

# Microsoft Entra ID SSO for Mealie

**Prerequisites:** Ability to create an app registration with delegated rights and access to the Mealie Docker volume or startup method. Mealie should be installed correctly.

Mealie makes recipe management and planning extremely easy. The Mealie software offers an OpenID interface, which means that Microsoft Entra ID can easily be used as an identity provider and permission manager.

## Create App Registration

First, an app registration including client secret must be created in Microsoft Entra ID. All settings can be left at the default values. Important settings are the Redirect URIs under the Authentication tab. Set these URIs to your external or internal domain on which Mealie is available. These URIs will be used for Microsoft Entra ID to know where to redirect the user in case of successful logins.

- **Type:** Single-page application
- **Redirect URIs:** <https://mealie.yourdomain.com/login>

- Overview
- Quickstart
- Integration assistant
- Diagnose and solve problems

Manage

Branding & properties

**Authentication**

Certificates & secrets

Token configuration

API permissions

Expose an API

App roles

Owners

Roles and administrators

Manifest

Support + Troubleshooting

New support request

### Platform configurations

Depending on the platform or device this application is targeting, additional configuration may be required such as redirect URIs, specific authentication settings, or fields specific to the platform.

+ Add a platform

**Single-page application** Quickstart Docs 🗑️

**Redirect URIs**

The URIs we will accept as destinations when returning authentication responses (tokens) after successfully authenticating or signing out users. The redirect URI you send in the request to the login server should match one listed here. Also referred to as reply URLs. [Learn more about Redirect URIs and their restrictions](#)

✓ 🗑️

[Add URI](#)

**Grant types**

✓ Your Redirect URI is eligible for the Authorization Code Flow with PKCE.

### Front-channel logout URL

This is where we send a request to have the application clear the user's session data. This is required for single sign-out to work correctly.

✓

### Implicit grant and hybrid flows

Request a token directly from the authorization endpoint. If the application has a single-page architecture (SPA) and doesn't use the authorization code flow, or if it invokes a web API via JavaScript, select both access tokens and ID tokens. For ASP.NET Core web apps and other web apps that use hybrid authentication, select only ID tokens. [Learn more about tokens.](#)

Select the tokens you would like to be issued by the authorization endpoint:

- Access tokens (used for implicit flows)
- ID tokens (used for implicit and hybrid flows)

Add the corresponding permissions for OpenID Connect as delegated permissions and grant admin consent for your tenant.

- **Permissions:** Delegated OpenId permissions (email, offline\_access, openid, profile)

- Overview
- Quickstart
- Integration assistant
- Diagnose and solve problems

Manage

Branding & properties

Authentication

Certificates & secrets

Token configuration

**API permissions**

Expose an API

App roles

Owners

### 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 LNC Freelancing

API / Permissions name	Type	Description	Admin consent req...	Status
Microsoft Graph (4) <span>⋮</span>				
email	Delegated	View users' email address	No	✓ Granted for LNC Freelancing <span>⋮</span>
openid	Delegated	Sign users in	No	✓ Granted for LNC Freelancing <span>⋮</span>
profile	Delegated	View users' basic profile	No	✓ Granted for LNC Freelancing <span>⋮</span>
User.Read	Delegated	Sign in and read user profile	No	✓ Granted for LNC Freelancing <span>⋮</span>

To view and manage consented permissions for individual apps, as well as your tenant's consent settings, try [Enterprise applications](#).

Create a client secret for the application and save the tenant ID, application ID and client secret in your password manager. You can find instructions for this information here: [Get app details and grant permissions to app registration](#)

# Setup Microsoft Entra ID login provider

With Mealie, the Microsoft Entra ID configurations can be set up using environment variables. Installation using Docker is mandatory for these instructions. The following environment variables enable the configuration of the OpenID integration. You can find more information on all of Mealie's environment variables here: [Backend Configuration - Mealie](#)

The relevant environment variables for OpenID are as follows:

- **OIDC\_AUTH\_ENABLED:** Enables authentication via OpenID Connect if set to true.
- **OIDC\_SIGNUP\_ENABLED:** Enables new users to be created when signing in for the first time with OIDC if set to true.
- **OIDC\_CONFIGURATION\_URL:** This is the URL to the configuration of the provider. Using Microsoft Entra ID this probably is: `https://login.microsoftonline.com/v2.0/.well-known/openid-configuration`
- **OIDC\_CLIENT\_ID:** This is the client id of your App Registration in Microsoft Entra ID.
- **OIDC\_CLIENT\_SECRET:** Here you have to provide the Client Secret from the App Registration.
- **OIDC\_PROVIDER\_NAME:** This is the display name for the button on the sign in screen.

**Side note:** As this type of configuration involves environment variables, these contents can also be transferred via the volume as an `.env` file, specified with a simple startup command or specified in another declarative context (Kubernetes manifest, Terraform, etc.).

## Docker compose example

This Docker Compose file shows a possible configuration for Mealie that authenticates using Microsoft Entra ID. In addition, a mail server is also specified for outgoing SMTP mail traffic.

Customize this content with your specifications and save the content in a normal `docker-compose.yml` file. As this is Docker Compose, the application can be started easily with the following command (in detach mode `-> -d`):

```
docker compose up -d
```

```
version: "3.7"
services:
  mealie:
    image: ghcr.io/mealie-recipes/mealie:latest
```

container\_name: <yourcontainername>

ports:

- "8600:<yourpublicport>"

volumes:

- <yourpersistentpath>:/app/data/

environment:

- ALLOW\_SIGNUP=true

- OIDC\_AUTH\_ENABLED=true

- OIDC\_SIGNUP\_ENABLED=true

- OIDC\_CONFIGURATION\_URL=https://login.microsoftonline.com/<yourtenantid>/v2.0/.well-known/openid-

configuration

- OIDC\_CLIENT\_ID=<yourclientid>

- OIDC\_CLIENT\_SECRET=<yourclientsecret>

- OIDC\_PROVIDER\_NAME=Microsoft Entra ID

- PUID=1000

- PGID=1000

- TZ=<yourtimezone>

- MAX\_WORKERS=1

- WEB\_CONCURRENCY=1

- BASE\_URL=https://<yourmealiedomain>

- SMTP\_HOST=<yoursmtppmailhost>

- SMTP\_PORT=587

- SMTP\_FROM\_EMAIL=<yoursmtppmail>

- SMTP\_USER=<yoursmtppmailuser>

- SMTP\_PASSWORD=<yoursmtppmailpassword>

- SMTP\_FROM\_NAME=<yourmailname>

restart: unless-stopped

---

Revision #3

Created 22 July 2024 08:32:48 by Luca Noah Caprez

Updated 22 July 2024 12:26:53 by Luca Noah Caprez