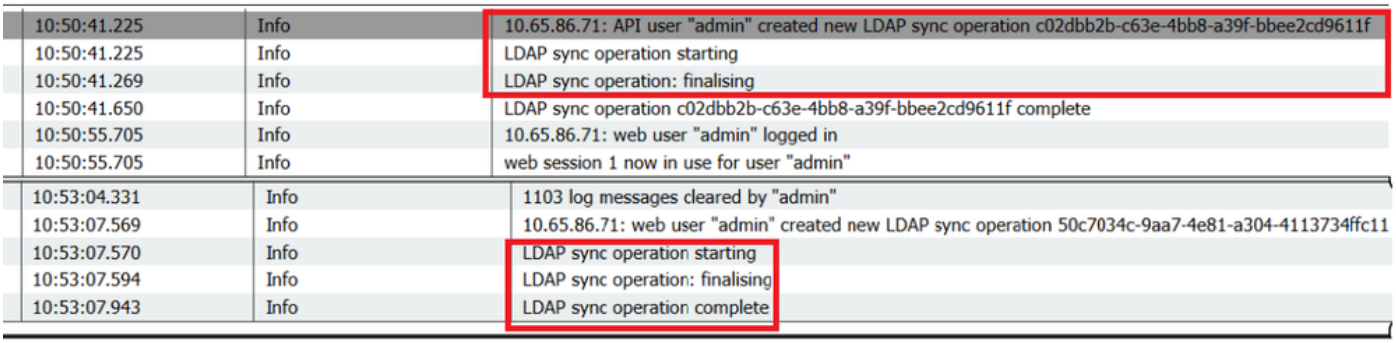
Configuration is complete. We can perform a full sync now.

Verify

Step 1. Send POST for /ldapSyncs from API and check event logs

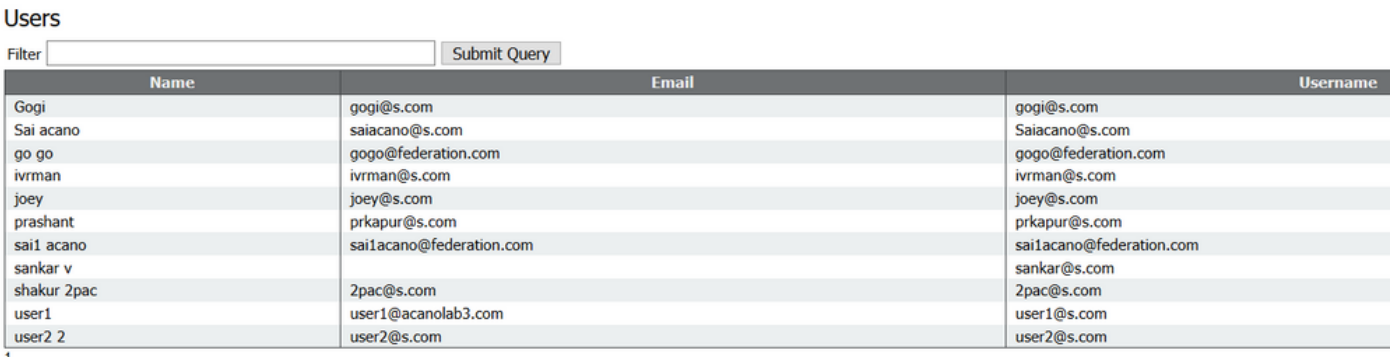


Step 2. Check in event logs if sync is completed.



10:50:41.225	Info	10.65.86.71: API user "admin" created new LDAP sync operation c02dbb2b-c63e-4bb8-a39f-bbee2cd9611f
10:50:41.225	Info	LDAP sync operation starting
10:50:41.269	Info	LDAP sync operation: finalising
10:50:41.650	Info	LDAP sync operation c02dbb2b-c63e-4bb8-a39f-bbee2cd9611f complete
10:50:55.705	Info	10.65.86.71: web user "admin" logged in
10:50:55.705	Info	web session 1 now in use for user "admin"
10:53:04.331	Info	1103 log messages cleared by "admin"
10:53:07.569	Info	10.65.86.71: web user "admin" created new LDAP sync operation 50c7034c-9aa7-4e81-a304-4113734ffc11
10:53:07.570	Info	LDAP sync operation starting
10:53:07.594	Info	LDAP sync operation: finalising
10:53:07.943	Info	LDAP sync operation complete

Step 3. Verify Users are synced from ldap source.



Users

Filter Submit Query

Name	Email	Username
Gogi	gogi@s.com	gogi@s.com
Sai acano	saiacano@s.com	Saiacano@s.com
go go	gogo@federation.com	gogo@federation.com
ivrman	ivrman@s.com	ivrman@s.com
joey	joey@s.com	joey@s.com
prashant	prkapur@s.com	prkapur@s.com
sai1 acano	sai1acano@federation.com	sai1acano@federation.com
sankar v		sankar@s.com
shakur 2pac	2pac@s.com	2pac@s.com
user1	user1@acanolab3.com	user1@s.com
user2 2	user2@s.com	user2@s.com

Troubleshoot

Verify API parameters and LDAP Attributes are accurate.

Taking packet captures from call Bridge helps in isolating connectivity issues with LDAP.