Azure AD Single Sign-On

Login to Peerdom with your Microsoft Azure AD credentials

A little background

Single sign-on (**SSO**) is an authentication scheme that allows a user to log in with a single ID and password to any of several related, yet independent, software systems, such as Peerdom. That is, employees can log in with one single set of credentials to get access to all corporate apps, websites, and data for which they have permission. SSO solves key problems by providing greater security and standardized password compliance. Your team will not have to create yet another account on yet another service – they can use a pre-existing, familiar username and password to log in to Peerdom.

Microsoft Azure Active Directory is a "universal platform to manage and secure identities": https://azure.microsoft.com/en-us/services/active-directory/

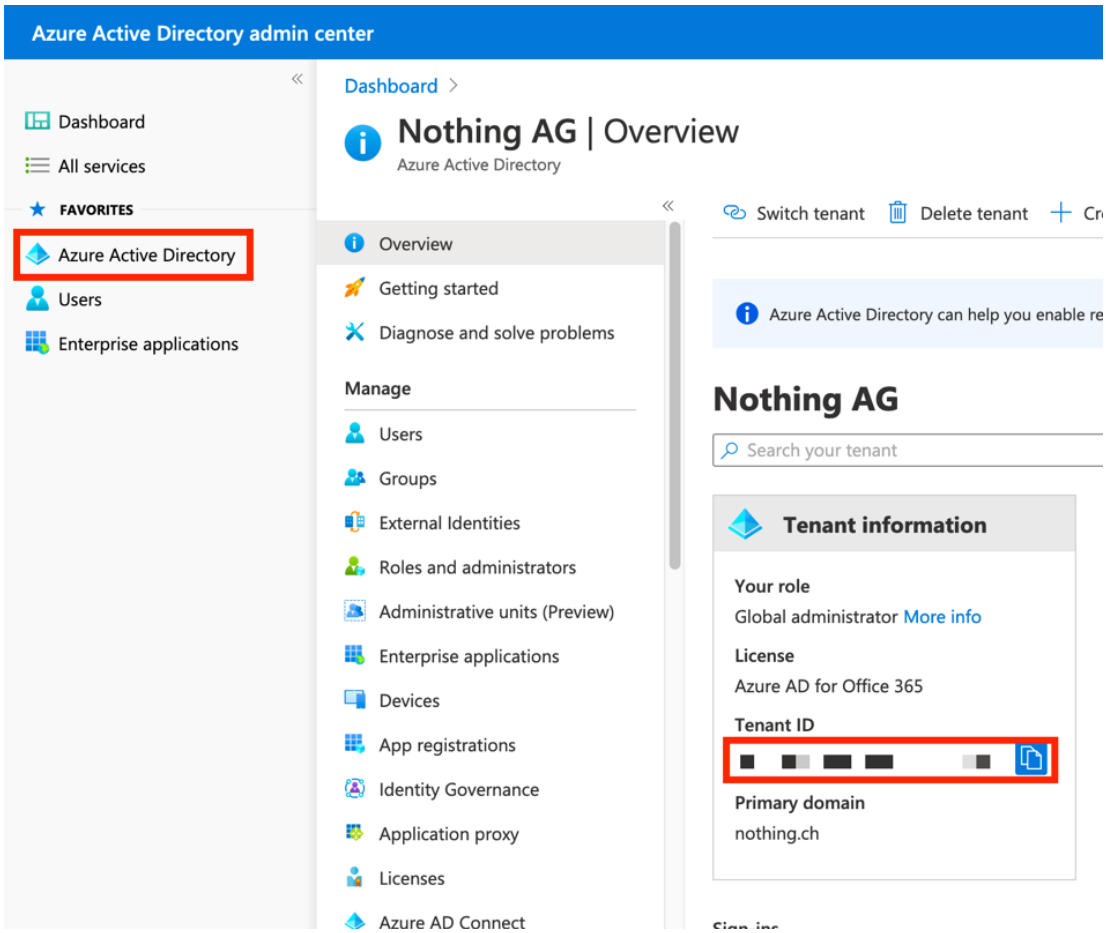
Peerdom supports the most recommended single-sign-on methods: *OpenID Connect* and *OAuth*.

What to do

1. Get your Tenant ID

We will need your **Azure Tenant ID** to activate SSO authentication. This can be found via the Active Directory Portal or the Azure Portal. In either case, the Tenant ID can be found in the *Azure Active Directory* > *Overview* section. Note your Tenant ID as you'll send it to us later.

Peerdom - c/o Nothing AG, Kirchstrasse 175, 3084 Wabern, Schweiz
hello@peerdom.org



The TenantID can be found under Tenant information in the Azure Active Directory Overview section

2. Register the Peerdom Sync App

In the *App Registrations* section, create a new application. Add two Redirect URIs:

1. **Type:** Web
   Value: https://backend.peerdom.org/auth/azure/return

2. **Type:** Web
   Value: http://localhost:3000/auth/azure/return

Note: the second Redirect URI will be used by us to get the initial refresh token. After we've setup the synchronization, this redirect URI can be removed.

3. Create Client secret

Enter the newly registered Peerdom Sync App (from step 2) and create a new Client Secret in the *Certificates & Secrets* section. Choose the option: Never expire. Note the Client Secret value for later.

4. Enable API permissions

Go to the API permissions tab on the Peerdom Sync App and add two permissions:

1. Microsoft Graph -> Application -> Directory.Read.All
2. Microsoft Graph -> Application -> User.Read.All

Approve the admin consent for both of these permissions on your organisation level by clicking on the button “Grant admin consent for ORGNAME”

5. Create a group to define peer synchronization

Peerdom synchronises with your Azure AD, meaning that your Peerdom directory will stay up to date with the current names and email addresses as defined in your Azure Active Directory. You will need to create a group with the users you’d like to synchronise. Create this group on Azure AD, add the members you’d like to appear on Peerdom, and write down the Group ID.

6. (Optional) Create a group to restrict logins

By default, Peerdom will accept all SSO log in attempts from your top-level domain. To restrict log in access to a particular group of users, you have two options:

1. Restrict access to your Peerdom Sync App Registration. To do so, follow the Azure AD documentation (under the section *App Registration*). The Application (client) ID you send us (see below) will enforce these restrictions.
2. Create a new group, add the users you’d like to give log in access to Peerdom, and write down the Group ID. This GroupID may be the same you created in step 5 for peer synchronization, but it can also differ.

7. Send us the SSO information

We need the following:

1. **Tenant ID**

2. **Application (client) ID**
   for the Peerdom sync app you registered

3. **Client secret value**
   for the Peerdom sync app

4. **Group ID**
   for peer synchronization

5. (Optional) **Group ID**
   for log in group restrictions; if different from #4 above

6. **Avatar synchronization: yes/no**
   Yes: synchronise images from Azure
   No: images are uploaded and managed in Peerdom

7. **Default access rights**
   When you add a new colleague to your Azure AD, a new peer will be created in Peerdom. We need to know what access rights to give them by default.

Member:	**View** content
Editor:	**Edit** content, invite/add/remove other peers
Owner:	Edit content, invite/add/remove, **administer** access rights