imc
part of Scheer

MS Teams Integration

imc Learning Suite

im-c.com

MS Teams Integration

imc Learning Suite

Author(s): Andreas Pohl, Dr. Peter Zönnchen
Date: 2020-10-19

Document	Description
Version	ILS 14.7
Status (Draft / Review / Finalisation)	Finalisation
Contact Person(s)	Andreas Pohl

History	Status	Who
2020-07-02	Draft	Andreas Pohl
2020-10-19	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.
- Configuration on back-end application along with Postman & Swagger document for using APIs.

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.

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:

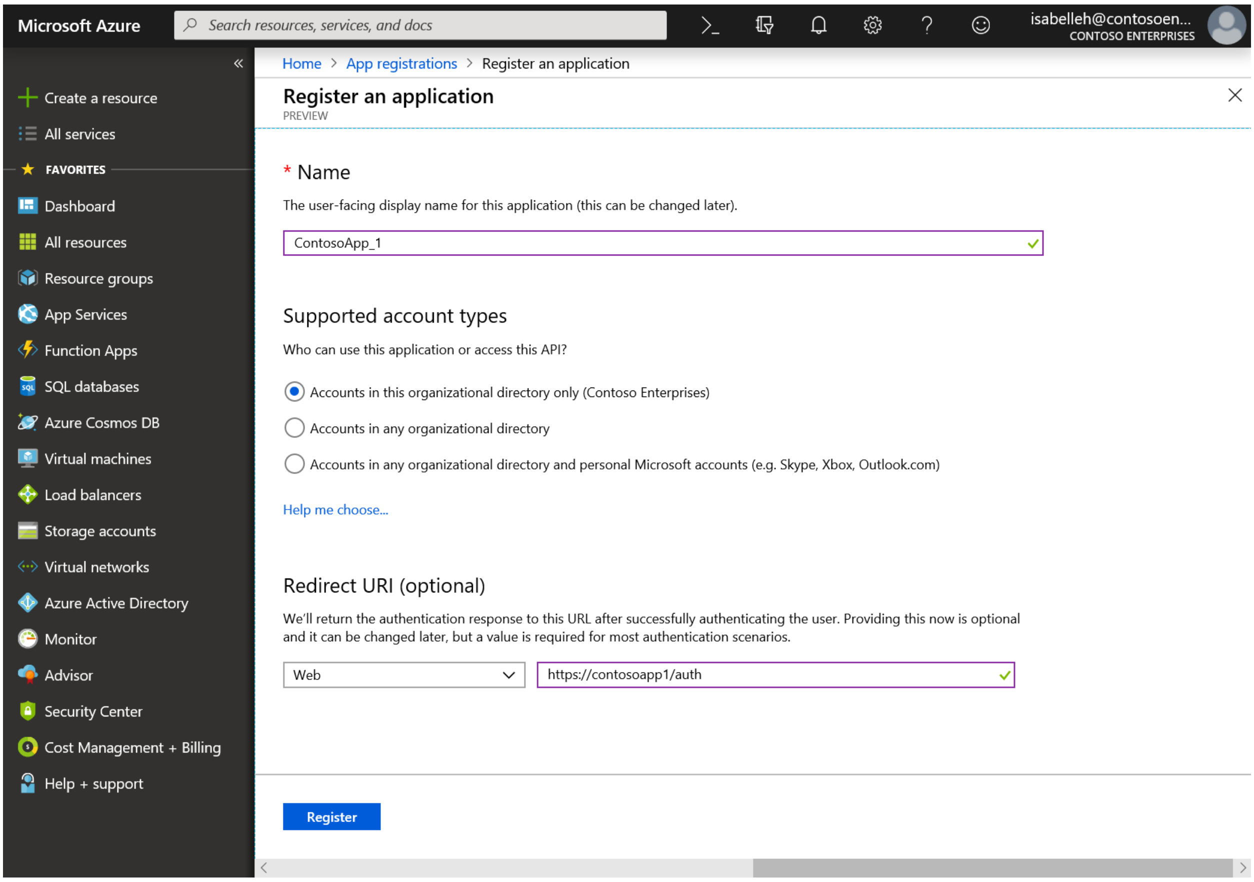
1. Sign in to the Azure portal using either a work or school account or a personal Microsoft account.
2. If your account gives you access to more than one tenant, select your account in the top right corner, and set your portal session to the Azure AD tenant that you want.

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**.



Register an application
PREVIEW

* Name

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

ContosoApp_1 ✓

Supported account types

Who can use this application or access this API?

