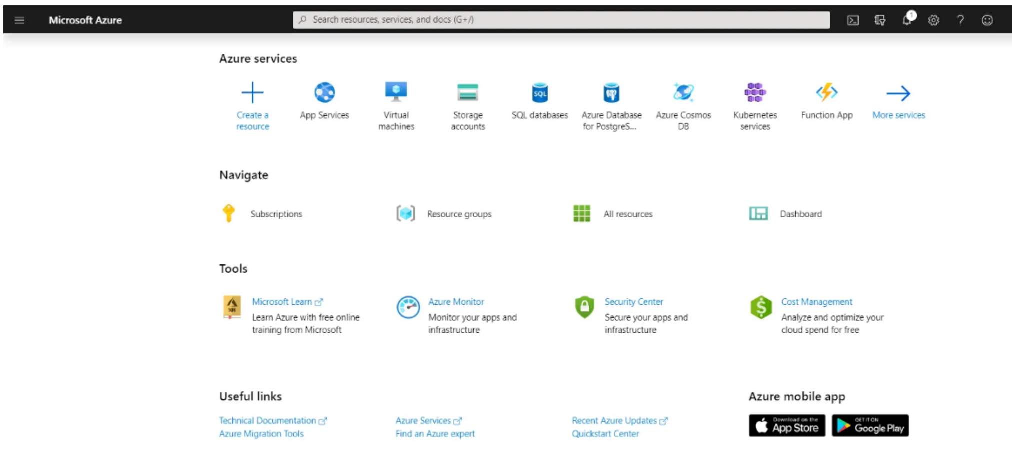
STEPS TO CONFIGURE AZURE AD SINGLE SIGN-ON (SSO) LOGIN INTO WORDPRESS

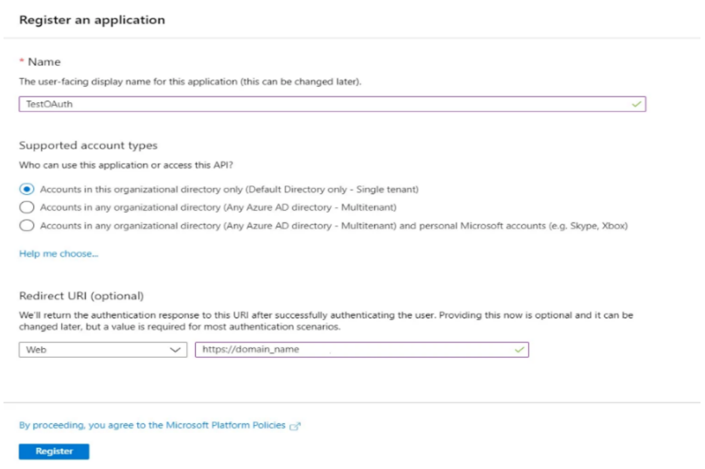
Step 1: Setup Azure Active Directory as OAuth Provider

Sign in to Azure portal.

Click on **App Services** and go to **Manage Azure Active Directory.**



In the left-hand navigation pane, click the **App registrations** service, and click **New registration**.



Register an application

* Name

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

TestOAuth

Supported account types

Who can use this application or access this API?

- [✓ Accounts in this organizational directory only (Default Directory 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://domain_name

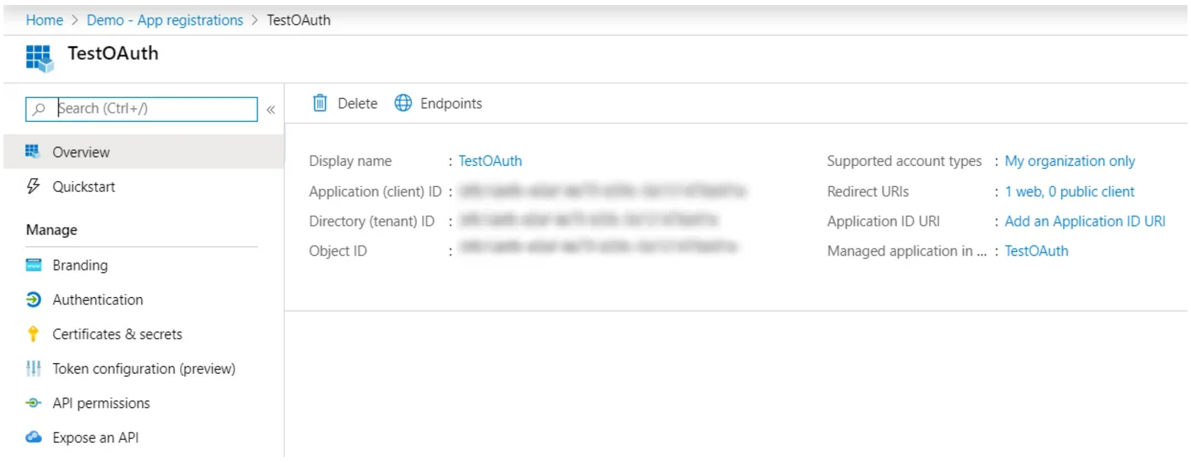
By proceeding, you agree to the Microsoft Platform Policies

Register

When the Create page appears, enter your application's registration information:

Name :	Name of your application.
Application type :	Select *"Web app / API"* for client applications and resource/API applications that are installed on a secure server. This setting is used for OAuth confidential web clients and public user-agent-based clients. The same application can also expose both a client and resource/API.
Sign-on URL :	For "Web app / API" applications, provide the base URL of your app. eg, https://<domain-name> might be the URL for a web app running on your local machine. Users would use this URL to sign in to a web client application.

