

# Microsoft Entra ID SSO for Bookstack

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

Bookstack offers an OpenID interface, which means that Microsoft Entra ID can easily be used as an identity provider for managing access and permission within Bookstack. The functionalities are more limited than other integrations. However, simple functionalities such as automatic user creation and email verification can be customized.

This guide is a compilation of the main documentation of Bookstack: [Third Party Authentication](#) · [BookStack \(bookstackapp.com\)](#)

## 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 Bookstack is available. These URIs will be used for Microsoft Entra ID to know where to redirect the user in case of successful logins.

- **Authentication Type:** Web
- **Redirect URIs:** <https://bookstack.yourdomain.com/login/service/azure/callback>

- 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

### Web

#### 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](#)

https://bookstack.yourdomain.com/login/service/azure/callback

Add URI

#### 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.

e.g. https://example.com/logout

#### 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)

#### Supported account types

Who can use this application or access this API?

- ☒ Accounts in this organizational directory only (LNC Freelancing only - Single tenant)
- ☐ Accounts in any organizational directory (Any Microsoft Entra ID tenant - Multitenant)

[Help me decide...](#)

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

### • Permissions: Delegated User permission (User.Read)

- 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)				
email	Delegated	View users' email address	No	✓ Granted for LNC Freelancing
openid	Delegated	Sign users in	No	✓ Granted for LNC Freelancing
profile	Delegated	View users' basic profile	No	✓ Granted for LNC Freelancing
User.Read	Delegated	Sign in and read user profile	No	✓ Granted for LNC Freelancing

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 Bookstack, 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.

The relevant environment variables for OpenID are as follows:

- **AZURE\_AUTO\_REGISTER:** If this setting is activated, user objects are automatically created when they authenticate for the first time via Microsoft Entra ID and the user does not yet exist in Bookstack.
- **AZURE\_AUTO\_CONFIRM\_EMAIL:** If activated this will skip the “Confirm email” setting as all email addresses are considered verified by Microsoft Entra ID.
- **AZURE\_TENANT:** Here you have to enter the Tenant ID of your Microsoft Entra ID Tenant.
- **AZURE\_APP\_ID:** This is the client id of your App Registration in Microsoft Entra ID.
- **AZURE\_APP\_SECRET:** Here you have to provide the Client Secret from the App Registration.

**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 Bookstack that authenticates using Microsoft Entra ID. In addition, the database container and mail settings are also specified.

Customize this content with your specifications and save the content in a normal docker-compose.yaml 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"
services:
  <yourbookstackcontainername>:
    image: lscr.io/linuxserver/bookstack
    container_name: <yourbookstackcontainername>
    environment:
```

- PUID=1000
- PGID=1000
- APP\_URL=https://<yourbookstackdomain>
- DB\_HOST=<yourmariadbcontainername>
- DB\_USER=<yourdbuser>
- DB\_PASS=<yourdbpassword>
- DB\_DATABASE=<yourdbname>
- MAIL\_HOST=<yourmailserver>
- MAIL\_PORT=587
- MAIL\_FROM\_NAME=<yourmailname>
- MAIL\_FROM=<yoursmtppmail>
- MAIL\_USERNAME=<yoursmtppmailuser>
- MAIL\_PASSWORD=<yoursmtppmailpassword>
- AZURE\_APP\_ID=<yourclientid>
- AZURE\_APP\_SECRET=<yourclientsecret>
- AZURE\_TENANT=<yourtenantid>
- AZURE\_AUTO\_REGISTER=true
- AZURE\_AUTO\_CONFIRM\_EMAIL=true

volumes:

- <yourpersistentpathforbookstack>:/config

ports:

- 6875:80

restart: unless-stopped

depends\_on:

- <yourmariadbcontainername>

<yourmariadbcontainername>:

image: lscr.io/linuxserver/mariadb

container\_name: <yourmariadbcontainername>

environment:

- PUID=1000
- PGID=1000
- MYSQL\_ROOT\_PASSWORD=<yourdbrootpassword>
- TZ=<yourtimezone>
- MYSQL\_DATABASE=<yourdbname>
- MYSQL\_USER=<yourdbuser>
- MYSQL\_PASSWORD=<yourdbpassword>

volumes:

- <yourpersistentpathformariadb>:/config

restart: unless-stopped

---

Revision #2

Created 21 July 2024 14:56:50

Updated 22 July 2024 12:27:08 by Luca Noah Caprez