

# IDP Login Configuration

The IDP (Identity Provider) login refers to the process of authenticating a user via an external service that manages identities.

The user can log in with an account from an external identity provider, e.g. Azure AD. This login approach needs to be configured in Keycloak backend depending on the requirements of the customers.

OMN does not manage the authentication and profiles of the IDP users since they are external identities, but the user could map them to work with OMN smoothly.

Currently, the following mappings are supported:

- [Mapping Activation for OMN Login](#) (Mapping the activation state)
- [Mapping IDP User Profile](#)
- [Mapping IDP User Groups](#)

According to the requirements of the customers, the IDP logins could be configured in the keycloak admin console. Keycloak supports both: OpenID Connect and SAML protocols. This guide will focus on the OpenID Connect protocol.

It needs the administrator access to the keycloak admin console for configuring the IDP logins. The credentials could be found in the `keycloak.env` file from the OMN server.

Furthermore, access to the MS Entra Admincenter (e.g. as Global Administrator) is required.

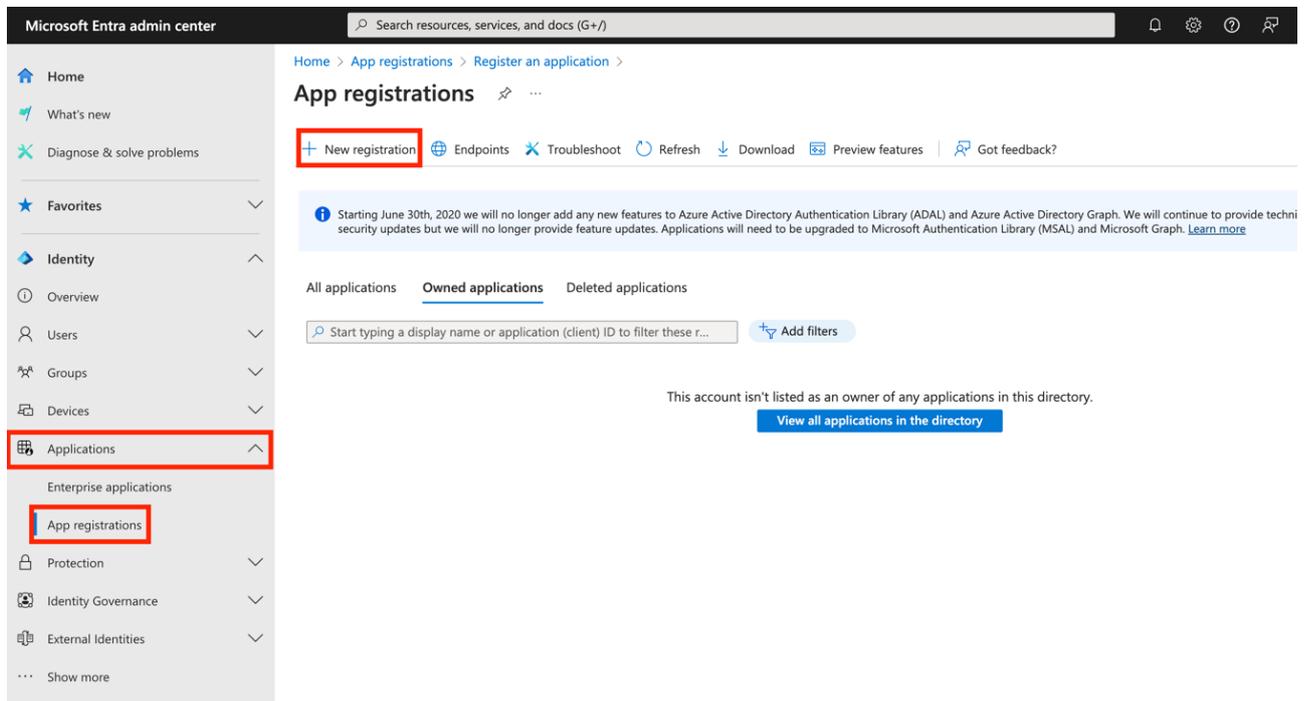
## IDP Configuration Example - Azure AD

To configure an IDP login with Azure AD, two setup steps are necessary: with the Microsoft Entra Admincenter (Azure AD) and the Keycloak Admin Console.

The steps in the following chapters should be followed.

### Microsoft Azure AD

1. Log in to your [admin console](#) and navigate to the Applications section:
2. Select [Applications](#) → [App Registrations](#) → [New registration](#)



3. Provide a name for the application you are registering and select the account type that you would like to be supported → **single-tenant**. Then click the register button.

Home > App registrations > Register an application > App registrations >

## Register an application

\* Name

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

Supported account types

Who can use this application or access this API?

- Accounts in this organizational directory only (Apollon GmbH&Co.KG only - Single tenant)
- Accounts in any organizational directory (Any Microsoft Entra ID tenant - Multitenant)
- Accounts in any organizational directory (Any Microsoft Entra ID tenant - 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.

Register an app you're working on here. Integrate gallery apps and other apps from outside your organization by adding from [Enterprise applications](#).

By proceeding, you agree to the [Microsoft Platform Policies](#)

4. After you have registered your application, you will be returned to the Application registration's overview page. You need to save your **Application (client) ID**, since you will need it later.

Home > KeyCloak > App registrations >

**KeyCloak**

Search << Delete Endpoints Preview features

Overview

- Quickstart
- Integration assistant
- Diagnose and solve problems

Manage

- Branding & properties
- Authentication
- Certificates & secrets**
- Token configuration
- API permissions
- Expose an API
- App roles
- Owners
- Roles and administrators
- Manifest

Got a second? We would love your feedback on Microsoft identity platform (previously Azure AD for developer). →

Essentials

Display name  
KeyCloak

Application (client) ID  
**251fd238-ee3e-467f-b80a-de515d8cb92e**

Object ID  
6eee5080-b415-4a7a-a190-b11cd1ce701d

Directory (tenant) ID  
2a349cbe-fdb0-4514-a478-35f81403deca

Supported account types  
[My organization only](#)

Client credentials  
[Add a certificate or secret](#)

Redirect URIs  
[Add a Redirect URI](#)

Application ID URI  
[Add an Application ID URI](#)

Managed application in local directory  
[KeyCloak](#)

Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure Active Directory Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. [Learn more](#)

[Get Started](#) [Documentation](#)

## 5. Create a Client Secret for your Application:

- On the left side, navigate to **Certificates and secrets**. Create a **New client secret**.