When finished, click **Register**. Azure AD assigns a unique Application ID to your application. Copy **Application ID** and the **Directory ID** , this will be your **Client ID** and **Tenent ID.**

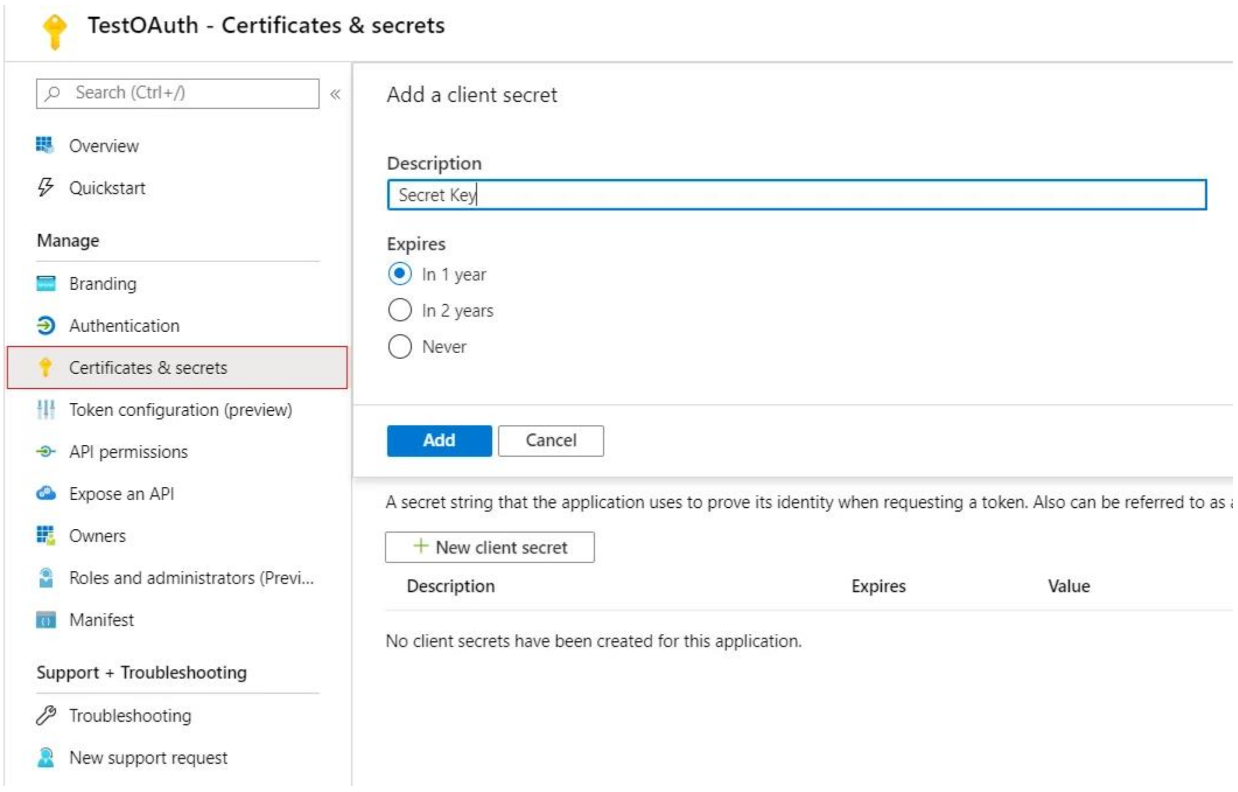

Home > Demo - App registrations > TestOAuth

TestOAuth

Display name : TestOAuth
Application (client) ID : [redacted]
Directory (tenant) ID : [redacted]
Object ID : [redacted]

Supported account types : My organization only
Redirect URIs : 1 web, 0 public client
Application ID URI : Add an Application ID URI
Managed application in ... : TestOAuth

Go to **Certificates and Secrets** from the left navigaton pane and click on **New Client Secret**. Enter description and expiration time and click on **ADD** option.



Add a client secret

Description: Secret Key

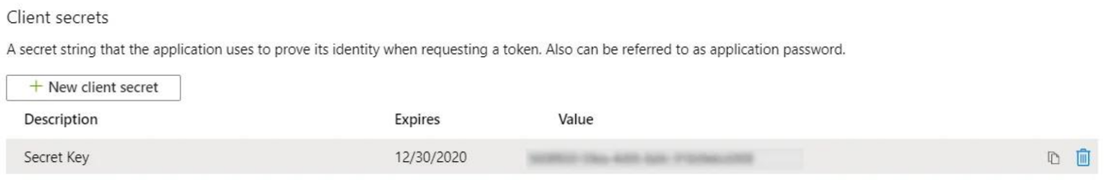
Expires: [✓ In 1 year] In 2 years Never

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as

Description	Expires	Value

No client secrets have been created for this application.

Copy value. This will be your **Secret key**.



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.

Description	Expires	Value
Secret Key	12/30/2020	[illegible]

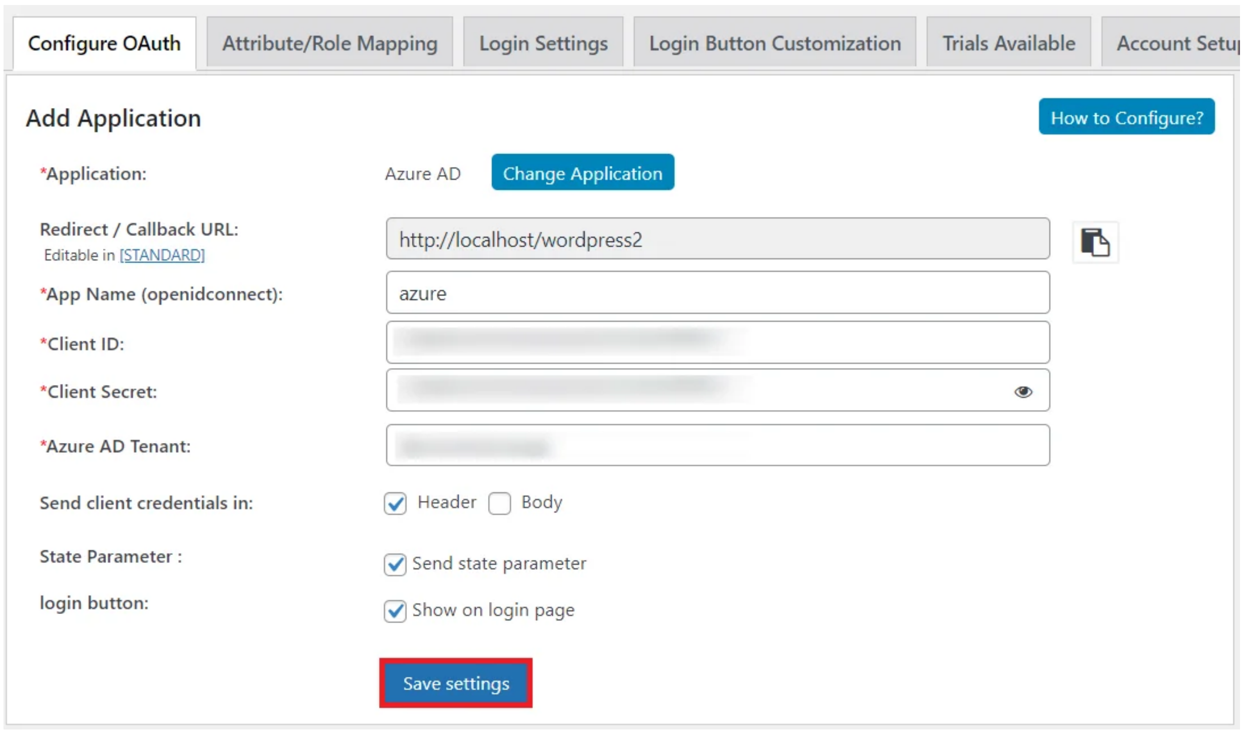
Azure AD Endpoints and Scope:

Scope:	openid
Authorize Endpoint:	https://login.microsoftonline.com/<tenant-id>/oauth2/authorize
Access Token Endpoint:	https://login.microsoftonline.com/<tenant-id>/oauth2/token
Get User Info Endpoint:	https://login.windows.net/common/openid/userinfo
Custom redirect URL after logout:[optional]	https://login.microsoftonline.com/common/oauth2/logout?post_logout_rediret-uri=<your URL>

You have successfully configured **Azure AD as OAuth Provider** for achieving user authentication with Azure AD Single Sign-On (SSO) login into your WordPress Site.

Step 2: Setup WordPress as OAuth Client

- Go to **Configure OAuth** tab and configure **App Name, Client ID, Client Secret, Tenant name and Policy from provided Endpoints**
- **OpenID** is already filled.
- Click on Save Settings to save the configuration.



Add Application

*Application:	Azure AD
Redirect / Callback URL: Editable in [STANDARD]	http://localhost/wordpress2
*App Name (openidconnect):	azure
*Client ID:	[redacted]
*Client Secret:	[redacted]
*Azure AD Tenant:	[redacted]
Send client credentials in:	[✓ Header] Body
State Parameter :	[✓ Send state parameter]
login button:	[✓ Show on login page]

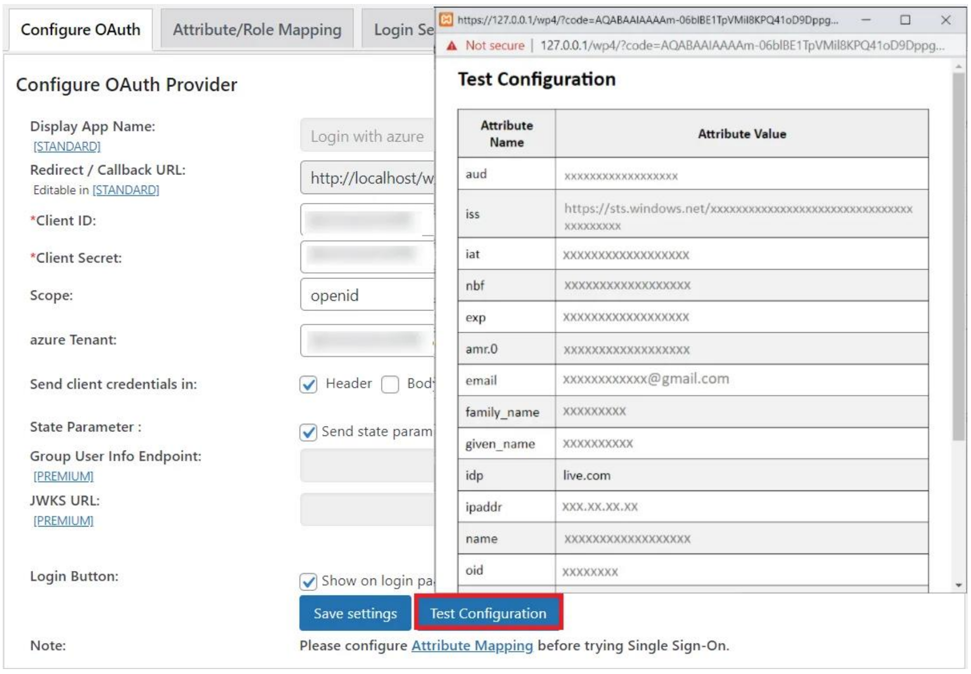
You have successfully configured **WordPress as OAuth Client** for achieving user authentication with Azure AD Single Sign-On (SSO) login into your WordPress Site.

Step 3: User Attribute Mapping

User Attribute Mapping is mandatory for enabling users to successfully login into WordPress. We will be setting up user profile attributes for WordPress using below settings.

Finding user attributes

Go to **Configure OAuth** tab. Scroll down and click on **Test Configuration**.



Configure OAuth
Attribute/Role Mapping
Login Se
https://127.0.0.1/wp4/?code=AQABAAIAAAAm-06bIBE1TpVMil8KPQ41oD9Dppg...
Not secure | 127.0.0.1/wp4/?code=AQABAAIAAAAm-06bIBE1TpVMil8KPQ41oD9Dppg...

Configure OAuth Provider

Display App Name: [STANDARD]	Login with azure
Redirect / Callback URL: Editable in [STANDARD]	http://localhost/w
*Client ID:	[redacted]
*Client Secret:	[redacted]
Scope:	openid
azure Tenant:	[redacted]
Send client credentials in:	✓ Header ☐ Bod
State Parameter :	✓ Send state param
Group User Info Endpoint: [PREMIUM]	
JWKS URL: [PREMIUM]	
Login Button:	✓ Show on login pa

Save settings | Test Configuration

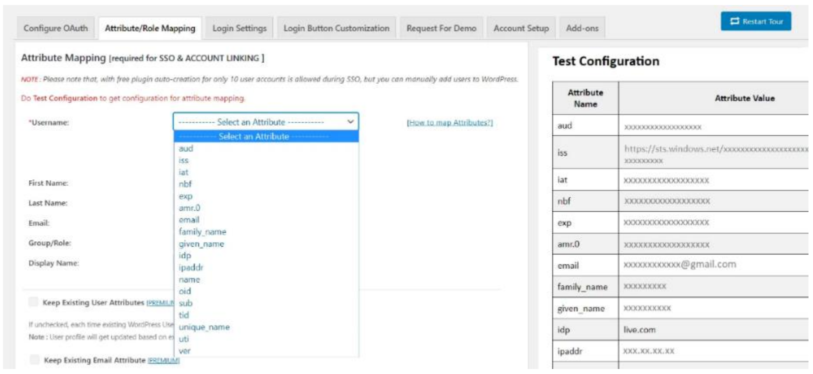
Note: Please configure Attribute Mapping before trying Single Sign-On.

