SINGLE SIGN-ON

FOR (APEX) APPLICATIONS

USING OAUTH2

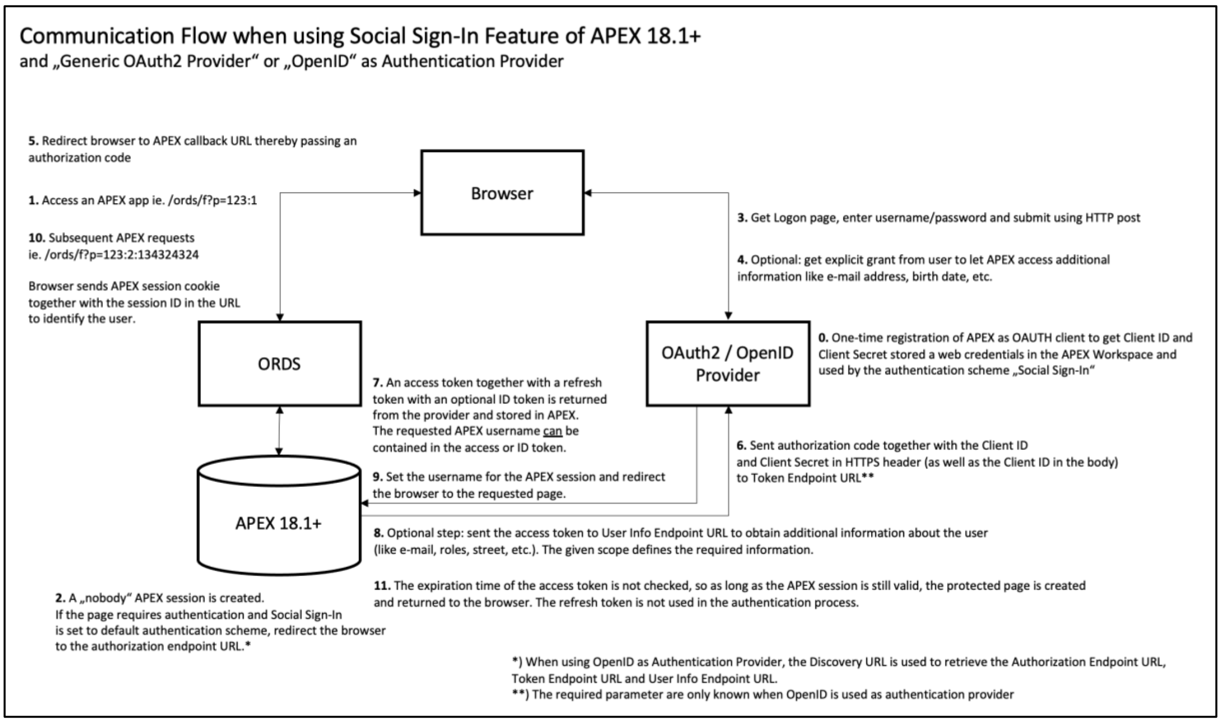
Author: Niels de Bruijn
Version: 3.0
Date: 26-JAN-2021

1 INTRODUCTION

A feature called “Social Sign-In” was introduced in APEX 18.1 which handles authentication based on the OAuth2 standard. Although the name suggests that it can only be used with Facebook & Co. as authentication provider, you can also use it with ie. Azure Active Directory or any other authentication provider that can handle OAuth2.

This document will explain the process flow for authentication based on OAuth2 as well as the implementation steps to undertake to achieve this goal when using Azure AD as authentication provider.

2 ARCHITECTURE & PROCESS FLOW



Communication Flow when using Social Sign-In Feature of APEX 18.1+
and „Generic OAuth2 Provider“ or „OpenID“ as Authentication Provider

5. Redirect browser to APEX callback URL thereby passing an authorization code

1. Access an APEX app ie. /ords/f?p=123:1

10. Subsequent APEX requests ie. /ords/f?p=123:2:134324324

Browser sends APEX session cookie together with the session ID in the URL to identify the user.

3. Get Logon page, enter username/password and submit using HTTP post

4. Optional: get explicit grant from user to let APEX access additional information like e-mail address, birth date, etc.

0. One-time registration of APEX as OAUTH client to get Client ID and Client Secret stored a web credentials in the APEX Workspace and used by the authentication scheme „Social Sign-In“

7. An access token together with a refresh token with an optional ID token is returned from the provider and stored in APEX. The requested APEX username can be contained in the access or ID token.

9. Set the username for the APEX session and redirect the browser to the requested page.

6. Sent authorization code together with the Client ID and Client Secret in HTTPS header (as well as the Client ID in the body) to Token Endpoint URL**

8. Optional step: sent the access token to User Info Endpoint URL to obtain additional information about the user (like e-mail, roles, street, etc.). The given scope defines the required information.

11. The expiration time of the access token is not checked, so as long as the APEX session is still valid, the protected page is created and returned to the browser. The refresh token is not used in the authentication process.

2. A „nobody“ APEX session is created.
If the page requires authentication and Social Sign-In is set to default authentication scheme, redirect the browser to the authorization endpoint URL.*

*) When using OpenID as Authentication Provider, the Discovery URL is used to retrieve the Authorization Endpoint URL, Token Endpoint URL and User Info Endpoint URL.
**) The required parameter are only known when OpenID is used as authentication provider

Here is what happens in detail when you access an APEX app protected by OAuth2 also known as “Authorization Code Flow”:

0. APEX is registered once as OAuth2 client at the provider. The given client ID (=app username) and client secret (=app password) are stored as “Web Credentials” on APEX workspace level.

1. The user accesses an APEX app ie. https://apex.mt-ag.com/ords/f?p=123:1.

2. APEX creates a session for “nobody” and checks if the page requires authentication. If so, the details of the current authentication scheme are retrieved from the metadata. If OpenID is used as authentication provider, the discovery URL is used to retrieve the endpoint/token/user info URL with its parameters. If “generic OAuth2 provider” was selected, these URLs have to be entered in APEX. The user is redirected to the authorization endpoint URL.

3. The user logs in ie. using username/password.

4. Depending on the provider used, the user may also need to explicitly consent access to resources like e-mail, street, ZIP code, etc.

5. After authentication, an authorization code was returned by the provider with which the browser is redirected to the APEX callback URL https://apex.mt-ag.com/ords/apex_authentication.callback.

6. APEX contacts the token endpoint URL over HTTPS thereby passing the authorization code, the client ID and client secret.

7. The provider returns an access token together with a refresh token, optionally an ID token as well. All tokens are stored in APEX.

8. Optionally, the user info endpoint URL is used to get additional information about the user. The provided "scope" determines the set of information requested. This additional information is returned in a JSON document. All attributes together with the ID- & access token are stored in `APEX_JSON.G_VALUES`.
Starting with APEX 20.2, the additional information can be mapped automatically to APEX items by APEX, so no coding is required to retrieve these values.

