

# Setting up Microsoft Teams in ScriptRunner

Using the ROPC workflow

Author: Michael Gall

Date: 2022-08-11

Version 1.0

## Table of Contents

<b>1</b>	<b>Introduction</b> .....	<b>4</b>
<b>2</b>	<b>Overview of the ROPC workflow</b> .....	<b>5</b>
<b>3</b>	<b>Installing PowerShell modules</b> .....	<b>6</b>
<b>4</b>	<b>Configuring the certificate</b> .....	<b>7</b>
4.1	Creating the certificate .....	7
4.2	Exporting the certificate.....	8
<b>5</b>	<b>Configuring the service principal</b> .....	<b>10</b>
5.1	Creating the service principal .....	10
5.2	Uploading the certificate.....	13
<b>6</b>	<b>Testing the connection in ScriptRunner</b> .....	<b>14</b>
<b>7</b>	<b>Customizing API permissions and ownership</b> .....	<b>16</b>
7.1	Adjusting API permissions .....	16
7.2	Creating member users in Azure.....	17
7.3	Adding member user as owner.....	17
<b>8</b>	<b>Completing the ScriptRunner configuration</b> .....	<b>18</b>
8.1	Creating a credential .....	18
8.2	Configuring the Microsoft Teams target.....	18
<b>9</b>	<b>Checklist</b> .....	<b>20</b>
<b>10</b>	<b>Possible error sources</b> .....	<b>21</b>
10.1	Conditional Access .....	21
10.2	Problems with the login.....	22
<b>11</b>	<b>Notes and references</b> .....	<b>23</b>
11.1	Notes .....	23
11.2	References .....	23

## Table of Figures

Figure 1: System drawing of the ROPC workflow .....	5
Figure 2: Overview of the installed PowerShell modules .....	6
Figure 3: Output of the PowerShell console after certificate creation.....	7
Figure 4: Local computer certificates in LocalComputer\My.....	8
Figure 5: Certificate export wizard - public key only .....	8
Figure 6: Certificate export wizard - X.509 (.CER) format.....	9
Figure 7: Azure Portal login page.....	10
Figure 8: App registration in the Azure Portal.....	10
Figure 9: Overview of enterprise applications in Azure AD.....	11
Figure 10: Registering a new service principal .....	11
Figure 11: Overview of the new service principal.....	12
Figure 12: Subpage in MS_TEAMS_ROPC.....	13
Figure 13: Security settings - certificate overview .....	13
Figure 14: Certificate upload in the Azure Portal .....	13
Figure 15: New Microsoft Graph target .....	14
Figure 16: Input form of the Microsoft Graph target .....	14
Figure 17: Connection test output .....	15
Figure 18: Default permissions for Microsoft Graph.....	16
Figure 19: API permission with admin consent.....	17
Figure 20: Adding the member user as an owner.....	17
Figure 21: Microsoft Teams target in ScriptRunner .....	18
Figure 22: Connection test output .....	19
Figure 23: Potential restrictions through conditional access policies .....	21
Figure 24: Sign-in logs in the Azure Portal.....	21
Figure 25: Error message in ScriptRunner portal - invalid tenant ID.....	22

# 1 Introduction

This document describes how to set up Microsoft Teams using ROPC workflows.

For the connection to work, the following requirements must be met:

- The ScriptRunner server is running in a [current version](#)<sup>\*1</sup>
- Microsoft Teams PowerShell module >= version 4.5.0 is installed
- Microsoft Graph PowerShell module >= version 1.10.0 is installed

The two modules required for Microsoft Teams and Microsoft Graph are installed in chapter 3

An account with global administrator privileges is required to set up and configure the service principal in Microsoft Azure.

The account used for the service principal is only required to be a member of the Azure tenant and must not be granted any other rights.

This document describes the individual steps:

- Preparation: Install PowerShell modules
- Create certificate for Microsoft Teams use case
- Set up service principal
- Upload certificate
- Test connection in ScriptRunner using Microsoft Graph
- Customize API permissions and ownerships
- Test connection with Microsoft Teams
- Possible errors and further resources

### Note

The procedure described here was successfully implemented as part of a proof of concept on August 3rd, 2022, and reflects the current status. The document will be updated periodically to account for future changes.

Please read and work through the document **completely**. If steps are omitted, this may result in the connection setup not working.

For feedback and questions, please contact [support@scriptrunner.com](mailto:support@scriptrunner.com).

---

<sup>1</sup>, tested on 2022-08-11 - current version PortalEdition R4, Build 1603

## 2 Overview of the ROPC workflow

This chapter contains a system drawing of the ROPC workflow.<sup>2</sup>

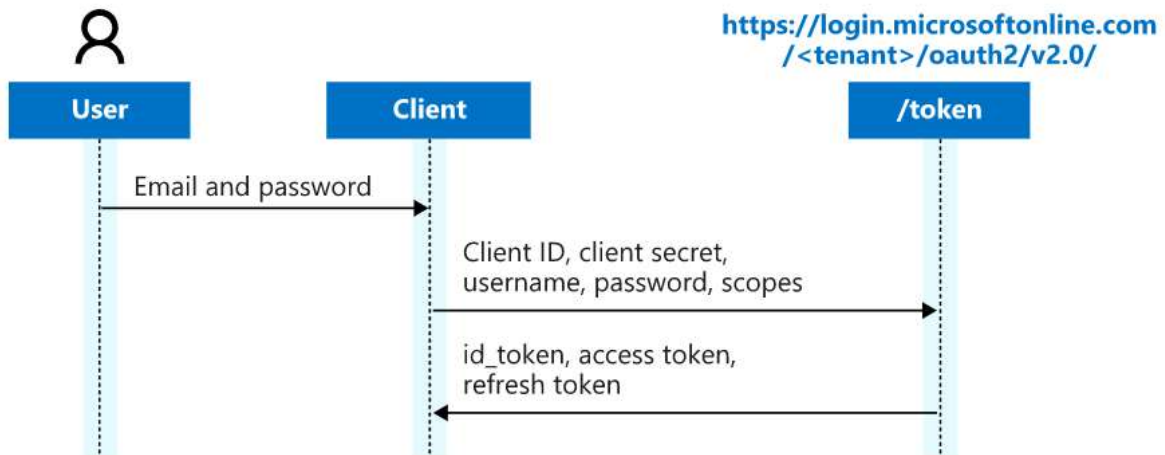


Figure 1: System drawing of the ROPC workflow

You will need the following data for setup:

- Azure username and password (default membership).
- Tenant ID or primary domain
- Application ID of the service principal
- Certificate thumbprint

To log in via the ROPC workflow, you need a username and password in addition to the certificate, tenant ID, and application ID.

For the setup in Azure AD, an Azure AD Premium P1 or P2 license is required. Users should have at least a Microsoft 365 E3 license associated. Microsoft Teams must be set up, otherwise API permissions cannot be granted.