Test Configuration

Attribute Name	Attribute Value
aud	xxxxxxxxxxxxxxxxxx
iss	https://sts.windows.net/xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxxx
iat	xxxxxxxxxxxxxxxxxx
nbf	xxxxxxxxxxxxxxxxxx
exp	xxxxxxxxxxxxxxxxxx
amr.0	xxxxxxxxxxxxxxxxxx
email	xxxxxxxxxxxx@gmail.com
family_name	xxxxxxxxx
given_name	xxxxxxxxxx
idp	live.com
ipaddr	xxx.xx.xx.xx
name	xxxxxxxxxxxxxxxxxx
oid	xxxxxxxx

You will see all the values returned by your OAuth Provider to WordPress in a table. If you don't see value for First Name, Last Name, Email or Username, make the required settings in your OAuth Provider to return this information.

Once you see all the values in Test Configuration, go to **Attribute / Role Mapping** tab, you will get the list of attributes in a Username dropdown.



Attribute Mapping [required for SSO & ACCOUNT LINKING]

NOTE : Please note that, with free plugin auto-creation for only 10 user accounts is allowed during SSO, but you can manually add users to WordPress.

Do Test Configuration to get configuration for attribute mapping.

*Username: ----------- Select an Attribute ----------- [How to map Attributes?]

----------- Select an Attribute -----------
aud
iss
iat
nbf
exp
amr.0
email
family_name
given_name
idp
ipaddr
name
oid
sub
tid
unique_name
uti
ver

First Name:
Last Name:
Email:
Group/Role:
Display Name:

☐ Keep Existing User Attributes [PREMIUM]

If unchecked, each time existing WordPress Use

Note : User profile will get updated based on e

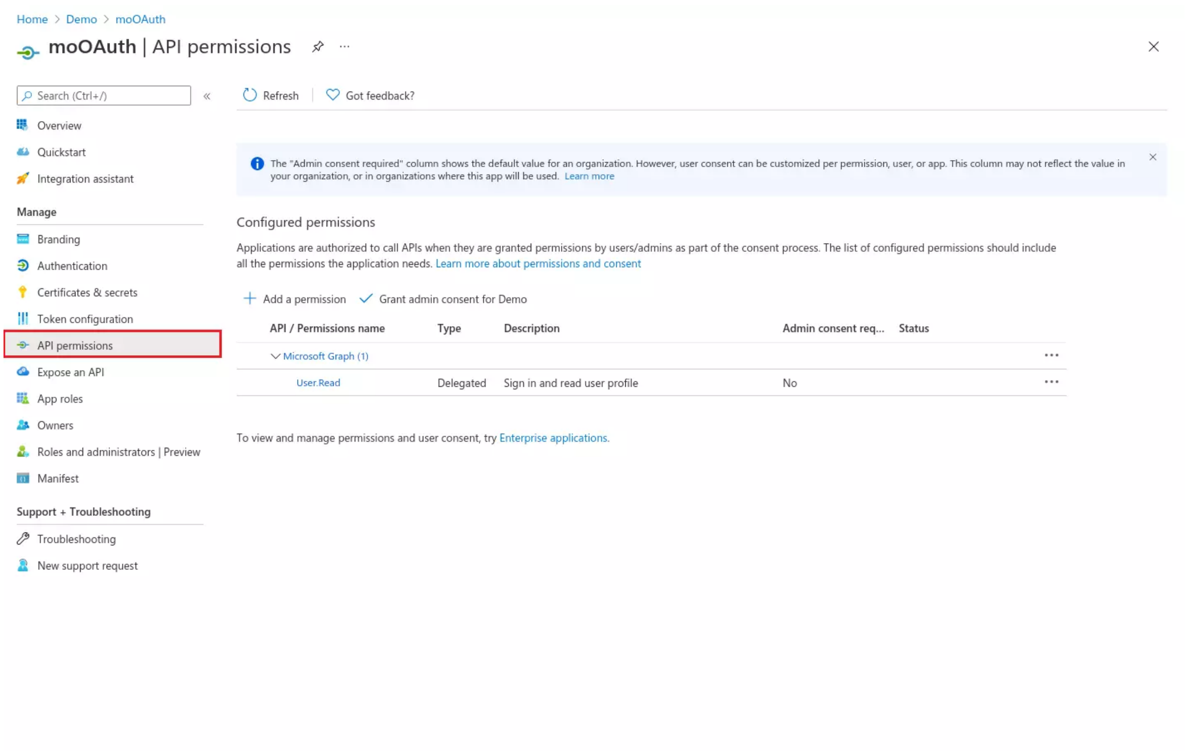
☐ Keep Existing Email Attribute [PREMIUM]

Test Configuration

Attribute Name	Attribute Value
aud	xxxxxxxxxxxxxxxxxx
iss	https://sts.windows.net/xxxxxxxxxxxxxxxxxxx xxxxxxxxx
iat	xxxxxxxxxxxxxxxxxx
nbf	xxxxxxxxxxxxxxxxxx
exp	xxxxxxxxxxxxxxxxxx
amr.0	xxxxxxxxxxxxxxxxxx
email	xxxxxxxxxxxx@gmail.com
family_name	xxxxxxxxx
given_name	xxxxxxxxxx
idp	live.com
ipaddr	xxx.xx.xx.xx

Step 4: Group Mapping [Premium Feature]

Go to Application → Select the application where you want to configure the group mapping. Now, Go to the **API Permissions** tab.



Home > Demo > moOAuth

moOAuth | API permissions

Search (Ctrl+/) · Refresh · Got feedback?

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

The "Admin consent required" column shows the default value for an organization. However, user consent can be customized per permission, user, or app. This column may not reflect the value in your organization, or in organizations where this app will be used. Learn more

