Kiuwan OAuth 2.0 and OIDC Integration

OAuth 2.0 is an authorization framework that allows software applications to access protected resources that belong to somebody or something. The software applications can obtain total or partial access to the protected resources, either on behalf of a resource owner by orchestrating an approval interaction between the resource owner and the HTTP service or by allowing the software applications to obtain access on their own behalf.

Additionally, OIDC adds a simple identity layer (adding end-users authentication-related data in the payload) on top of the OAuth 2.0, making simple integration with external systems that support some of the grant types defined in OAuth 2.0 specification (OAuth 2.0 Authorization Framework) i.e. most major online services nowadays.

Contents:

Web Browser Single Sign-On

When configured, a Kiuwan user can authenticate against an OAuth/OIDC server, and access Kiuwan resources.
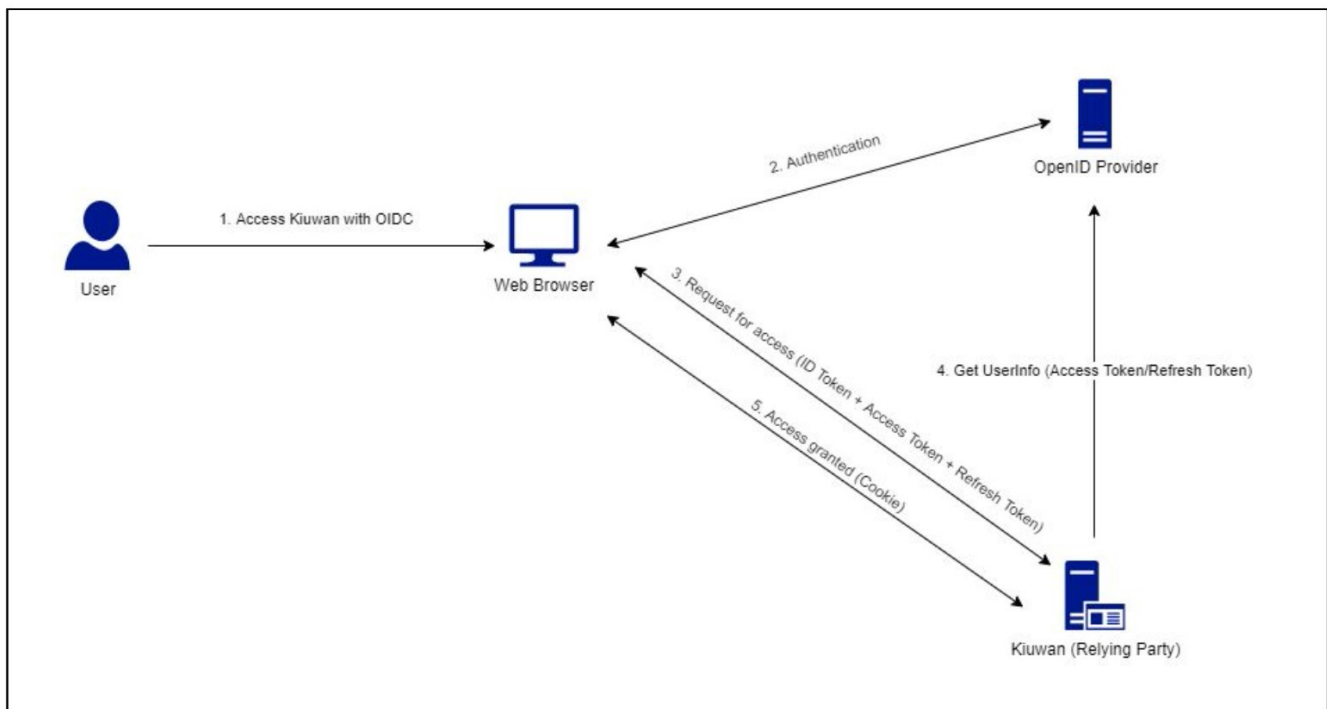
Before the user can start using Web Browser SSO flow, the administrator of the OpenID Provider in their organization must first have registered Kiuwan as a Relying Party, and subsequently, the account owner of the Kiuwan user account must have registered their OpenID Provider in Kiuwan.

Configuring OpenID Provider

The following describes how to setup Kiuwan as a registered application on Azure ADFS, any other OAuth2.0/Open ID Provider with a similar configuration.

To enable the authentication through an OAuth/OIDC server when configuring Kiuwan as a client application is required to be obtained from OAuth Issuer the following:

1. **URL of the issuer**: The base URL provided by the OAuth provider that will be used for authentication and exchange of tokens.
2. **OIDC Metadata URL**: e.g. .well-know/opendi-configuration URL while not mandatory by OpenID connect standard, the auto-configuration enables clients to derive all server configuration.

3. **Application ID**: The name of the Kiuwan application as registered in the Oauth2.0 / OIDC Server.
4. **Client Secret**: Kiuwan OIDC Client demands encrypted tokens, so a client secret, produced by the Oauth2.0/OIDC Server must be provided.
5. **The username mapping**: The name of the OAuth/OIDC token field that enables Kiuwan to obtain the corresponding user in Kiuwan. For instance, if a user email is used to define the Kiuwan username, then the user mapping could be "e-mail". Any other field described by the Oauth /OIDC server claims can be used, as long as the resulting value matches the Kiuwan user. It is common for the issuer server to include in the authentication token the field preferred_username, which usually maps to the username in the Identity Provider OAuth issuer.
In Azure AD, preferred_username matches the Unique Principal Name, e.g. the unique username of the identity within the AD definition.

