4. External OAuth Provider Overview

1. Intro

This guide tries to provide a more in-depth description of the configurations in the External OAuth Provider module

2. Types Of records

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 intance 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.

Please check the Configuring Security Groups for Inbound Email on more information in the differences between the two and how to use them properly.

3. OAuth2 Client lib

KiyoCRM is using the OAuth 2.0 Client from The PHP League. It uses the Generic Provider for all the providers.

4. Base Config_

On the base configuration, depicted below, you can find all the fields that are essential for the OAuth Provider to work. This may vary depending on the provider that you will be using, but most of these field are common for most providers.

suitecrm-oauth-provider-base-config.png

5. Extra Configurations

These are extra configurations that you may need to add depending on the provider. All the below are merged with any existing defaults.

1. Extra Provider Params: are parameters to inject to the provider when initializing. These may be used on all requests
2. Get Token Request grant type: the type of grant to use in the request to retrieve the token
3. Get Token Request options: Extra options that can be sent on the request to retrieve the token
4. Refresh Token Request Grant Type: the type of grant to use in the request to refresh the token
5. Refresh Token Request Options: Extra options that can be sent on the request to refresh the token

suitecrm-oauth-providers-extra-config.png

6. Mapping Configurations

These define how to map the fields received on the token, to the fields where they are store in KiyoCRM. All the except for token type have defaults.

In this configuration it is possible to retrieve values nested within an array. For that you can define the path to the field using the key names and using `.` to join them. E.g. Microsoft connector uses the following mapping to get the token type `values.token_type`

suitecrm-oauth-provider-mapping-config.png

7. Connectors

Connectors are the part of the code that is responsible for managing the requests to the provider to do the authentication

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.

8. Internal Configuration Format

In order to understand how/when the fields from the External OAuth Providers module are used, it may help to understand the internal structure.

The following code snippet is an example of the configurations without the shortcuts from the frontend.

```
[
    'type' => 'Generic',
    'client_id' => '...',
    'client_secret' => '....',
    'redirect_uri' => '....',
    'authorize_url_options' => [
        'scope' => '...',
    ],
    'extra_provider_params' => [
        'scopes' => '...',
        'urlAuthorize' => '...',
        'urlAccessToken' => '...',
    ],
    'get_token_request_grant' => 'authorization_code', // optional
    'get_token_request_options' => [], // optional
    'refresh_token_request_grant' => 'refresh_token', // optional
    'refresh_token_request_options' => [], // optional
    'token_mapping' => [ // optional
        'access_token' => '...',
        'expires_in' => '...',
        'refresh_token' => '...',
        'token_type' => ''
    ]
]
```

From the above example you can see that:

1. `scope` is injected into both the `authorize_url_options` and the `extra_provider_params`
2. `urlAuthorize` is injected into `extra_provider_params`
3. `urlAccessToken` is injected into `extra_provider_params`

Revision #4
Created 20 May 2025 10:44:30 by Admin
Updated 22 May 2025 09:55:40 by Admin