● Accounts in this organizational directory only (Contoso Enterprises)
○ Accounts in any organizational directory
○ Accounts in any organizational directory and personal Microsoft accounts (e.g. Skype, Xbox, Outlook.com)

Help me choose...

Redirect URI (optional)

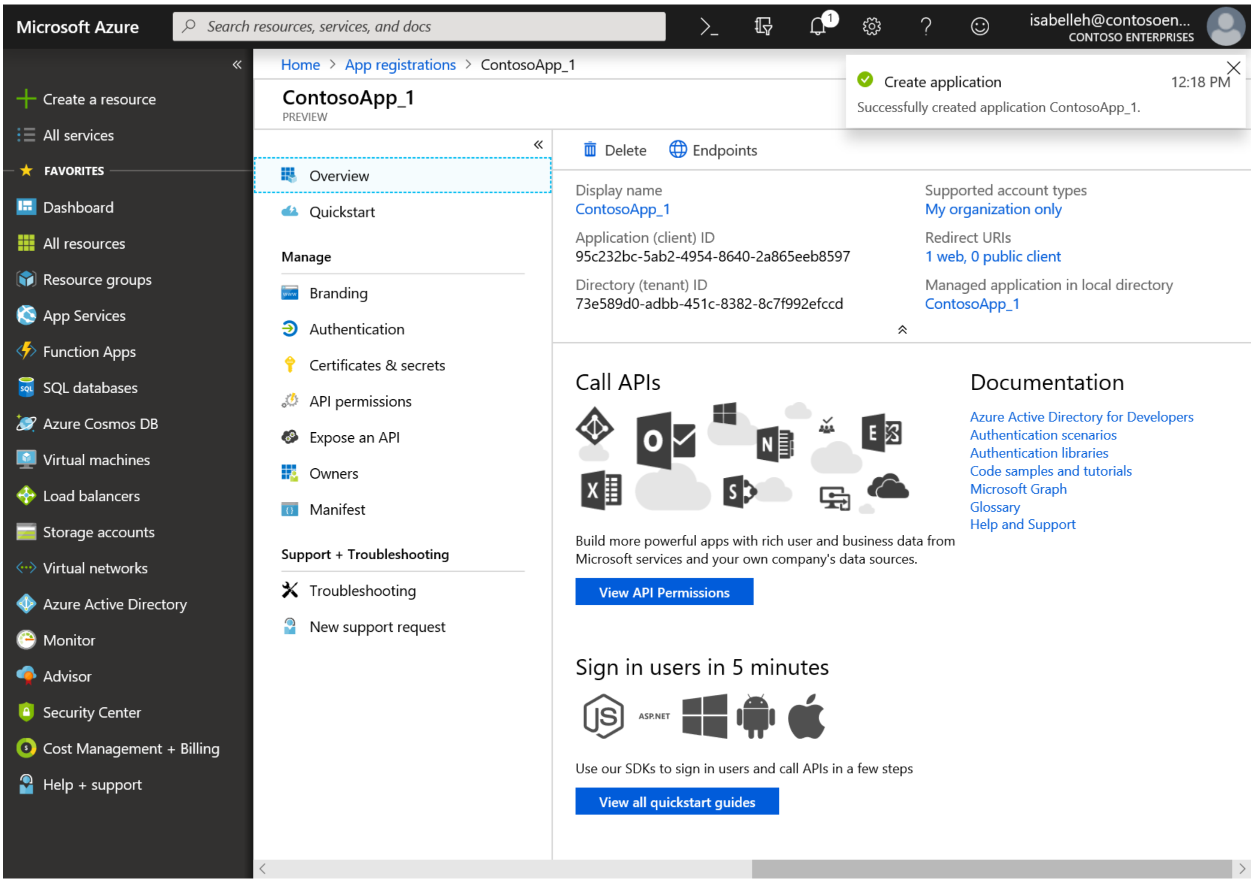
We'll return the authentication response to this URL after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

Web | https://contosoapp1/auth ✓

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.



Microsoft Azure   Search resources, services, and docs   isabelleh@contosoen... CONTOSO ENTERPRISES

- Create a resource
- All services
- FAVORITES
- Dashboard
- All resources
- Resource groups
- App Services
- Function Apps
- SQL databases
- Azure Cosmos DB
- Virtual machines
- Load balancers
- Storage accounts
- Virtual networks
- Azure Active Directory
- Monitor
- Advisor
- Security Center
- Cost Management + Billing
- Help + support

Home > App registrations > ContosoApp_1

ContosoApp_1
PREVIEW

Create application   12:18 PM
Successfully created application ContosoApp_1.