Configured permissions

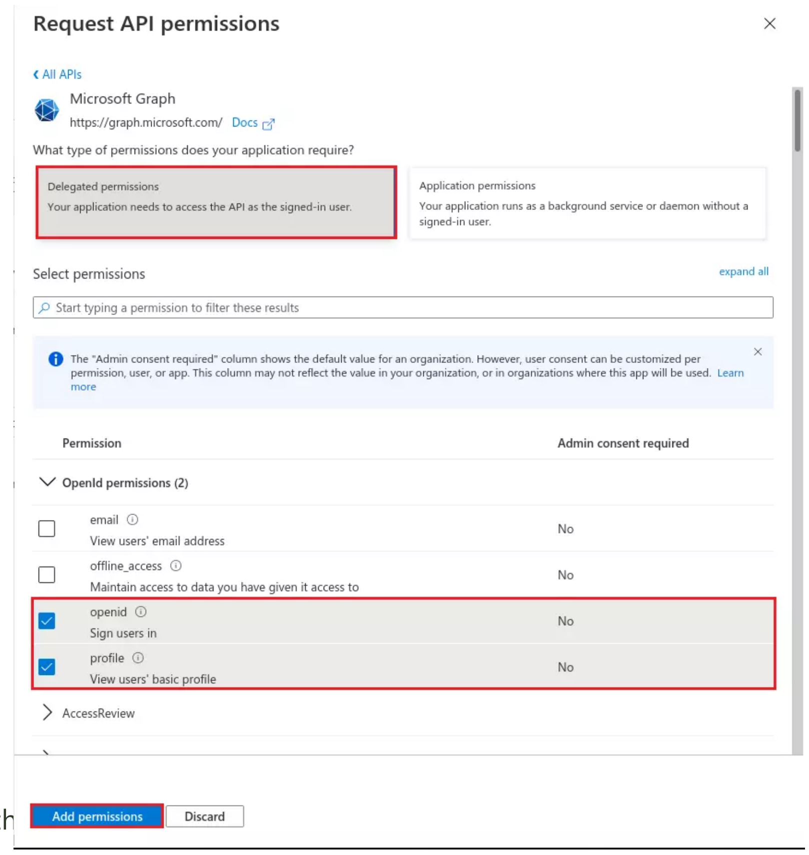
Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. Learn more about permissions and consent

+ Add a permission ✓ Grant admin consent for Demo

API / Permissions name	Type	Description	Admin consent req...	Status
Microsoft Graph (1)				
User.Read	Delegated	Sign in and read user profile	No	

To view and manage permissions and user consent, try Enterprise applications.

Click on the **Add permission** button, and then **Microsoft Graph API -> Delegated Permissions** and select **openid, Profile** scope and click on the **Add Permissions** button.



Request API permissions

< All APIs

Microsoft Graph
https://graph.microsoft.com/ Docs

What type of permissions does your application require?

Delegated permissions
Your application needs to access the API as the signed-in user.

Application permissions
Your application runs as a background service or daemon without a signed-in user.

Select permissions · expand all

Start typing a permission to filter these results

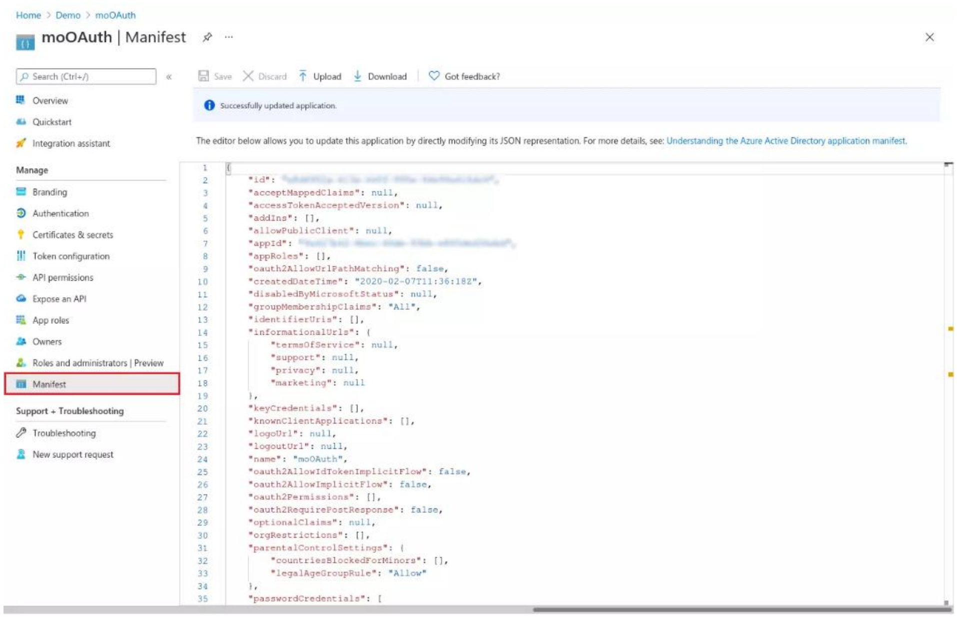
The "Admin consent required" column shows the default value for an organization. However, user consent can be customized per permission, user, or app. This column may not reflect the value in your organization, or in organizations where this app will be used. Learn more

	Permission	Admin consent required
	OpenId permissions (2)	
☐	email — View users' email address	No
☐	offline_access — Maintain access to data you have given it access to	No
☑	openid — Sign users in	No
☑	profile — View users' basic profile	No
	AccessReview	

Add permissions · Discard

Click on th

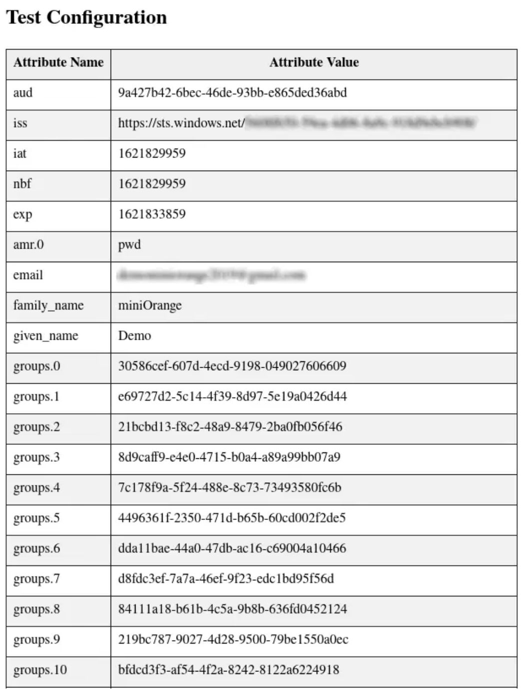
Go to Manifest tab and find **groupMembershipClaims** and changes it’s value to **"All"** and click on the **save** button.



Successfully updated application.

