

MS Teams Extender

An extender that uses Microsoft Graph API to add additional MS Teams interactions to NWC, this extender only exposes a subset of the functionality available in the Microsoft Graph API. For queries, feel free to get in [contact](#).

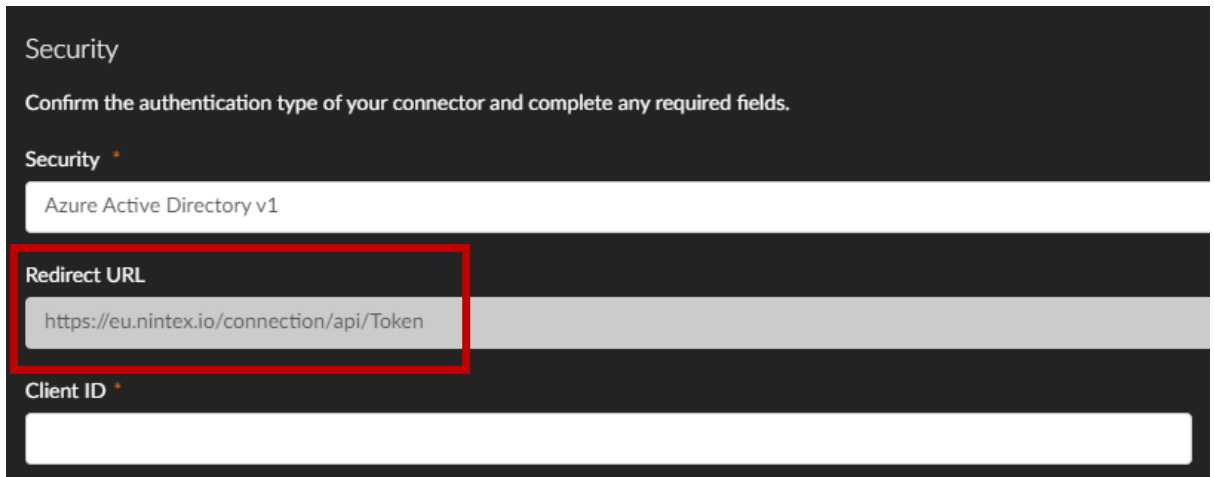
[Create AAD App Registration](#)..... 2

[Methods](#) 3

[Usage](#) 4

Create AAD App Registration

1. Follow the [quickstart](#) article to create an App Registration to be used to manage you MS Graph permissions
2. Use the redirect URI that is given when creating a new Xtension on NWC



Security

Confirm the authentication type of your connector and complete any required fields.

Security *

Azure Active Directory v1

Redirect URL

https://eu.nintex.io/connection/api/Token

Client ID *

3. Copy the Client ID (Application ID on the Overview page) and the Client Secret (on the Certificates & secrets page) and use these in your Xtension
4. Grant the application the following permissions (both delegated & application where needed):

User.ReadWrite.All
TeamsTab.ReadWrite.All
TeamsTab.Create
TeamsTab.ReadWrite.All
ChannelMessage.Send
ChannelMessage.Read.All
Group.Read.All
TeamMember.Read.All
ChannelMember.ReadWrite.All
Channel.Delete.All
Channel.ReadBasic.All
Channel.Create
Team.ReadBasic.All
TeamSettings.Read.AllTeamSettings.Read.All
TeamSettings.ReadWrite.AllTeamSettings.ReadWrite.All

Methods

<i>Method</i>	Description
<i>Add Channel Member</i>	Add users to a private channel
<i>Create Document Library Tab</i>	Create a tab in a channel
<i>Create Excel Tab</i>	
<i>Create OneNote Tab</i>	
<i>Create PDF Tab</i>	
<i>Create PowerPoint Tab</i>	
<i>Create SharePoint Tab</i>	
<i>Create Website Tab</i>	
<i>Create Word Tab</i>	
<i>Create a Channel</i>	Create a standard or private channel
<i>Delete Channel</i>	Move a channel to the recycle bin (to be removed after 30 days)
<i>Delete Tab</i>	Remove a tab
<i>Get Channel</i>	Retrieve information for a specified channel
<i>Get Channel Files Folder</i>	Get the corresponding SharePoint Files folder of a channel
<i>Get Channel Messages</i>	Return all root-level messages
<i>Get Channels</i>	Retrieve all channels in a team
<i>Get Tabs</i>	Retrieve all tabs in a channel
<i>Get Team Members</i>	Retrieve a collection of all members of a team
<i>Get User</i>	Retrieve information on a specific user
<i>Send Adaptive Card</i>	Send a Teams Adaptive Card (see Cards Reference)

Usage

Note: Only select methods are listed due to complexity of setup

Add Channel Member

1. Get the user ID by executing the Get User step and providing the principal name and specifying an output variable:

The screenshot shows a workflow in the Nintex editor. The workflow starts with a 'Start event: Component workfl...' step, followed by a 'Get User' step, and ends with a 'Workflow complete' step. The 'Get User' step is highlighted. To the right, the configuration panel for the 'Get User' step is visible. It shows the connection 'MS Teams Extension', the user 'janed@ntxte09.com', and the output variable 'User'.