9. APEX retrieves the username from the tokens (access/ID), or the data from the user info endpoint. APEX then authenticates the user in the session and redirects to the originally requested page (step 1).

10. A subsequent browser request for an APEX page now includes an APEX session cookie (next to the session ID in the URL) with which APEX can identify the user with. As long as the APEX session is valid, the user is returned the requested page.

11. The access token is not checked for ie. expiration. Although obtained, the refresh token has no usage for the authentication process as implemented in APEX 20.2.

3 IMPLEMENTATION STEPS

3.1 PREREQS

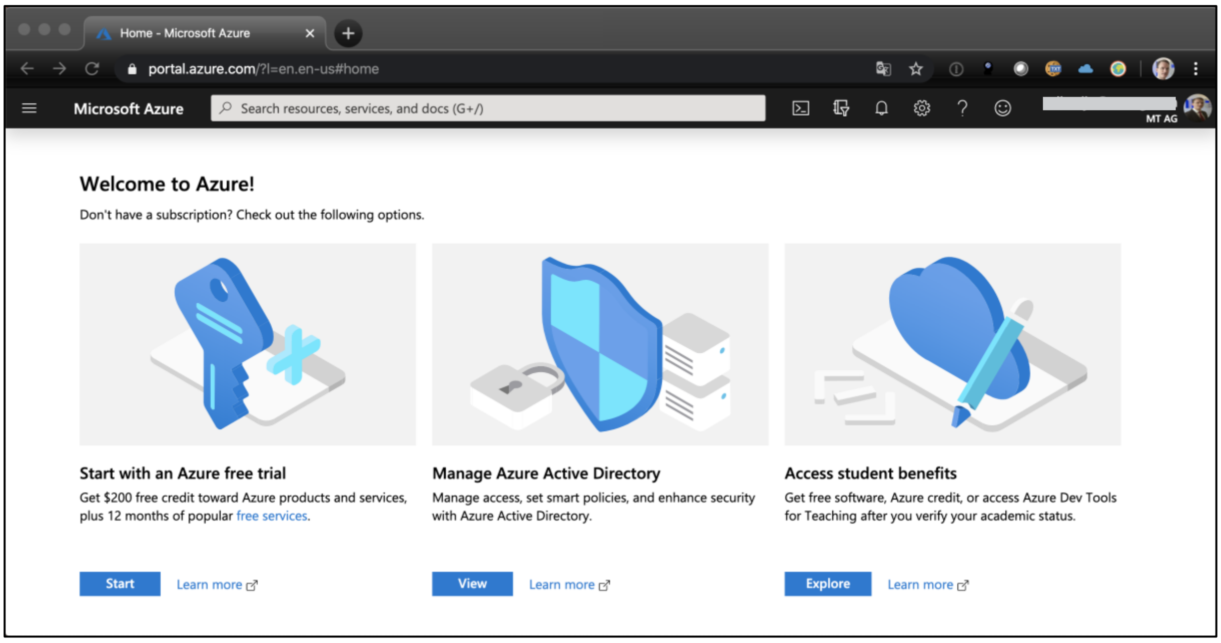
Before starting the implementation, make sure that the username in Azure Active Directory (Azure AD) is the same as used in your local users table (used for authorization purposes).

To start the configuration, you will need Oracle APEX 18.1+ running in Oracle Database 12.2+ and Azure AD.

All screenshots of Oracle APEX in this document were taken using version 20.2.

3.2 REGISTER THE APEX ENVIRONMENT AT AZURE AD

Logon to https://portal.azure.com.


Home - Microsoft Azure
portal.azure.com/?l=en.en-us#home
Microsoft Azure
Search resources, services, and docs (G+/)
MT AG

Welcome to Azure!

Don't have a subscription? Check out the following options.

Start with an Azure free trial

Get $200 free credit toward Azure products and services, plus 12 months of popular free services.

Start   Learn more

Manage Azure Active Directory

Manage access, set smart policies, and enhance security with Azure Active Directory.

View   Learn more

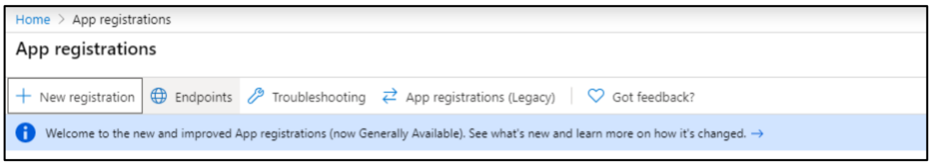
Access student benefits

Get free software, Azure credit, or access Azure Dev Tools for Teaching after you verify your academic status.

Explore   Learn more

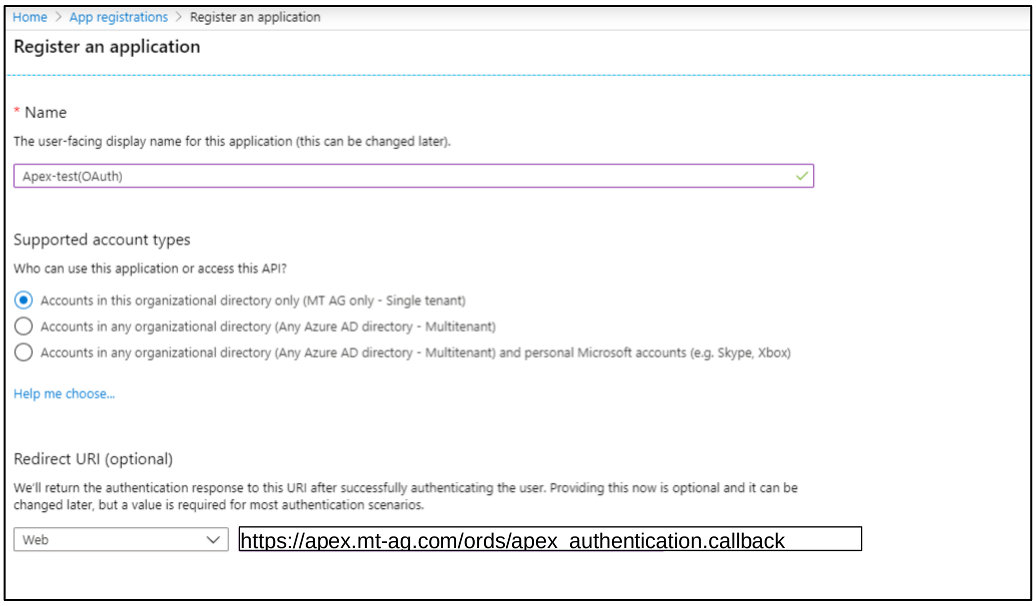
Click on "View" at "Manage Azure Active Directory".

Within "App registrations", click on "New registration".



