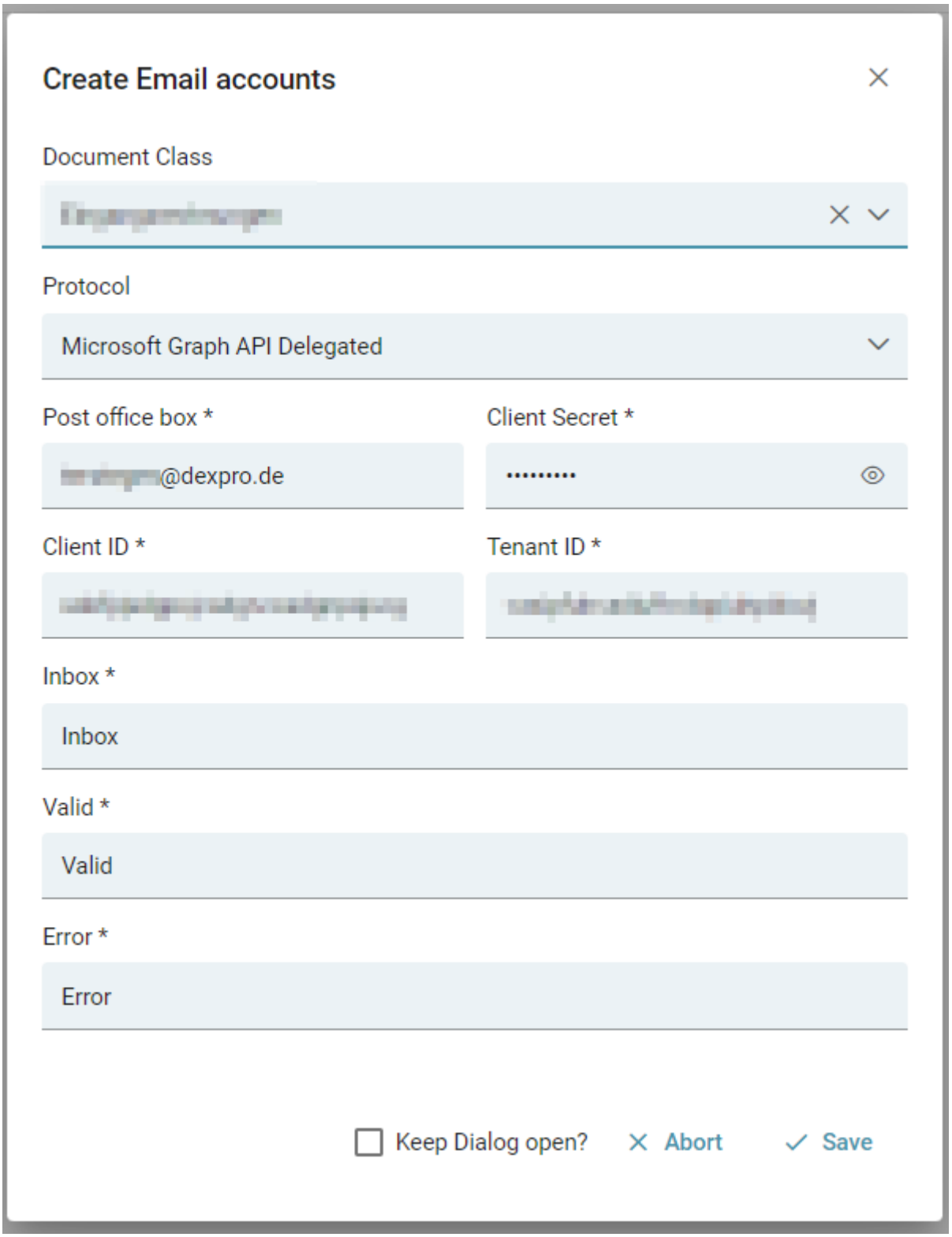
Configuration Authentication Code Flow (delegated) MS Graph API [ENG]