The editor below allows you to update this application by directly modifying its JSON representation. For more details, see: Understanding the Azure Active Directory application manifest.

```
{
    "id": [redacted]
    "acceptMappedClaims": null,
    "accessTokenAcceptedVersion": null,
    "addIns": [],
    "allowPublicClient": null,
    "appId": [redacted]
    "appRoles": [],
    "oauth2AllowUrlPathMatching": false,
    "createdDateTime": "2020-02-07T11:36:18Z",
    "disabledByMicrosoftStatus": null,
    "groupMembershipClaims": "All",
    "identifierUris": [],
    "informationalUrls": {
        "termsOfService": null,
        "support": null,
        "privacy": null,
        "marketing": null
    },
    "keyCredentials": [],
    "knownClientApplications": [],
    "logoUrl": null,
    "logoutUrl": null,
    "name": "moOAuth",
    "oauth2AllowIdTokenImplicitFlow": false,
    "oauth2AllowImplicitFlow": false,
    "oauth2Permissions": [],
    "oauth2RequirePostResponse": false,
    "optionalClaims": null,
    "orgRestrictions": [],
    "parentalControlSettings": {
        "countriesBlockedForMinors": [],
        "legalAgeGroupRule": "Allow"
    },
    "passwordCredentials": [
```



Test Configuration

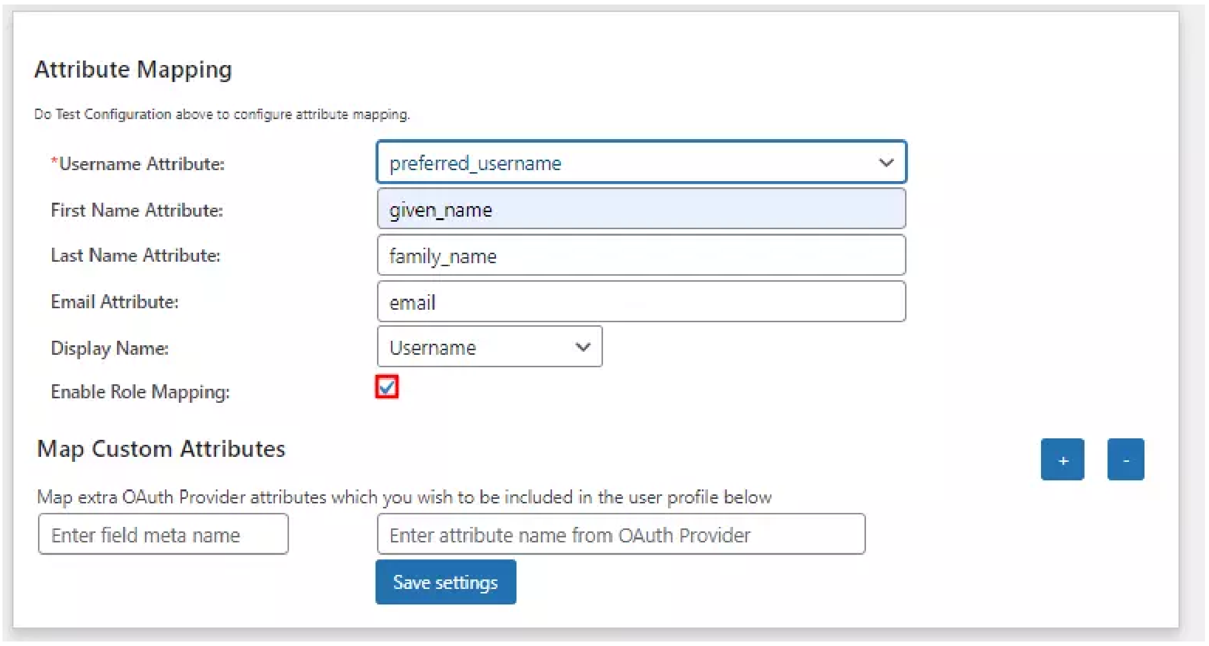
Attribute Name	Attribute Value
aud	9a427b42-6bec-46de-93bb-e865ded36abd
iss	https://sts.windows.net/[redacted]
iat	1621829959
nbf	1621829959
exp	1621833859
amr.0	pwd
email	[redacted]
family_name	miniOrange
given_name	Demo
groups.0	30586cef-607d-4ecd-9198-049027606609
groups.1	e69727d2-5c14-4f39-8d97-5e19a0426d44
groups.2	21bcbd13-f8c2-48a9-8479-2ba0fb056f46
groups.3	8d9caff9-e4e0-4715-b0a4-a89a99bb07a9
groups.4	7c178f9a-5f24-488e-8c73-73493580fc6b
groups.5	4496361f-2350-471d-b65b-60cd002f2de5
groups.6	dda11bae-44a0-47db-ac16-c69004a10466
groups.7	d8fdc3ef-7a7a-46ef-9f23-edc1bd95f56d
groups.8	84111a18-b61b-4c5a-9b8b-636fd0452124
groups.9	219bc787-9027-4d28-9500-79be1550a0ec
groups.10	bfdcd3f3-af54-4f2a-8242-8122a6224918

Now you would be able to get the group's value in the Test configuration window. You can follow the role mapping section to map the groups to WordPress users while Azure AD SSO <link>.

Step 5: Role Mapping [Premium Feature].

Click on **"Test Configuration"** and you will get the list of Attribute Names and Attribute Values that are sent by your OAuth provider.

From the Test Configuration window, map the Attribute Names in the Attribute Mapping section of the plugin. Refer to the screenshot for more details.



Attribute Mapping

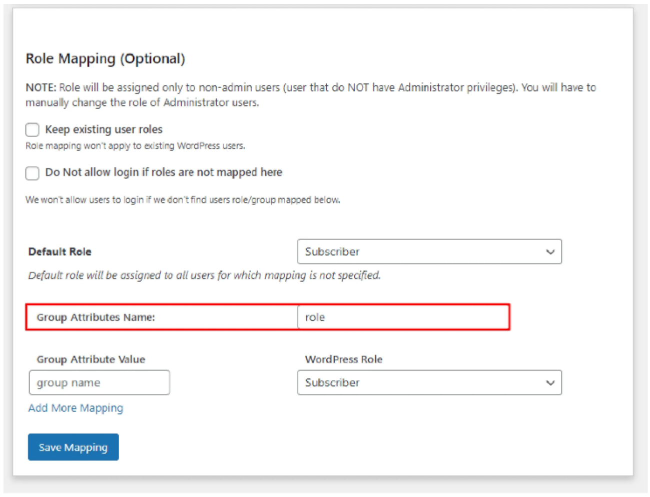
Do Test Configuration above to configure attribute mapping.