- Overview
- Quickstart

Manage

- Branding
- Authentication
- Certificates & secrets
- API permissions
- Expose an API
- Owners
- Manifest

Support + Troubleshooting

- Troubleshooting
- New support request

Delete   Endpoints

Display name ContosoApp_1	Supported account types My organization only
Application (client) ID 95c232bc-5ab2-4954-8640-2a865eeb8597	Redirect URIs 1 web, 0 public client
Directory (tenant) ID 73e589d0-adbb-451c-8382-8c7f992efccd	Managed application in local directory ContosoApp_1

Call APIs

Build more powerful apps with rich user and business data from Microsoft services and your own company's data sources.

View API Permissions

Documentation

- Azure Active Directory for Developers
- Authentication scenarios
- Authentication libraries
- Code samples and tutorials
- Microsoft Graph
- Glossary
- Help and Support

Sign in users in 5 minutes

Use our SDKs to sign in users and call APIs in a few steps

View all quickstart guides

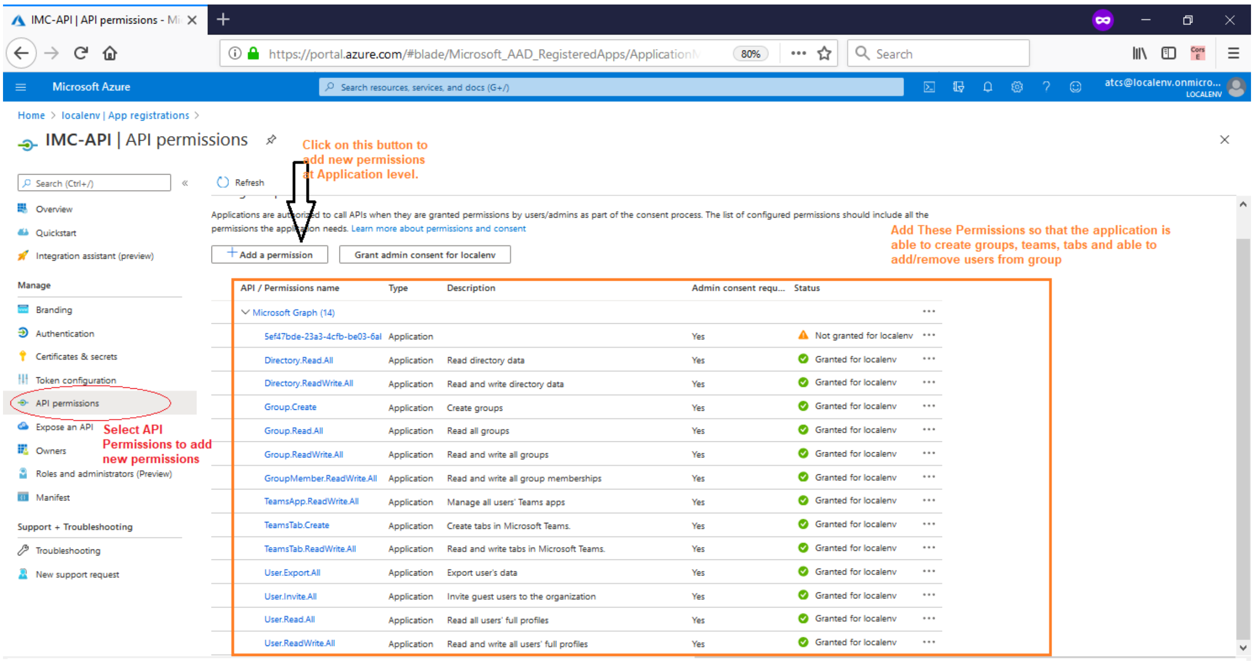
Fig. 4.2: New registered application – Overview.