Registering Kiuwan as an Oauth/OIDC client in Azure AD

1. Login to your Azure Portal and navigate to **Active Directory > App Registrations**.
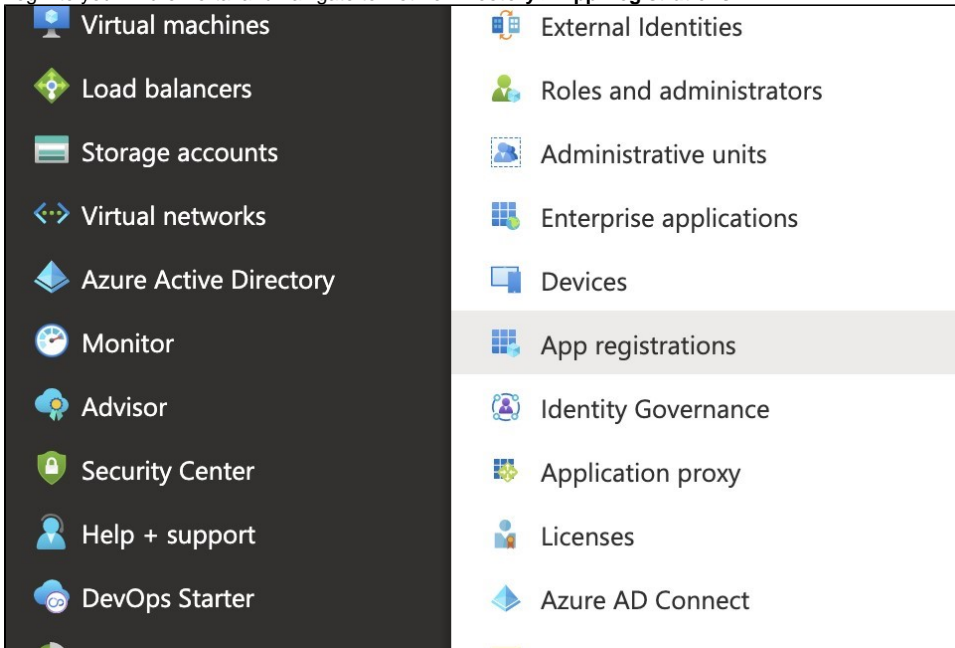
2. Select **New registration.**
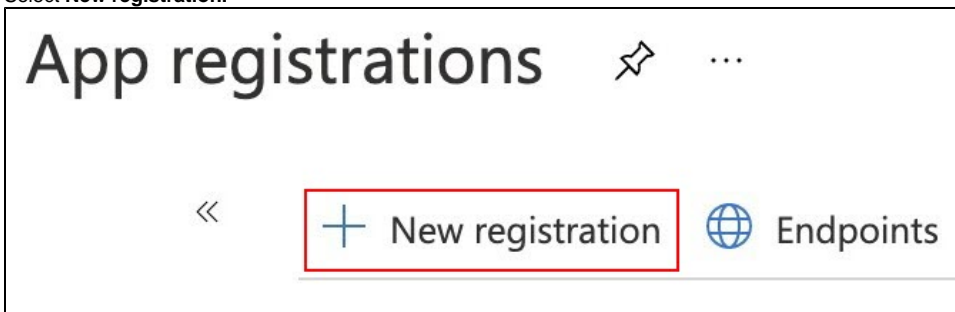
3. Fill in the required information

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 (`DOMAIN` only - Single tenant)

( ) Accounts in any organizational directory (Any Azure AD directory - Multitenant)

( ) Accounts in any organizational directory (Any Azure AD directory - 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.

[Web   ⌄] [e.g. https://example.com/auth   ]

4. Configure the Redirect URI as follows:
   a. Add the following URI as the redirect URI: https://www.kiuwan.com/saas/oidc/auth
   b. The endpoint OAuth2/OIDC server will redirect, within the Kiuwan server, after the successful authentication of a specific user.
   c. If you have a local installation of Kiuwan, instead of www.kiuwan.com enter the corresponding URL, or IP address, of the local Kiuwan server.
   d. After defining this Application Registration, the following is obtained:

foobar 📌 ···

Search (Cmd+/)	«	🗑 Delete ⊕ Endpoints ⊞ Preview features

⌃ Essentials

Display name	: foobar		Supported account types	: My organization only
Application (client) ID	: ea6913b8-06d4-4176-b8e0-d253190e041f		Redirect URIs	: 1 web, 0 spa, 0 public client
Directory (tenant) ID	: a661771f-a950-4156-b52b-ad98deec4638		Application ID URI	: Add an Application ID URI
Object ID	: 86ce1fb8-bb63-482c-b883-26ea4e51e15d		Managed application in l...	: foobar

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

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

Overview
Quickstart
Integration assistant

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

Support + Troubleshooting
Troubleshooting
New support request

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

Microsoft identity platform
Authentication scenarios
Authentication libraries
Code samples
Microsoft Graph
Glossary
Help and Support

5. Configure Client Secret.
   a. After creating the registration, select the **Certificates & Secrets** option, and **New Client Secret.**

	Thumbprint	Start date	Expires	ID
Certificates & secrets				
Token configuration	No certificates have been added for this application.			
API permissions				
Expose an API				
App roles	**Client secrets**			
Owners	A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.			
Roles and administrators	Preview				
Manifest	[+ New client secret]			

Support + Troubleshooting

Description	Expires	Value	ID
Troubleshooting			
New support request	No client secrets have been created for this application.		

b. Enter the needed values and click **Add.**

Add a client secret

Description

Expires
- In 1 year
- In 2 years
- Never

[Add] [Cancel]

c. A new secret for this Client Application ID is generated.

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.

+ New client secret

Description	Expires	Value	ID
fooobar secret	3/16/2022		200a67d1-eb82-4565-8968-69554d0b94b5

Copy this value to a temporary location as it will be needed later and after the screen is closed the access to the secret value is lost. Ideally, paste it directly into the OAuth/OIDC configuration at Kiuwan described in the next chapter.

In the previous steps, you have created a new registry for Kiuwan to integrate with OAuth/OIDC server.

The following information was generated, that would be needed in configuring Kiuwan to use this registry:

Application ID:

Delete   Endpoints   Preview features

∧ Essentials

Display name   : foobar

Application (client) ID : ea6913b8-06d4-4176-b8e0-d253190e041f

Application Secret:

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.