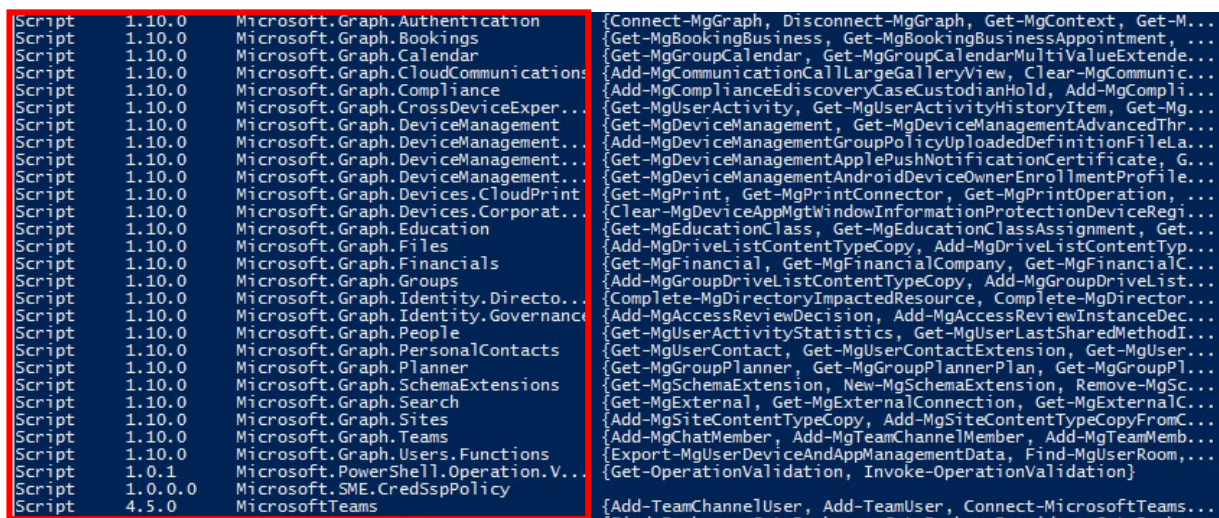
<sup>2</sup> , verified 2022-08-11 - drawing taken from: <https://docs.microsoft.com/de-de/azure/active-directory/develop/v2-oauth-ropc>

### 3 Installing PowerShell modules

Before starting the setup, verify that the relevant modules for Microsoft Teams and Microsoft Graph are already installed. To do so, run the following PowerShell command:

```
Get-Module -ListAvailable
```

You will get an overview of the installed modules as an output. This list should include both Microsoft Graph and Microsoft Teams.



Script 1.10.0	Microsoft.Graph.Authentication	{Connect-MgGraph, Disconnect-MgGraph, Get-MgContext, Get-M...
Script 1.10.0	Microsoft.Graph.Bookings	{Get-MgBookingBusiness, Get-MgBookingBusinessAppointment, ...
Script 1.10.0	Microsoft.Graph.Calendar	{Get-MgGroupCalendar, Get-MgGroupCalendarMultiValueExtende...
Script 1.10.0	Microsoft.Graph.CloudCommunications	{Add-MgCommunicationCallLargeGalleryView, Clear-MgCommunic...
Script 1.10.0	Microsoft.Graph.Compliance	{Add-MgComplianceDiscoveryCaseCustodianHold, Add-MgCompli...
Script 1.10.0	Microsoft.Graph.CrossDeviceExper...	{Get-MgUserActivity, Get-MgUserActivityHistoryItem, Get-Mg...
Script 1.10.0	Microsoft.Graph.DeviceManagement	{Get-MgDeviceManagement, Get-MgDeviceManagementAdvancedThr...
Script 1.10.0	Microsoft.Graph.DeviceManagement...	{Add-MgDeviceManagementGroupPolicyUploadedDefinitionFileLa...
Script 1.10.0	Microsoft.Graph.DeviceManagement...	{Get-MgDeviceManagementApplePushNotificationCertificate, G...
Script 1.10.0	Microsoft.Graph.DeviceManagement...	{Get-MgDeviceManagementAndroidDeviceOwnerEnrollmentProfile...
Script 1.10.0	Microsoft.Graph.Devices.CloudPrint	{Get-MgPrint, Get-MgPrintConnector, Get-MgPrintOperation, ...
Script 1.10.0	Microsoft.Graph.Devices.Corporat...	{Clear-MgDeviceAppMgtWindowInformationProtectionDeviceRegi...
Script 1.10.0	Microsoft.Graph.Education	{Get-MgEducationClass, Get-MgEducationClassAssignment, Get...
Script 1.10.0	Microsoft.Graph.Files	{Add-MgDriveListContentTypeCopy, Add-MgDriveListContentTyp...
Script 1.10.0	Microsoft.Graph.Financials	{Get-MgFinancial, Get-MgFinancialCompany, Get-MgFinancialC...
Script 1.10.0	Microsoft.Graph.Groups	{Add-MgGroupDriveListContentTypeCopy, Add-MgGroupDriveList...
Script 1.10.0	Microsoft.Graph.Identity.Directo...	{Complete-MgDirectoryImpactedResource, Complete-MgDirector...
Script 1.10.0	Microsoft.Graph.Identity.Governanc...	{Add-MgAccessReviewDecision, Add-MgAccessReviewInstanceDec...
Script 1.10.0	Microsoft.Graph.People	{Get-MgUserActivityStatistics, Get-MgUserLastSharedMethodID...
Script 1.10.0	Microsoft.Graph.PersonalContacts	{Get-MgUserContact, Get-MgUserContactExtension, Get-MgUser...
Script 1.10.0	Microsoft.Graph.Planner	{Get-MgGroupPlanner, Get-MgGroupPlannerPlan, Get-MgGroupPl...
Script 1.10.0	Microsoft.Graph.SchemaExtensions	{Get-MgSchemaExtension, New-MgSchemaExtension, Remove-MgSc...
Script 1.10.0	Microsoft.Graph.Search	{Get-MgExternal, Get-MgExternalConnection, Get-MgExternalC...
Script 1.10.0	Microsoft.Graph.Sites	{Add-MgSiteContentTypeCopy, Add-MgSiteContentTypeCopyFromC...
Script 1.10.0	Microsoft.Graph.Teams	{Add-MgChatMember, Add-MgTeamChannelMember, Add-MgTeamMemb...
Script 1.10.0	Microsoft.Graph.Users.Functions	{Export-MgUserDeviceAndAppManagementData, Find-MgUserRoom, ...
Script 1.0.1	Microsoft.PowerShell.Operation.V...	{Get-OperationValidation, Invoke-OperationValidation}
Script 1.0.0.0	Microsoft.SME.CredSspPolicy	
Script 4.5.0	MicrosoftTeams	{Add-TeamChannelUser, Add-TeamUser, Connect-MicrosoftTeams...

Figure 2: Overview of the installed PowerShell modules

Should any of the mentioned PowerShell modules be missing, install them with the following commands:

```
Install-Module -Name MicrosoftTeams -Scope AllUsers
Install-Module -Name Microsoft.Graph -Scope AllUsers
```

## 4 Configuring the certificate

This chapter describes how to create and export the certificate.