*Username Attribute:	preferred_username
First Name Attribute:	given_name
Last Name Attribute:	family_name
Email Attribute:	email
Display Name:	Username
Enable Role Mapping:	✓

Map Custom Attributes

Map extra OAuth Provider attributes which you wish to be included in the user profile below

Enable Role Mapping: To enable Role Mapping, you need to map Group Name Attribute. Select the attribute name from the list of attributes which returns the roles from your provider application.
Eg: Role



Role Mapping (Optional)

NOTE: Role will be assigned only to non-admin users (user that do NOT have Administrator privileges). You will have to manually change the role of Administrator users.

☐ Keep existing user roles
Role mapping won't apply to existing WordPress users.

☐ Do Not allow login if roles are not mapped here

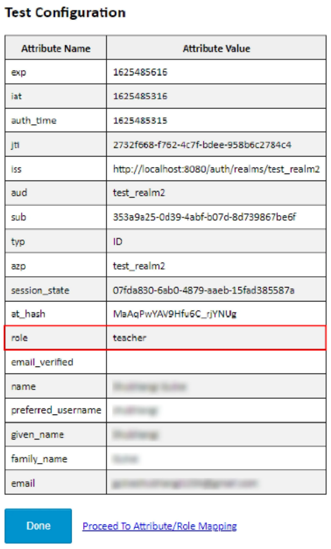
We won't allow users to login if we don't find users role/group mapped below.

Default Role	Subscriber

Default role will be assigned to all users for which mapping is not specified.

Group Attributes Name:	role

Group Attribute Value	WordPress Role
group name	Subscriber

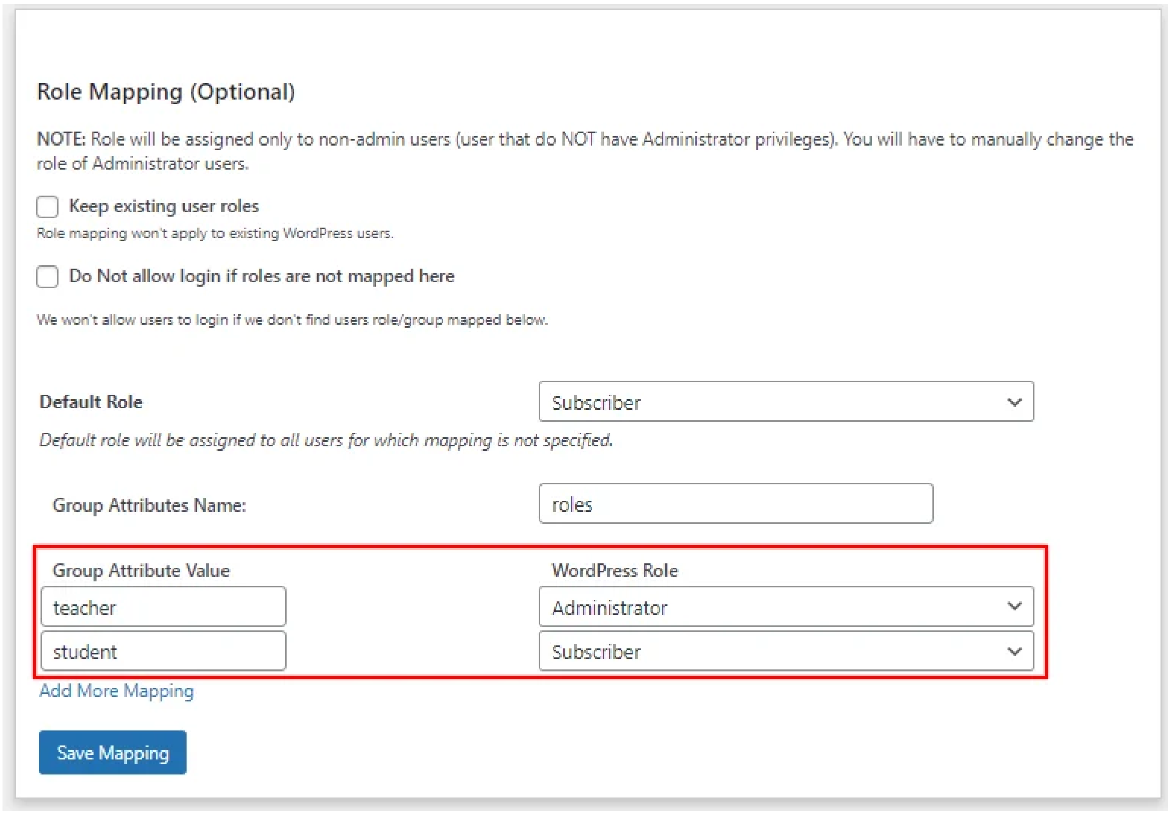


Test Configuration

Attribute Name	Attribute Value
exp	1625485616
iat	1625485316
auth_time	1625485315
jti	2732f668-f762-4c7f-bdee-958b6c2784c4
iss	http://localhost:8080/auth/realms/test_realm2
aud	test_realm2
sub	353a9a25-0d39-4abf-b07d-8d739867be6f
typ	ID
azp	test_realm2
session_state	07fda830-6ab0-4879-aaeb-15fad385587a
at_hash	MaAqPwYAV9Hfu6C_rjYNUg
role	teacher
email_verified	
name	[redacted]
preferred_username	[redacted]
given_name	[redacted]
family_name	[redacted]
email	[redacted]

Assign WordPress role to the Provider role: Based on your provider application, you can allocate the WordPress role to your provider roles. It can be a student, teacher, administrator or any other depending on your application. Add the provider roles under Group Attribute Value and assign the required WordPress role in front of it under WordPress Role.

For example, in the below image. Teacher has been assigned the role of Administrator & Student is assigned the role of Subscriber.


Role Mapping (Optional)

NOTE: Role will be assigned only to non-admin users (user that do NOT have Administrator privileges). You will have to manually change the role of Administrator users.

☐ Keep existing user roles

Role mapping won't apply to existing WordPress users.

☐ Do Not allow login if roles are not mapped here

We won't allow users to login if we don't find users role/group mapped below.

Default Role: Subscriber

Default role will be assigned to all users for which mapping is not specified.

Group Attributes Name: roles

Group Attribute Value	WordPress Role
teacher	Administrator
student	Subscriber

Once you save the mapping, the provider role will be assigned the WordPress administrator role after SSO.
Example: As per the given example, Users with role ‘teacher’ will be added as Administrator in WordPress and ‘student’ will be added as Subscriber.

