1. How to configure a Microsoft OAuth Provider

1. Intro

This guide will explain how to configure a client on Microsoft Azure and then use the information from Microsoft Azure to configure an OAuth Provider in KiyoCRM.

2. Register KiyoCRM App in Microsoft Store

The following steps assume that you are configuring a provider for a company account that will be shared among many users.

2.1 Go to App Registrations

Go to https://portal.azure.com/ and login

On the home page you should see a screen like the following

azure-homepage.png

Check if there is a link to **App Registrations** on the home page, should look something like the following

azure-services.png

Otherwise, go to **More services** and then search for the **App Registrations** link

more-services.png

azure-services-app-registration-entry.png

The Azure **App Registrations** page looks like the following

azure-app-registrations.png

2.2 Create a New App Registration

On the **App Registrations** page click on the "New Registration" link

azure-new-app-registration-link.png

You should then see the App registration creation page

On the registration page complete the following:

1. Add a name for the registration like "KiyoCRM"
2. Select one of the "Supported account types" depending on your needs
    1. **Note** In this example we are going to use "Single tenant", that does not mean it is the one to be used for every scenario.
    2. You should select the option that is appropriate for your use case
3. Set a "Redirect URI". This should be similar to: `https://<your-kiyocrm-instance-host>/index.php?entryPoint=setExternalOAuthToken`
    1. **Note** Azure only allows https connections to host that aren't the localhost. (the azure interface will alert you to it)
        1. A host like `http://my-kiyocrm.com/index.php?entryPoint=setExternalOAuthToken` is not valid
        2. A host like `https://my-kiyocrm.com/index.php?entryPoint=setExternalOAuthToken` is valid
        3. A host like `http://localhost/index.php?entryPoint=setExternalOAuthToken` is valid

Screenshot 1.png

After filling in the information proceed with the registration

After the registration you should be re-directed to the "detail view" of the registration.

There you will find **Application (client) ID** that is the **Client Id** that will need to be configured later on KiyoCRM.

azure-client-id.png

2.3. Create a client secret

The next step to take is to generate a "client secret" that will be used by KiyoCRM to communicate with Microsoft.

On your App Registration main page click "Add a certificate or secret".

azure-add-secret.png

This should take you to the "Certificates & secrets" page

Here you can generate a new secret by clicking "New Client Secret"

azure-new-client-secret.png

A sidebar should open where you can place the name of the secret you want to give to you secret and its duration. After setting the name and the duration click "Add".

Screenshot 2.png

This should generate a new line on the "Certificates & secrets" page table

Screenshot 2025-05-22 154034.png

You should copy the "Value" by (clicking the icon next to it) for your new secret and store it in some safe place. You won't be able to access this value again from the Microsoft Azure interface. The secret will be required on the KiyoCRM documentation.

azure-secret-value.png

We recommend noting the expiry date for your Client Secret and replacing the secret on a regular basis. Once the secret expires, your email connections will fail with an error message in the logs similar to `OAuthAuthorizationService::hasConnectionTokenExpired | Access token has expired` If you see this error, generate a new secret in Azure and update it within KiyoCRM to ensure continued access.

Now go back to your App Registration main page, by clicking "Overview" on the sidebar.

2.4. Define the authorized scopes

Our next step is to configure the scopes that KiyoCRM will be able to access.

Back on your App Registration main page

Click on "View API" permissions

azure-view-api-permissions-link.png

This should take you to the "Api Permissions" page

Now lets add the scopes that we allow. Click "Add a permission"

azure-add-permission-link.png

This should open a sidebar

azure-add-permission-sidebar-clean.png

The permissions we want are under the Microsoft Graph. So click on "Microsoft Graph".

azure-microsoft-graph-permissions-link.png

After clicking you should be prompted with which kind of permission you want to use.

azure-microsoft-graph-permission-types.png

Go ahead and click "Delegate permissions"

azure-delegated-permissions-link.png

You should now see a new "Select permissions" section.

In the search bar type `offline_access`, that should show the offline_access permission. Select it add then click "Add permissions"

azure-add-offline-access-permission.png

Repeat the process for the following permissions: . `IMAP.AccessAsUser.All` . `User.Read`

azure-add-imap-permission.pngazure-add-user-read-permission.png

After you add all the above permissions, your permissions table should look something like the following:

azure-api-permissions-page-with-values.png

Now go back to your App Registration main page, by clicking "Overview" on the sidebar.

