imc
part of Scheer

MS Teams Integration

LMS Teams Integration Prerequisites
imc Learning Suite

im-c.com

MS Teams Integration

LMS Teams Integration Prerequisites
imc Learning Suite

Author(s): Andreas Pohl
Date: 2022-03-30

Document	Description
Version	ILS 14.12
Status (Draft / Review / Finalisation)	Finalisation
Contact Person(s)	Andreas Pohl

History	Status	Who
2020-07-02	Draft	Andreas Pohl
2021-11-10	Review	Lia Ghiţă
2022-03-30	Review	Nadine Gohr
2022-03-30	Finalisation	Dr. Peter Zönnchen

imc
information multimedia communication AG
Headquarter Saarbrücken
Scheer Tower, Uni-Campus Nord
D-66123 Saarbrücken
T. +49 681 9476-0 | Fax -530
info@im-c.com
im-c.com

Content

1 Description

This document provides information on integration steps for MS Teams APIs and is divided among three sections as:

- Creation of new account on Azure portal along with steps to register a new application.
- Adding permissions to the registered application.

2 Prerequsites / Limitations

- The service (social-integration-backend) needs to be configured to be part of the delivery package.
- A MS Teams tenant is needed (usually included in an Office 365 subscription / tenant).
- A technical user / app needs to be created to access the Microsoft Graph API with specifc permissions.
- The frontend URL need to be configured in the client properties.
- Only internal users belonging to the company's tenant can be added to the teams - no guests.
- The following MS Teams administration rights are required:
  `Group.Read.All`, `Group.ReadWrite.All`, `TeamsTab.ReadWrite`
  (see https://docs.microsoft.com/en-us/graph/permissions-reference for more details).

3 Functional Summary

- Sharing of visual representation of courses into Teams and Channels.
- Creation of an individual Teams Team per Course from within Course's Creation.
- Representation of the Course Content (Syllabus) inside the Course‘s Team.

4 Configuration

4.1 Creation of the technical user / app

4.1.1 System Requirements

Application / Web Server

Product	Version	Fix Level
Tomcat	9	

4.1.2 Technology Stack

The following table provides a list of the technologies used for the implementation of MS Teams API's.

Name	Version	Comments
Java	1.8	
Spring boot	2.1.8	
Gradle	4.10	
Microsoft Graph SDK	1.0	
Junit	4.12	
lombok	1.18.10	

4.1.3 Integration Steps

Register New Application on Azure

To register a new application using the Azure Portal follow these steps:

1. Sign in to the Azure portal using either a work or school account or a personal Microsoft account.
2. If your account gives you access to more than one tenant, select your account in the top right corner, and set your portal session to the Azure AD tenant that you want.

3. In the left-hand navigation pane, select the **Azure Active Directory** service, and then select **App registrations** > **New registration**.
4. When the **Register an application** page appears, enter your application's registration information:
   - **Name** - Enter a meaningful application name that will be displayed to users of the app.
   - **Supported account types** - Select which accounts you would like your application to support.

Supported Account Types	Description
Accounts in this organizational directory only	Select this option if you're building a line-of-business (LOB) application. This option is not available if you're not registering the application in a directory. This option maps to Azure AD only single-tenant. This is the default option unless you're registering the app outside of a directory. In cases where the app is registered outside of a directory, the default is Azure AD multi-tenant and personal Microsoft accounts.
Accounts in any organizational directory	Select this option if you would like to target all business and educational customers. This option maps to an Azure AD only multi-tenant. If you registered the app as Azure AD only single-tenant, you can update it to be Azure AD multi-tenant and back to single-tenant through the Authentication blade.
Accounts in any organizational directory and personal Microsoft accounts	Select this option to target the widest set of customers. This option maps to Azure AD multi-tenant and personal Microsoft accounts. If you registered the app as Azure AD multi-tenant and personal Microsoft accounts, you cannot change this in the UI. Instead, you must use the application manifest editor to change the supported account types

- **Redirect URI** (optional) - Select the type of app you're building, **Web** or **Public client** (mobile & desktop), and then enter the redirect URI (or reply URL) for your application. This option is optional and can be skipped.