Home > KeyCloak > App registrations > KeyCloak

**KeyCloak | Certificates & secrets**

Search << Got feedback?

Overview

- Quickstart
- Integration assistant
- Diagnose and solve problems

Manage

- Branding & properties
- Authentication
- Certificates & secrets**
- Token configuration
- API permissions
- Expose an API
- App roles
- Owners
- Roles and administrators
- Manifest

Support + Troubleshooting

- New support request

Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Application registration certificates, secrets and federated credentials can be found in the tabs below.

Certificates (0) **Client secrets (0)** Federated credentials (0)

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

**+ New client secret**

Description	Expires	Value	Secret ID
No client secrets have been created for this application.			

- Add a description for your new secret and set an expiration for it.

Home > Keycloak > App registrations > Keycloak

**Keycloak | Certificates & secrets**

Search << Got feedback?

Overview  
Quickstart  
Integration assistant  
Diagnose and solve problems

Manage

Branding & properties  
Authentication  
Certificates & secrets  
Token configuration  
API permissions  
Expose an API  
App roles  
Owners  
Roles and administrators  
Manifest

Support + Troubleshooting  
New support request

**Add a client secret**

Description: Keycloak Secret

Expires: Recommended: 180 days (6 months)

+ New client secret

Description	Expires
No client secrets have been created for this application.	

Add Cancel

- Please save the Value and the Secret ID for later.

Home > Keycloak > App registrations > Keycloak

**Keycloak | Certificates & secrets**

Search << Got feedback?

Overview  
Quickstart  
Integration assistant  
Diagnose and solve problems

Manage

Branding & properties  
Authentication  
Certificates & secrets  
Token configuration  
API permissions  
Expose an API  
App roles  
Owners  
Roles and administrators  
Manifest

Support + Troubleshooting  
New support request

Credentials enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Certificates (0) **Client secrets (1)** Federated credentials (0)

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

+ New client secret

Description	Expires	Value	Secret ID
Keycloak Secret	3/18/2025	6t78Q~3nYcf.RcAPpMXyehC254EYr...	bda2491e-d252-4c2a-9e57-8878e0c...

## 6. Add the Redirect URI generated from Keycloak to the Azure Application Registration.

- Select Add a Redirect URI

Home > KeyCloak

Search

Delete Endpoints Preview features

Overview

- Quickstart
- Integration assistant
- Diagnose and solve problems

Manage

- Branding & properties
- Authentication
- Certificates & secrets
- Token configuration
- API permissions
- Expose an API
- App roles
- Owners
- Roles and administrators
- Manifest

Support + Troubleshooting

- New support request

Essentials

Display name  
KeyCloak

Application (client) ID  
251fd238-ee3e-467f-b80a-de515d8cb92e

Object ID  
6eee5080-b415-4a7a-a190-b11cd1ce701d

Directory (tenant) ID  
2a349cbe-fdb0-4514-a478-35f81403deca

Supported account types  
My organization only

Client credentials  
0 certificate, 1 secret

Redirect URIs  
Add a Redirect URI

Application ID URI  
Add an Application ID URI

Managed application in local directory  
KeyCloak

Get Started Documentation

Microsoft identity platform  
Help and Support  
Microsoft Graph  
Authentication libraries

Code samples  
Authentication scenarios  
Glossary

- Select Add a platform → Web

Home > KeyCloak | Authentication

Search

Got feedback?

Overview

- Quickstart
- Integration assistant
- Diagnose and solve problems

Manage

- Branding & properties
- Authentication
- Certificates & secrets
- Token configuration
- API permissions
- Expose an API
- App roles
- Owners
- Roles and administrators
- Manifest

Support + Troubleshooting

- New support request

Platform configurations

Depending on the platform or device this application is targeted, you can configure redirect URIs, specific authentication settings, or fields specific to the platform.

+ Add a platform

Supported account types

Who can use this application or access this API?

Accounts in this organizational directory only (Apollon Gm)

Accounts in any organizational directory (Any Microsoft En)

Help me decide...

Advanced settings

Allow public client flows

Enable the following mobile and desktop flows:

- App collects plaintext password (Resource Owner Password Grant)
- No keyboard (Device Code Flow) [Learn more](#)
- SSO for domain-joined Windows (Windows Integrated Authentication)

App instance property lock

Configure platforms

Web applications

Web  
Build, host, and deploy a web server application. .NET, Java, Python

Single-page application  
Configure browser client applications and progressive web applications. Javascript.