+ New client secret

Description	Expires	Value	ID
fooobar secret	3/16/2022		200a67d1-eb82-4565-8968-69554d0b94b5

Additionally, you would also need to collect the following endpoints:

OpenID Connect metadata document

https://login.microsoftonline.com/⬛⬛⬛⬛⬛v2.0/.well-known/openid-configuration

OAuth 2.0 authorization endpoint (v2)

https://login.microsoftonline.com/⬛⬛⬛⬛⬛/oauth2/v2.0/authorize

Registering Kiuwan as an OAuth/OIDC client in Windows ADFS 2016

Similar to Azure DevOps, as with any other OAuth/OIDC issuer or identity provider, the configuration of ADFS follows the specifics defined for that Idp. However, information needs to be collected during the setup of Kiuwan as a registered app, as detailed in section Summary:

- **Application ID:** As manually entered in ADFS.
- **Application Secret:** Produced by ADFS during the setup.
- **ADFS issuer URL:** The base URL where Kiuwan is redirected to authenticate the user in ADFS.
  (https://{YOUR_ADFS_HOSTNAME}/adfs)
- **ADFS Metadata URL:** The URL in ADFS providing the openid-configuration.
  (https://{YOUR_ADFS_HOSTNAME}/adfs/.well-known/openid-configuration)

The process of registering Kiuwan as an OAuth/OICD application is straightforward.

1. Start **ADFS Management**.

2. Right-click the **Application Groups** folder from the left side of the window, and select **Add Application Group.**
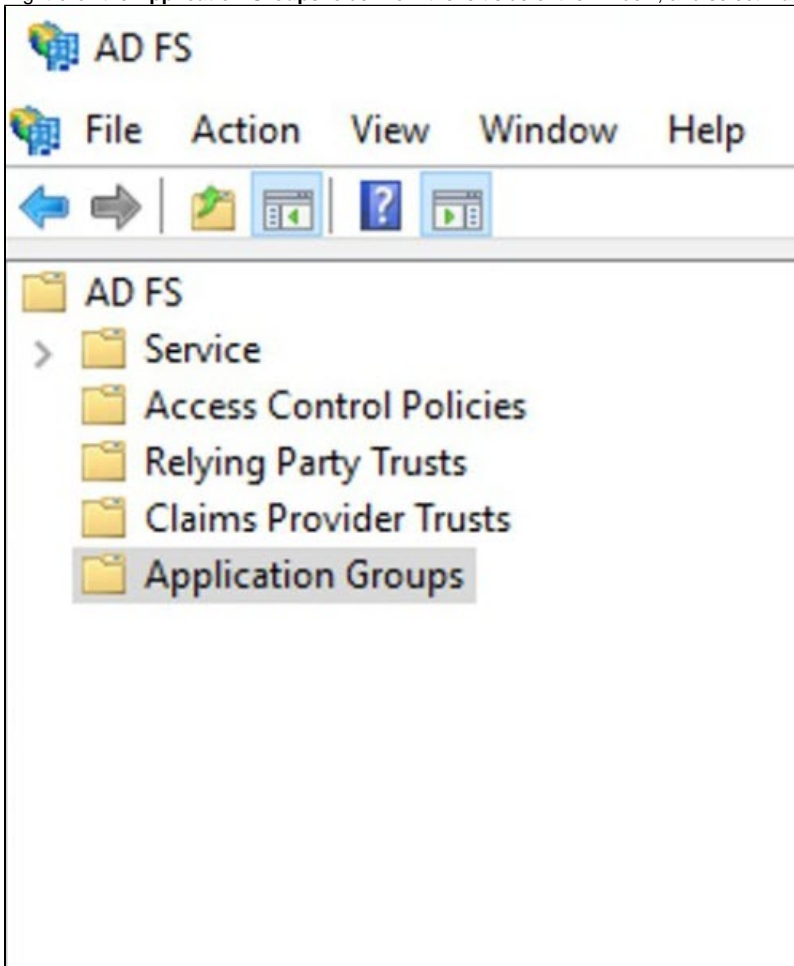
3. Fill in the required **Application Group Name**, **Description**, select the option **Server application accessing web API**, and click **Next**.
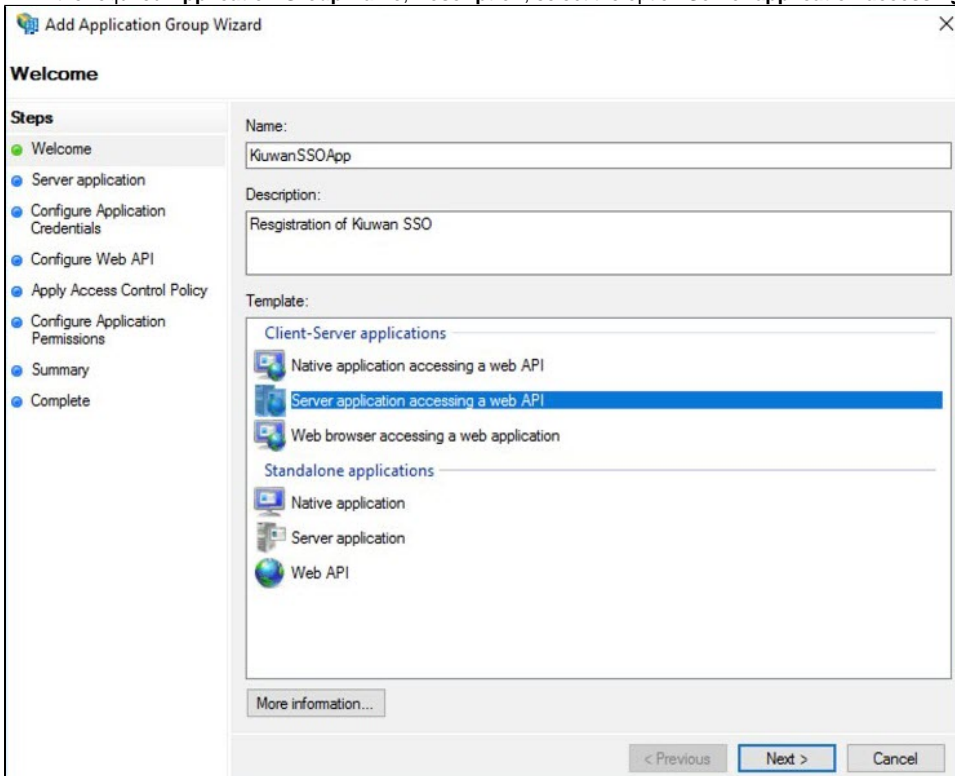
4. Enter the **Name** of the Server Application and collect – and save in a temporary location the **Client Identifier** generated automatically. This information will be available for later consultation but can be copied now and pasted to the Kiuwan SSO configuration form as described in Click **Next**.