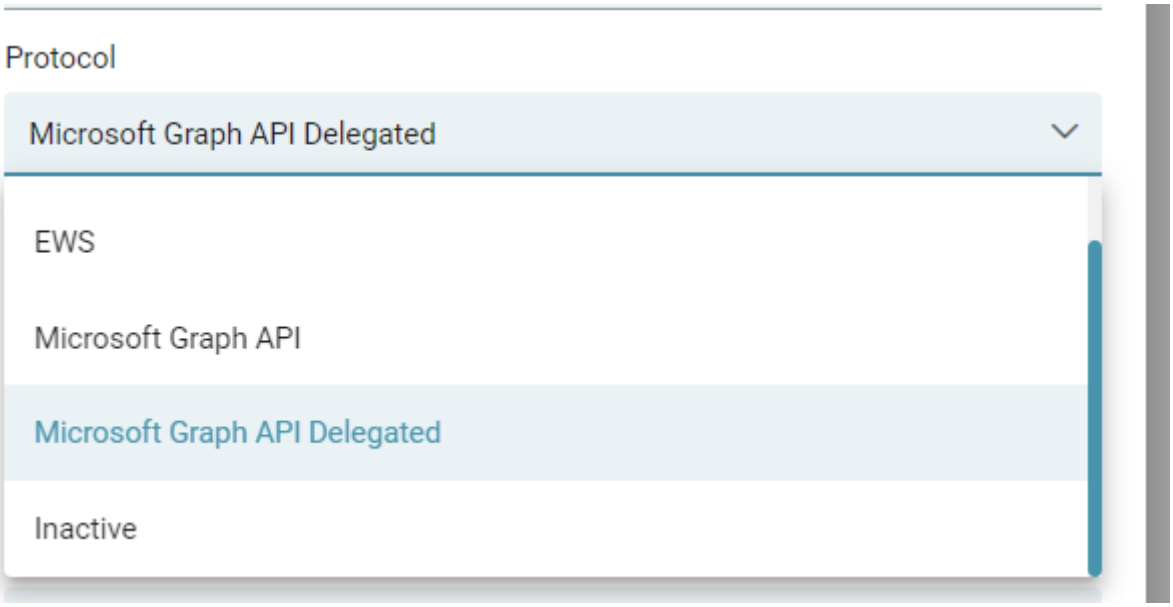
Configuration in Squeeze

This configuration interface is provided as of version 2.3.0.

If you want to use this authentication method, make sure that the Internal job refresh-tokens is enabled. This is documented further down in the article.



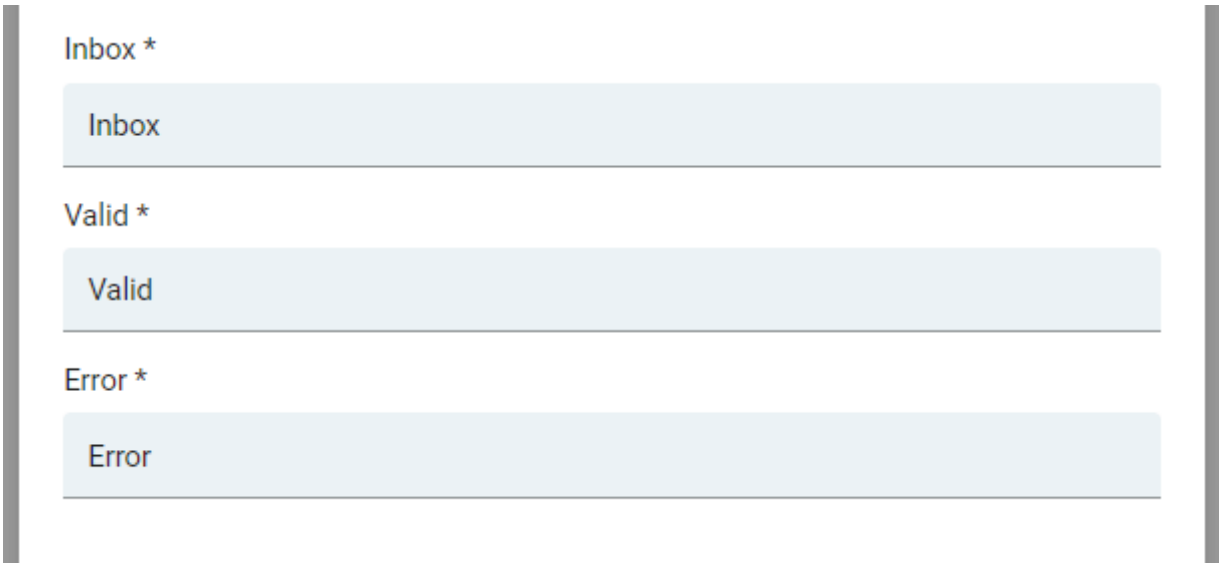
As you can see now, in the configuration interface under Protocol you can choose two types of connection to Microsoft(MS) Graph API:



For the delegated authorization process we select **Microsoft Graph API Delegated**. In the next step we fill in the Client ID, Client Secret, Tenant ID and the rest of the fields.

Folder - Configuration

In the folder configuration you need to specify three folders



1. **Inbox**
   This folder is regularly checked to import the emails it contains.
2. **Exported**
   The successfully imported emails are moved to this folder.
3. **Error**
   The emails that could not be imported (e.g. missing attachments) are moved to this folder.