### 4.1 Creating the certificate

Create a self-signed certificate using the following PowerShell command:

```
New-SelfSignedCertificate -DnsName <FQDN> -FriendlyName  
MS_TEAMS_ROPC -KeyExportPolicy Exportable -KeySpec Signature
```

After the certificate has been created, the console displays the certificate thumbprint. Make a copy of the certificate thumbprint for later use.



```
PS C:\Users\Administrator.SCRIPTRUNNER> New-SelfSignedCertificate -dnsname [redacted] -FriendlyName MS_TEAMS_ROPC -KeyExportPolicy Exportable -KeySpec Signature  
  
PSParentPath: Microsoft.PowerShell.Security\Certificate:\LocalMachine\My  
  
Thumbprint                               Subject  
-----  
FC05048941CF [redacted] CN=VMSr01.scriptrunner.tan  
  
PS C:\Users\Administrator.SCRIPTRUNNER>
```

Figure 3: Output of the PowerShell console after certificate creation

#### Note

Be sure to create the certificate with the **-KeySpec Signature** parameter, otherwise the certificate cannot be used to establish the connection.

By default, the certificate is created in the **Cert:\LocalMachine\My** store.

## 4.2 Exporting the certificate

Export the certificate you just created using the Certificate Manager in the Microsoft Management Console (MMC).

Exporting the public key is sufficient since the private key cannot be imported into Azure.

Right-click the certificate and click **All Tasks > Export...**

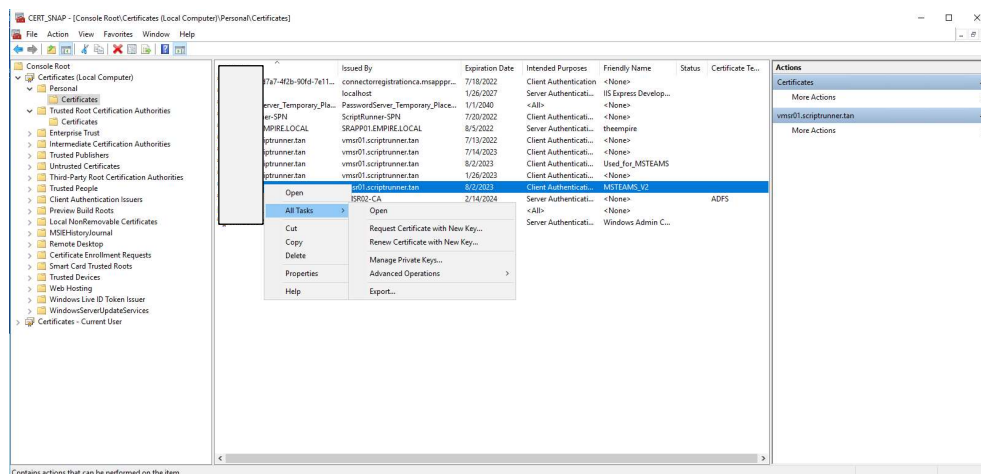


Figure 4: Local computer certificates in LocalComputer\My.

In the wizard, enable the **No, do not export the private key** option.

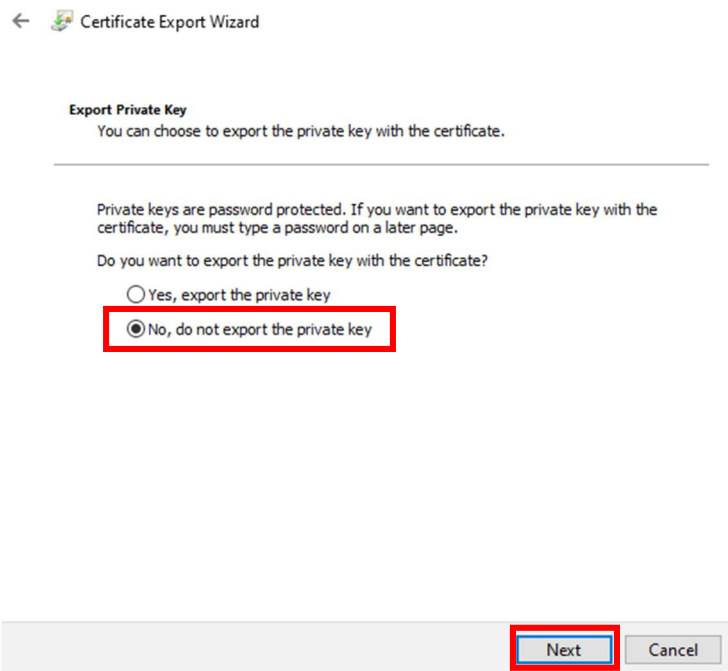


Figure 5: Certificate keys export wizard - public key only



Export the certificate in X.509 (.CER) format.

←  Certificate Export Wizard

### Export File Format

Certificates can be exported in a variety of file formats.

Select the format you want to use:

- DER encoded binary X.509 (.CER)
- Base-64 encoded X.509 (.CER)
- Cryptographic Message Syntax Standard - PKCS #7 Certificates (.P7B)
  - Include all certificates in the certification path if possible
- Personal Information Exchange - PKCS #12 (.PFX)
  - Include all certificates in the certification path if possible
  - Delete the private key if the export is successful
  - Export all extended properties
  - Enable certificate privacy
- Microsoft Serialized Certificate Store (.SST)

Next

Cancel

Figure 6: Certificate export wizard - X.509 (.CER) format

## 5 Configuring the service principal

This chapter describes the setup and configuration of the service principal.

### 5.1 Creating the service principal

Setup the service principal in Azure Active Directory. Log in to **portal.azure.com** and click **Azure Active Directory**.

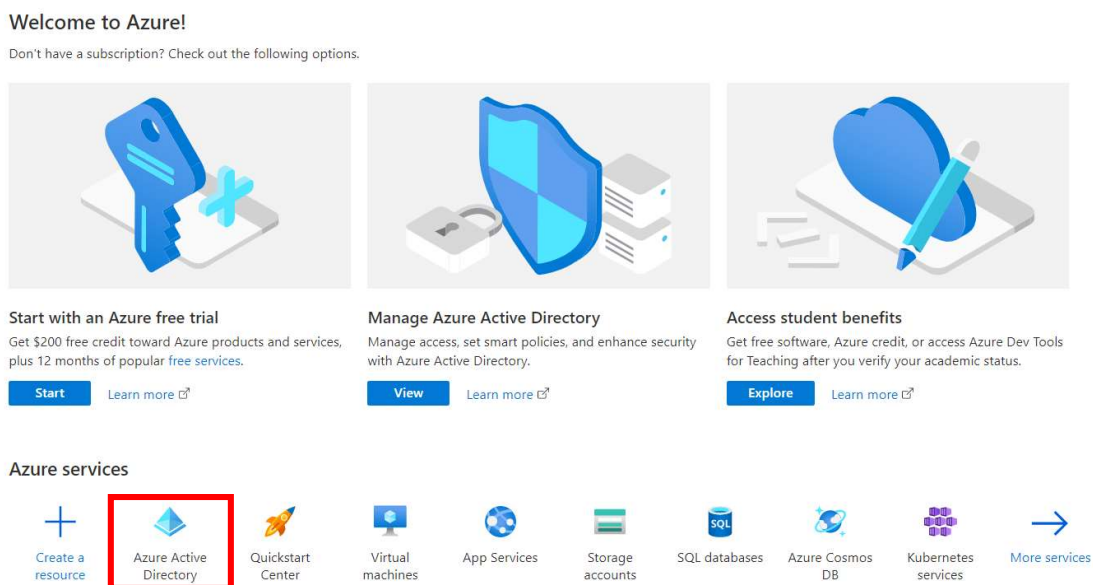


Figure 7: Azure Portal login page

In the left navigation bar on the overview page, click **App registrations**.