5. Next, add the Kiuwan Login entry point URI as a valid URI for redirection, e.g. https://www.kiuwan.com/saas/oidc/auth or the corresponding if you are using Kiuwan On-Premises or a custom installation instead of the standard Kiuwan cloud setup. Click **Next**.

6. On the **Configure Application Credentials** tab, check the option **Generate a shared secret**, this will create the corresponding secret to be copied into the Kiuwan OAuth configuration panel, as described in Setting up Kiuwan SSO with OAuth/Open ID Connect. Click **Next**.

✅ Copy the generated the shared secret for a temporary location as it will not be available again, and if needed it will need to be regenerated.

7. Configure the **Web API for ADFS** by entering the name of the API and a description for the identifier of this Web API in ADFS. Click **Next**.

8. Select the **Permit Everyone** option as a policy for Access Control and click **Next**.

9. Configure Application Permissions, select **OpenID** and **Profile** as Permitted scopes. If needed or desired additional scopes can be added to the list, this configures the available information about the authenticated user to be passed to Kiuwan.
If more information is required in Kiuwan to match the ADFS user with the Kiuwan user, it is allowed to select or even define new scopes. As minimal both OpenID (default) and profile scopes are needed by Kiuwan. Click **next**, to confirm the summary information.

10. Click **Close** to complete the process.

As stated, the application group and corresponding client are now configured.

Registering Kiuwan as an OAuth/OIDC client in OKTA

To use OKTA as an OAuth SSO identity provider, it is required to register Kiuwan in OKTA.

Follow these instructions to register Kiuwan in OKTA:

1. Go to Applications and select **Add Application**.
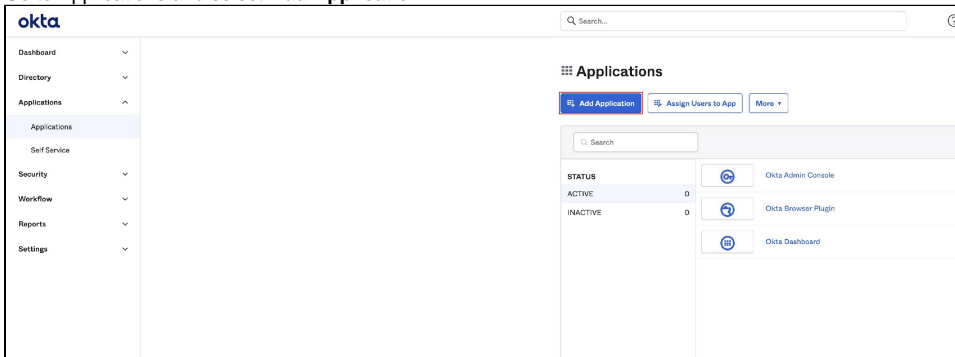
2. Click **Create New App**.
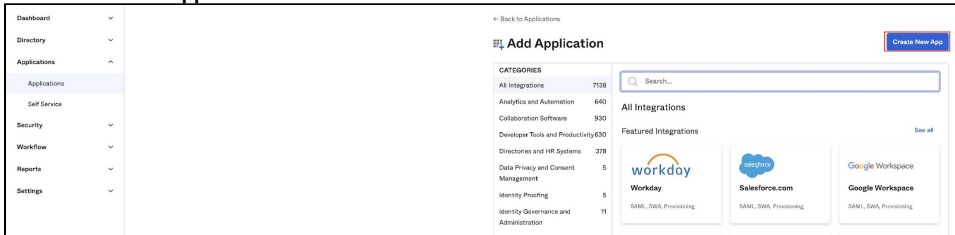
3. On the **Create a New Application Integration** window, select **Web** as Platform entry and select **OpenID** Connect as a Sign on method. Click **Create**.

4. In the following screen fill in the required information, adding https://www.kiuwan.com/saas/oidc/auth as a Redirect URI. Click **Save**.

5. Collect the needed information to configure Kiuwan, the application ID, and client secret. Note that the latest will not be available anymore after passing the following screen:

Client Credentials

Edit

Client ID

OoasumtxmQ7dx5Zt25d6

Public identifier for the client that is required for all OAuth flows.

Client secret

●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●   👁

Secret used by the client to exchange an authorization code for a token. This must be kept confidential! Do not include it in apps which cannot keep it secret, such as those running on a client.

General Settings

Edit

Okta domain

shiftleft-kiuwan-sso.okta.com

APPLICATION

Application name	Kiuwan SSO Web App
Application type	Web
Allowed grant types	Client acting on behalf of itself

☐ Client Credentials

Client acting on behalf of a user

☑ Authorization Code
☐ Refresh Token
☐ Implicit (Hybrid)

6. On the tab collect the issuer URI needed for configuring Kiuwan.

General	**Sign On**	Assignments	Okta API Scopes

Settings

Sign on methods

The sign-on method determines how a user signs into and manages their credentials for an application. Some sign-on methods require additional configuration in the 3rd party application.

Application username is determined by the user profile mapping. Configure profile mapping

◉ OpenID Connect

Token Credentials   Edit

Signing credential rotation ⍰   Automatic

OpenID Connect ID Token   Edit

Issuer	https://shiftleft-kiuwan-sso.okta.com
Audience	0oasumtxmQ7dx5Zt25d6
Claims	Claims for this token include all user attributes on the app profile.
Groups claim type	Filter
Groups claim filter ⍰	None ⮥ Using Groups Claim

OKTA allows for additional configuration and provides extensive support, make sure you tailor the configuration to your needs.