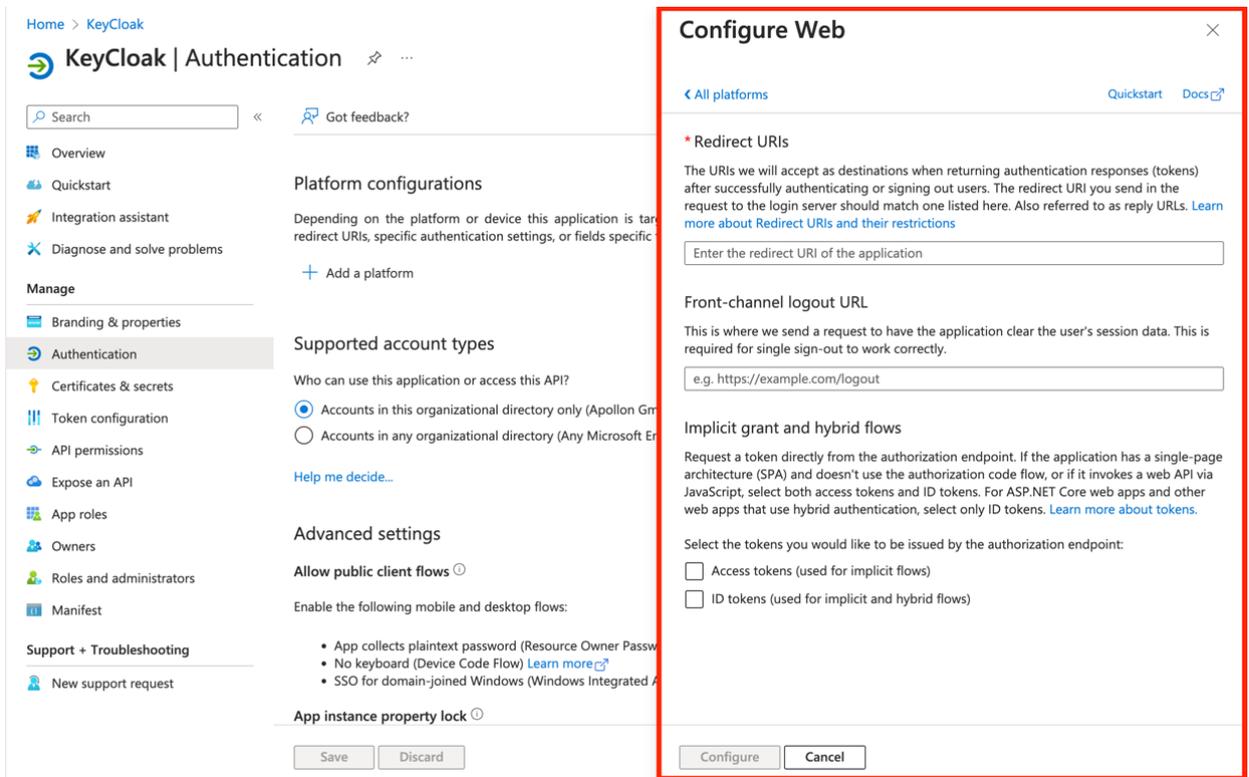
Mobile and desktop applications

iOS / macOS  
Objective-C, Swift, Xamarin

Android  
Java, Kotlin, Xamarin

Mobile and desktop applications  
Windows, UWP, Console, IoT & Limited-entry Devices, Classic iOS + Android

- Add your Redirect URI you saved when configuring your identity provider in Keycloak and paste it into the Redirect URI option for the platform you're configuring.



# Keycloak

## 1. Create OpenID Connect v1.0 as an Identity Provider in your Realm

- Log in to the Keycloak admin console and make sure the **OMN** realm is selected.
- Create Realm and Client in Keycloak Administration Console

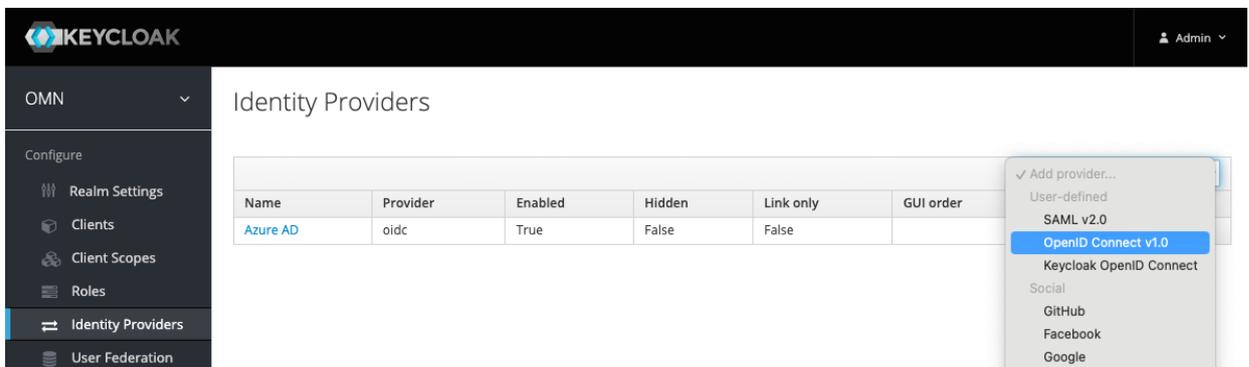
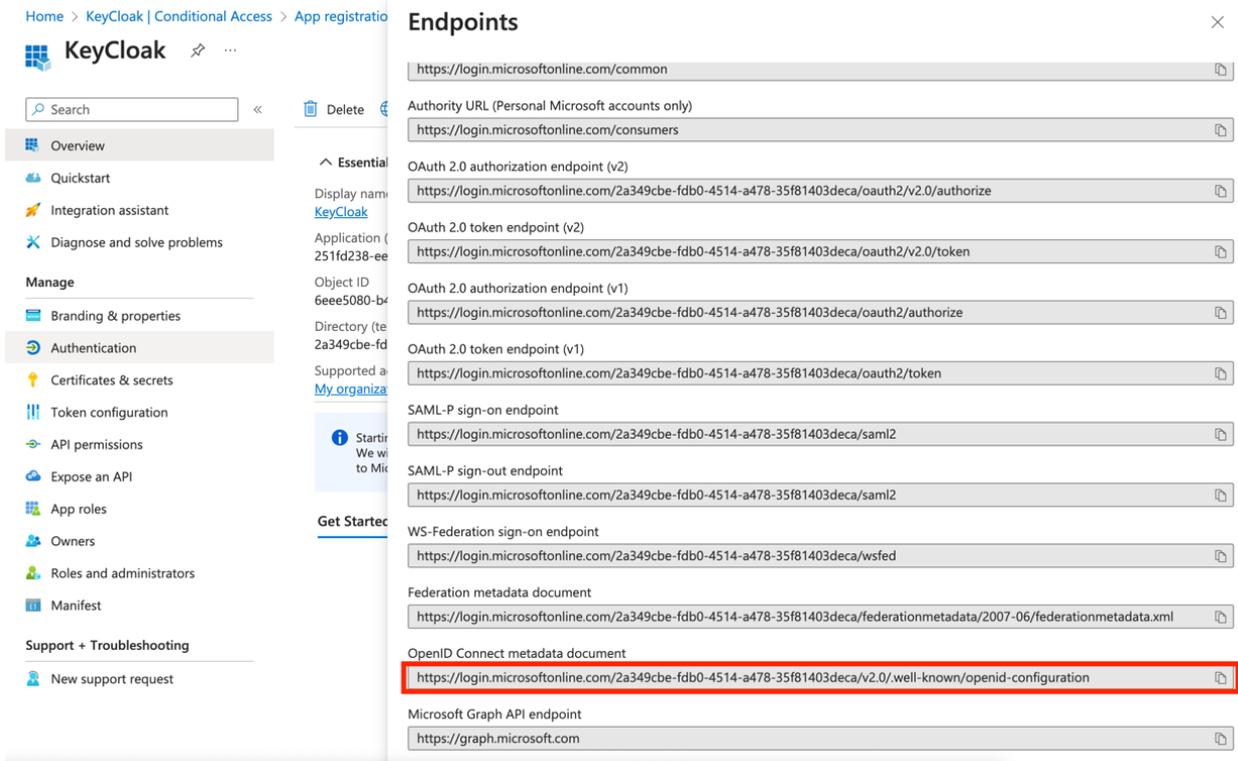


Figure 1. Create IDP with OpenID Connect Protocol

- Select **Identity Providers** → **Add provider** → **OpenID Connect v1.0**



- Under **Import External IDP Config** you need to add the External link from Entra AAD. For this go to **Endpoints** and copy the **OpenID Connect metadata document** link.