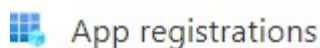


Figure 8: App registration in the Azure Portal

This page gives you an overview of all applications, owned applications and deleted applications.

Click **+ New registration** to create a new service principal.

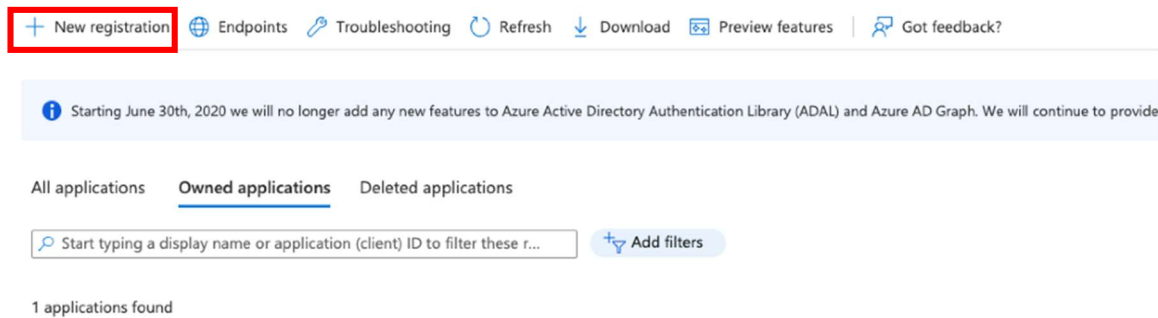


Figure 9: Overview of enterprise applications in Azure AD

Assign a name. For the other settings, the default can remain selected.

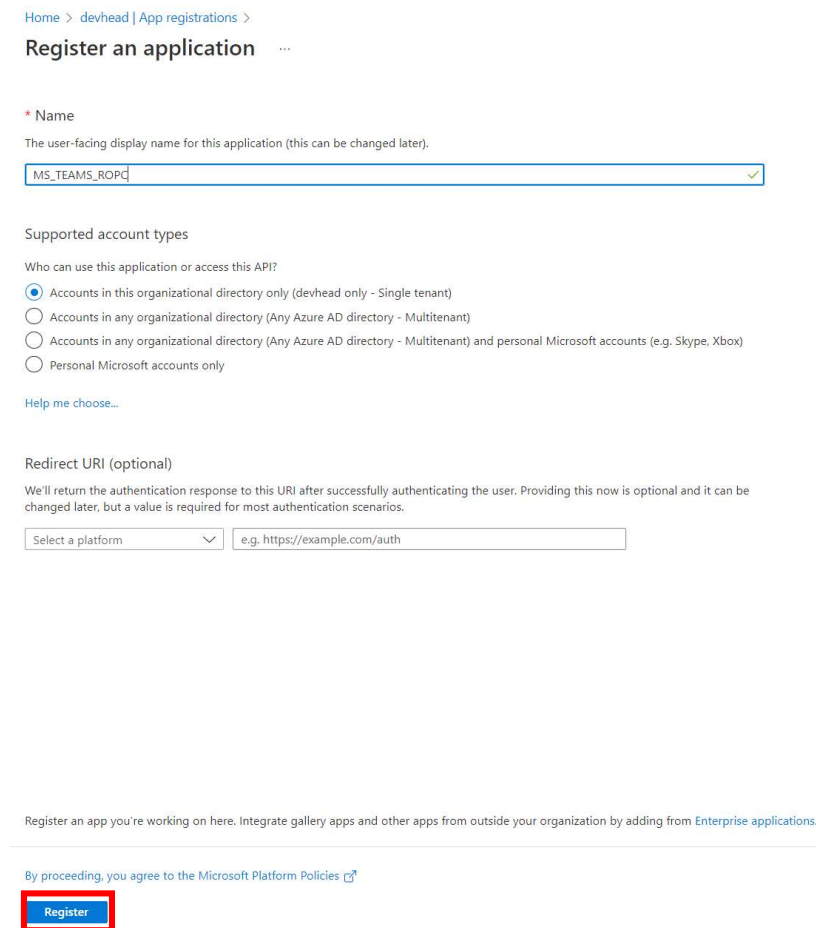


Figure 10: Registering a new service principal

Switch back to the overview and display the application ID and the tenant ID. You will need both to set up the connection.

 Delete  Endpoints  Preview features

### ^ Essentials

Display name : [MS TEAMS ROPC](#)

Application (client) ID : 67c04b6a-886a-4b9e-

Object ID : 1124eac4-6a04-4e15-b3f7-

Directory (tenant) ID : 557f8ff5-

Supported account types : [My organization only](#)

Figure 11: Overview of the new service principal

## 5.2 Uploading the certificate

To upload your certificate in the Azure Portal, click **Certificates & Secrets** in the left navigation bar.

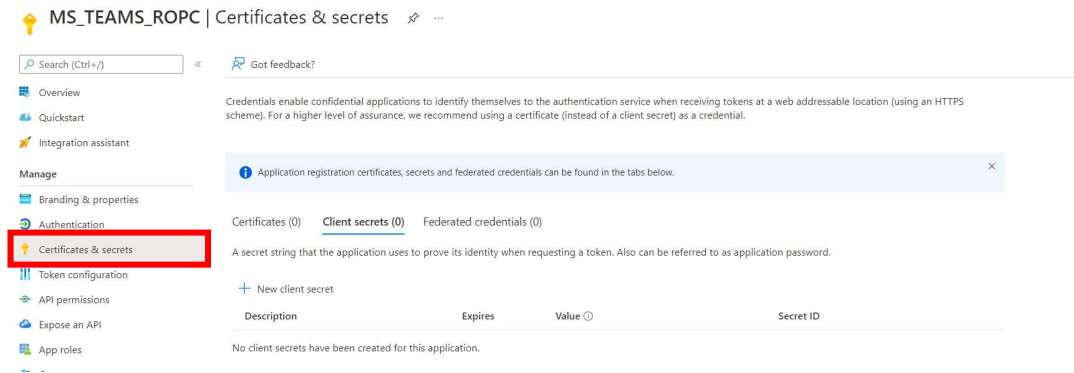


Figure 12: Subpage in MS\_TEAMS\_ROPC

In the **Certificates** section, click **Upload certificate**.

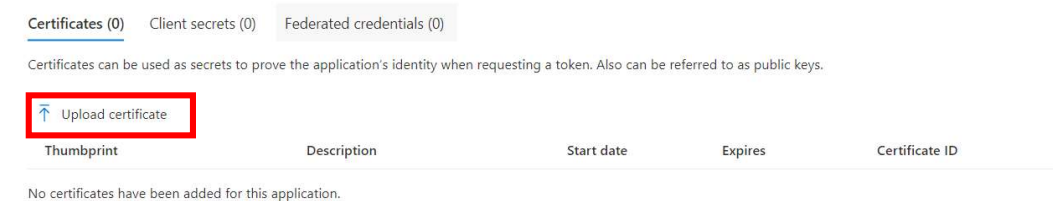


Figure 13: Security settings - certificate overview

An area for uploading the certificate opens on the right side.

