Azure App Registration for Calendar Synchronization

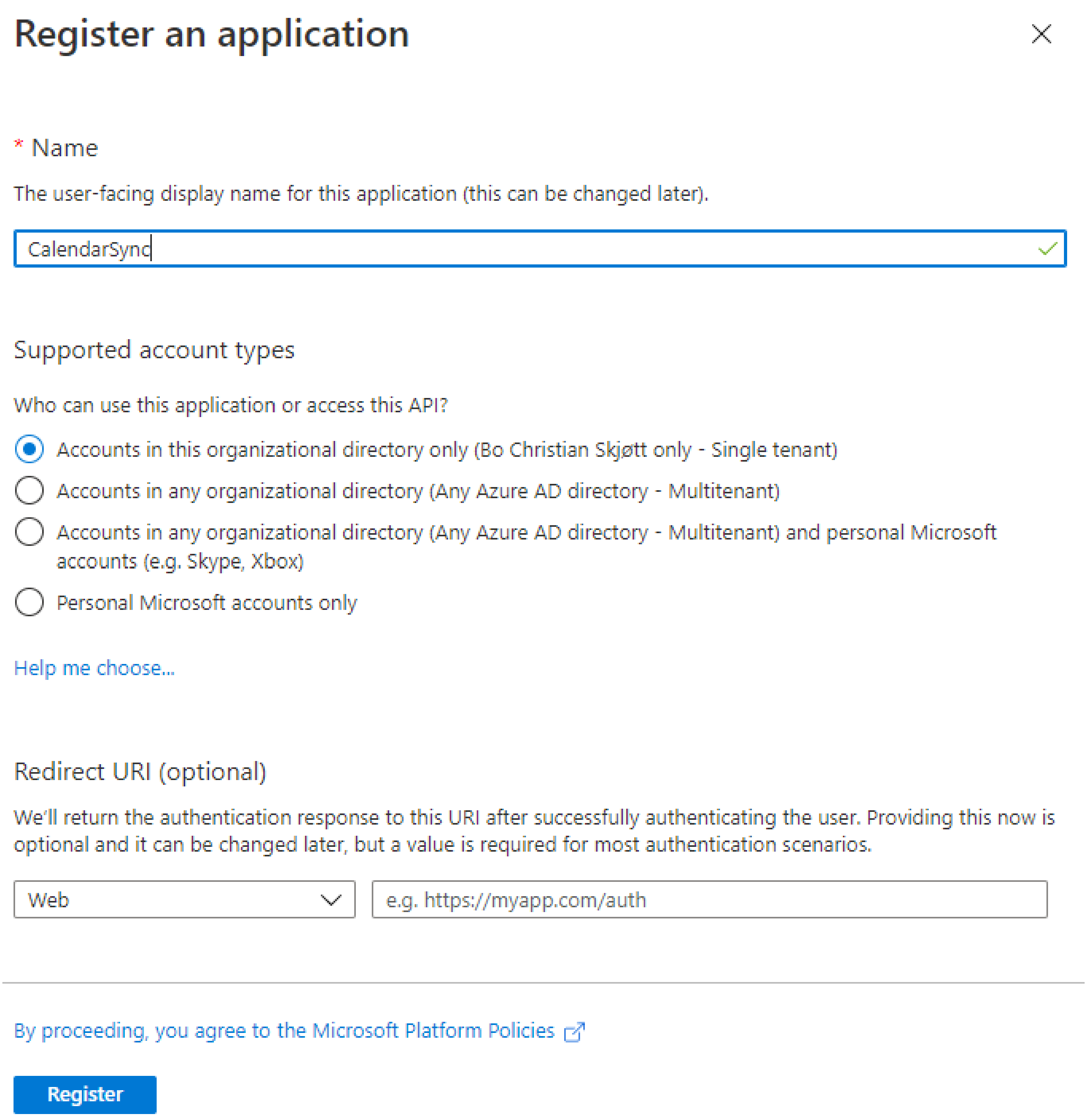
In order to be able to synchronize Microsoft calendar appointments via the Microsoft Grap API an Application must be created in Azure.

App creation in Azure portal

Go to your Active Directory in the Azure portal and follow the steps below.

Step 1. Register an App

Click App registrations, New registration.



Register an application

* Name

The user-facing display name for this application (this can be changed later).

CalendarSync

Supported account types

Who can use this application or access this API?

● Accounts in this organizational directory only (Bo Christian Skjøtt only - Single tenant)

○ Accounts in any organizational directory (Any Azure AD directory - Multitenant)

○ Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)

○ Personal Microsoft accounts only

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.