Welcome to the new and improved App registrations (now Generally Available). See what's new and learn more on how it's changed. →

Fill out the form as shown below and click on "register". Make sure that the Redirect URI points to your APEX instance. It should look something like this:
https://apex.mycompany.com/ords/apex_authentication.callback



Register an application

* Name

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

Apex-test(OAuth)

Supported account types

Who can use this application or access this API?

- [✓ Accounts in this organizational directory only (MT AG 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)

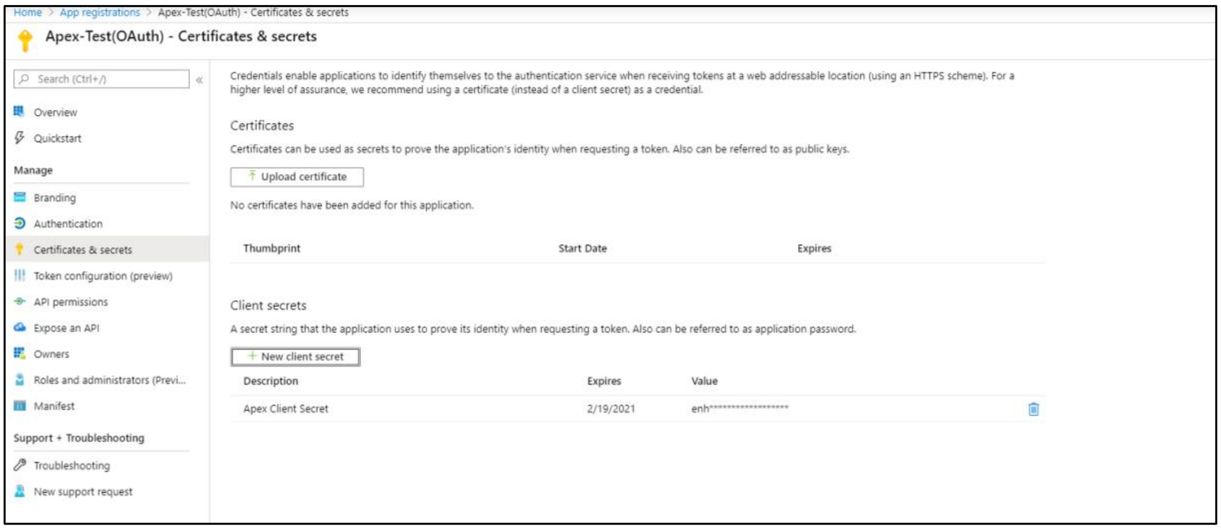
Help me choose...

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   https://apex.mt-ag.com/ords/apex_authentication.callback

Within “Certificates & secrets”, click on “New client secret”. Note the client secret provided.


Home > App registrations > Apex-Test(OAuth) - Certificates & secrets

Apex-Test(OAuth) - Certificates & secrets

Credentials enable 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

Certificates can be used as secrets to prove the application's identity when requesting a token. Also can be referred to as public keys.

Upload certificate

No certificates have been added for this application.

Thumbprint	Start Date	Expires

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
Apex Client Secret	2/19/2021	enh******************

3.3 ENABLE OUTBOUND (SSL) COMMUNICATION FROM APEX

Logon with `sys as sysdba` to the APEX database and execute the following script with the correct APEX version permitting the database to contact Azure AD when requesting an access token:

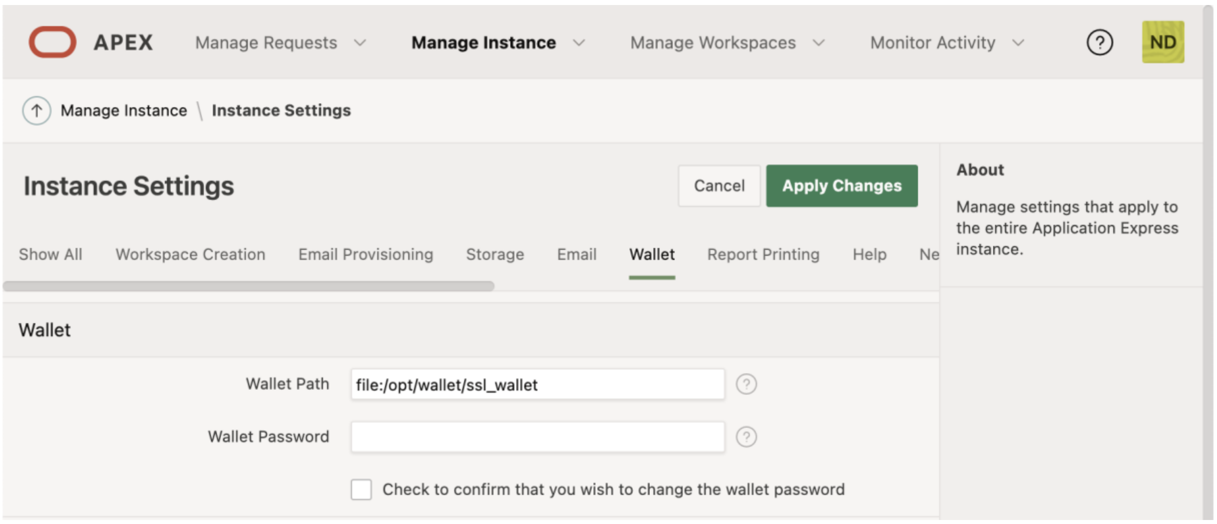
```
BEGIN
  DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE
  ( host => 'login.microsoftonline.com'
  , ace  => xs$ace_type
            ( privilege_list => xs$name_list('connect')
            , principal_name => 'APEX_200200'
            , principal_type => xs_acl.ptype_db
            )
  );
  DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE
  ( host => 'graph.microsoft.com'
  , ace  => xs$ace_type
            ( privilege_list => xs$name_list('connect')
            , principal_name => 'APEX_200200'
            , principal_type => xs_acl.ptype_db
            )
  );

  select privilege
  ,      grant_type
  ,      principal
  ,      host
  from   dba_host_aces
  where  principal = 'APEX_200200'
  ;

END;
/
```

Both `login.microsoftonline.com` as well as `graph.microsoft.com` are using SSL. When using the Oracle Database as SSL client, we need to create a SSL wallet and make APEX aware of it.
Get the top root certificate and copy it on the server. For the script below to run properly, we assume that the file is called `Baltimore_Cybertrust_Root.crt` and is stored in the directory `/opt/wallet`.

On the database server where APEX was installed, run:

```
cd /opt/wallet
orapki wallet create -wallet ssl_wallet -pwd mysecretpassword -auto_login
orapki wallet add -wallet ssl_wallet -cert Baltimore_Cybertrust_Root.crt -pwd mysecretpassword
-trusted_cert
orapki wallet display -wallet ssl_wallet -pwd mysecretpassword
chown -R oracle:oinstall ssl_wallet
```

Make the created wallet known to APEX. Logon to the internal workspace of APEX. Within "instance settings" go to the section "Wallet" and enter the file path of the wallet:



Manage settings that apply to the entire Application Express instance.