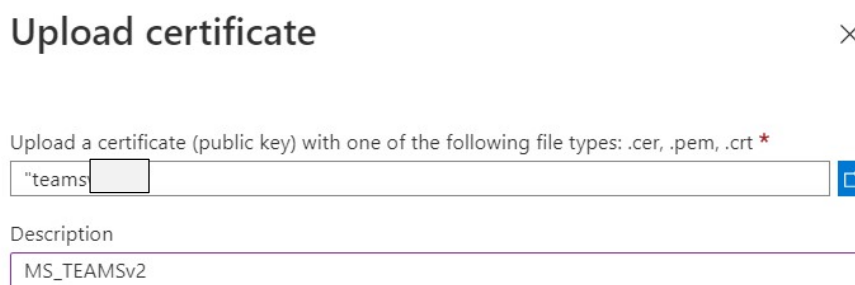


Figure 14: Certificate upload in the Azure Portal

Once the certificate has been uploaded, the setup of the service principal for configuration with Microsoft Graph is complete.

When you have made all configurations, perform a connection test in ScriptRunner via Microsoft Graph. Test whether it is possible to establish a connection (see chapter 6).

## 6 Testing the connection in ScriptRunner

Next, set up the target system in ScriptRunner. Open the ScriptRunner Portal and go to **Configuration > Targets**. Click **Create > Microsoft 365** and assign a display name to the target. In the **Microsoft services** section, add the **Microsoft Graph** service.

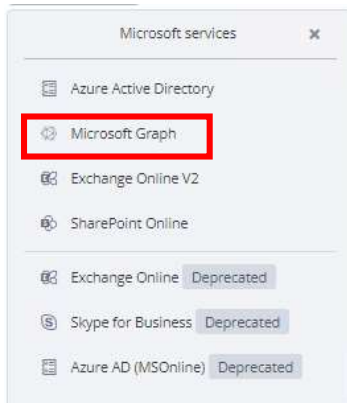


Figure 15: New Microsoft Graph target

Enter the **Tenant ID**, the **Application ID**, and the **Certificate thumbprint** in the service settings. The **Credential** field must be left blank!

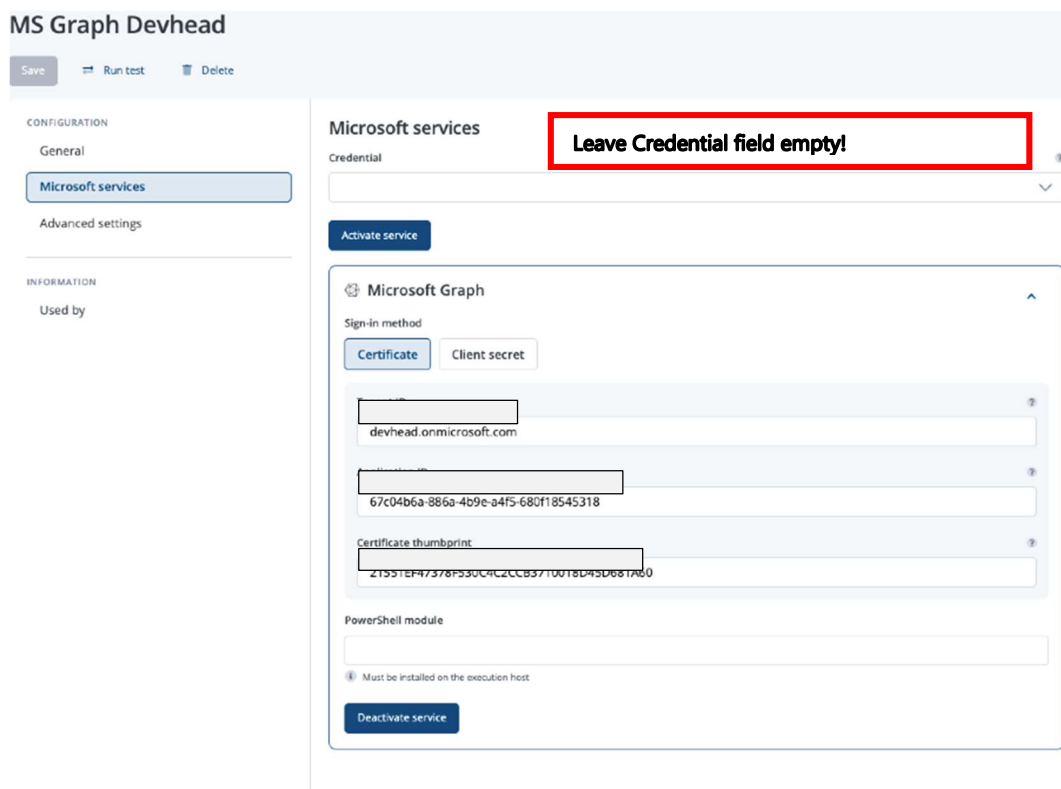
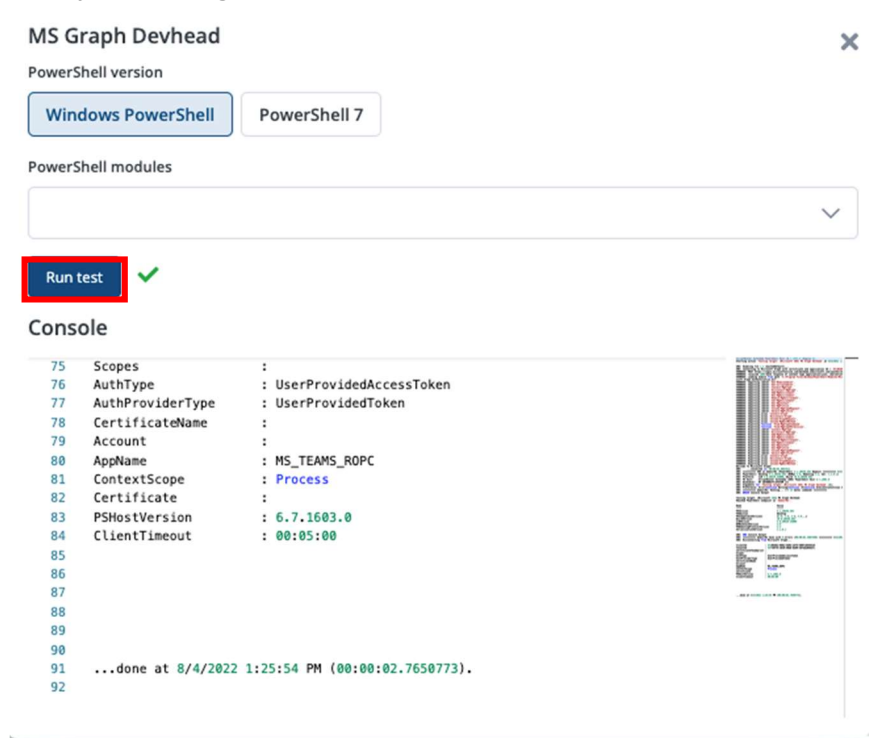


Figure 16: Input form of the Microsoft Graph target

Save your settings. Click **Run test** to perform a connection test.