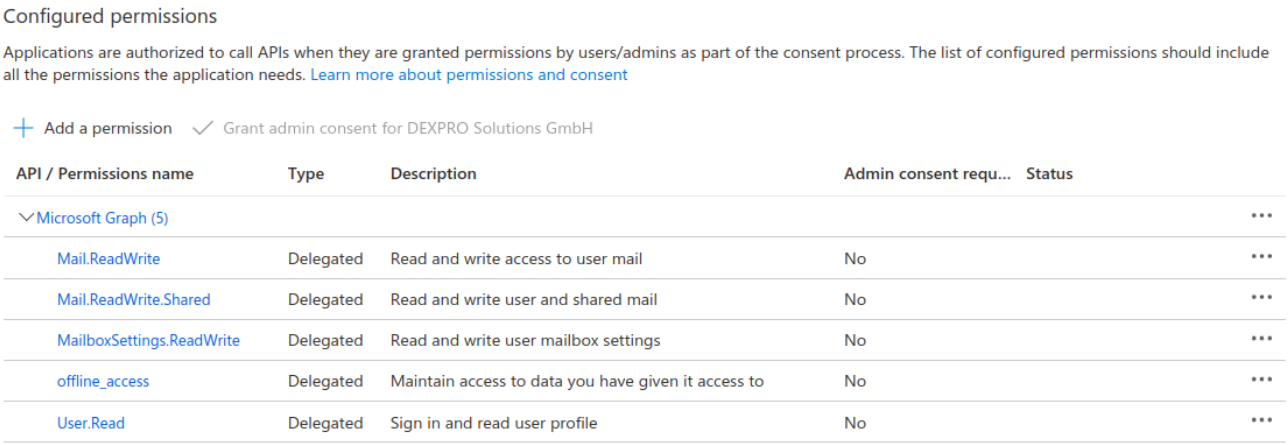
When defining the folders, please make sure that the names of the folders must be unique,

because the folders are searched in the directory structure of the mailbox.
If a configured folder is not unique, this can lead to a different folder being used than the desired one.

Configuration in AAD (Azure Active Directory)

Other necessary steps for a smooth process are the setup of an Azure Active Directory Application with a Client Secret. The global registration of a mail service of "Dexpro" in the Azure Active Directory of the customer is not offered for the time being.

In addition, it must be ensured that the application has the following scopes.

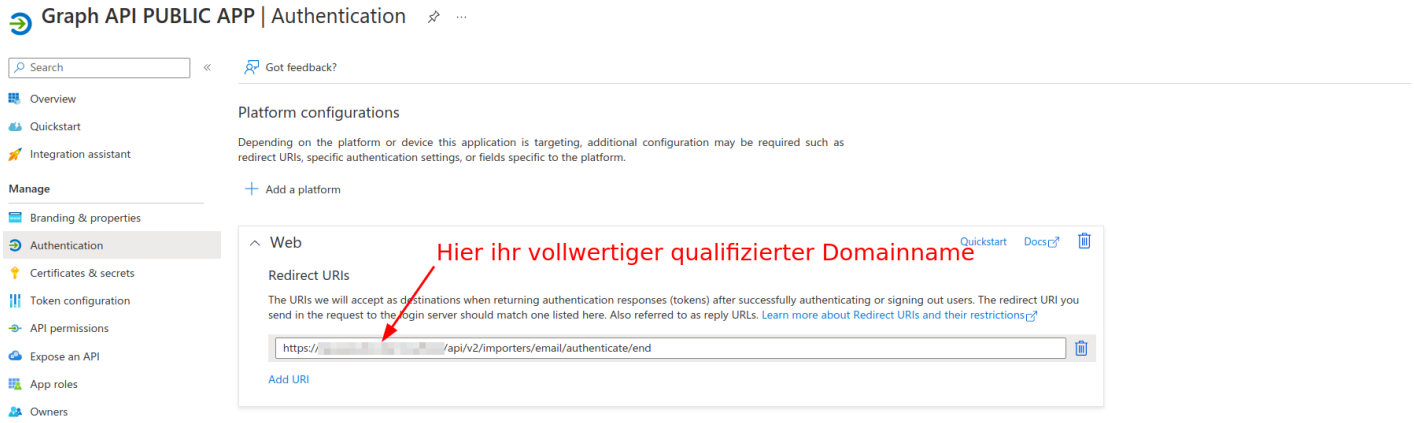


Unlike in the Permission for Client Credential Flow, any permission here is limited to the authorizing user.

Note: If a shared mailbox is to be accessed, the shared mailbox must be entered under Mailbox and the Application Permission Mail.Read.Write.Shared must be configured.