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**:
   - `Directory.Read.All`
   - `Directory.ReadWrite.All`
   - `Group.Create`
   - `Group.Read.All`
   - `Group.ReadWrite.All`
   - `GroupMember.ReadWrite.All`

- `TeamsApp.ReadWrite.All`
- `TeamsTab.Create`
- `TeamsTab.ReadWrite.All`
- `User.Export.All`
- `User.Invite.All`
- `User.Read.All`
- `User.ReadWrite.All`


IMC-API | API permissions - Mi
https://portal.azure.com/#blade/Microsoft_AAD_RegisteredApps/Application
80%
Search
Microsoft Azure
Search resources, services, and docs (G+/)
atcs@localenv.onmicro...
LOCALENV
Home > localenv | App registrations >
IMC-API | API permissions
Search (Ctrl+/)
Overview
Quickstart
Integration assistant (preview)
Manage
Branding
Authentication
Certificates & secrets
Token configuration
API permissions
Expose an API
Owners
Roles and administrators (Preview)
Manifest
Support + Troubleshooting
Troubleshooting
New support request
Refresh
Add a permission
Grant admin consent for localenv

Click on this button to add new permissions at Application level.

Select API Permissions to add new permissions

Add These Permissions so that the application is able to create groups, teams, tabs and able to add/remove users from group

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

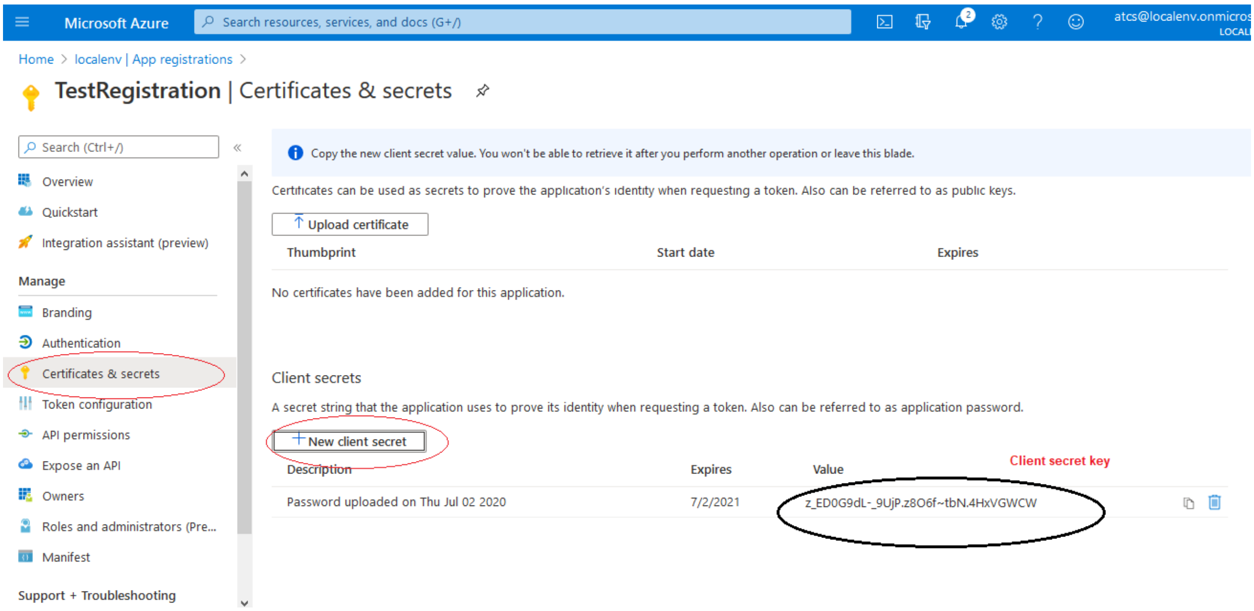
API / Permissions name	Type	Description	Admin consent requ...	Status
Microsoft Graph (14)				
5ef47bde-23a3-4cfb-be03-6al	Application		Yes	Not granted for localenv
Directory.Read.All	Application	Read directory data	Yes	Granted for localenv
Directory.ReadWrite.All	Application	Read and write directory data	Yes	Granted for localenv
Group.Create	Application	Create groups	Yes	Granted for localenv
Group.Read.All	Application	Read all groups	Yes	Granted for localenv
Group.ReadWrite.All	Application	Read and write all groups	Yes	Granted for localenv
GroupMember.ReadWrite.All	Application	Read and write all group memberships	Yes	Granted for localenv
TeamsApp.ReadWrite.All	Application	Manage all users' Teams apps	Yes	Granted for localenv
TeamsTab.Create	Application	Create tabs in Microsoft Teams.	Yes	Granted for localenv
TeamsTab.ReadWrite.All	Application	Read and write tabs in Microsoft Teams.	Yes	Granted for localenv
User.Export.All	Application	Export user's data	Yes	Granted for localenv
User.Invite.All	Application	Invite guest users to the organization	Yes	Granted for localenv
User.Read.All	Application	Read all users' full profiles	Yes	Granted for localenv
User.ReadWrite.All	Application	Read and write all users' full profiles	Yes	Granted for localenv

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:



Copy the new client secret value. You won't be able to retrieve it after you perform another operation or leave this blade.