**MS Graph Devhead** [X]

PowerShell version

Windows PowerShell | PowerShell 7

PowerShell modules

**Run test** ✓

**Console**

```
75 Scopes :  
76 AuthType : UserProvidedAccessToken  
77 AuthProviderType : UserProvidedToken  
78 CertificateName :  
79 Account :  
80 AppName : MS_TEAMS_ROPC  
81 ContextScope : Process  
82 Certificate :  
83 PSHostVersion : 6.7.1603.0  
84 ClientTimeout : 00:05:00  
85  
86  
87  
88  
89  
90  
91 ...done at 8/4/2022 1:25:54 PM (00:00:02.7650773).  
92
```

Figure 17: Connection test output

Once the connection has been successfully established, API permissions and ownership can be set up.

## 7 Customizing API permissions and ownership

This chapter describes the final steps required to perform a login in the ROPC workflow.

### 7.1 Adjusting API permissions

Open the settings of the registered service principal in the Azure Portal. In the left navigation bar, click **API Permissions**.

By default, only the **User.Read** permission is provided here:

Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. [Learn more about permissions and consent](#)

Grant admin consent for devhead

API / Permissions name	Type	Description	Admin consent requ...	Status
Microsoft Graph (1)				
User.Read	Delegated	Sign in and read user profile	No	...

To view and manage permissions and user consent, try [Enterprise applications](#).

Figure 18: Default permissions for Microsoft Graph

Click **+Add a permission** to add the following permissions as **Delegated** type:

- **Microsoft Graph**
  - App.Catalog.ReadWrite.All
  - Group.ReadWrite.All
  - User.Read
  - User.Read.All
- **Skype and Teams Tenant Admin AP**
  - user\_impersonation

Once all permissions are set correctly, click **Grant admin consent for <name>**.



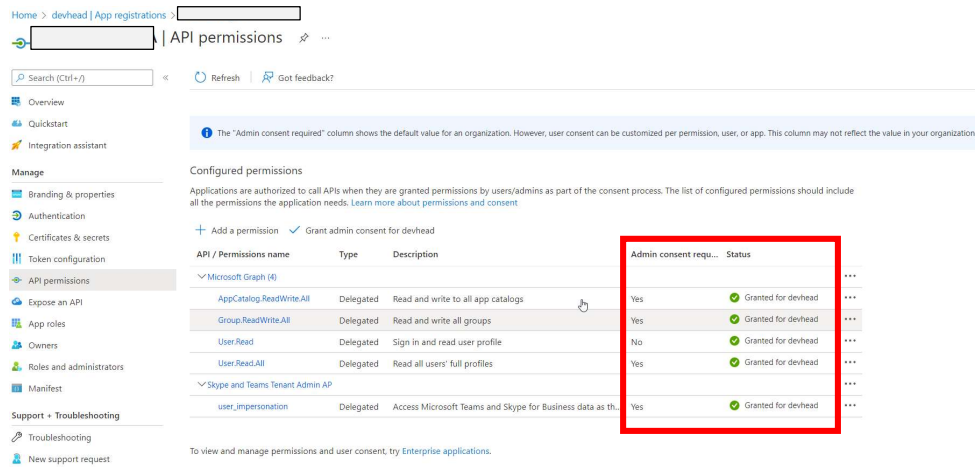


Figure 19: API permission with admin consent

## 7.2 Creating member users in Azure

You still need a separate owner. Do not use the global administrator account, but instead a member account.

It is entirely sufficient to set up a standard user without any additional rights.

## 7.3 Adding member user as owner

This user must be added as an owner. In the left navigation bar, click **Owners**.

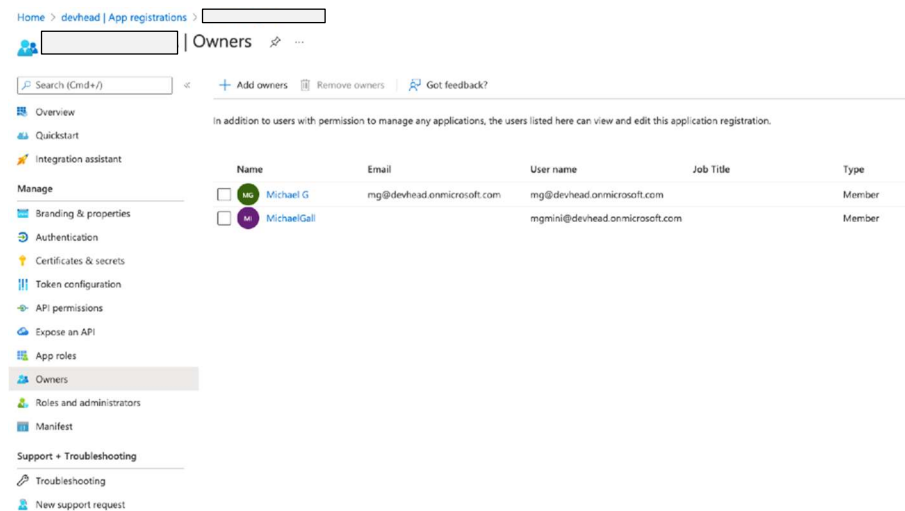


Figure 20: Adding the member user as an owner

## 8 Completing the ScriptRunner configuration

Store the credential of the user account in ScriptRunner and complete the configuration.