Web

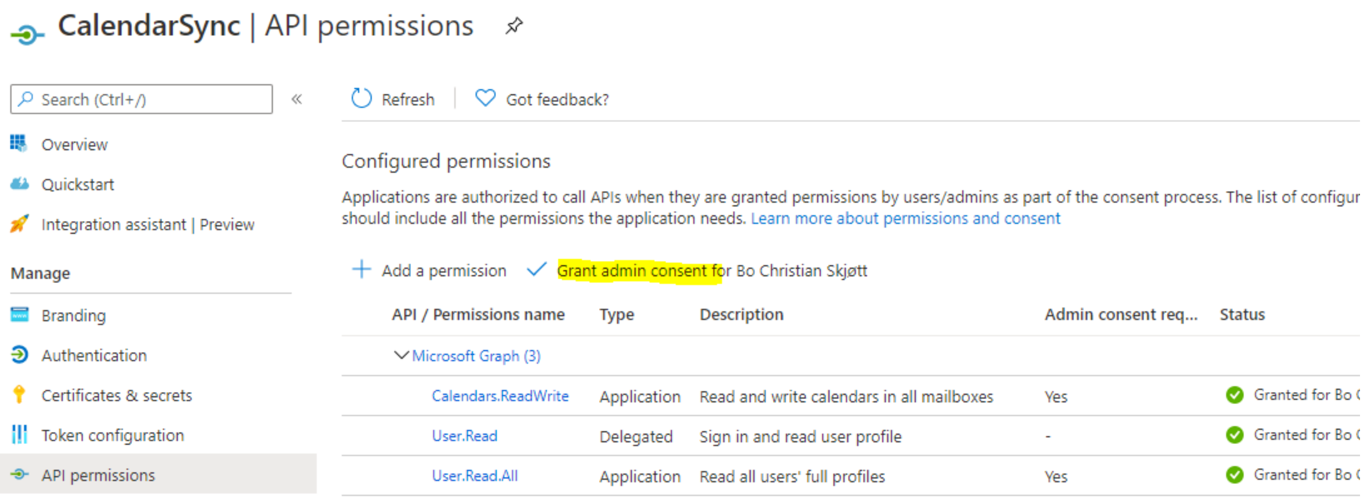
e.g. https://myapp.com/auth

By proceeding, you agree to the Microsoft Platform Policies

Click the Register button

Step 2. Add API permissions

Add the **User.Read.All** and **Calendar.ReadWrite** Graph API permissions



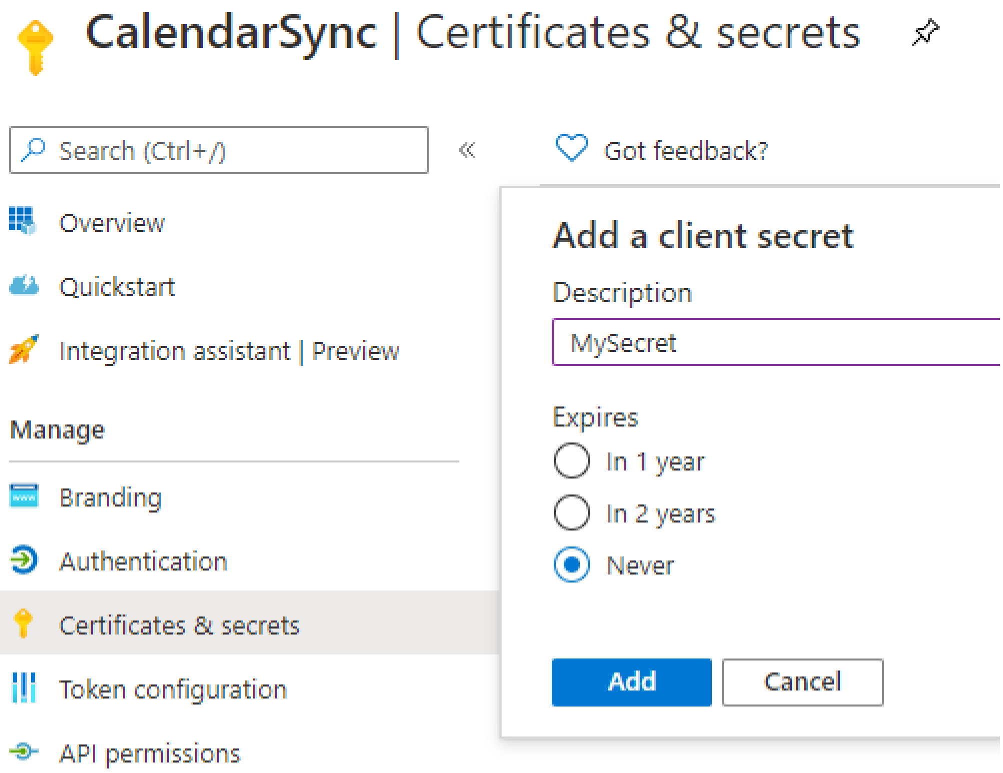
Configured permissions

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

API / Permissions name	Type	Description	Admin consent req...	Status
Microsoft Graph (3)				
Calendars.ReadWrite	Application	Read and write calendars in all mailboxes	Yes	✓ Granted for Bo (
User.Read	Delegated	Sign in and read user profile	-	✓ Granted for Bo (
User.Read.All	Application	Read all users' full profiles	Yes	✓ Granted for Bo (

Click on the **"Grant admin consent for ..."** button.