Certificates can be used as secrets to prove the application's identity when requesting a token. Also can be referred to as public keys.

Thumbprint	Start date	Expires

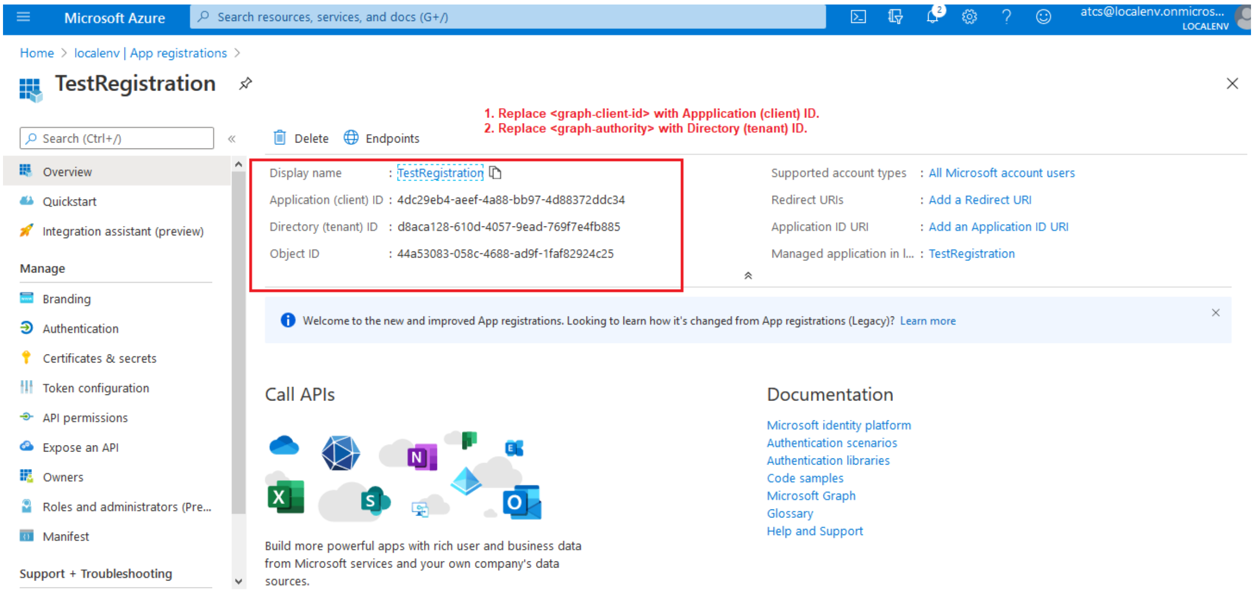
No certificates have been added for this application.

Client secrets

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

Description	Expires	Value
Password uploaded on Thu Jul 02 2020	7/2/2021	z_ED0G9dL-_9UjP.z8O6f~tbN.4HxVGWCW

Fig. 4.4: *Certificates & secrets*



1. Replace <graph-client-id> with Appplication (client) ID.
2. Replace <graph-authority> with Directory (tenant) ID.

Display name	: TestRegistration	Supported account types	: All Microsoft account users
Application (client) ID	: 4dc29eb4-aeef-4a88-bb97-4d88372ddc34	Redirect URIs	: Add a Redirect URI
Directory (tenant) ID	: d8aca128-610d-4057-9ead-769f7e4fb885	Application ID URI	: Add an Application ID URI
Object ID	: 44a53083-058c-4688-ad9f-1faf82924c25	Managed application in l...	: TestRegistration

Welcome to the new and improved App registrations. Looking to learn how it's changed from App registrations (Legacy)? Learn more

Build more powerful apps with rich user and business data from Microsoft services and your own company's data sources.