5. When finished, select **Register**.

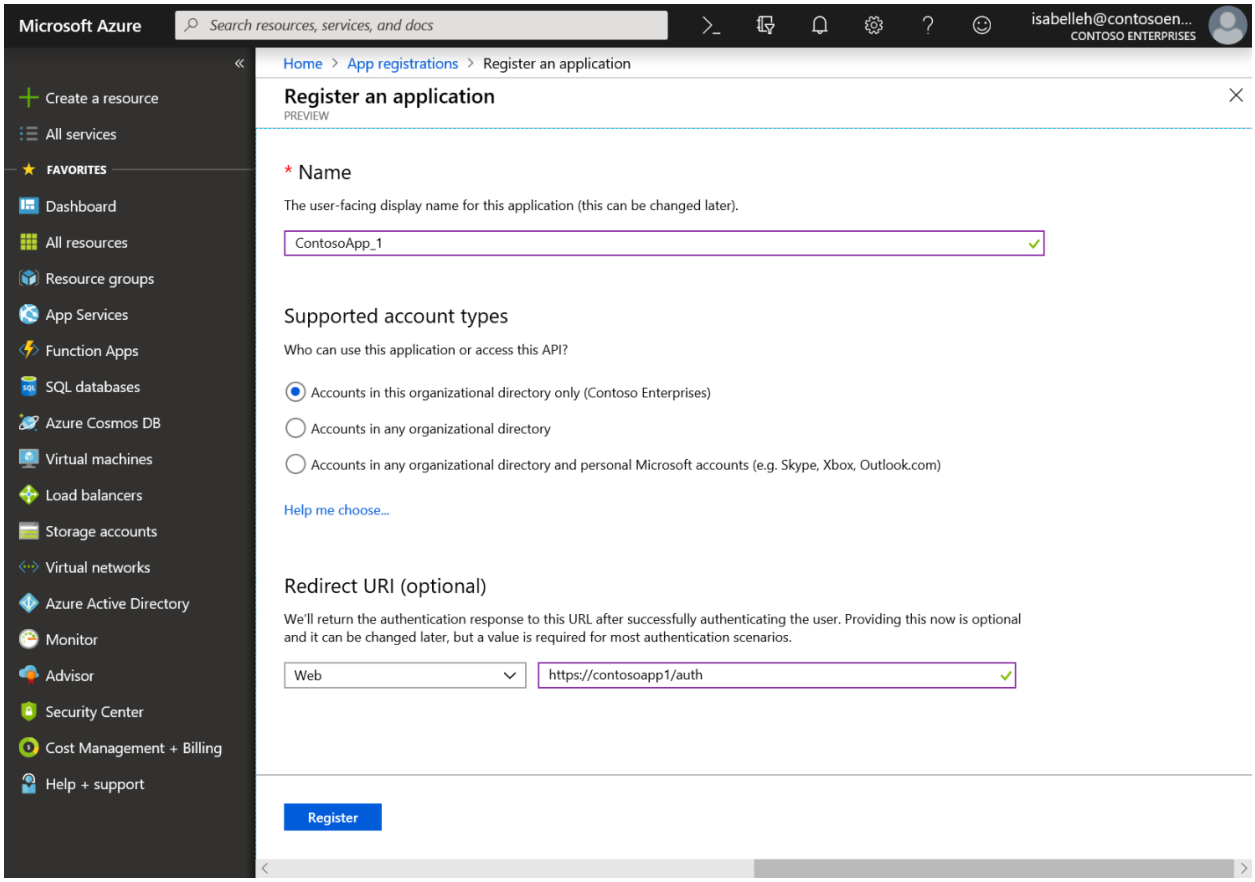

Fig. 4.1: Register an application.

Azure AD assigns a unique application (client) ID to your app, and you're taken to your application's Overview page. To add additional capabilities to your application, you can select other configuration options including branding, certificates and secrets, API permissions, and more.