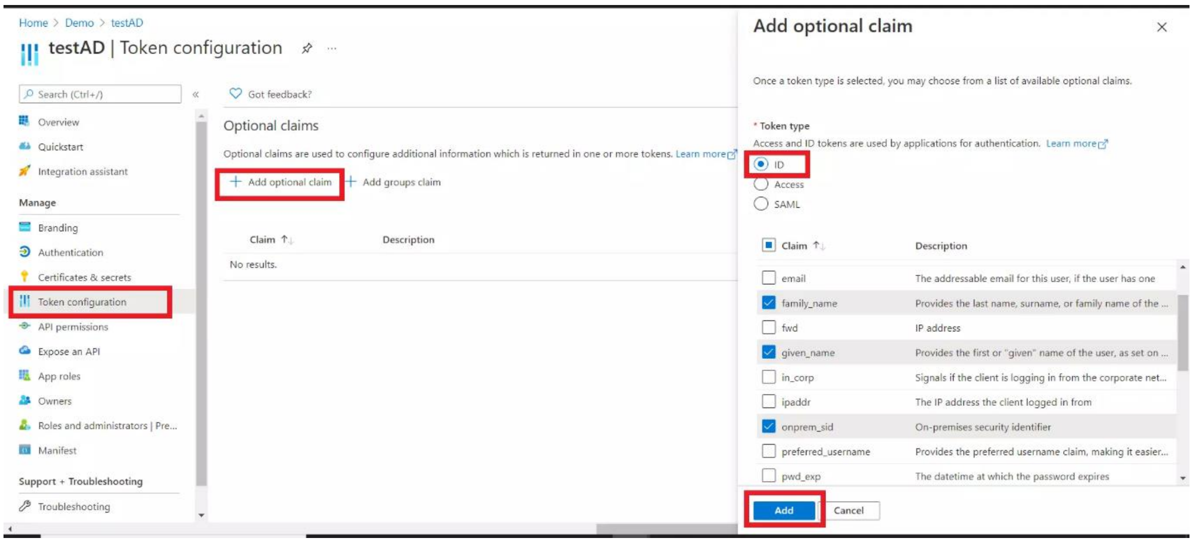
Step 6: Custom Attribute Mapping [Premium]

Go to your application in Azure Active Directory and select **Token configuration** from the left menu.

Click on **Add optional claim** and then select **ID** from the right section.

Now choose all the attributes you want to fetch while SSO (e.g family_name, given_name, etc) and click on **Add** button.

You might see a popup to **Turn on the Microsoft Graph profile permission (required for claims to appear in token)**, enable it, and click on Add button.



Optional claims

Optional claims are used to configure additional information which is returned in one or more tokens. Learn more

Claim	Description
No results.	

Add optional claim

Once a token type is selected, you may choose from a list of available optional claims.

* Token type

Access and ID tokens are used by applications for authentication. Learn more

[✓ ID] Access SAML

	Claim	Description
☐	email	The addressable email for this user, if the user has one
✓	family_name	Provides the last name, surname, or family name of the ...
☐	fwd	IP address
✓	given_name	Provides the first or "given" name of the user, as set on ...
☐	in_corp	Signals if the client is logging in from the corporate net...
☐	ipaddr	The IP address the client logged in from
✓	onprem_sid	On-premises security identifier
☐	preferred_username	Provides the preferred username claim, making it easier...
☐	pwd_exp	The datetime at which the password expires

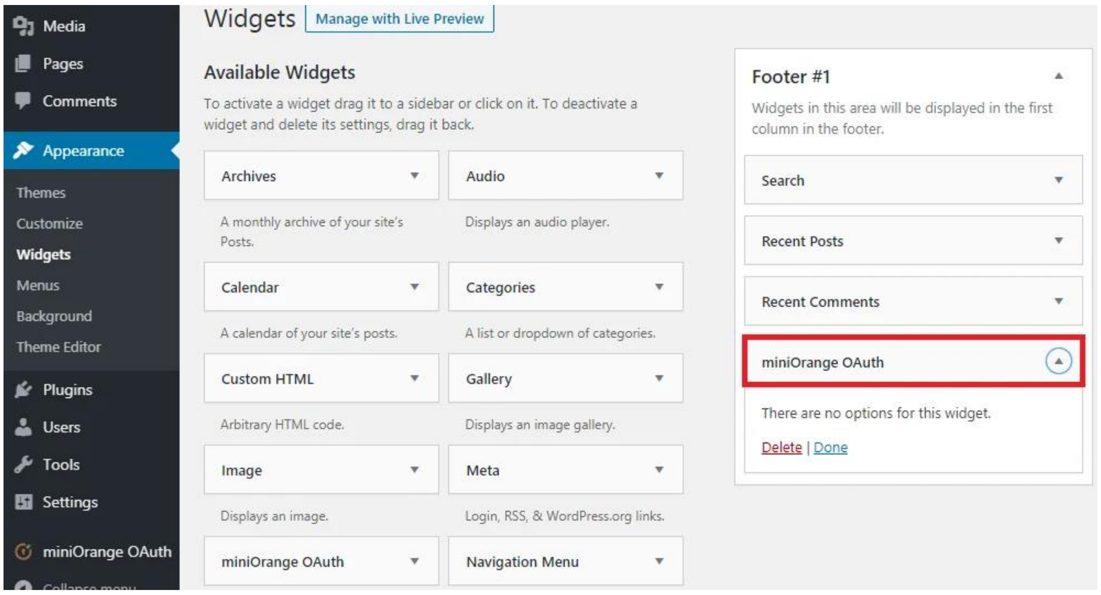
Step 7: Login Settings / Sign In Settings

The settings in Single Sign-On (SSO) Settings tab define the user experience for Single Sign-On (SSO). To add a Azure AD login widget on your WordPress page, you need to follow the below steps.

- **Sign in settings for wordpress 5.7 and before :**

  Go to **WordPress Left Panel** > **Appearances** > **Widgets**.

  Select **miniOrange OAuth**. Drag and drop to your favourite location and save.



Available Widgets

To activate a widget drag it to a sidebar or click on it. To deactivate a widget and delete its settings, drag it back.

A monthly archive of your site's Posts.

Displays an audio player.

A calendar of your site's posts.

A list or dropdown of categories.

Arbitrary HTML code.

Displays an image gallery.

Displays an image.

Login, RSS, & WordPress.org links.

Widgets in this area will be displayed in the first column in the footer.

There are no options for this widget.

- **Sign in settings for WordPress 5.8 :**

  Go to **WordPress Left Panel** > **Appearances** > **Widgets**.

  Select **miniOrange OAuth**. Drag and drop to your favourite location and save.

  Open your WordPress page and you can see the Azure AD SSO login button there. You can test the Azure AD Single Sign-On (SSO) now.