Fig. 4.5: *New client secret.*

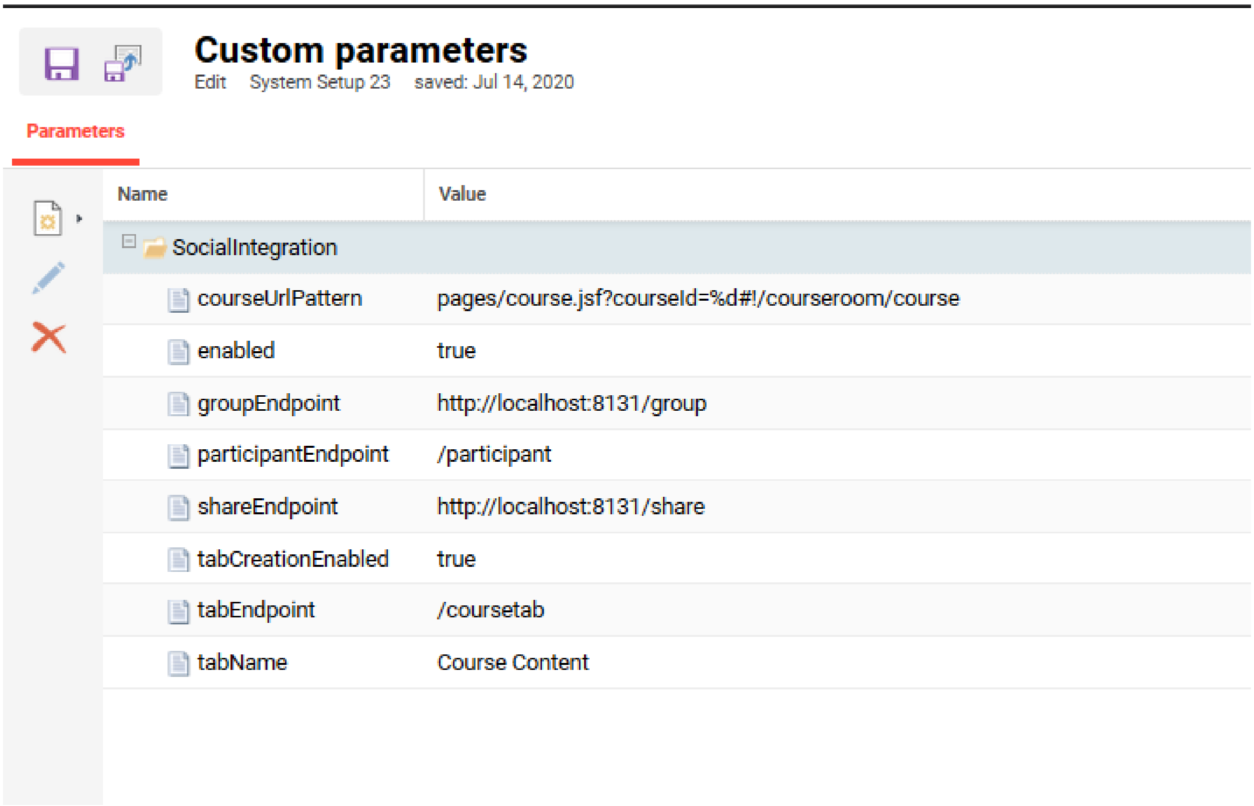
4.1.6 REST API Documentation

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 Service configuration inside the ILS

At the current state, the configuration is done in "Custom parameters" and will be moved to an individual configuration section in the configuration manager.
Please see the following screen shot for an example configuration:



Name	Value
SocialIntegration	
courseUrlPattern	pages/course.jsf?courseId=%d#!/courseroom/course
enabled	true
groupEndpoint	http://localhost:8131/group
participantEndpoint	/participant
shareEndpoint	http://localhost:8131/share
tabCreationEnabled	true
tabEndpoint	/coursetab
tabName	Course Content

Fig. 4.6: “Custom parameters” inside the ILS.

The detailed explanation of the parameters is given in below table:

Parameter / Division	Description	Example Value
SocialIntegration	Division to hold the needed.	`Parameters`
enabled	Used to generally enable or disable the integration	`true`
shareEndPoint	API endpoint used for sharing; including external domain.	`https://achievemore.im-c.com/share`
groupEndpoint	API endpoint to create groups (i.e. teams); including internal domain and port.	`http://localhost:8083/group`
tabCreationEnabled	Used to enable or disable tab creation; currently only generally.	`true`
tabName	Used to define the name of the channel tab.	`Course Content`
tabEndpoint	API endpoint to create a tab with a website URL in the general channel.	`/coursetab`
participantEndpoint	API endpoint to add and remove participants.	`/participant`

4.3 Known Issues

- MS Teams owners are created from Administrator and Tutors on the first saving, they are currently not updated later → Fix for ILS 14.5.1.
- Tab name only in one language; will be moved to a bundle → Fix in ILS 14.5.1.
- Teams are not removed when course is deleted / archived - Teams can be deleted by owners inside Teams.