Remember that the shared mailbox will also be added to your Email Enabled security group if you have one set up.

Another significant difference to the Client Credential Flow is the use of a **Redirect URI**.
This Redirect URI is created in the AzureActive directory under the Application in the Authentication tab:

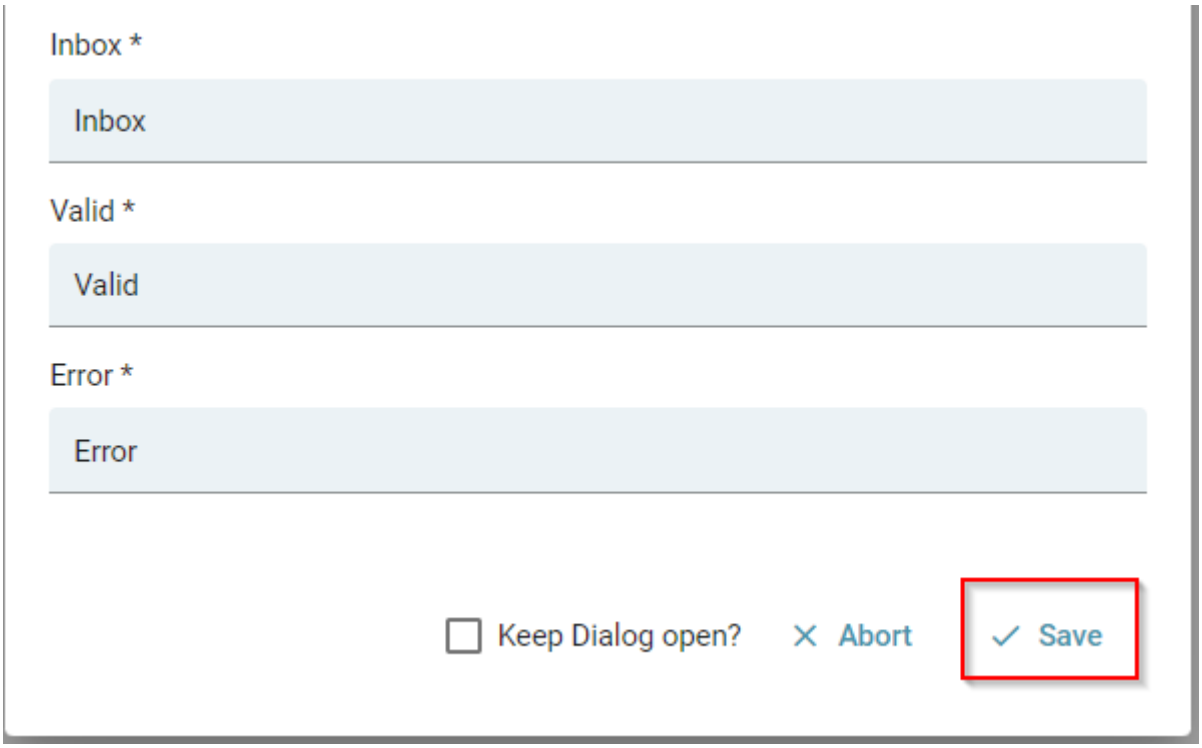


> Scheme Redirect URI:
> https://Ihr.vollwertiger.DomainName/api/v2/importers/email/authenticate/end

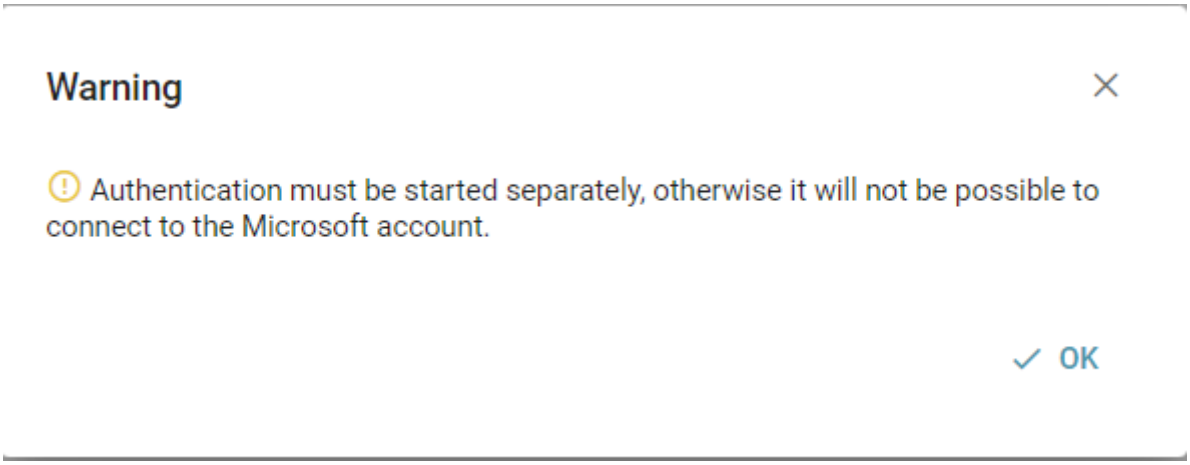
Note: A Squeeze client always requires an Azure Active Directory application with a redirect URI. This means that you must create an application for each client.