2. Execute the Add Channel Member step, provide a Team ID, Channel ID (this method only works on a **private channel**), and replace the {user_id} text in the User Binding field with the Id field of the Get User step's output object

The screenshot shows a workflow in the Nintex editor. The workflow starts with a 'Start event: Component workfl...' step, followed by a 'Get User' step, then an 'Add Channel Member' step, and ends with a 'Workflow complete' step. The 'Add Channel Member' step is highlighted. To the right, the configuration panel for the 'Add Channel Member' step is visible. It shows the connection 'MS Teams Extension', the team ID '2c40affa-1688-4285-afe8-7c03088cf4d4', the channel ID '19:6a8b8aca65a244ab9c8e4ecbb065f8d4@thread.ta cv2', and the member binding 'https://graph.microsoft.com/v1.0/users(*{Id})'.

Send Adaptive Card

1. Visit <https://adaptivecards.io/designer/> and create your adaptive card (change the Host App to MS Teams to get the correct schema version). Copy the payload for later use
2. Create a variable in your workflow called *Message Content*

The screenshot shows the Nintex workflow editor interface. On the left, the 'Workflow' pane is visible with 'Start event' and 'Context' sections. On the right, the 'Create variable' dialog is open. It features a '+ Create variable' button at the top. Below it, a variable named 'Message Content' is selected, indicated by a blue 'x' icon. The 'Name' field contains 'Message Content'. The 'Default value' field is empty. There is an unchecked 'Output' checkbox with an information icon. Below these fields, it states 'Variable type: Text' and 'Consumed by: 0 workflow actions'. A blue 'Save' button is at the bottom right of the dialog.

3. Add a "Set a variable value" step to your workflow and pass the following value to the variable you created:

`<attachment id="74d20c7f34aa4a7fb74e2b30004247c5"></attachment>`

The screenshot shows the Nintex workflow editor with a workflow diagram on the left and the 'Set a variable value' configuration pane on the right. The workflow diagram includes a 'Start event: Component workfl...' step, followed by a 'Set a variable value' step (marked with a blue 'x=?' icon), and finally a 'Workflow complete' step. The configuration pane for the 'Set a variable value' step shows the 'Variable *' dropdown set to 'Message Content'. The 'Value *' field contains the XML payload: `<attachment id="74d20c7f34aa4a7fb74e2b30004247c5"></attachment>`.

Note: The ID value can be any value as long as it corresponds to an attachment in the step below.

4. Add the “Send Adaptive Card” step and configure the Team ID and Channel ID. Add *html* as the Content Type and the variable we created in step 1 to the Content field. Optionally you can add a subject to your message as well.

The screenshot shows a workflow in the Nintex editor. The workflow steps are: 'Start event: Component workfl...', 'Set a variable value', 'Send Adaptive Card', and 'Workflow complete'. The 'Send Adaptive Card' step is selected, and its configuration panel is open on the right. The configuration panel is titled 'Microsoft Teams Extension' and includes the following fields:

- Connection ***: MS Teams Extension
- Team ***: Team ID (2c40affa-1688-4285-afe8-7c03088cf4d4)
- Channel ***: Channel ID (19:6a8b8aca65a244ab9c8e4ecbb065f8d4@thread.ta cv2)
- Message**: (empty)
- Body ***: (empty)
- Content Type**: html
- Content**: Message Content

5. Click on the *Add item* link under Attachment and paste the id of your attachment markup as in step 3. For Content Type use *application/vnd.microsoft.card.adaptive* (or you can use different schemas given in the Card Reference documentation link above) and paste the payload of the card you created in step 1

The screenshot shows the same workflow in the Nintex editor, but the 'Send Adaptive Card' step is now configured with attachment details. The configuration panel is titled 'Send Adaptive Card' and 'Microsoft Teams Extension'. The configuration includes the following fields:

- Attachment**: (empty)
- Item ...**: Adaptive Card Attachment
- ID ***: 74d20c7f34aa4a7fb74e2b30004247c5
- Content Type**: application/vnd.microsoft.card.adaptive
- Content**: {"type": "AdaptiveCard", "body": [{"type": "TextBlock", "size": "Medium", "weight": "Bolder", "text": "\${title}},{type": "ColumnSet", "columns": [{"ty