2.5. Define the Return URI

Our next step is to set the configurations on the "Authentication" like the Return URI and others.

Back on your App Registration main page

Click the "Redirect URIs" link

azure-redirect-uri-link.png

This should have taken you to the "Authentication" page.

Here you should see the Return URI that you previously configured

azure-redirect-uri-list.png

After that enable the "Access tokens (used for implicit flows)" option on the Implicit grant and hybrid flows section

azure-enable-access-token.png

Now go back to your App Registration main page, by clicking "Overview" on the sidebar.

2.6. Retrieve the endpoint information

The last step you need to take is to retrieve the following endpoints:

1. OAuth 2.0 authorization endpoint
2. OAuth 2.0 authorization endpoint (v2)

Back on your App Registration main page

Click in the "Endpoints" link.

azure-endpoints-link.png

This should open a sidebar like the following

azure-endpoints-sidebar.png

From the sidebar copy and take note of the following endpoints, that will be required to configure KiyoCRM

azure-required-endpoints.png

3. Configure the Microsoft Provider in KiyoCRM

In the following steps we are going to configure a provider that can be used by multiple users within KiyoCRM. This scenario only makes sense when you have registered an app in azure for accounts that share the same domain, usually accounts that are not @outlook accounts or similar.

Since we are going to configure a Group OAuth provider, the following steps are needed to be done by an admin user

Login into KiyoCRM as an admin user and go to the admin panel

image.png

On the admin panel search for "External OAuth Providers"

image.png

Click on the "External OAuth Providers", that should take you to the module list view

suitecrm-external-oauth-provider-module.png

As an admin you can create two types of records:

1. Personal
    1. These are only accessible by the user that created them
    2. Personal records are meant to be used when configuring access to personal accounts in exiting providers, without a custom domain. I.e. when for instance configuring access to your `@gmail` or `@outlook` accounts. These can have a shared group configuration as for each used the "Client Id", "Client Secret" and other fields are going to be unique per account.
2. Group
    1. Records that will be used by many users
    2. Group records are meant to be use by everyone that have accounts that share the same domain, lie `@example-inc.onmicrosoft.com`. The "Client ID", "Client Secret" and the other fields are going to be the same for all accounts that use this domain.

As mentioned before in this guide we are going to configure a provider that will be used by multiple users, so we are going to create a group record.

Click on the "New Group OAuth Provider" which should take you to the create view.

Add a meaningful name to your provider, it can be just "Microsoft", the name of your domain, or something that helps you identify and differentiate the provider from other possible providers.

Then select the "Microsoft" on "Connector" field

suitecrm-connector-selection.png

The Microsoft connector type works in the same way as the "Generic" connector type. The difference is that it has several built in defaults. This spares you some configuration steps and makes the process of configuring an External OAuth provider easier.

Add the "Scopes" that you want to access, the same ones you've configured on Azure's API Permissions page.

1. `offline_access`
2. `https://outlook.office.com/IMAP.AccessAsUser.All`
3. `User.Read`

suite_crm_oauth_provider_scopes.png

Set the "Client Id" that was generated in Azure

suitecrm-oauth-provider-client-id.png

Set the "Client Secret" that was generated in Azure

suitecrm-oauth-provider-client-secret.png

Set the "Authorize URL" that you've copied from the Azure endpoints (OAuth 2.0 authorization endpoint (v2))

azure-authorize-url.pngsuitecrm-oauth-provider-authorize-url.png

Set the "Access Token URL" that you've copied from the Azure endpoints (OAuth 2.0 token endpoint (v2))

azure-access-token-url.pngsuitecrm-oauth-provider-access-token-url.png

Your record should now look something like the following:

suitecrm-oauth-provider-filled-example.png

Since many of the other fields have defaults and the Microsoft connector adds other defaults, these are all the field that you should need to configure.

You can now save the record

4. Configure User Inbound emails

Now users should be able to use the OAuth Provider you have created for authenticating with Microsoft.

Check the How to configure Inbound Email with OAuth guide for the steps users need to take to configure their inbound emails using an OAuth connection.

Revision #7
Created 20 May 2025 09:17:21 by Admin
Updated 22 May 2025 10:31:26 by Admin