Setting up Kiuwan SSO with OAuth/Open ID Connect

To set up Kiuwan SSO with OAuth/OpenID connect follow these instructions:

1. As Account Administrator open the **Account Management** section:
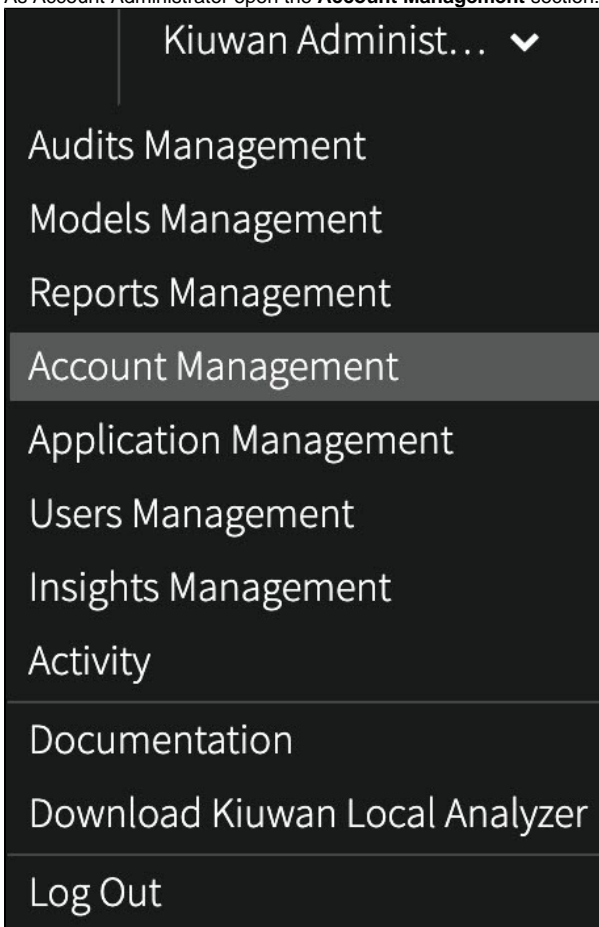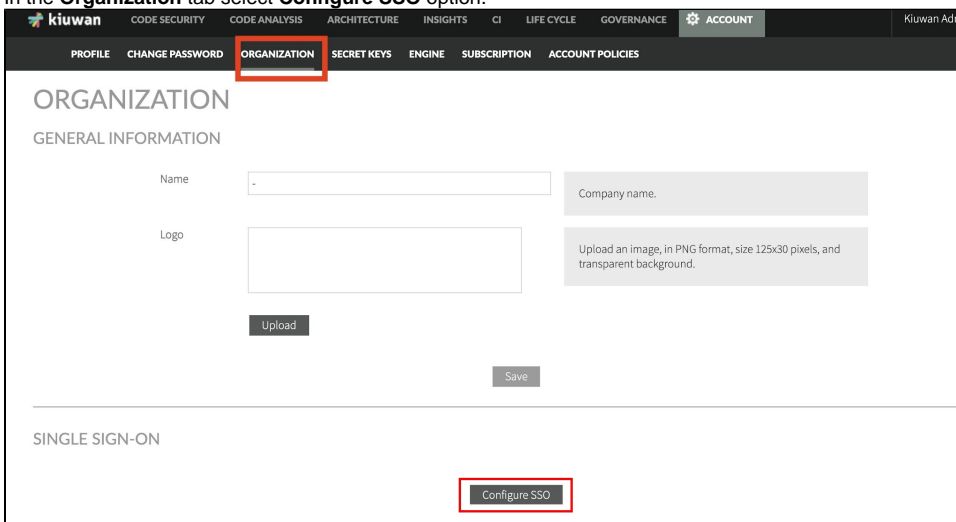

2. In the **Organization** tab select **Configure SSO** option.

3. Select **OAuth2/OpenID Connect** option and confirm.

SSO TYPE   ✖

SELECT A SSO TYPE TO CONFIGURE FOR YOUR ORGANIZATION

○ SAML

◉ Oauth2/OpenID Connect

Select SSO Type

4. Read the following text explaining the implications when activating SSO, click **Continue** to proceed to the OAuth2/OpenID Connect setup.

SSO CONFIGURATION   ✖

This page allows you to activate the access by SSO to the users of your organization. Nevertheless, before you start with the activation process, you should keep in mind the following:

- By activating the SSO in your account, all users of your account will be automatically migrated to your own domain to avoid conflict with other usernames in other Kiuwan accounts.
- After this migration, all users of your account must use a new URL for the Login, leaving the login URL that you have been using until now. This new URL will be communicated to you in the next steps of this page.
- In order to continue using the Kiuwan Local Analyzer, API REST, Kiuwan for Developers, or any other plugin that needs to request for some data to Kiuwan, you must change the configuration and indicate the DOMAIN ID in their respective configuration screens. This DOMAIN ID will be provided when you activate the SSO.
- Once activated the SSO, you must communicate to all your users the new login URL and your DOMAIN ID.
- Once SSO is activated, it is NOT possible to disable it or re-migrate users to the previous Kiuwan domain.
- Event though the activation process is completed, you will need to register Kiuwan as SP in your IdP. Till then, you can not use SSO.

To continue with the activation process, click on 'Continue'.

Continue

5. Fill in the information, as gathered during Configuring OpenID Provider, and confirm entered data by selecting **Enable new IdP**.