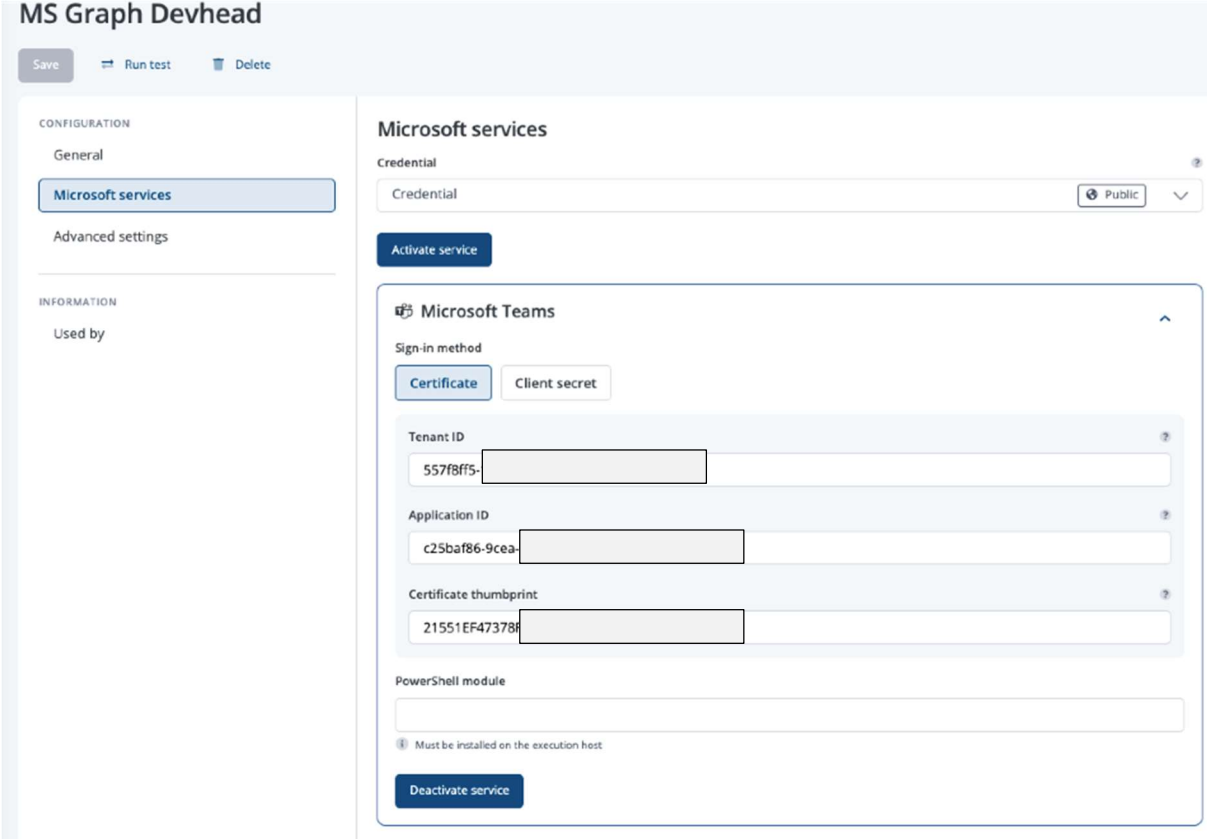
### 8.1 Creating a credential

Open the ScriptRunner Portal. In the **Credentials** section, enter the credential of the Azure account (member account).

### 8.2 Configuring the Microsoft Teams target

Create a Microsoft Teams target in the same way as explained in chapter 6 and copy the data from the Microsoft Graph target.

In the **Credential** field, select the credential you just created.



The screenshot shows the 'MS Graph Devhead' configuration page in ScriptRunner. The page is divided into two main sections: 'CONFIGURATION' and 'INFORMATION'. The 'CONFIGURATION' section has a sidebar with 'General' and 'Microsoft services' (selected). The 'Microsoft services' section is titled 'Microsoft services' and contains a 'Credential' dropdown menu with a 'Public' button and a 'Delete' icon. Below this is an 'Activate service' button. The 'Microsoft Teams' section is expanded, showing the 'Sign-in method' with 'Certificate' and 'Client secret' options. The 'Tenant ID' field contains '557f8ff5-'. The 'Application ID' field contains 'c25baf86-9cea-'. The 'Certificate thumbprint' field contains '21551EF47378f'. Below these fields is a 'PowerShell module' field. At the bottom of the section is a 'Deactivate service' button and a note: 'Must be installed on the execution host'.

Figure 21: Microsoft Teams target in ScriptRunner

Save your settings. Click **Run test** to perform a connection test.

### MS Graph Devhead ✕

PowerShell version

**Windows PowerShell** PowerShell 7

PowerShell modules

**Run test** ✓

### Console

```
3838
3839
3840
3841
3842
3843
3844
3845
3846 SRX: END Console Output
3847 SRX: ***** demosr01 done with 0 errors (00:00:08.6405807) ***** 8/4/2022
3848 SRX: Disconnecting from Microsoft Teams...
3849
3850
3851
3852
3853
3854 ...done at 8/4/2022 2:08:55 PM (00:00:09.3419472).
3855
```

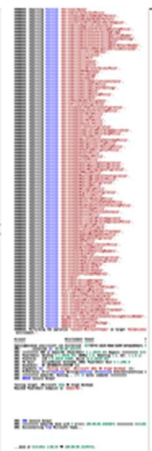


Figure 22: Connection test output

## 9 Checklist

### Checking requirements

- The ScriptRunner server is running in a [current version](#)\* (at least Portal Edition R4 Build 1603)
- Microsoft Teams PowerShell module >= version 4.5.0 is installed
- Microsoft Graph PowerShell module >= version 1.10.0 is installed

### Embedding/creating a certificate

- If available: Embed in Microsoft Azure
- If not yet available: Create a self-signed certificate with **-KeySpec Signature**
- Export the public key only

### Creating a service principal in Azure

- Create a service principal in Azure
- Save application ID, tenant ID and certificate thumbprint
- Upload certificate

### Creating a Microsoft Graph target system

- Create a Microsoft Graph target system in ScriptRunner using the data from the service principal
- **Credential** field must remain empty
- Run connection test

### Customizing API permissions

- See chapter 7.1

### Creating restricted Azure account (tenant member)

- See chapter **Fehler! Verweisquelle konnte nicht gefunden werden.**

### Adding service principal as owner

- See chapter **Fehler! Verweisquelle konnte nicht gefunden werden.**

### Finishing ScriptRunner configuration

- Create a member account under **Credentials**
- Transfer data from the MS Graph target to the MS Teams target
- In the **Credential** field, select the Azure account
- Perform connection test

## 10 Possible error sources

This chapter describes possible error sources and their solutions.

### 10.1 Conditional Access

Make sure that there are no rules in **Conditional Access** that restrict access.

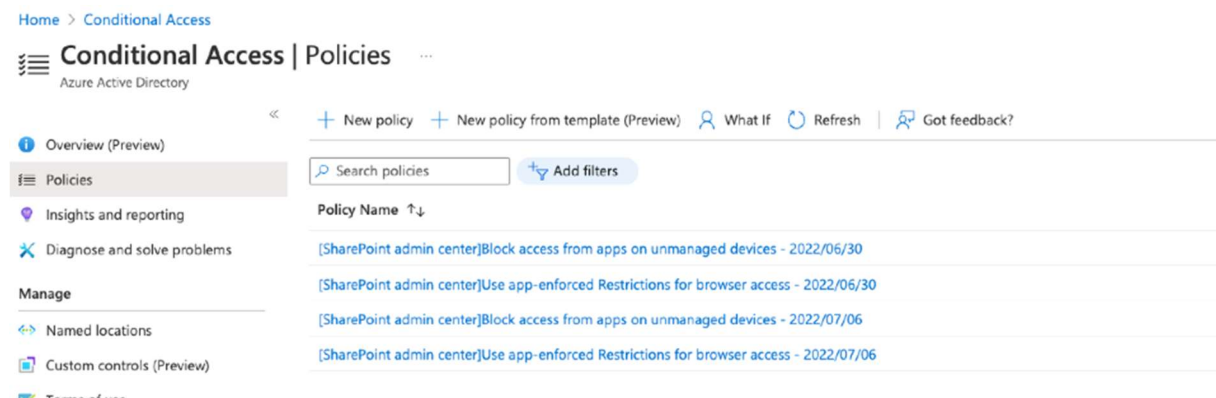


Figure 23: Potential restrictions through conditional access policies

You can check whether such a rule actually blocks access using **Monitoring > Sign-in logs**. Use the wizard to recreate the error in question.