### ∨ Import External IDP Config ?

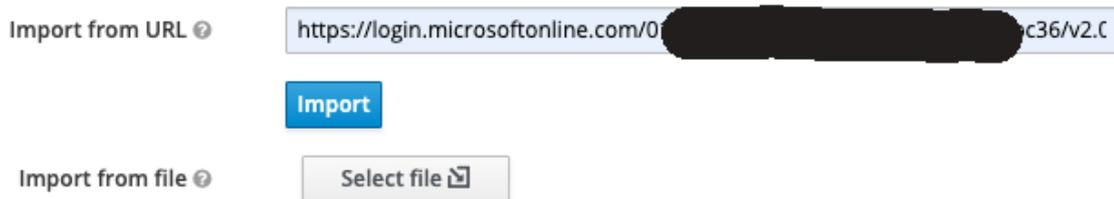


Figure 2. Import the metadata from IDP URL

2. Import the IDP configuration from the Azure AD metadata URL (Ask the customer for the metadata URL):
  - Fill the field **Import from URL** with the information you can find [here](#), for example.
  - Click on the **Import** button.
3. Fill the **Alias** field with the expected IDP name, e.g. **azuread**, the **Redirect URI** field will be filled automatically.
4. Use the generated **Redirect URI** to register an application in the Azure AD side for getting the **Client ID** and **Client Secret** from Azure AD (Ask the customer for help).
5. Fill the **Client ID** and **Client Secret** fields with the values from the Azure AD application configuration.
6. Set **First Login Flow** to **OMN First Broker Login**
7. Set **Post Login Flow** to **IDP-PostLogin**
8. Set **Client Authentication** to **Client Secret sent as post**



Then check the following points:

- The access token and ID token should be returned successfully.
- The user profile data should be included in the ID token.
- The user groups should be included in the ID token.
- The `login_hint` claim should be included in the ID token (optional).

You can use <https://jwt.io> to decode the ID token and check above mentioned points.

## Mapping Activation for OMN Login

Normally in OMN a new created user is not to be activated by default.

In the case of IDP login, the users could be activated automatically by mapping to the built-in role `omn-login-role`.

Go to the **Mappers** tab and create a new mapper with the following settings:

- **Name:** `omn-login-role-mapper`
- **Sync Mode:** `Force`
- **Mapper Type:** `Claim to Role`
- **Claim:** `groups`  
⇒ The claim name which provides the user groups from Azure AD
- **Claim Value:** `your-target-group-from-azure-ad`  
⇒ The group name whose members should be activated in OMN automatically.
- **Role:** `omn-login-role`

The screenshot shows the configuration page for the 'omn-login-role-mapper' in the OMN Identity Provider Mappers tab. The breadcrumb path is 'Identity Providers > azuread > Identity Provider Mappers > Omn-login-role-mapper'. The form fields are as follows:

ID	109bce5f-0bd4-4488-871c-2d0aa0c1ef76
Name *	omn-login-role-mapper
Sync Mode Override *	force
Mapper Type	Claim to Role
Claim	groups
Claim Value	ef7a3f68-8f69-4eff-9212-7ea54535401f
Role	omn-login-role

Buttons: Save, Cancel, Select Role

Figure 4. Mapping Activation State

If there are multiple groups in Azure AD which should be activated in OMN, you can create multiple mappers with different **Claim Value** settings for each group.

After that, all users from the configured groups will be activated automatically and others have to be activated manually by the administrator.



This automatic activation will be only executed at the first IDP login. After that, the user could be activated or deactivated only by administrator manually.

## Mapping IDP User Profile

An OMN user has own profile properties which could be also mapped from the IDP user profile data.

OMN has the following standard profile properties:

- userName
- firstName
- lastName
- email
- phone
- fax
- zip
- address
- city
- country
- department
- company

Keycloak normally does the mapping automatically for the standard claims like `username`, `email`, `given_name`, `family_name`, etc.

Others, e.g. `company`, should be mapped manually following the next steps:

Go to the **Mappers** tab and create a new mapper with the following settings:

- **Name:** `omn-user-info-company`
- **Sync Mode:** `Force`
- **Mapper Type:** `Attribute Importer`
- **Claim:** `companyName`  
⇒ The claim name matching the `company name` from Azure AD
- **User Attribute Name:** `company`  
⇒ The OMN user profile property name from the above properties list

## Omn-userinfo-company

ID	<input type="text" value="fac361cc-f343-4f4f-8735-4754240699c2"/>
Name * 	<input type="text" value="omn-userinfo-company"/>
Sync Mode Override * 	<input type="text" value="force"/>
Mapper Type 	<input type="text" value="Attribute Importer"/>
Claim 	<input type="text" value="companyName"/>
User Attribute Name 	<input type="text" value="company"/>
	<input type="button" value="Save"/> <input type="button" value="Cancel"/>

Figure 5. Mapping User Profile

After that, the `company` property will be filled with the value from the Azure AD user profile. The same could be done for other user profile properties.



The user profile mapping will be executed at each IDP login so that the user profile data will be always up-to-date.

## Mapping IDP User Groups

OMN has user groups with different access rights, that could be also mapped from the IDP user groups. For example, the user from the group `admin` in Azure AD could be mapped to the `admin` group in OMN.

Before the mapping, the groups should be synchronized from the OMN LDAP into Keycloak.

Go to User Federation → `omn-ldap` → Mappers → `omn-ldap-group`, and click on `Sync LDAP Groups to Keycloak` button.

