

PUBLIC

DOCUMENT VERSION: 2023-11-17

Virtual Learning Service Configuration Microsoft Teams



Table of Contents

Overview	3
Configuration Within Microsoft Azure	4
Register the Application	4
Generate a Client Secret	5
Set Microsoft Graph API Permissions	6
Configure the CsApplicationAccessPolicy for Application Authentication	7
Prerequisites	7
Configuration Steps	7
Configuration Within SuccessFactors Learning	9
Create a New VLS Configuration File	9
Update the Microsoft Teams VLS Configuration File with a New Client Secret	

Overview

This guide provides detailed configuration information to enable SuccessFactors Learning to use Microsoft Teams as a Virtual Learning Service for instructor-led classes.

Note: It is strongly recommended that you work with a Microsoft partner to register your Microsoft Teams app and complete the required tasks within the Microsoft Azure portal.

Considerations

- The instructor's ID is their Microsoft Teams email address. No password is required.
- Users can join a Microsoft Teams meeting through a URL, but only users that are registered for the class receive credit for attendance.
- If users have an alternate email address in Microsoft Azure, you **must** provide it in the user profile for attendance to be properly processed. You also need to specify which custom field contains the alternate email address in the VLS configuration for Microsoft Teams (Learning Administration > System Administration > Configuration > VLS Configuration).

Configuration Within Microsoft Azure

Register the Application

- 1. Log in to your Azure portal (https://portal.azure.com).
- 2. Navigate to Azure Active Directory (AD).
- 3. Navigate to App registrations and then choose New registration.
- 4. Enter the desired name.
- 5. For the **Supported account** types, select the first or second option. Personal Microsoft accounts aren't supported.

Home > sapsfsflearning >
Register an application
* Name
The user-facing display name for this application (this can be changed later).
Learning-Teams-Integration
Supported account types
who can use this application of access this API?
Accounts in this organizational directory only (sapsistearning only - single tenant)
Accounts in any organizational directory (Any Azure AD directory - Multitenant)
Help me choose
Redirect URI (optional)
We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be
changed later, but a value is required for most authentication scenarios.
Select a platform V e.g. https://example.com/auth
Register an app you're working on here. Integrate gallery apps and other apps from outside your organization by adding from Enterprise applications.
By proceeding, you agree to the Microsoft Platform Policies 🖻
Register

6. Choose Register.

- 7. Copy the values in the **Application (client) ID** and **Directory (tenant) ID** fields and then save in a secure, temporary location. These values are required when setting up the integration in SuccessFactors Learning Administration.
 - Note: A service principal is generated when an app is created in Azure AD. The credentials (application ID, directory ID, and app secret) associated with the service principal are used to authenticate the integration with your Azure AD. This allows SuccessFactors Learning to connect to your Microsoft Graph APIs.

Generate a Client Secret

- 1. Within your Azure portal, navigate to **Certificates & secrets** and then choose **New client secret** in the **Client secrets** section.
- 2. Complete the client secret properties as desired, then choose Add.

Microsoft Azure	و جر	Search resources, services	i, and docs (G+/)		🛛 😡 🖓	⑦ R shashi@sapsfsflearning sapsfsflearning (sapsfsflearning	. 0
Home > saps/sflearning > Learning-Teams-Integration		Add a client secret		×			
🔶 Learning-Teams-Integration Certificates & secrets 👒 🐇							
	_				Description	LMS CLIENT SECRET	
Search (Cmd+/) «	R Got feedback?				Expires	Recommended: 6 months	\sim
K Overview	Credentials enable confidential application	s to identify themselves t	o the authentication service when receiving tokens	at a web addressable location (using an H			
duickstart	start scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.						
🚀 Integration assistant							
Manage	 Application registration certificates, se 	crets and federated creden	tials can be found in the tabs below.				
Branding							
Output Authentication	Certificates (0) Client secrets (1)	Federated credentials	(0)				
Certificates & secrets	A secret string that the application uses t	o prove its identity when	requesting a token. Also can be referred to as appl	ication password.			
Token configuration	+ New client secret						
API permissions	Description	Expires	Value ①	Secret ID			
Co Expose an API	LMS-APIs	6/17/2022	CLIENT_SECRET_VALUE	CLIENT_SECRET_ID]		
App roles							
Roles and administrators Preview							
Manifest							
Support + Troubleshooting							
P Troubleshooting							
New support request							
					Add Cancel		

- 3. Copy the client secret in the **Value** column and then save in a secure, temporary location. This value is required when setting up the integration in SuccessFactors Learning Administration.
 - Note: When you navigate away from this page, the client secret value is masked and can't be copied.
 If you haven't copied the secret value, you'll need to create new client secret.