After confirming the SSO data, a new email is sent with the activation confirmation code, alongside the instructions explaining how to log in to

To continue with SSO activation process, please enter the following activation code in the screen:

Activation Code:▮▮▮▮▮▮▮▮

Remember that after SSO activation, you have to use the following link to log into Kiuwan using your username and password:

Login URL: http://▮▮▮▮▮▮▮▮/saas/web/login.html?
domain=93a8096f169c96794d6922ea25d7678ee98a41c1f5026f865568f5724165768ff1fdbb80392ad983fc5487a55fcaadd5d51f9971a69887503e6d476b48df2f29

And this is your DOMAIN ID, necessary to use the Kiuwan Local Analyzer, Kiuwan REST API, and Kiuwan plugins:

DOMAIN ID:
93a8096f169c96794d6922ea25d7678ee98a41c1f5026f865568f5724165768ff1fdbb80392ad983fc5487a55fcaadd5d51f9971a69887503e6d476b48df2f29

Thanks for using Kiuwan. Have fun analyzing!
The Kiuwan Team

Kiuwan after the SSO is activated.

**6.** Copy the confirmation code and enter it in the confirmation form.

⚠ When the option **Disable Login with password for all my users** is checked, it prevents any user registered in your account, with the exception of Account Owner user, to be able to login using Kiuwan managed identity username/password.

Users will be able to log in only through defined OAuth/OIDC server authentication.

7. After entering the activation code a confirm the activation of SSO, the following confirmation dialogue displays:

SSO CONFIGURATION   ✖

Congratulations! SSO is now activated in your account.

Close
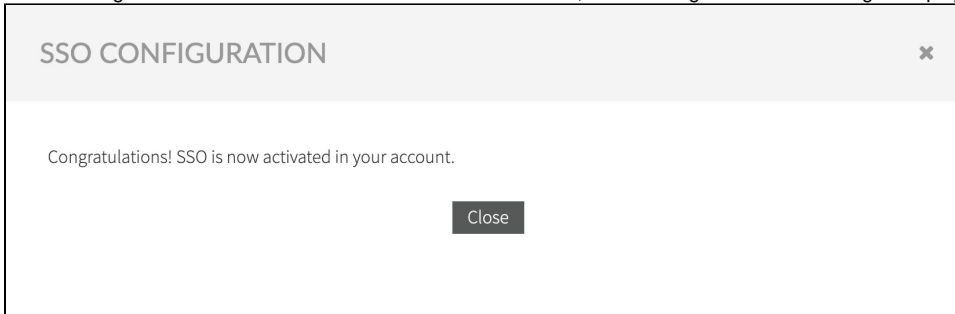
Login to Kiuwan after OAuth SSO is activated

As explained in the confirmation email received during the configuration process, and as shown, after SSO setup in the Organization tab in the Account Management section, after SSO is completed the login must comply with the following URL:

https://<kiuwan.server.url>/saas/login?domain=<the.created.domain>&**sso=on**

This link opens the following page, click login to redirect the user to the configured authentication page at the OAuth/OIDC identity management system.

Welcome to Kiuwan
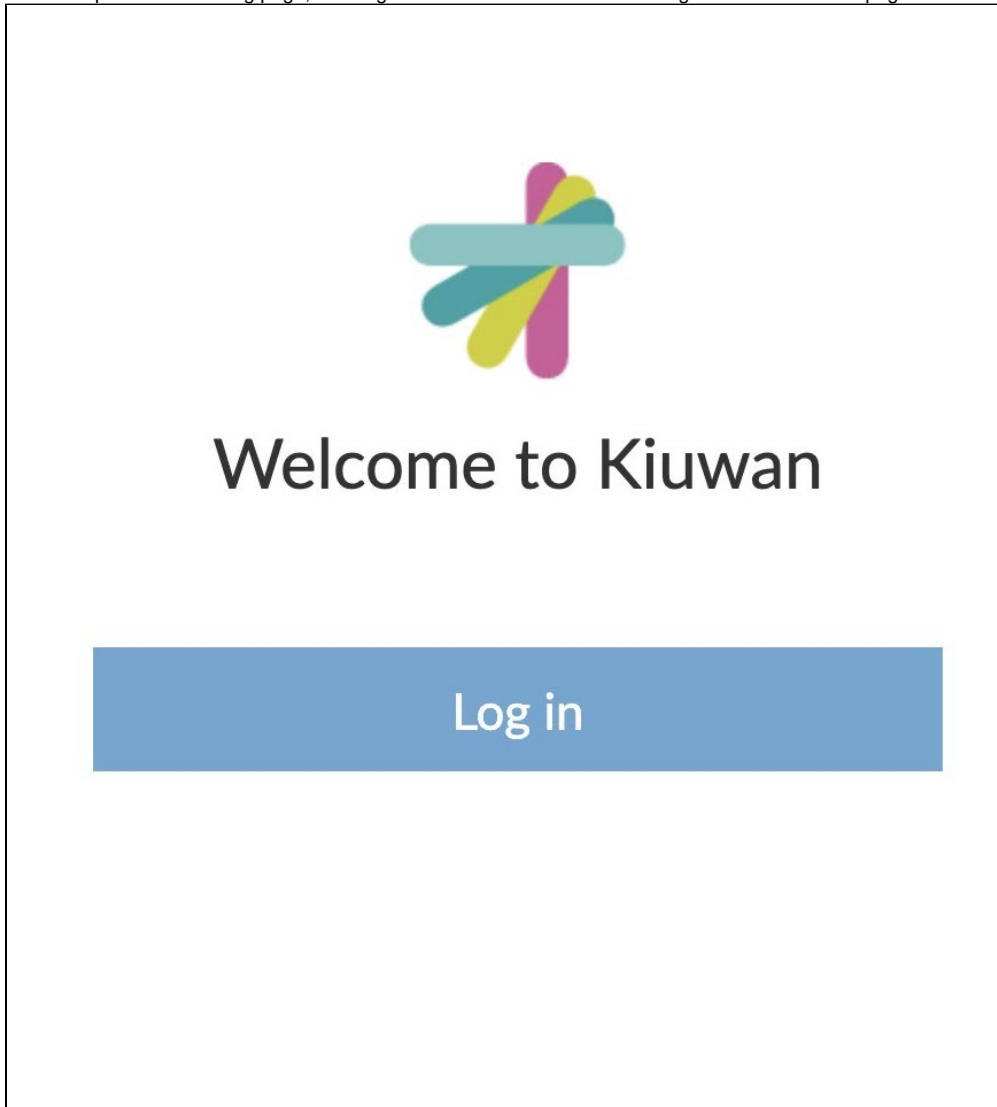
Log in

Successful authentication will enable access to Kiuwan, with the corresponding authorization configured for the logged-in user.