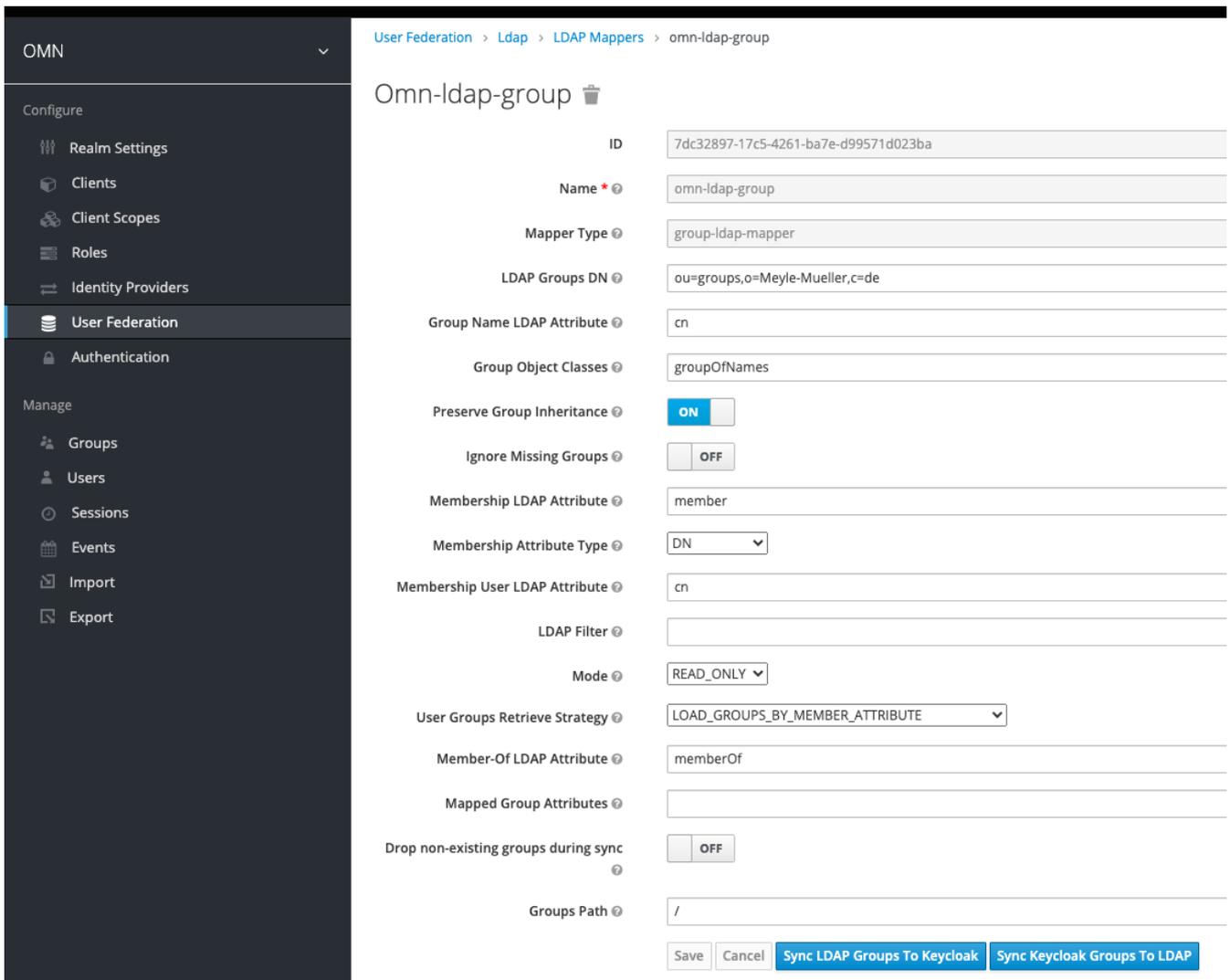


Figure 6. Sync OMN Groups

Go to **Identity Providers** → **Azure AD**, then open the **Mappers** tab and create a new mapper with the following settings:

- **Name:** `omn-group-admin-mapper`
- **Sync Mode:** **Force**
- **Mapper Type:** **Advanced Claim to Group**
- **Claims:** ⇒ add a claim key-value pair key: `groups` ⇒ The claim name which provides the user groups from Azure AD value: `admin-IDP` ⇒ The IDP group name which needs to be mapped to the OMN group
- **Group:** `manager-gui`  
⇒ Choose the target OMN group by clicking on the **Select Group** button

The same could be done for other groups.