Authorization Process Authentication Code Flow

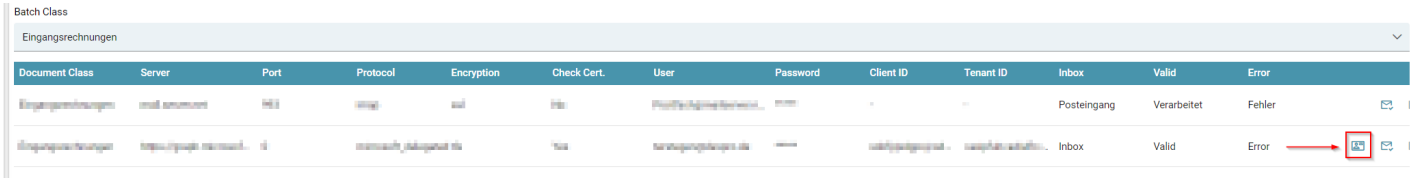
Once you have configured your application in the AAD and entered all the necessary data in the configuration interface, you can save your configuration.



Now another dialog box opens:

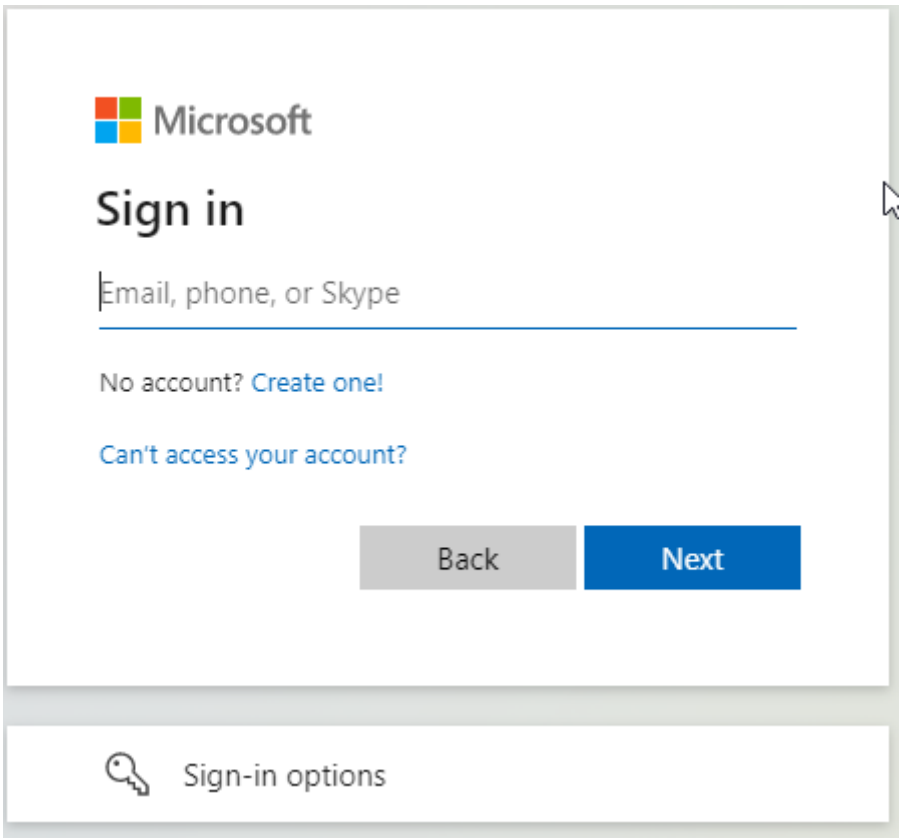


The system has saved your configuration, but you need to start the process of authentication separately using a button in the email configuration. For this purpose please select the following button:

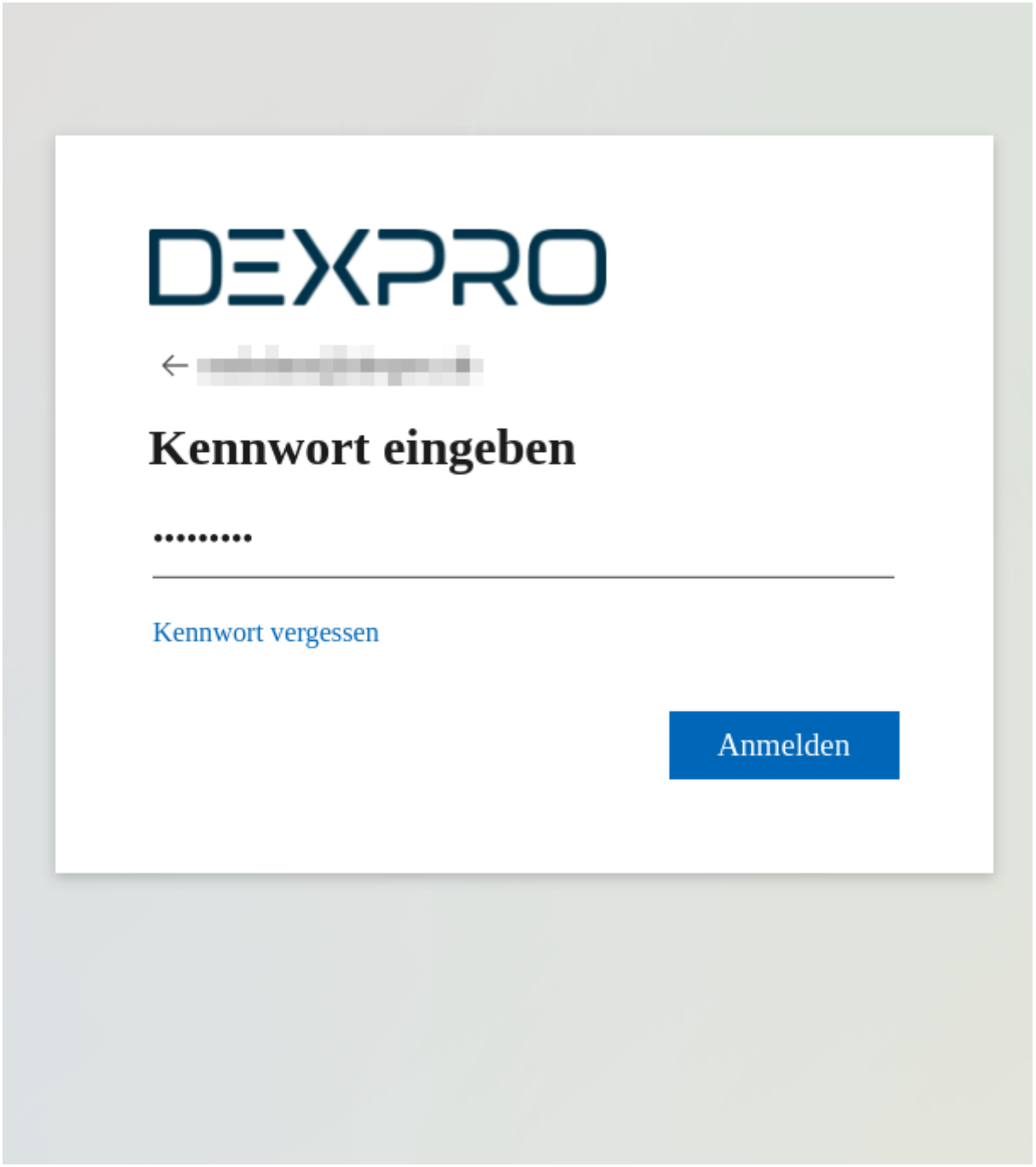


After clicking this button, you will now be redirected to the Microsoft interface. There, the authentication process begins. Follow Microsoft's instructions:

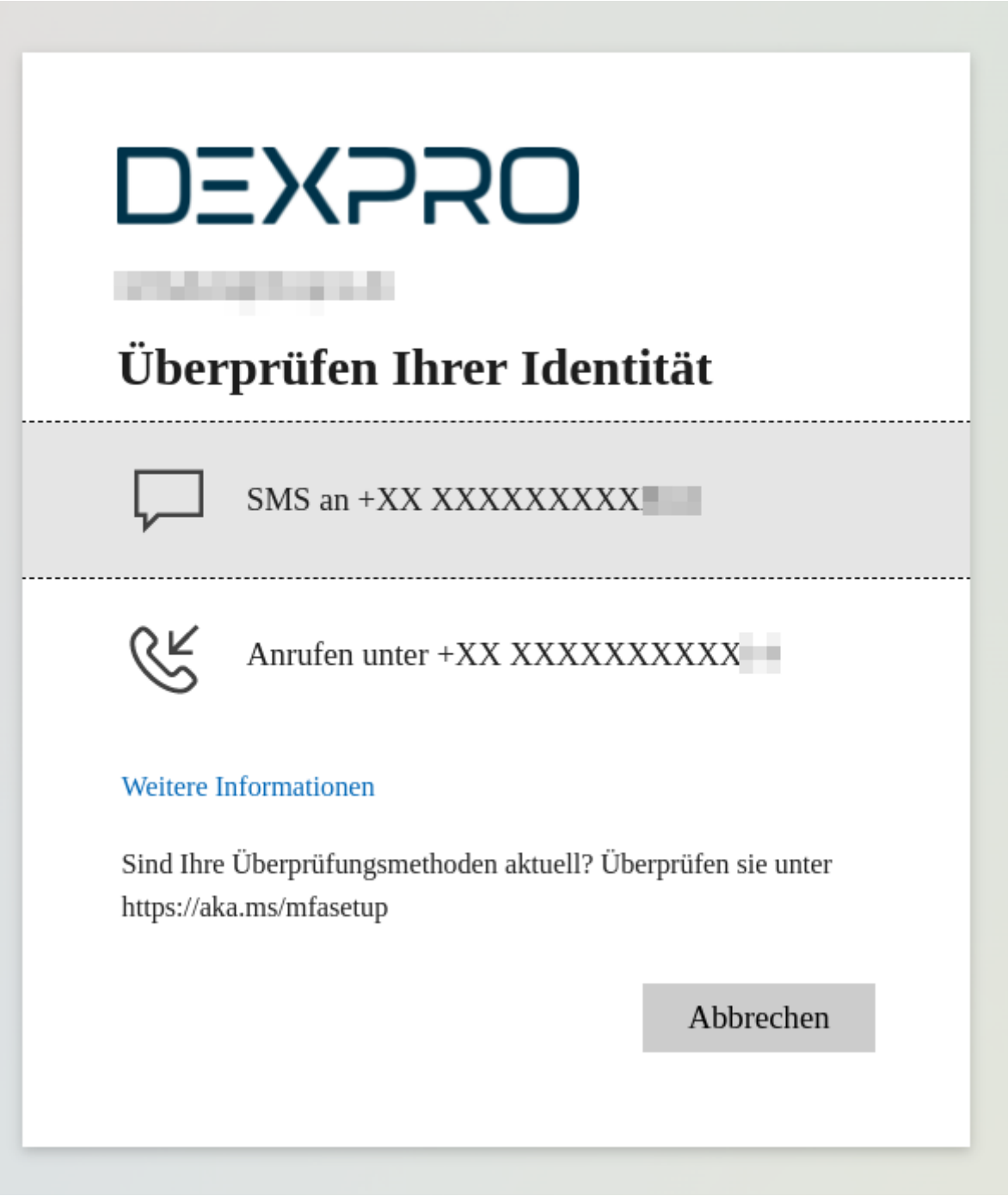
Enter your email address.



Enter your password:

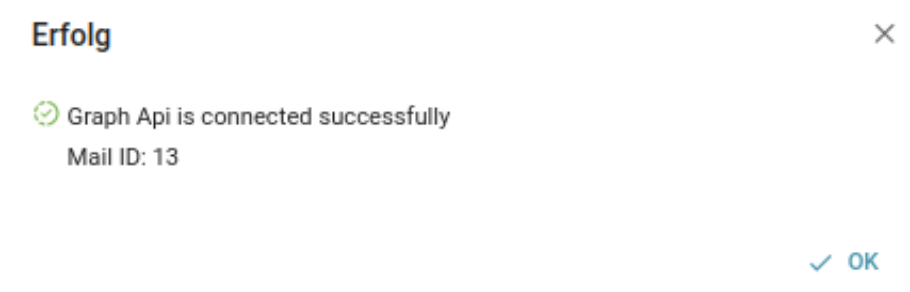


Depending on the company, you may also be asked to perform 2-factor authentication:



Once you are authenticated, you will be prompted for permissions and asked to confirm those permissions. After your confirmation, Microsoft will immediately redirect you back to the Squeeze interface.