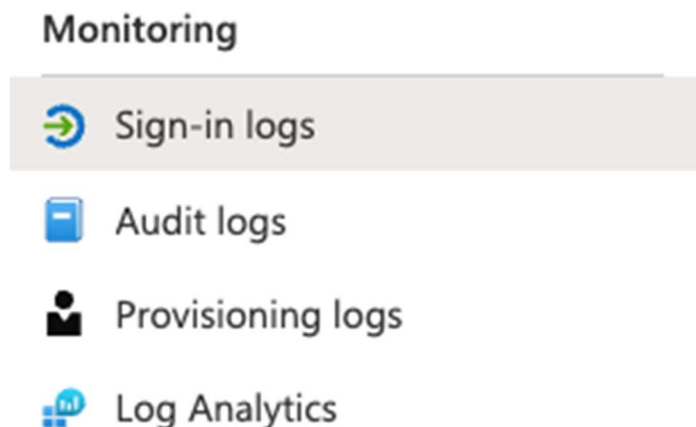


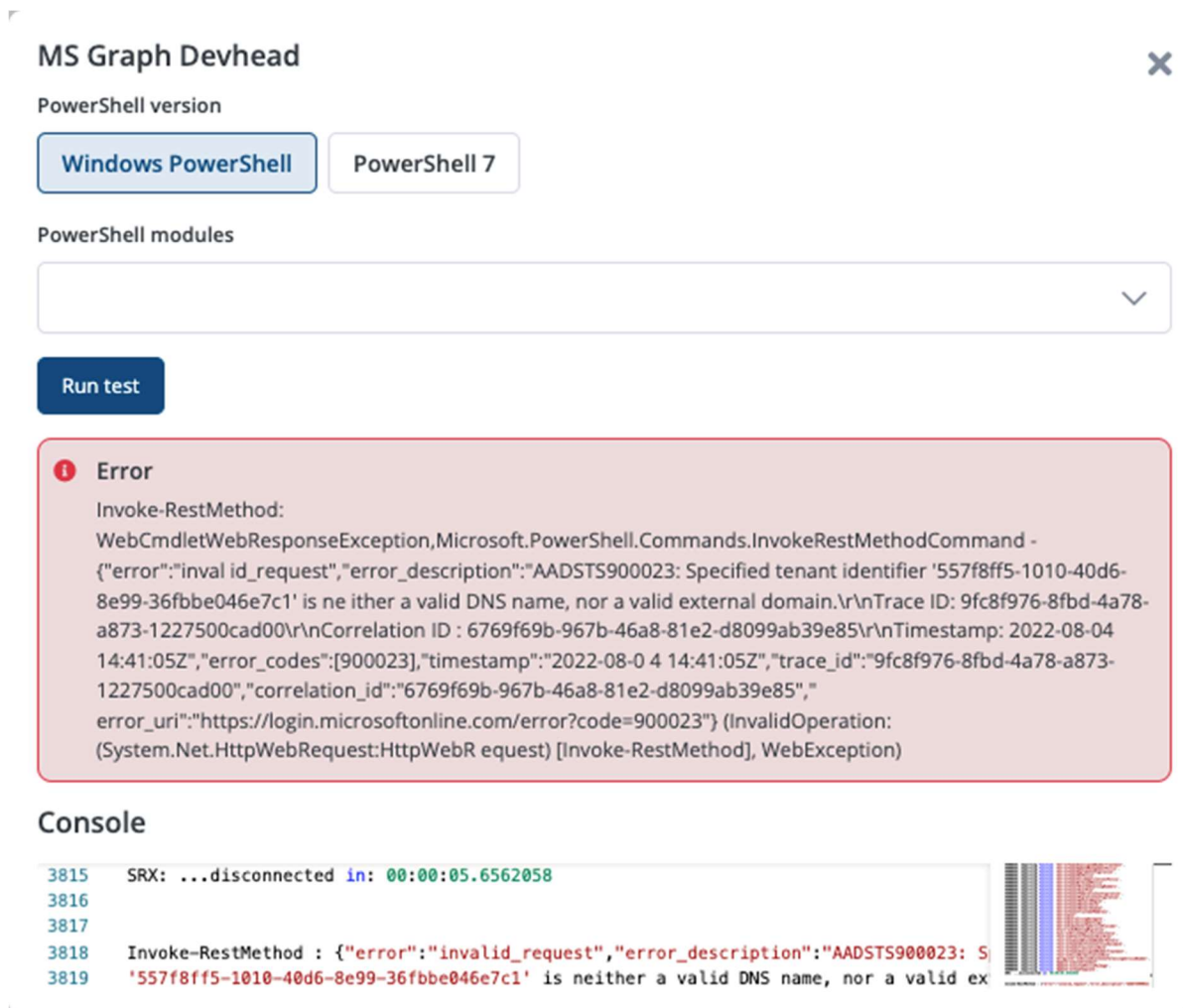
Figure 24: Sign-in logs in the Azure Portal

## 10.2 Problems with the login

Use the connection test in the target configuration to troubleshoot certificate problems. In all cases, the error messages indicate the problem. Common error sources are:

- The relevant PowerShell modules are missing
- The certificate thumbprint is not correct
- The tenant ID or application ID is incorrect

The error message is displayed in the upper area.



**MS Graph Devhead** ✕

PowerShell version

Windows PowerShell PowerShell 7

PowerShell modules

Run test

**Error**

Invoke-RestMethod:  
WebCmdletWebResponseException, Microsoft.PowerShell.Commands.InvokeRestMethodCommand -  
{\"error\": \"invalid\_request\", \"error\_description\": \"AADSTS900023: Specified tenant identifier '557f8ff5-1010-40d6-8e99-36fbbe046e7c1' is neither a valid DNS name, nor a valid external domain.\\r\\nTrace ID: 9fc8f976-8fbd-4a78-a873-1227500cad00\\r\\nCorrelation ID : 6769f69b-967b-46a8-81e2-d8099ab39e85\\r\\nTimestamp: 2022-08-04 14:41:05Z\", \"error\_codes\": [900023], \"timestamp\": \"2022-08-04 14:41:05Z\", \"trace\_id\": \"9fc8f976-8fbd-4a78-a873-1227500cad00\", \"correlation\_id\": \"6769f69b-967b-46a8-81e2-d8099ab39e85\", \"error\_uri\": \"https://login.microsoftonline.com/error?code=900023\"} (InvalidOperation: (System.Net.HttpWebRequest:HttpWebRequest) [Invoke-RestMethod], WebException)

**Console**

```
3815 SRX: ...disconnected in: 00:00:05.6562058
3816
3817
3818 Invoke-RestMethod : {\"error\": \"invalid_request\", \"error_description\": \"AADSTS900023: S
3819 '557f8ff5-1010-40d6-8e99-36fbbe046e7c1' is neither a valid DNS name, nor a valid ex
```

Figure 25: Error message in ScriptRunner portal - invalid tenant ID

## 11 Notes and references

### 11.1 Notes

The following Microsoft Azure users were used in this tutorial:

- [mg@devhead.onmicrosoft.com](mailto:mg@devhead.onmicrosoft.com) -> Tenant / global administrator
- [mgmini@devhead.onmicrosoft.com](mailto:mgmini@devhead.onmicrosoft.com) -> Simple user account in the tenant

Tenant information, tenant ID, application ID, and certificate thumbprints have been blurred.

### 11.2 References

#### **Microsoft Teams - ROPC Login:**

<https://docs.microsoft.com/de-de/azure/active-directory/develop/v2-oauth-ropc>

#### **Github Office\_Docs:**

<https://github.com/MicrosoftDocs/office-docs-powershell/blob/main/teams/teams-ps/teams/Connect-MicrosoftTeams.md>