## Add Identity Provider Mapper

Name \*

Sync Mode Override \*

Mapper Type

Claims

Key	Value	Actions
groups	ef7a3f68-8f69-4eff-9212-7ea54535401f	Add

Regex Claim Values  OFF

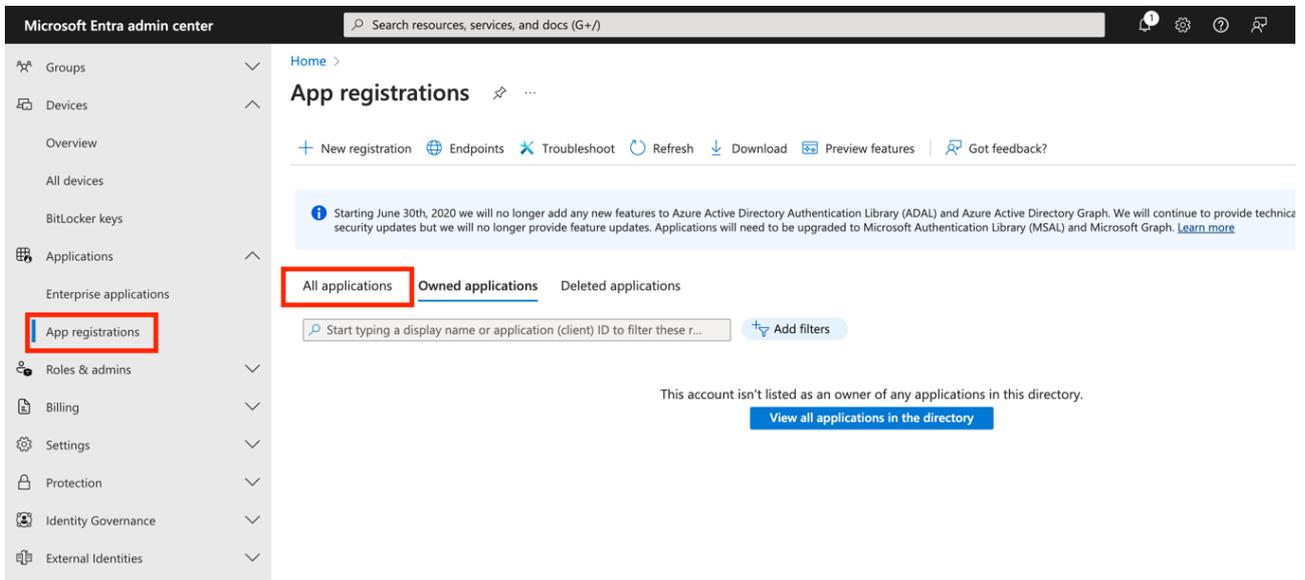
Group

Figure 7. Mapping Group

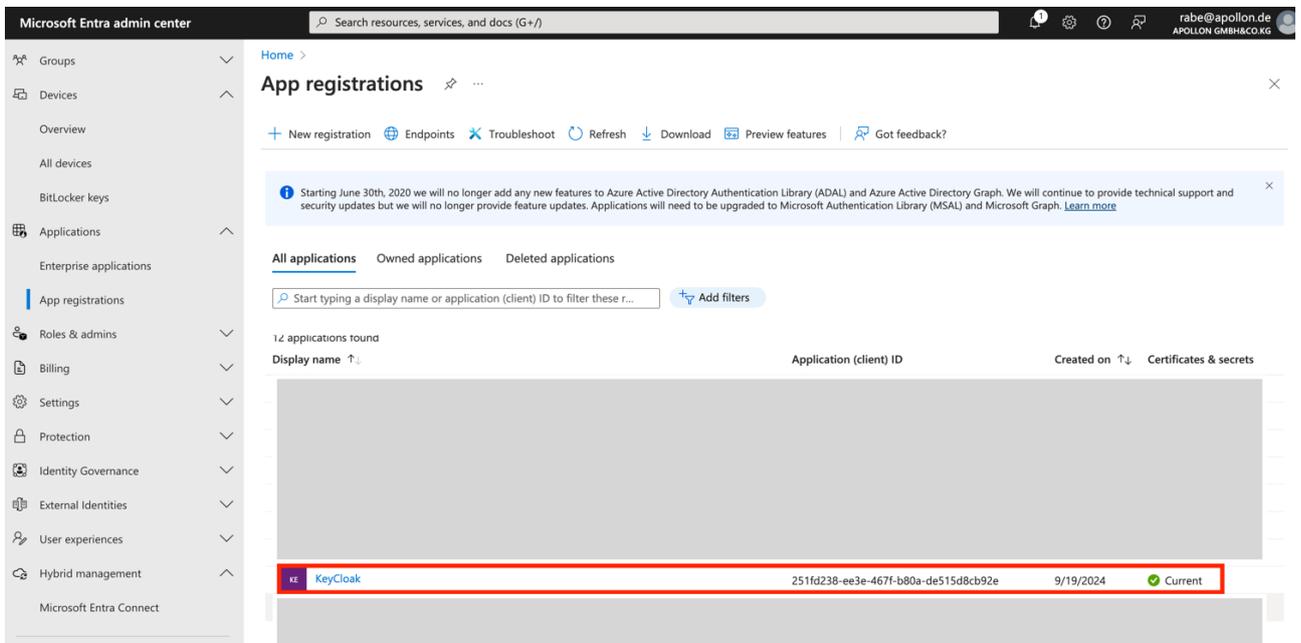
# Configure ID Token with additional user information (Azure AD)

The following example shows the configuration of additional claims. In this case, we configure the IDP Token endpoint to print the user groups, additionally.

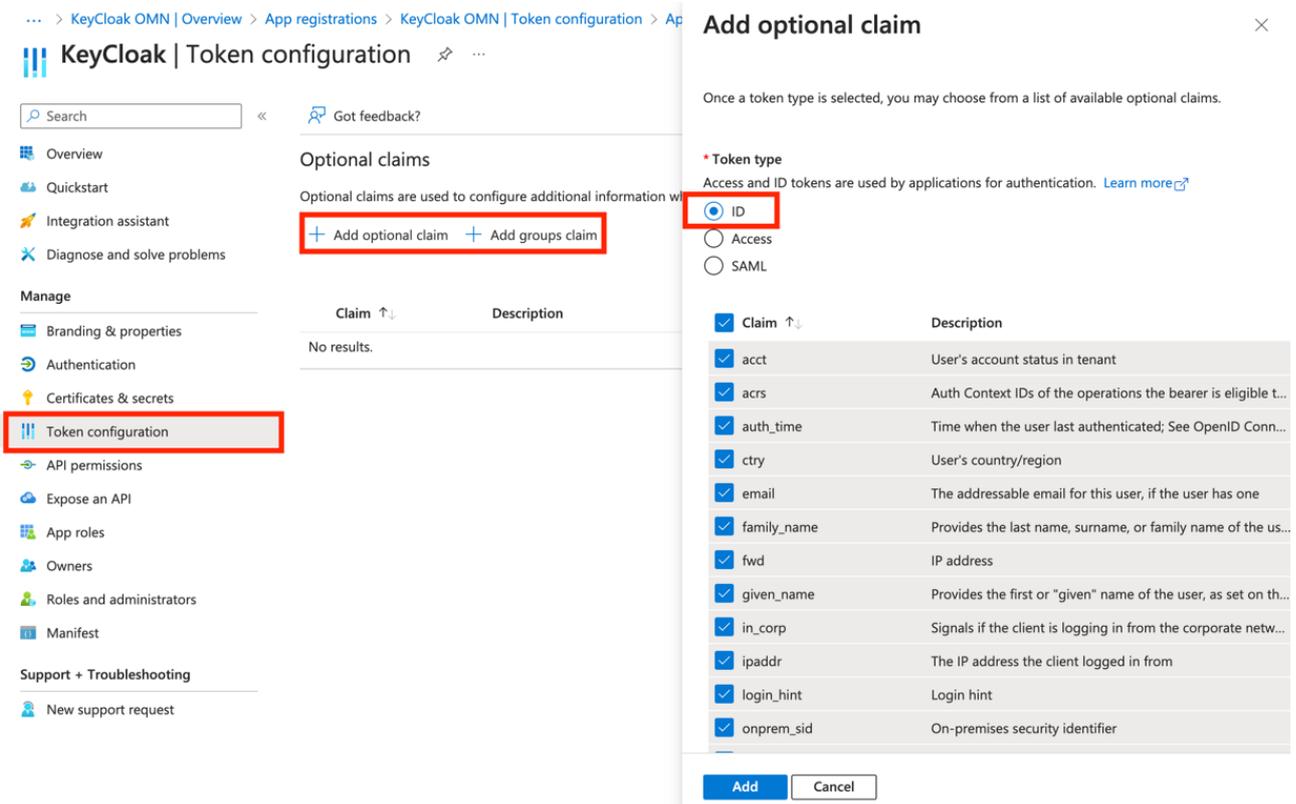
1. Log in to your [admin console](#) and navigate to the Applications section:
2. Select [Applications](#) → [App Registrations](#) → [All applications](#)



3. Select the [Keycloak](#) app instance created previously.



- Go to **Token configuration** → **Add optional Claim** → **ID** → Select needed claims, e.g. Claim all and Click **Add** Button.