Login in with Kiuwan defined username and password

When the user is configured to login with Username and Password in Kiuwan, as shown below:

It is possible to login into Kiuwan bypassing the SSO, this is particularly useful for accessing Kiuwan through the provided REST API, or when SSO is unavailable. To do so, use the following construct to access the Kiuwan console application:

https://<kiuwan.server.url>/saas/login?domain=<the.created.domain>&**sso=off**

It opens the standard Kiuwan Login screen:

In either case, after setting up SSO, the domain must always be set during the authentication process, otherwise, Kiuwan will not be able to identify the corresponding user.

Setup KLA to use Oauth2 SSO

After an account is configured to use SSO through OAuth2, this configuration becomes also available to KLA.

To enable KLA to use OAuth2/OIDC in the login form follow these steps:

1. Select the **Advanced** option.
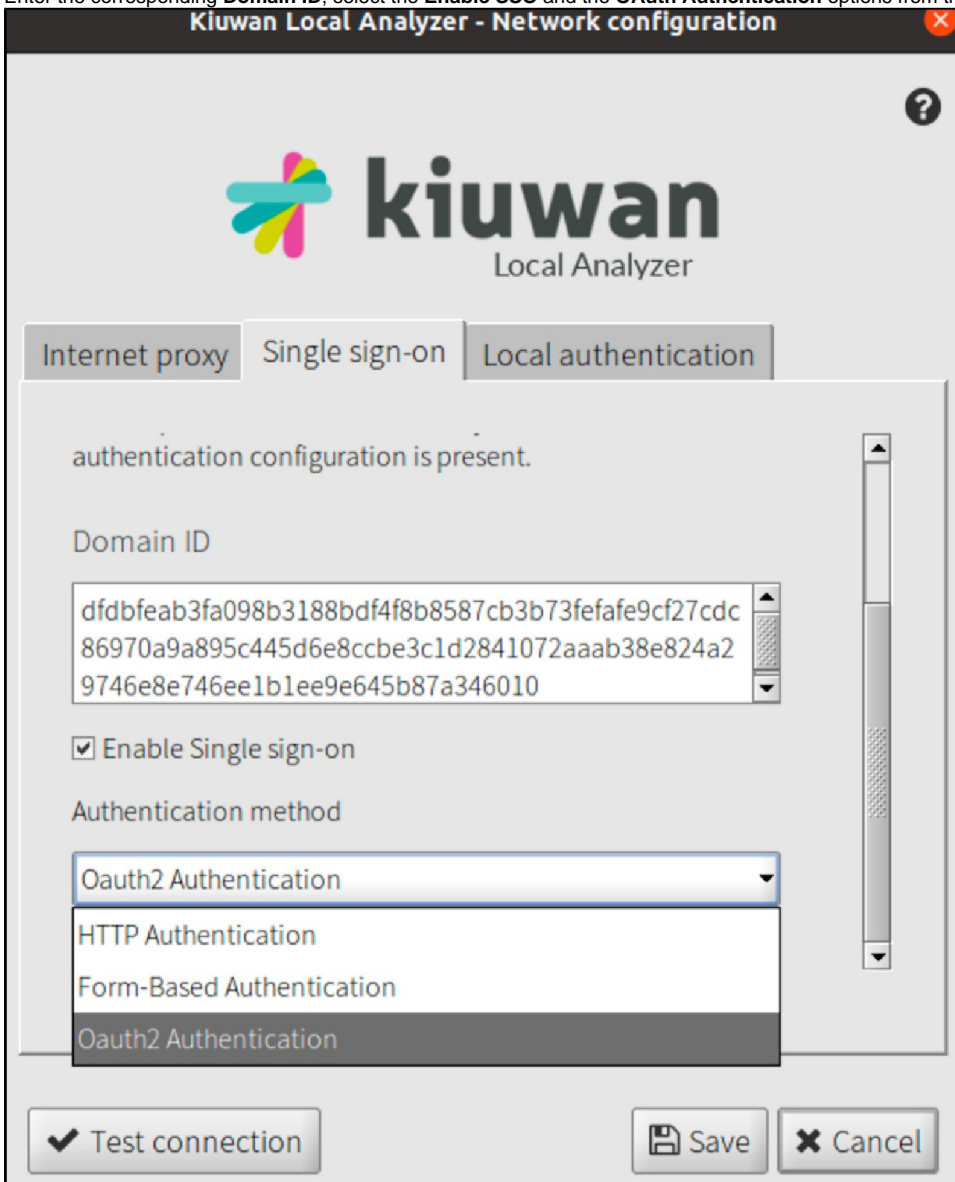
2. Enter the corresponding **Domain ID**, select the **Enable SSO** and the **OAuth Authentication** options from the pull-down list.

KLA is now configured to use the OAuth authentication as configured for that Kiuwan Account.