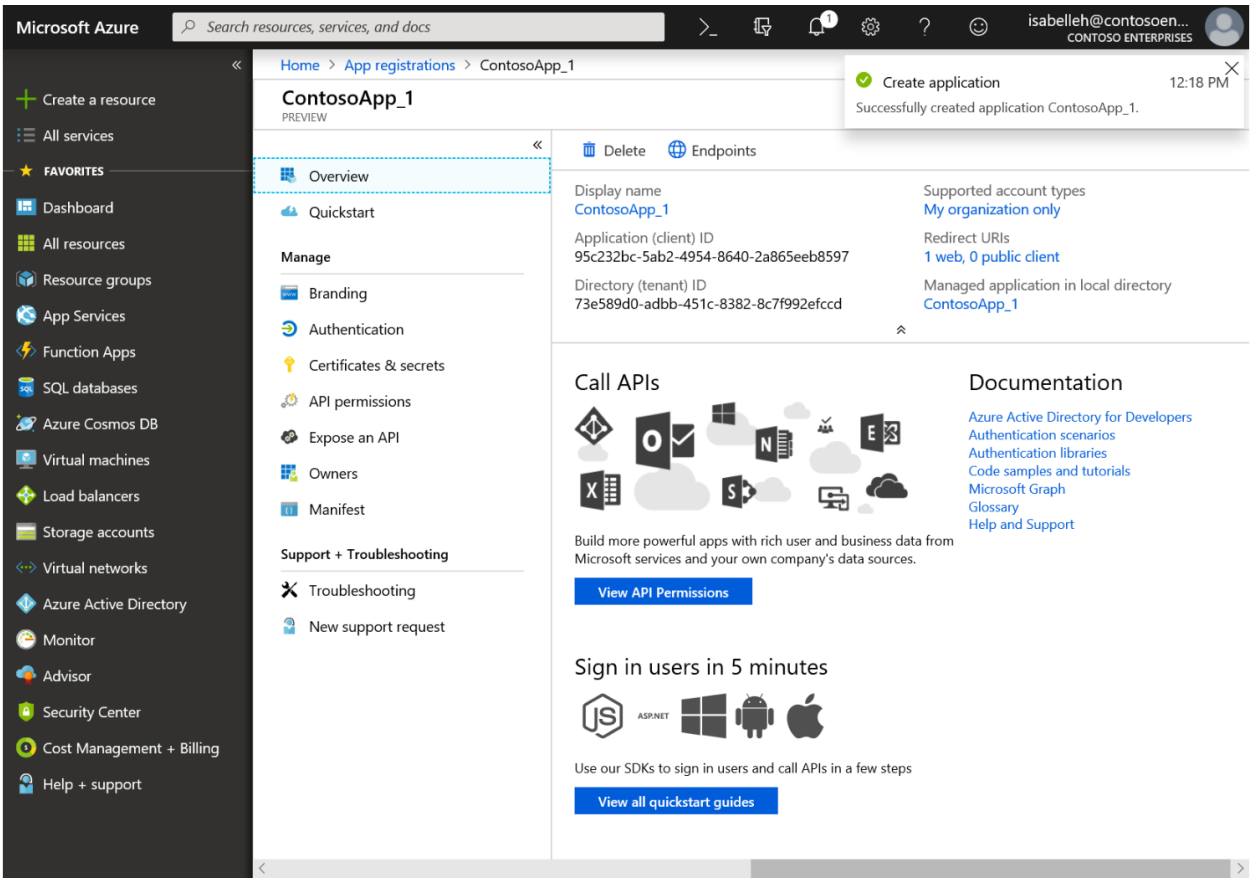

Fig. 4.2: New registered application – Overview.

4.1.4 Adding Permissions on Registered Application

Permissions are required to the registered application so as to create a new group, add and remove members from a group and adding tabs to channels. To add permissions, perform below steps:

1. In the left-hand navigation pane, select the Azure Active Directory service, and then select App registrations. This will list all the registered applications. Now select the newly registered application. This will redirect to the overview page of application.
2. In left-hand navigation pane select API Permissions.
3. Click on Add a permission button and assign below permissions to your app at Application level and Grant admin concent:
   - `Group.Create`
   - `Group.Read.All`
   - `Group.ReadWrite.All`
   - `GroupMember.ReadWrite.All`
   - `TeamsTab.Create`
   - `TeamsTab.ReadWrite.All`
   - `User.Read.All`

