User and Access control

Snow Owl supports role based access control by organizing **Users** into a set of **Roles**. A **Role** is essentially a group that has a set of assigned **Permissions**. **Permissions** allow users to do certain things in the system. If **User** *A* belongs to a **Role** called *R* that has a permission called *P*, then user *A* can perform operations that require permission *P*.

Permissions

- browse:* - to be able to browse/search/view terminology content
- edit:* - to be able to create/modify/delete terminology content
- import:* - to be able to modify terminology content via import operation
- export:* - to be able to export terminology content into various formats
- version:* - to be able to create a terminology version
- promote:* - to be able to merge task changes to the terminology working branch (aka MAIN)

Default Roles

Snow Owl comes with a set of predefined **Roles** that can be used to organize your users in order to restrict them from performing something they should not be allowed to do. Each role has a default set of assigned permissions

- Browser - browse, export
- Editor - browse, edit, import, export
- Reviewer - browse, export, promote
- Administrator - browse, edit, import, export, promote, version

Administrator role is a special **Role** that is required by Snow Owl. It must be kept as is in order to run Snow Owl without issues. Snow Owl uses this role to identify users who can version and release content of a terminology and edit anything on the terminology’s working branch (aka MAIN or equivalent).

Customizing access control

Snow Owl must be configured to use *LDAP* as external source of user and access control information in order to customize it to your needs.

IMPORTANT

If you're using an ApacheDS LDAP service (as mentioned in the install guide earlier than 5.10.11), then please switch to the more secure and robust OpenLDAP service. To install OpenLDAP please see the provided `ldap/docker-compose.yml` file in the distribution archive or this Dockerfile (https://github.com/osixia/docker-openldap/blob/stable/image/Dockerfile) if you would like to install OpenLDAP manually. Also you can install OpenLDAP manually by following a guide matching your deployment environment (CentOS 7.x/Ubuntu 14.04/Other).

Once you have a running OpenLDAP service you can connect to it with the ApacheDS Studio you've used for the ApacheDS LDAP service.

Customize schema

Snow Owl requires a permission schema object to be present in the LDAP schema in order to represent permissions in LDAP:

1. Connect to the running OpenLDAP service via an LDAP client using the `cn=admin,cn=config` user and `config` password.
2. Open the `permission_schema.ldif` in the ApacheDS Studio via Open File… menu
3. Execute the LDIF file against the OpenLDAP service

Import default configuration

After successfully modifying the schema we can import Snow Owl's permissions and the default roles.

1. Connect to the running OpenLDAP service via an LDAP client using the `cn=admin,dc=snowowl,dc=b2international,dc=com` user and configured admin password.
2. Open and execute the following LDIF files:
   a. `permissions.ldif` - contains permissions recognized by Snow Owl
   b. `roles.ldif` - contains default roles
   c. `pm.ldif` - contains default permission → role mapping

Add new user

See Section `Creating a new user from Directory Studio` in Snow Owl installation guide on how to add new users to the LDAP repository.

Modify access control

To customize access control, first connect to your running OpenLDAP service using the `cn=admin,dc=snowowl,dc=b2international,dc=com` user and configured admin password.

Create new Role

1. Create a new child entry under `dc=snowowl,dc=b2international,dc=com` using the *Administrator* Role as template
2. Change the name of your new **Role**
3. Finish the wizard

Modify Roles

Select the **Role** you would like to edit

- You can edit name by double clicking on it then changing it
- You can add new permissions by **Right Click** → **New Attribute**
- You can remove permissions by **Right Click** → **Delete Value** on a permissionId row
- You can assign users via **Right Click on uniqueMember** → **New Value**
- You can unassign users via **Right Click on a uniqueMember row** → **Delete Value**