The interface will now show you if Squeeze is now authorized or if an error has occurred:

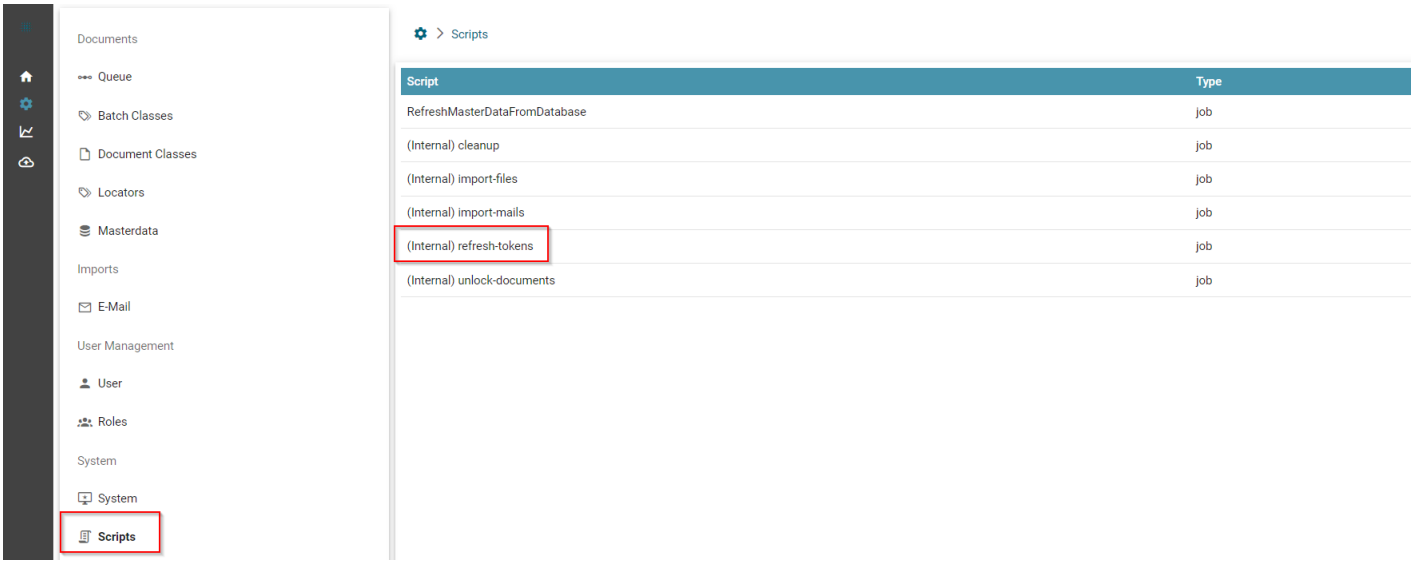


Now you can test the mail connection by clicking the Mail Connection Test button in the interface.

Job-Configuration

The use of this authentication method requires that a separate job is activated, which keeps the access data for the Graph API up-to-date in the background.

For this purpose, we find the new script named refresh-tokens in the administration under **"Scripts"**:



You can run this script manually at any time.

> You need to set up a job for this script that runs frequently enough to prevent the access data from expiring. By default, refresh tokens from Microsoft are valid for 1 day, so you should try to refresh the access data several times a day.
>
> We recommend running the job every hour:
> Cron expression for the job: */50 * * * *.

How to configure jobs is documented here:

- Artikel über Job Ausführungen in Squeeze
- Artikel über Job Steuerungen

FAQ

What happens if the job fails?	If the job did not run due to unforeseen reasons you have to re-authenticate using the authentication button in the email configuration.
Can I just change my mailbox in the configuration and do I have to do anything after that?	As soon as you change or save the configuration, the application prompts you to initiate a new authentication process. Reject this request. You still have the option to press the authentication button.
Can I use my Azure Active Directory Application for multiple Squeeze clients?	No, using one Application (AAD) for multiple clients is not possible as long as the clients are accessible under other domains. **Example:** 1 client: test.client.squeeze.one 2 client: test2.client.squeeze.one In this example we cannot use an application, because we can only address one client per application. Authentication with Microsoft would only run on one client at a time. You have the possibility to use multiple mailboxes with the same application (AAD), there are no limits.

My access through AzureTokens constantly expires, even though I have configured the job correctly.	In this case the server time configuration and the time configuration of the Squeeze application might differ. **Example:** Server Time: CEST (UTC +2) Squeeze Time: UTC (UTC +0) In this case, the tokens that were generated are stored in CEST. If, in this scenario, the Squeeze application checks whether a token needs to be refreshed, the applicationwill consider the token as not yet expired because itassumes the token is valid in the context of Squeeze. This is because Squeeze checks the token in the past with UTC +0, but the token was issued with CEST (UTC +2). Therefore, the token has already expired in CEST but not in UTC, which Squeeze checks for. The problem has been fixed in version 2.5. if you don't have this version yet, try to synchronize the server time with the application time. This can be done by configuring the date.timezone in the php.ini. See: https://www.php.net/manual/en/datetime.configuration.php

Revision #9
Created 1 March 2023 14:15:33 by Tim Glaesner
Updated 6 October 2023 08:03:40 by Phillip Langer