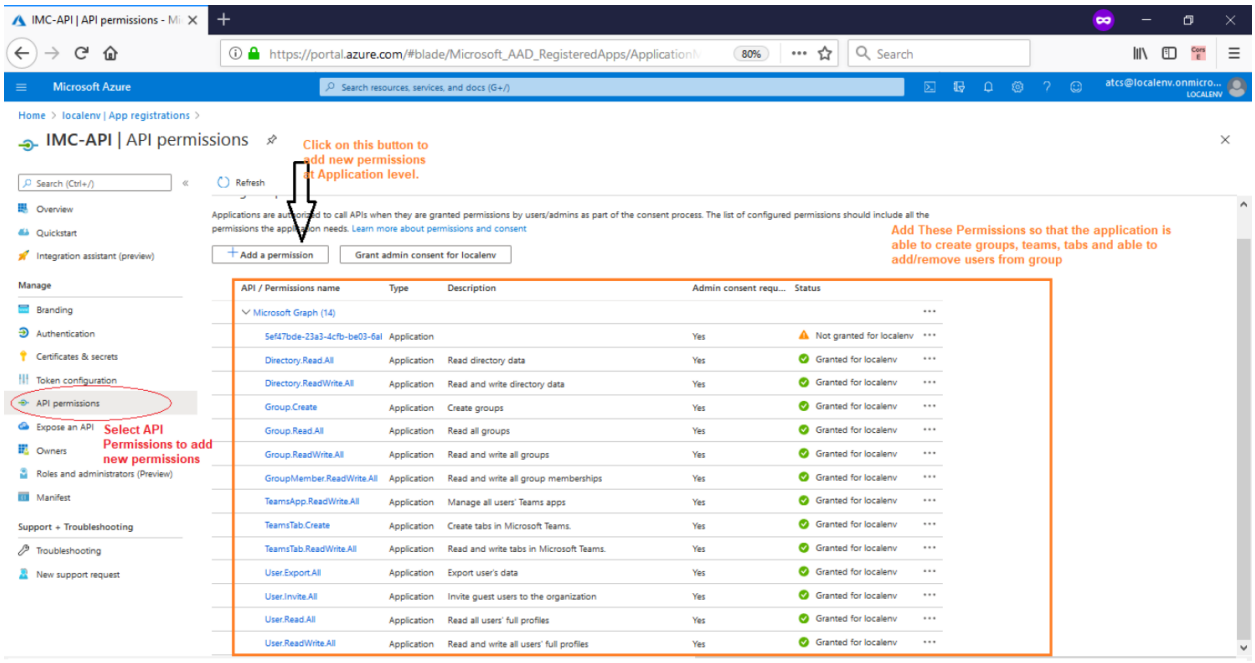

Fig. 4.3: API permissions.

4.1.5 MS Teams application configuration

To configure Back-end application follow below steps:

1. Under the project root directory go to **`src > main > resources`** and edit these properties files (application-dev.properties, application-prod.properties and application-test.properties).
2. Replace `<graph-client-id>` with **Application (client) ID** of the registered application.
3. Replace `<graph-authority>` with **Directory (tenant) ID** of the registered application.
4. Replace `<graph-client-secret>` with **client secret key**. To generate client secret, navigate to overview page of application and select **Certificates & secrets**. Under Client secrets click on **New Client Secret** Button and select **Add**. This will generate key. Refer Screenshot for more info:

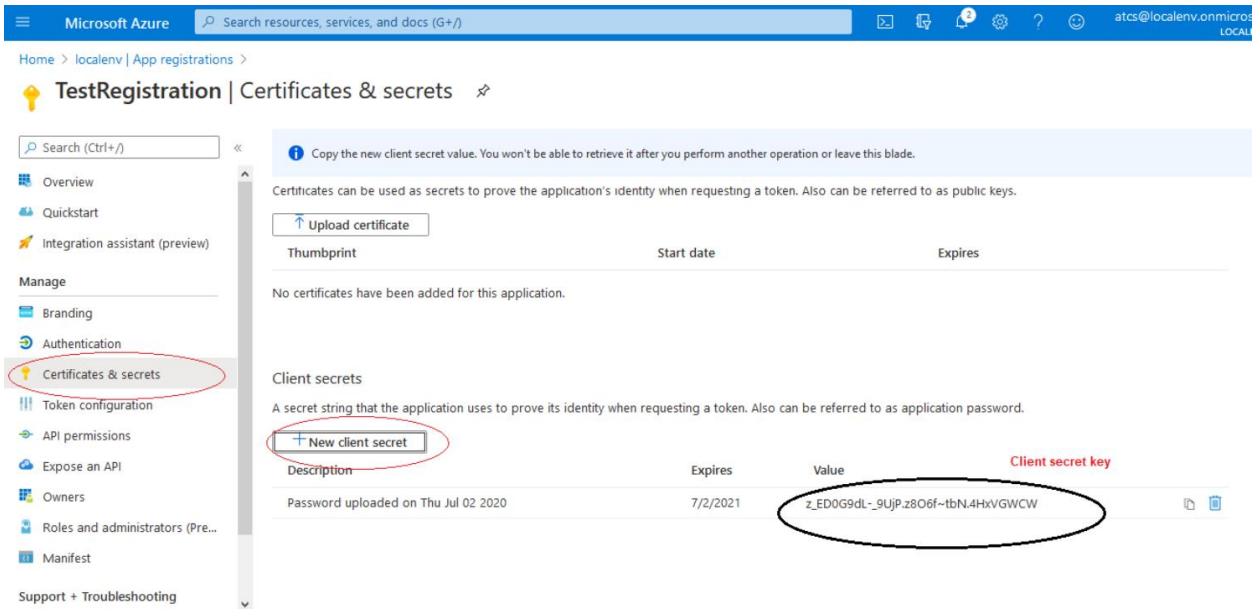

Fig. 4.4: Certificates & secrets.

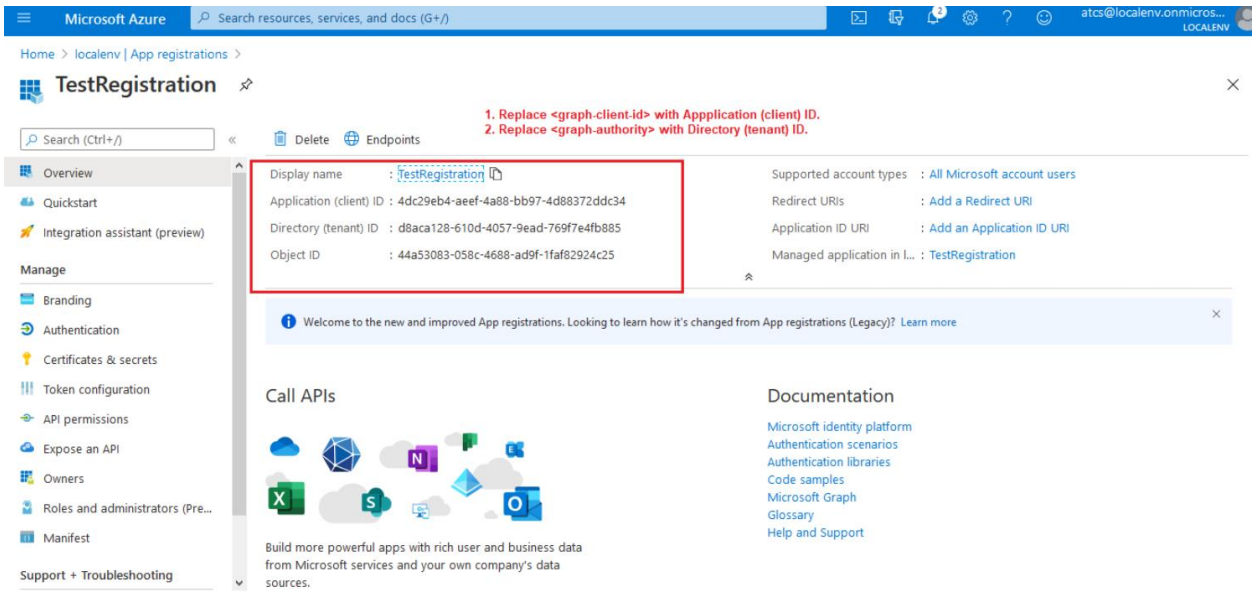

Fig. 4.5: New client secret.

4.1.6 REST API Documentation

Documentation of API's for teams can be found in Back-end project structure under `doc` folder in root directory.

- For Postman collection refer file: `IMC_Teams_Postman_Collection.json`.
- For Swagger documentation refer file: `swagger-document.json`.

4.2 Known Issues

- MS Teams owners are created from Administrator and Tutors on the first saving, they are currently not updated later
- Tab name only in one language; will be moved to a bundle
- Teams are not removed when course is deleted / archived - Teams can be deleted by owners inside Teams.