Wallet

Wallet Path: file:/opt/wallet/ssl_wallet

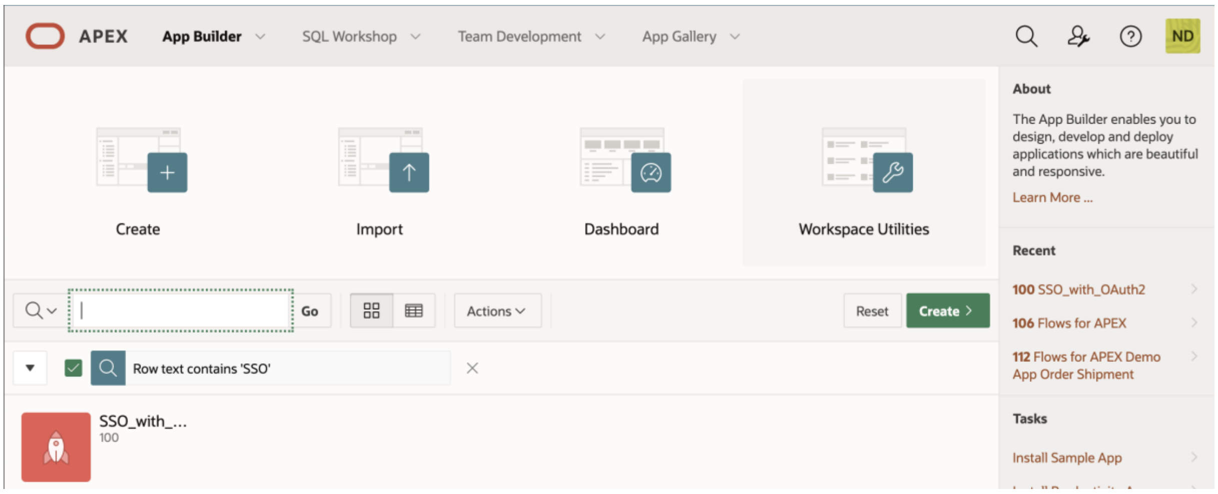
Wallet Password:

☐ Check to confirm that you wish to change the wallet password

As we have created a wallet with the option "auto login", APEX doesn't have to be aware of the password as it will use the *.sso file.

3.4 CREATE WEB CREDENTIALS IN THE APEX WORKSPACE

Logon to the APEX workspace containing the app that should be configured for OAuth2 authentication against Azure AD.



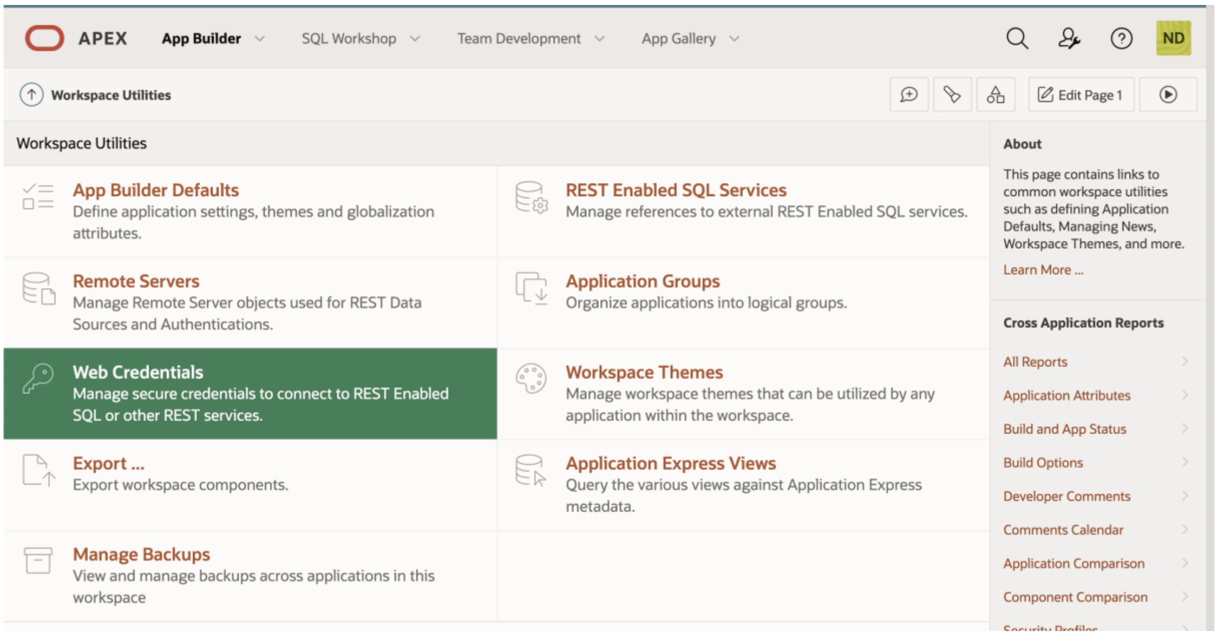
The App Builder enables you to design, develop and deploy applications which are beautiful and responsive.

Row text contains 'SSO'

SSO_with_...
100

100 SSO_with_OAuth2
106 Flows for APEX
112 Flows for APEX Demo App Order Shipment

Click on "Workspace Utilities".



Workspace Utilities

App Builder Defaults Define application settings, themes and globalization attributes.

REST Enabled SQL Services Manage references to external REST Enabled SQL services.

Remote Servers Manage Remote Server objects used for REST Data Sources and Authentications.

Application Groups Organize applications into logical groups.

Web Credentials Manage secure credentials to connect to REST Enabled SQL or other REST services.

Workspace Themes Manage workspace themes that can be utilized by any application within the workspace.

Export ... Export workspace components.

Application Express Views Query the various views against Application Express metadata.