3. Click **Test Connection** and KLA will redirect the authentication procedure to the native web browser:
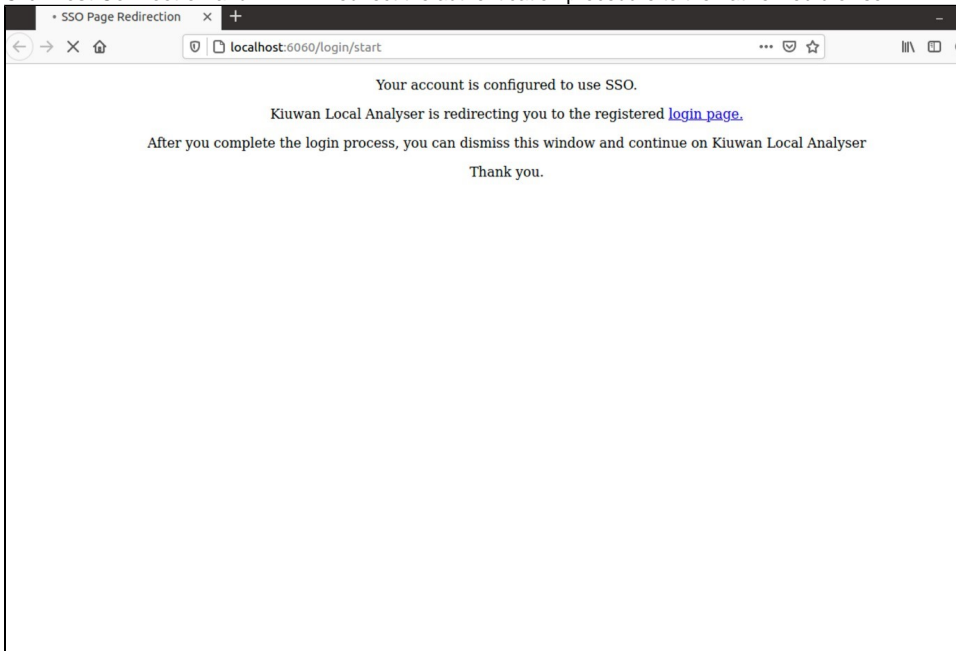
The above page is temporarily presented informing the user that the login process is underway, and finally the regular authentication process for the configured Identity provided is initiated.

After the authentication process is completed, the browser window can be discarded as KLA has collected the needed information to proceed.
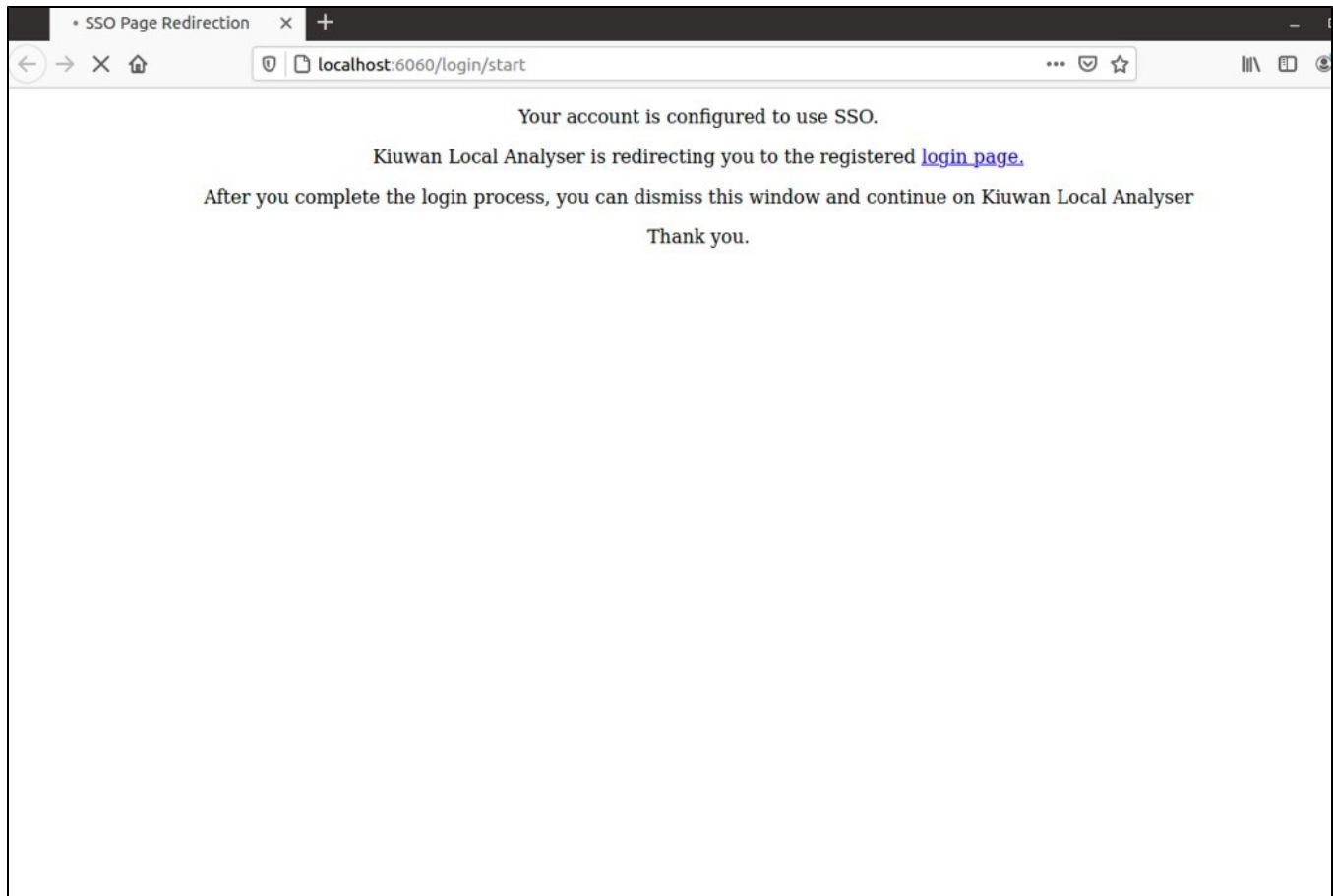
Login to KLA after OAuth SSO is activated

When KLA is configured to use SSO with OAuth, the Login form will show an indication that SSO is activated.

By selecting **Login** in the login form, KLA starts a new browser window to initiate the login flow.

Initially, the browser displays the following information:

Eventually, it would be redirected to the regular Authentication page of the configured Identity provider for the selected account.

Upon successful completion of the authentication process, the browser window can be dismissed, and KLA will be available to be used according to the user's corresponding capabilities.