- Go to **Add groups claim** → select e.g. **Security groups** and the token ID with e.g. **Group ID**. Then Click the **Add** button.

Home > KeyCloak

## KeyCloak | Token configuration

Search  Got feedback?

- Overview
- Quickstart
- Integration assistant
- Diagnose and solve problems

**Manage**

- Branding & properties
- Authentication
- Certificates & secrets
- Token configuration**
- API permissions
- Expose an API
- App roles
- Owners
- Roles and administrators
- Manifest

**Support + Troubleshooting**

- New support request

Optional claims

Optional claims are used to configure additional information with tokens.

+ Add optional claim **+ Add groups claim**

Claim	Description
auth_time	Time when the user last authenticated
ctry	User's country/region
email	The addressable email for this user
family_name	Provides the last name, surname
fwd	IP address
given_name	Provides the first or "given" name
in_corp	Signals if the client is logging in
ipaddr	The IP address the client logged in
login_hint	Login hint
onprem_sid	On-premises security identifier
preferred_username	Provides the preferred username

### Edit groups claim

Adding the groups claim applies to Access, ID, and SAML token types. [Learn more](#)

**Select group types to include in Access, ID, and SAML tokens.**

- Security groups
- Directory roles
- All groups (includes 3 group types: security groups, directory roles, and distribution lists)
- Groups assigned to the application (recommended for large enterprise companies to avoid exceeding the limit on the number of groups a token can emit)

**Customize token properties by type**

^ ID

- Group ID
- sAMAccountName
- NetBIOSDomain\sAMAccountName
- DNSDomain\sAMAccountName
- On Premises Group Security Identifier
- Emit groups as role claims

^ Access

^ SAML

**Add** **Cancel**

## 6. Go to API permissions → Add a permission

Enterprise applications | All applications > KeyCloak OMN | Overview > App registrations > KeyCloak | API permissions > App registrations > KeyCloak

## KeyCloak | API permissions

Search  Refresh Got feedback?

- Overview
- Quickstart
- Integration assistant
- Diagnose and solve problems

**Manage**

- Branding & properties
- Authentication
- Certificates & secrets
- Token configuration
- API permissions**
- Expose an API
- App roles
- Owners
- Roles and administrators
- Manifest

**Support + Troubleshooting**

- New support request

Granting tenant-wide consent may revoke permissions that have already been granted tenant-wide for that application. Permissions that users have already granted on their own behalf aren't affected. [Learn more](#)

The "Admin consent required" column shows the default value for an organization. However, user consent can be customized per permission, user, or app. This column may not reflect the value in your organization, or in organizations where this app will be used. [Learn more](#)

**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](#)

+ Add a permission **✓ Grant admin consent for Apollon GmbH&Co.KG**

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

To view and manage consented permissions for individual apps, as well as your tenant's consent settings, try [Enterprise applications](#).

Home > Keycloak OMN | API permissions > App registrations > Keycloak

Keycloak | API permissions

Search Refresh Got feedback?

Overview  
Quickstart  
Integration assistant  
Diagnose and solve problems

Manage

Branding & properties  
Authentication  
Certificates & secrets  
Token configuration  
API permissions  
Expose an API  
App roles  
Owners  
Roles and administrators  
Manifest

Support + Troubleshooting  
New support request

Granting tenant-wide consent may revoke permissions that have already been granted tenant-wide for that application. Permissions that users have already granted on their own behalf aren't affected. [Learn more](#)

The "Admin consent required" column shows the default value for an organization. However, user consent can be customized per permission, user, or app. This column may not reflect the value in your organization, or in organizations where this app will be used. [Learn more](#)

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](#)

+ Add a permission Grant admin consent for Apollon GmbH&Co.KG

API / Permission	Type	Description	Admin consent req...	Status
Microsoft Graph (1)				
User.Read	Delegated	Sign in and read user profile	No	Granted for Apollon Gm...

To view and manage consented permissions for individual apps, as well as your tenant's consent settings, try [Enterprise applications](#).

- Go to Microsoft Graph → add a permission
- Go to Microsoft Graph → Delegated permission is selected.

## Request API permissions



Microsoft Graph

<https://graph.microsoft.com/> [Docs](#)

What type of permissions does your application require?

Delegated permissions

Your application needs to access the API as the signed-in user.

Application permissions

Your application runs as a background service or daemon without a signed-in user.

Select permissions

[expand all](#)



The "Admin consent required" column shows the default value for an organization. However, user consent can be customized per permission, user, or app. This column may not reflect the value in your organization, or in organizations where this app will be used. [Learn more](#)



Permission	Admin consent required
OpenId permissions (2)	
<input checked="" type="checkbox"/> email <small> ⓘ</small> View users' email address	No
<input type="checkbox"/> offline_access <small> ⓘ</small> Maintain access to data you have given it access to	No
<input type="checkbox"/> openid <small> ⓘ</small> Sign users in	No
<input type="checkbox"/> profile <small> ⓘ</small>	

Update permissions

Discard

9. There you need the following permission set:

OpenId permissions: check that **email** and **profile** is selected.

## Request API permissions



### OpenId permissions (2)

<input checked="" type="checkbox"/>	email ⓘ View users' email address	No
<input type="checkbox"/>	offline_access ⓘ Maintain access to data you have given it access to	No
<input type="checkbox"/>	openid ⓘ Sign users in	No
<input checked="" type="checkbox"/>	profile ⓘ View users' basic profile	No

> AccessReview

> Acronym

> AdministrativeUnit

> AgreementAcceptance

> Agreement

> Analytics

> APIConnectors

Update permissions

Discard

10. In **User** check that **User.Read** is selected.

# Request API permissions



▼ User (1)