Manage Backups View and manage backups across applications in this workspace

About

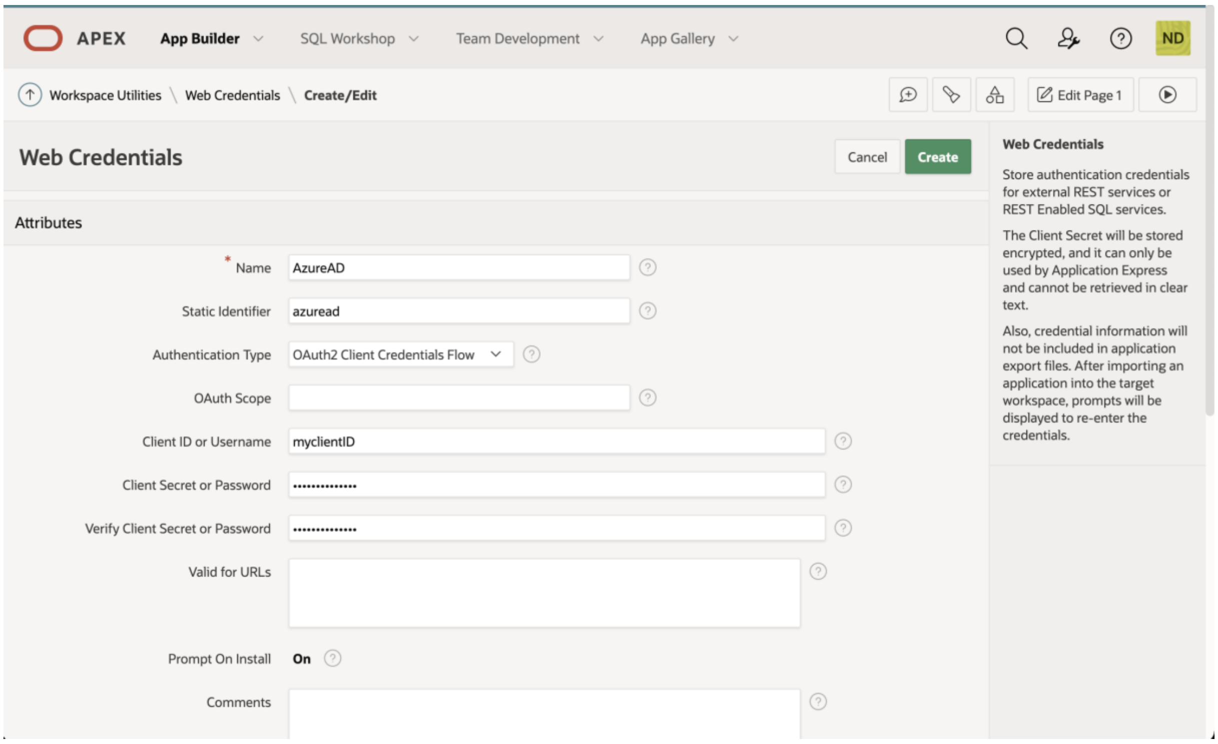
This page contains links to common workspace utilities such as defining Application Defaults, Managing News, Workspace Themes, and more.

Learn More ...

Cross Application Reports

- All Reports
- Application Attributes
- Build and App Status
- Build Options
- Developer Comments
- Comments Calendar
- Application Comparison
- Component Comparison

Click on “Web Credentials”.



Web Credentials

Attributes

* Name	AzureAD
Static Identifier	azuread
Authentication Type	OAuth2 Client Credentials Flow
OAuth Scope	
Client ID or Username	myclientID
Client Secret or Password	••••••••••••••
Verify Client Secret or Password	••••••••••••••
Valid for URLs	
Prompt On Install	On
Comments	

Web Credentials

Store authentication credentials for external REST services or REST Enabled SQL services.

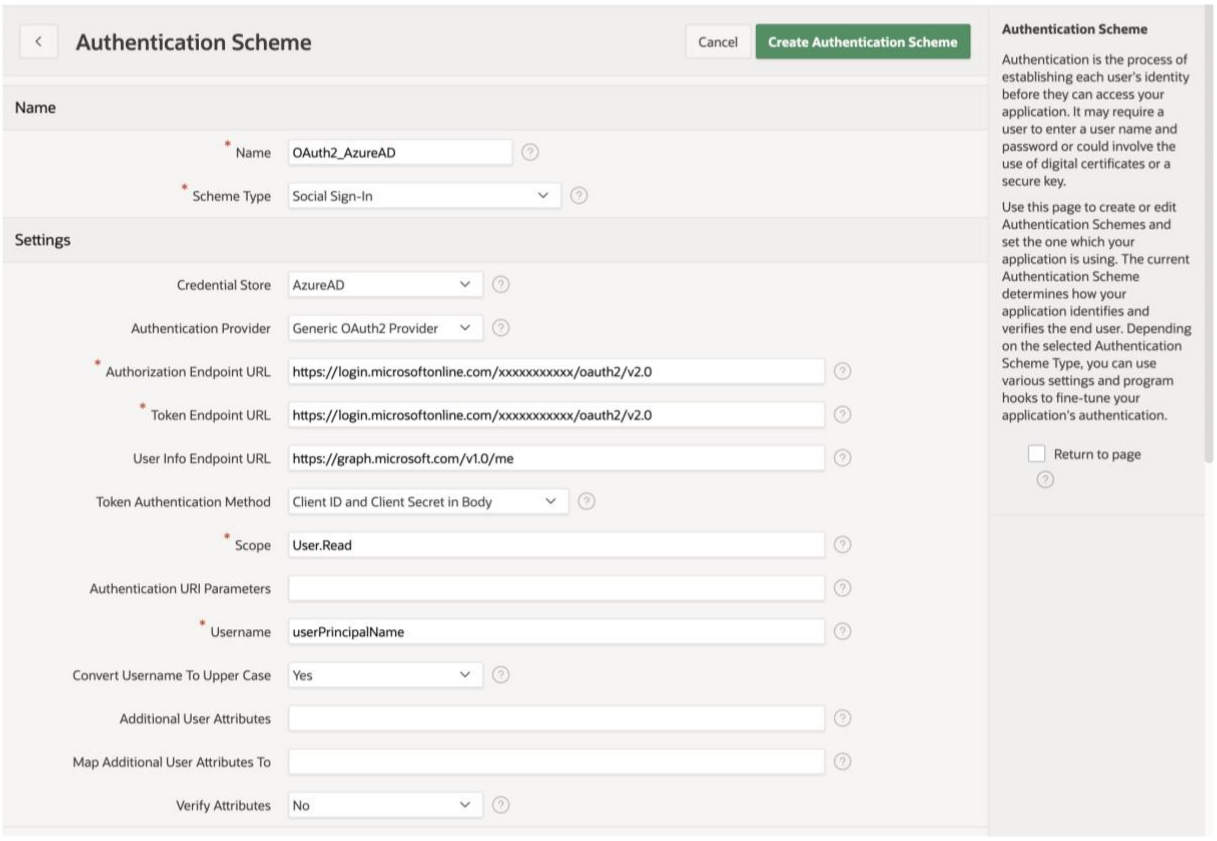
The Client Secret will be stored encrypted, and it can only be used by Application Express and cannot be retrieved in clear text.

Also, credential information will not be included in application export files. After importing an application into the target workspace, prompts will be displayed to re-enter the credentials.

Enter the client ID and client secret given by Azure AD.