Set Microsoft Graph API Permissions

1. Within your Azure portal, navigate to API permissions and then choose Add a permission.



2. Choose Microsoft Graph and then choose Application permissions.

Microsoft Azure	Search resources, services, and docs (G+/)		🖂 🕼 🖉 🛞 🕐 🗟 shashi@sapsfsflearning 🧕
Home > sapstiftearning > Learning-Teams-Integration 		Request API permissions	×
Search (Cmd+/) « Overview	🕐 Refresh 🔰 🖗 Got feedback?	Microsoft Graph https://graph.microsoft.com/ Docs @ What type of permissions does your application require?	
Quickstart Integration assistant	The "Admin consent required" column shows the default value for an organization. However, user consent can be comore	Delegated permissions Your application needs to access the API as the signed-in user.	Application permissions Your application runs as a background service or daemon without a signed-in user.
Branding Authentication	Configured permissions Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent all the permissions the application needs. Learn more about permissions and consent	Select permissions P Start typing a permission to filter these results	expand all
Certificates & secrets	+ Add a permission 🗸 Grant admin consent for sapsfsflearning	Permission	Admin consent required
API permissions	API / Permissions name Type Description	> AccessReview	
Expose an API	sapimsintegration Delegated sapimsintegration	> AdministrativeUnit	
App roles	Microsoft Graph (2) OnlineMeetings.ReadWrite.All Application Read and create online meetings	> AgreementAcceptance	
Boles and administrators Preview	User.Read.All Application Read all users' full profiles	> Agreement	
Support + Troubleshooting	To view and manage permissions and user consent, try Enterprise applications.	> APIConnectors	
Troubleshooting		> AppCatalog	
New support request		> Application	
		> AnnRaleAccianment	
		Add permissions Discard	

3. Select the following permissions:

Required Permission	Purpose
OnlineMeetings.ReadWrite.All	Used by the Learning API to schedule, update, and delete meetings.
User.Read.All	Used by the Learning API to enroll and withdraw users from a virtual time slot.
OnlineMeetingArtifact.Read.All	Used by the Learning API to process meeting attendance.

- 4. Choose Add permissions.
- 5. Navigate to API permissions and choose Grant admin consent for [your Azure tenant name] and then choose Yes. The Status column should display "Granted for [your Azure tenant name]" for all API permissions.

Configure the CsApplicationAccessPolicy for Application Authentication

The CsApplicationAccessPolicy policy controls which users the integration can act on behalf of. It is important that the integration can act on behalf of instructors so that it can retrieve meeting attendance information.

Prerequisites

- The steps need to be performed by an administrator with admin access to Microsoft Teams.
- You must have Powershell 5.1 or later. You can check your version by running \$Host.version within PowerShell. Microsoft has installation instructions here: https://docs.microsoft.com/en-us/skypeforbusiness/set-up-your-computer-for-windows-powershell/download-and-install-windows-powershell-5-1
- You must have Microsoft Teams PowerShell Module installed. Microsoft has installation instructions here: https://docs.microsoft.com/en-us/microsoftteams/teams-powershell-install

Configuration Steps

1. In Powershell, connect to your Microsoft Teams using the Connect-MicrosoftTeams command. This will open a browser and allow you to log in with your Microsoft credentials. After successful login, the running PowerShell instance will be able to administer the Microsoft Teams instance that is associated with the account you used to log in to Microsoft.

- 2. Create a policy that contains the app id for the registered app you're working with.
 - a. Save the Azure registered app id (GUID/UUID) into a variable: \$appId = "your-app-id"
 - b. Save the policy name into a variable: \$policyName = "Your MS Teams Policy"
 - c. Create a new policy: new-csapplicationaccesspolicy -identity \$policyName
 - d. Assign the registered application to the policy: Set-CsApplicationAccessPolicy identity \$policyName -appids @{add=\$appId }
- 3. Assign that policy either globally or to individual users who will be instructors in the integration:
 - To assign the policy globally to all users use this command: Set-CsApplicationAccessPolicy
 -identity "Global" -appIds @{add=\$appid}
 - To assign the policy to an individual user only use this command: Grant-CsApplicationAccessPolicy -identity "instructorUsername@example.com" -PolicyName \$policyName

Note: You need to run this command for each instructor separately, you cannot add multiple instructors in a single command.

Configuration Within SuccessFactors Learning

When you configure the Virtual Learning Service (VLS), you create an API connection to a virtual meeting room vendor. That connection manages the virtual training sessions: learners can launch them from their assignments and learning administrators can schedule them from the administration environment.

Create a New VLS Configuration File

- 1. Go to SuccessFactors Learning Administration > System Administration > Configuration > System Configuration and edit LMS_ADMIN.
- 2. In the LMS_ADMIN file, find vleEnabled and change its value to true. If you have created previous VLS configuration files, this will already be set to true.
- 3. Go to System Administration > Configuration > VLS Configuration and choose Add New.
- 4. Select Microsoft Teams from the VLS Vendor dropdown list to populate the default configuration template.
- 5. Edit the file and paste the values you copied from the Azure portal into the corresponding elements, as described in the following table:

Element	Value to Use in the File
api_key	The client ID, from the Application ID field on overview page in the Microsoft Azure portal.
password	The client secret, generated in the Microsoft Azure portal. Note: It's stored as an encrypted value so if you don't have it, you'll need to generate it.
tenant_id	The tenant ID, from the Directory ID field on the overview page in the Microsoft Azure portal.
application_id	The client ID, from the Application ID field on overview page in the Microsoft Azure portal.

- 6. Provide a value in the VLS ID field (required) and in the Description field (optional).
- 7. The save the configuration, choose **Add**.

Update the Microsoft Teams VLS Configuration File with a New Client Secret

Each time the client secret expires, you need to generate a new secret in the Azure portal and then update the VLS configuration file in SuccessFactors Learning with the new value.

- 1. Generate a Client Secret.
- 2. Go to SuccessFactors Learning Administration > System Administration > Configuration > VLS Configuration and search for and edit your Microsoft Teams configuration file.
- 3. In the password element, paste the new client secret.
- 4. To save, choose Apply Changes.