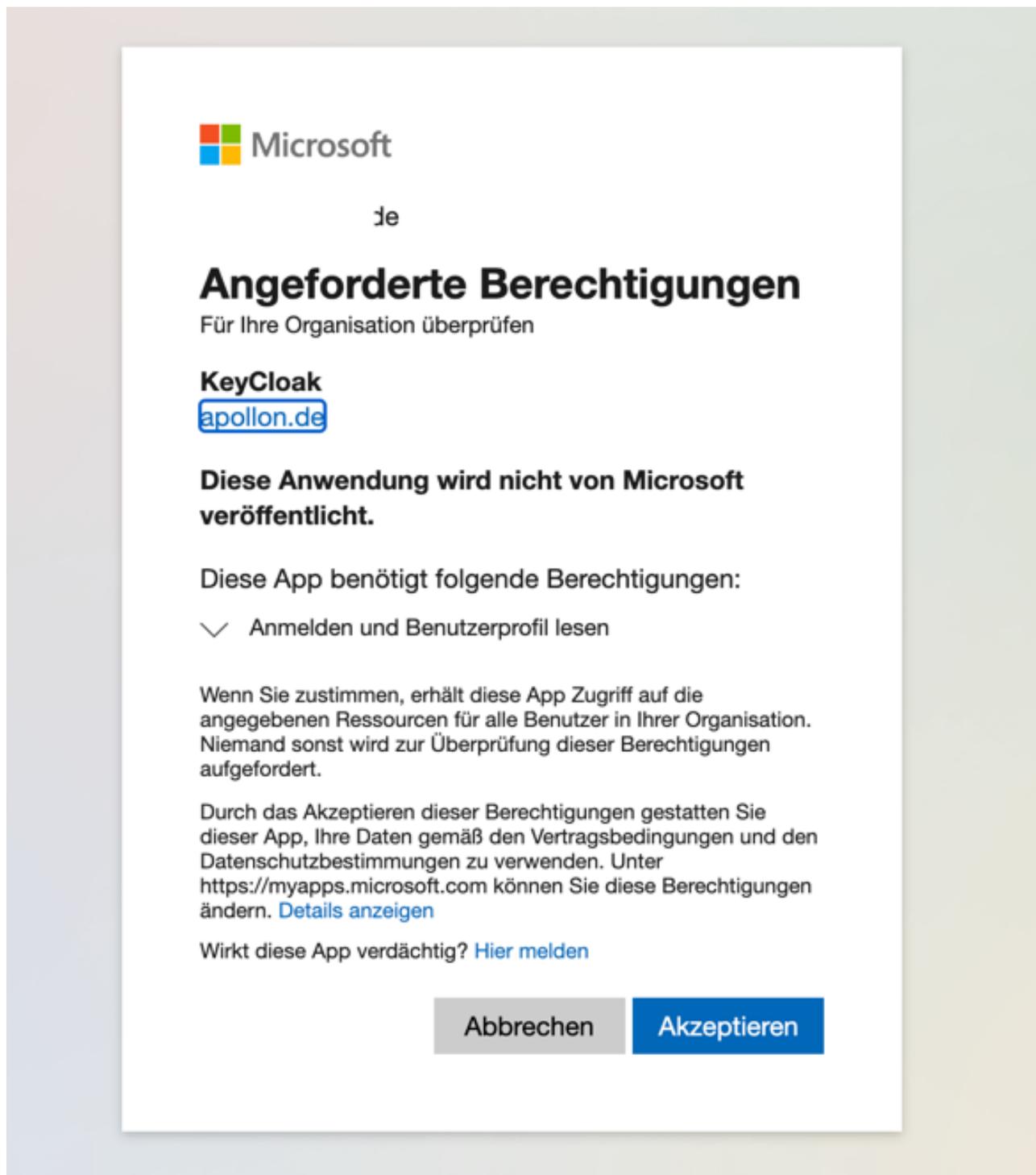
<input type="checkbox"/>	User.DeleteRestore.All ⓘ Delete and restore users	Yes
<input type="checkbox"/>	User.EnableDisableAccount.All ⓘ Enable and disable user accounts	Yes
<input type="checkbox"/>	User.Export.All ⓘ Export user's data	Yes
<input type="checkbox"/>	User.Invite.All ⓘ Invite guest users to the organization	Yes
<input type="checkbox"/>	User.ManageIdentities.All ⓘ Manage user identities	Yes
<input checked="" type="checkbox"/>	User.Read ⓘ Sign in and read user profile	No
<input type="checkbox"/>	User.Read.All ⓘ Read all users' full profiles	Yes
<input type="checkbox"/>	User.ReadBasic.All ⓘ Read all users' basic profiles	No
<input type="checkbox"/>	User.ReadWrite ⓘ Read and write access to user profile	No
<input type="checkbox"/>	User.ReadWrite.All ⓘ Read and write all users' full profiles	Yes
<input type="checkbox"/>	User.RevokeSessions.All ⓘ Revoke all sign in sessions for a user	Yes

11. Navigate to **Applications** → **Enterprise applications** → **Permissions** and **Grant the Admin Consent**...

The screenshot shows the Azure portal interface. On the left, the navigation pane has 'Applications' and 'Enterprise applications' highlighted with a red box. The main content area shows the 'Keycloak | Permissions' page for an 'Enterprise Application'. A red box highlights the 'Grant admin consent for Apollon GmbH&Co.KG' button. Below this, there is a table of permissions granted to the application.

API Name	Claim value	Permission	Type	Granted through	Granted by
Microsoft Graph	email	View users' email address	Delegated	Admin consent	An administrator
Microsoft Graph	profile	View users' basic profile	Delegated	Admin consent	An administrator
Microsoft Graph	User.Read	Sign in and read user pr...	Delegated	Admin consent	An administrator

12. Accept the registration.



The screenshot shows a Microsoft consent dialog box. At the top left is the Microsoft logo. Below it, the text 'je' is centered. The main heading is 'Angeforderte Berechtigungen' (Requested Permissions) in bold, followed by the subtitle 'Für Ihre Organisation überprüfen' (Check for your organization). The application name 'KeyCloak' is displayed, with the URL 'apollon.de' in a blue box below it. A warning message states: 'Diese Anwendung wird nicht von Microsoft veröffentlicht.' (This application is not published by Microsoft). Below this, it says 'Diese App benötigt folgende Berechtigungen:' (This app needs the following permissions:). A single permission is listed: 'Anmelden und Benutzerprofil lesen' (Sign in and read user profile), preceded by a checkmark icon. A paragraph explains that by consenting, the app will have access to resources for all users in the organization, and that no one else is required to check these permissions. Another paragraph states that by accepting, the user allows the app to use their data according to terms and conditions, with a link to 'https://myapps.microsoft.com' for changing permissions. At the bottom, there is a link 'Wirkt diese App verdächtig? Hier melden' (Is this app suspicious? Report here). At the very bottom are two buttons: 'Abbrechen' (Cancel) in a grey box and 'Akzeptieren' (Accept) in a blue box.



This is just an example of a test configuration. More information for [Configure optional claims](#) from Microsoft.