3.5 CREATE AN AUTHENTICATION SCHEME FOR THE APEX APP

Select the APEX application in the workspace, click on “Shared Components” and create a new “Authentication Schema” based on Scheme Type “Social Sign-In” and Authentication Provider “Generic OAuth2 Provider”. The details entered should look like below:



Authentication Scheme

Name

* Name	OAuth2_AzureAD
* Scheme Type	Social Sign-In

Settings

Credential Store	AzureAD
Authentication Provider	Generic OAuth2 Provider
* Authorization Endpoint URL	https://login.microsoftonline.com/xxxxxxxxxxx/oauth2/v2.0
* Token Endpoint URL	https://login.microsoftonline.com/xxxxxxxxxxx/oauth2/v2.0
User Info Endpoint URL	https://graph.microsoft.com/v1.0/me
Token Authentication Method	Client ID and Client Secret in Body
* Scope	User.Read
Authentication URI Parameters	
* Username	userPrincipalName
Convert Username To Upper Case	Yes
Additional User Attributes	
Map Additional User Attributes To	
Verify Attributes	No

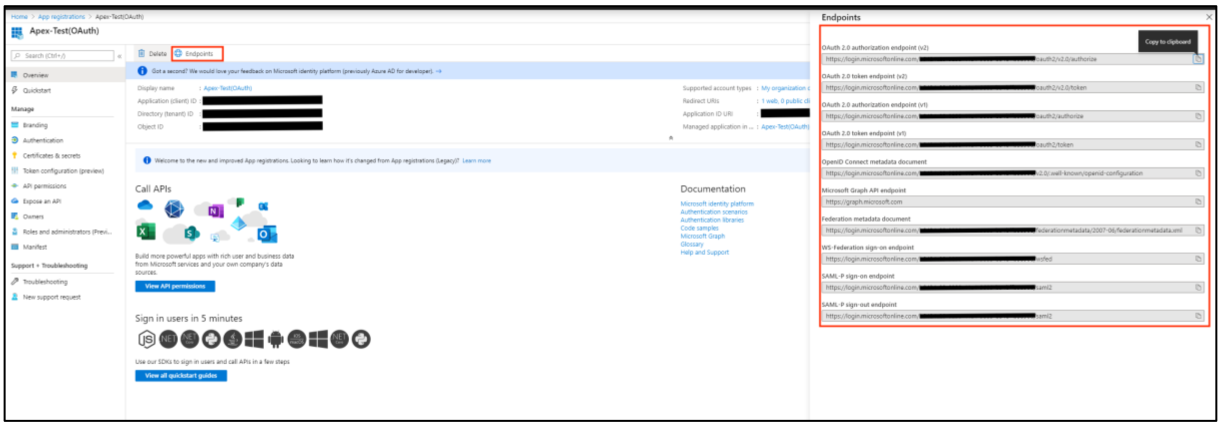
Authentication Scheme

Authentication is the process of establishing each user's identity before they can access your application. It may require a user to enter a user name and password or could involve the use of digital certificates or a secure key.

Use this page to create or edit Authentication Schemes and set the one which your application is using. The current Authentication Scheme determines how your application identifies and verifies the end user. Depending on the selected Authentication Scheme Type, you can use various settings and program hooks to fine-tune your application's authentication.

☐ Return to page

You can get the Endpoint URLs from Azure AD by clicking on “Endpoints” after choosing the app registration:



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

Display name	Apex-Test(OAuth)
Application (client) ID	[redacted]
Directory (tenant) ID	[redacted]
Object ID	[redacted]
Supported account types	My organization
Redirect URIs	1 web, 0 public cli
Application ID URI	[redacted]
Managed application in ...	Apex-Test(OAuth)

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

Build more powerful apps with rich user and business data from Microsoft services and your own company's data sources.

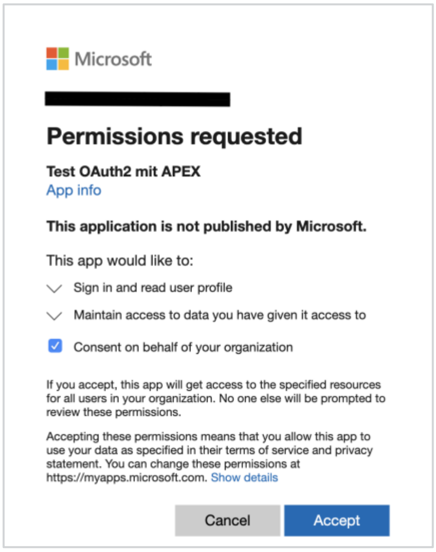
Use our SDKs to sign in users and call APIs in a few steps

Endpoints

OAuth 2.0 authorization endpoint (v2)	https://login.microsoftonline.com/[redacted]/oauth2/v2.0/authorize
OAuth 2.0 token endpoint (v2)	https://login.microsoftonline.com/[redacted]/oauth2/v2.0/token
OAuth 2.0 authorization endpoint (v1)	https://login.microsoftonline.com/[redacted]/oauth2/authorize
OAuth 2.0 token endpoint (v1)	https://login.microsoftonline.com/[redacted]/oauth2/token
OpenID Connect metadata document	https://login.microsoftonline.com/[redacted]/v2.0/.well-known/openid-configuration
Microsoft Graph API endpoint	https://graph.microsoft.com
Federation metadata document	https://login.microsoftonline.com/[redacted]/federationmetadata/2007-06/federationmetadata.xml
WS-Federation sign-on endpoint	https://login.microsoftonline.com/[redacted]/wsfed
SAML-P sign-on endpoint	https://login.microsoftonline.com/[redacted]/saml2
SAML-P sign-out endpoint	https://login.microsoftonline.com/[redacted]/saml2

After the creation of the 2^{nd} authentication scheme, it automatically becomes the default scheme.

Run the APEX application. After authentication, you will need to explicitly accept that APEX, as a 3th party app, is allowed to read out your user profile to get the username (in this case `userPrincipalName`):



Microsoft

[redacted]

Permissions requested

Test OAuth2 mit APEX
App info

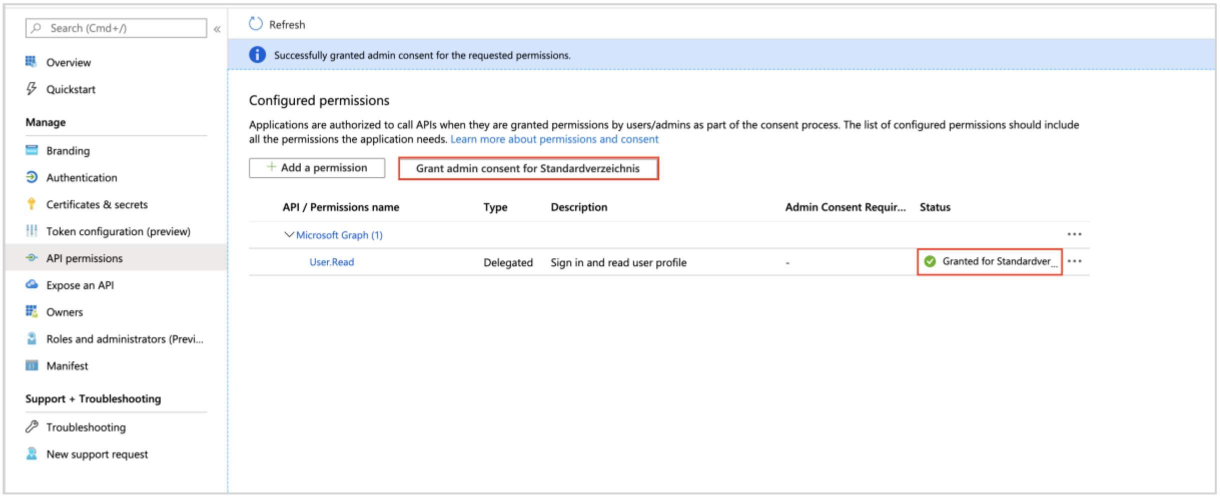
This application is not published by Microsoft.