Step 3. Add a Client Secret



Add a client secret

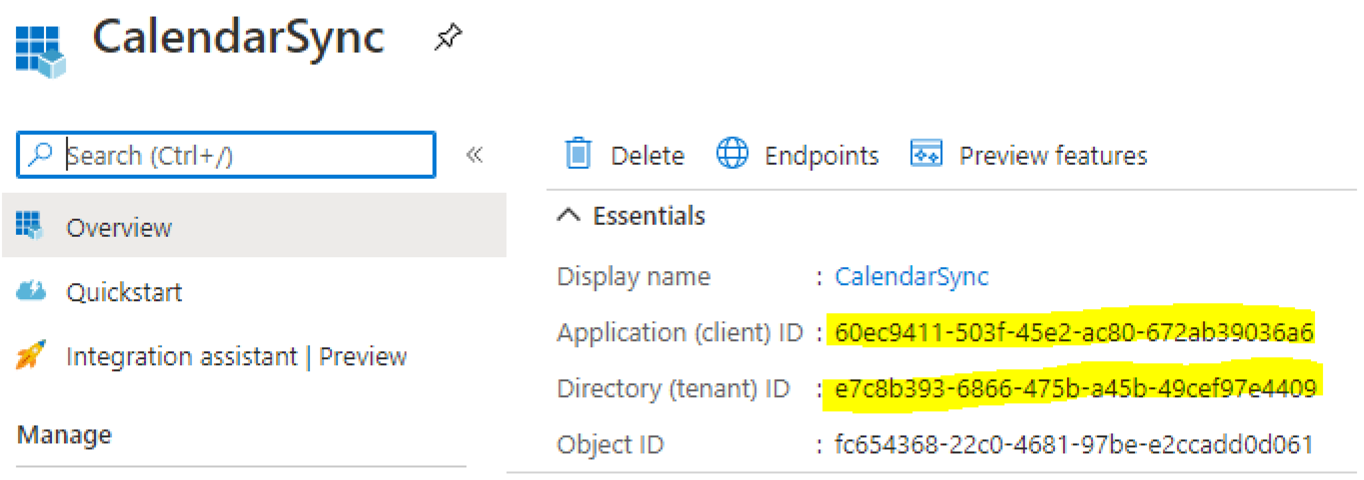
Description: MySecret

Expires
○ In 1 year
○ In 2 years
● Never

Take a copy of the generated secret. It is only shown during creation.

Step 4. Get the Client and Tenant IDs for the application

Go to the Application Overview page and copy the Client ID and Tenant ID. You need to enter these in the Novus Configuration UI along with the Client Secret.



Display name : CalendarSync

Application (client) ID : 60ec9411-503f-45e2-ac80-672ab39036a6

Directory (tenant) ID : e7c8b393-6866-475b-a45b-49cef97e4409

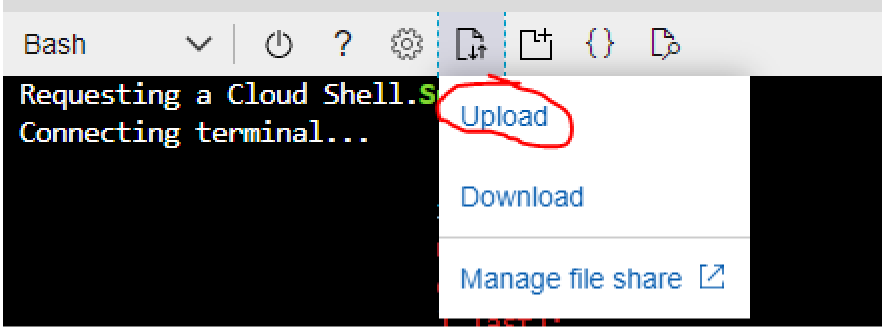
Object ID : fc654368-22c0-4681-97be-e2ccadd0d061

App creation using Azure CLI commands

The steps above can also be done with the Azure CLI commands below.

Copy the **requiredResourceAccess.json** file to the Azure storage (clouddrive) used by Azure CLI.

If you are using the **Cloud Shell** in the Azure Portal then you can click on the Upload File button in its menubar as shown below



```
Requesting a Cloud Shell.S
Connecting terminal...
```

Create an application with this command

```
az ad app create \
--display-name CalendarSync \
--password VerySecretWord#1234 \
--end-date 2100-12-31 \
--required-resource-accesses requiredResourceAccess.json
```

Replace the password with your choice.

Grant admin consent for the requested API permissions with this command

```
az ad app permission admin-consent --id 00000000-0000-0000-0000-000000000000
```

where 00000000-0000-0000-0000-000000000000 must be replaced with the actual ID of the application created above.

(Note: the **az ad app permission admin-consent** fails with an exception)