This app would like to:
- Sign in and read user profile
- Maintain access to data you have given it access to
- ☑ Consent on behalf of your organization

If you accept, this app will get access to the specified resources for all users in your organization. No one else will be prompted to review these permissions.

Accepting these permissions means that you allow this app to use your data as specified in their terms of service and privacy statement. You can change these permissions at https://myapps.microsoft.com. Show details

Note: as an admin of Azure AD, you can grant consent for all users to prevent this message from being shown to the user. Just click on the button “Grant admin consent…” in the “API permissions” section of the app registration:



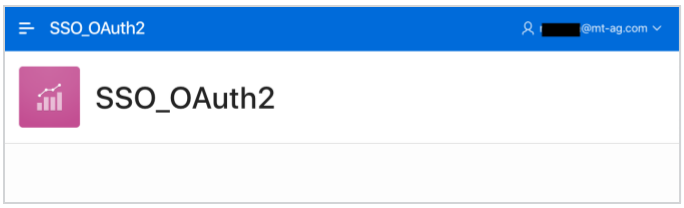
Successfully granted admin consent for the requested permissions.

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

API / Permissions name	Type	Description	Admin Consent Requir...	Status
Microsoft Graph (1)				
User.Read	Delegated	Sign in and read user profile	-	✓ Granted for Standardver...

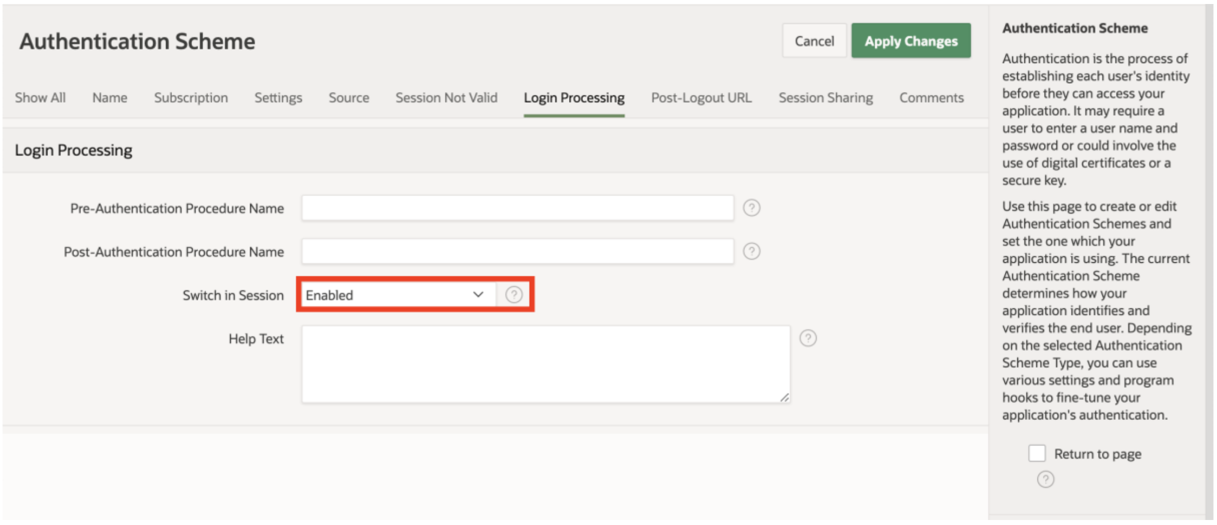
After the consent, the APEX app is displayed with the user logged on. In APEX the current user can be retrieved using the `:APP_USER` variable.



3.6 SWITCHING AUTHENTICATION SCHEME IN RUNTIME

Since not all users are registered in Azure AD, there should be a separate URL in place with which a user can logon locally. In this case, username and password are stored in a custom table. The following steps describe how to switch between authentication schemes in runtime. It will not cover how to setup custom authentication, as this is already running in production.

For BOTH authentication schemes, enable the option “Switch in Session” in the section “Login Processing”:



Login Processing

Pre-Authentication Procedure Name

Post-Authentication Procedure Name

Switch in Session: Enabled

Help Text

Authentication Scheme

Authentication is the process of establishing each user's identity before they can access your application. It may require a user to enter a user name and password or could involve the use of digital certificates or a secure key.

Use this page to create or edit Authentication Schemes and set the one which your application is using. The current Authentication Scheme determines how your application identifies and verifies the end user. Depending on the selected Authentication Scheme Type, you can use various settings and program hooks to fine-tune your application's authentication.

When calling the APEX app with only the application ID, the current authentication scheme will be used. For example: https://apex.mt-ag.com/ords/f?p=114329.

A specific authentication scheme can be requested through the URL. In our example this is:

APEX authentication:
`https://apex.mt-ag.com/ords/f?p=114329:1:<SESSION_ID>:APEX_AUTHENTICATION=APEX_AUTH`

AZURE AD authentication:
`https://apex.mt-ag.com/ords/f?p=114329:1:<SESSION_ID>:APEX_AUTHENTICATION=AZUREAD_AUTH`

Note: due to a “bug” in APEX 19.2, it is not possible to call the URL with the parameter APEX_AUTHENTICATION directly. You first have to call a public APEX page (to get an APEX session) before switching the authentication. For APEX 20.2, this bug was fixed.

3.7 LOGGING OUT

To log out of Azure AD, call the URL:
`https://login.microsoftonline.com/<Tenant_ID>` or `common/oauth2/logout`

After that, the user should close the browser to explicitly terminate the APEX session.

Note: logging out of Azure AD will also logout the user for all apps connected to that Azure AD tenant. It is not possible to logout of a specific app residing on the same tenant.

3.8 DEBUGGING

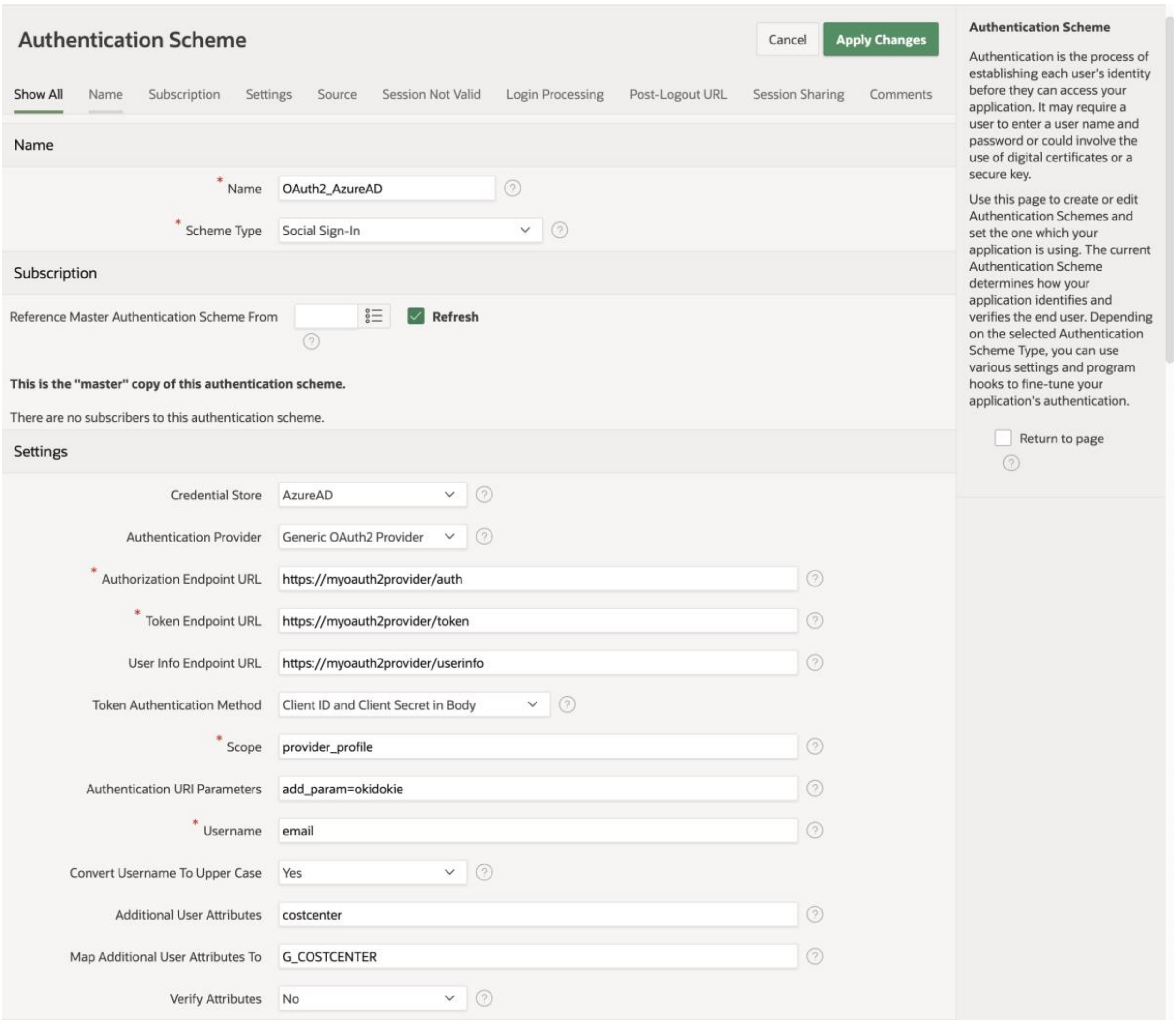
Something doesn’t work as planned and you need to get more info about what is going on under the hood? Here is a great blog post about debugging all requests:

https://chrisonoracle.wordpress.com/2020/04/03/debugging-apex-authentication-issues/

4 GLOSSARY

4.1 ATTRIBUTES GENERIC OAUTH2 AUTHENTICATION SCHEME

The values shown below are just for example purposes. Some of them are only available in APEX 20.2.



Authentication Scheme

Name

* Name	OAuth2_AzureAD
* Scheme Type	Social Sign-In

Subscription

Reference Master Authentication Scheme From ✓ Refresh

This is the "master" copy of this authentication scheme.

There are no subscribers to this authentication scheme.

Settings

Credential Store	AzureAD
Authentication Provider	Generic OAuth2 Provider
* Authorization Endpoint URL	https://myoauth2provider/auth
* Token Endpoint URL	https://myoauth2provider/token
User Info Endpoint URL	https://myoauth2provider/userinfo
Token Authentication Method	Client ID and Client Secret in Body
* Scope	provider_profile
Authentication URI Parameters	add_param=okidokie
* Username	email
Convert Username To Upper Case	Yes
Additional User Attributes	costcenter
Map Additional User Attributes To	G_COSTCENTER
Verify Attributes	No

Authentication Scheme

Authentication is the process of establishing each user's identity before they can access your application. It may require a user to enter a user name and password or could involve the use of digital certificates or a secure key.

Use this page to create or edit Authentication Schemes and set the one which your application is using. The current Authentication Scheme determines how your application identifies and verifies the end user. Depending on the selected Authentication Scheme Type, you can use various settings and program hooks to fine-tune your application's authentication.

☐ Return to page

<u>Credential Store</u>: the web credentials to use for app-authentication at the authentication provider (APEX in this case).

<u>Authorization Endpoint URL</u>: the URL of the authentication provider where the user can authenticate himself and optionally can give his consent to let the application access additional information.

<u>Token Endpoint URL</u>: the URL contacted by APEX to get the access/ID/refresh tokens.

<u>User Info Endpoint URL</u>: the URL contacted by APEX to get additional user information returned as a JSON Web Token (JWT).

Token Authentication Method: the way the APEX server makes a request to the authentication provider.

Scope: all requests against the authentication provider must have

Authentication URI Parameters: if the authentication provider requires additional parameters for the authorization endpoint, these can be specified here.

Username: the name of the username variable to be found in the returned tokens (access/ID) as used by APEX to replace “nobody” in the current session.

Convert Username To Upper Case: determines if `:APP_USER` is returned in lower/upper case.

Additional User Attributes: additional information to be retrieved based on the given scope(s).

Map Additional User Attributes To: in which APEX item(s) should the user attribute(s) be stored?

In the given example, `G_COSTCENTER` is an application item in the app.

Verify Attributes: if an authentication provider lets you change ie. the e-mail address and APEX uses this for authentication, then you want to make sure that this e-mail address first got verified by the user (normally by clicking on a link in an e-mail sent to the new address). If the new address wasn’t verified yet, the authentication provider can send a special attribute for this to APEX. When setting “Verify Attributes” to “yes”, APEX will only accept the attribute if the provider tells APEX that the attribute “<attribute>_verified” is set to “true”.
If this attribute is also used as “Username”, then APEX rejects to create a session.

Disclaimer:
Just to make sure: MT AG is not responsible for any damage, outages or loss of profit resulting from the usage of this document. Use it at your own risk. Have fun!