

Setting up Authentication in Design Room ONE with Keycloak

This document describes how to setup and use authentication and user management in Design Room ONE by means of Keycloak Integration. This functionality is provided as EXPERIMENTAL. It is included for users' evaluation and feedback and is not recommended for production use. Please see the list of [know limitations](#) in the end of the document.

Table of Contents

- Keycloak Server Installation..... 2**
 - Downloading Keycloak..... 2
 - Starting Keycloak Server..... 2
- Keycloak Server Configuration 2**
 - Creating a Realm Admin 2
 - Configuring Master Realm..... 3
 - Configuring Drone Realm 7
- Setting up Users and Roles..... 11**
 - Creating Users..... 11
 - Managing Access with Roles..... 14
 - Importing Users from Active Directory 16
- Setting Up Design Room ONE Server 21**
 - Starting the Design Room ONE server..... 22
 - Login into the Design Room ONE server with the new user 23
- Exporting Models with Design Room ONE Integration Plugin 23**
 - Prerequisites: 23
- Known Limitations..... 29**

Keycloak Server Installation

Downloading Keycloak

Use the following link:

<https://downloads.jboss.org/keycloak/7.0.1/keycloak-7.0.1.zip>

Starting Keycloak Server

Unzip the downloaded file keycloak-7.0.1.zip into an installation directory of your choice. We will refer to this installation directory as `KEYCLOAK_INSTALL_DIR`.

Run the standalone version of the server in `KEYCLOAK_INSTALL_DIR/bin`

For Linux systems use the command: `standalone.sh -b 0.0.0.0`

For Windows: `standalone.bat -b 0.0.0.0`

This script uses `KEYCLOAK_INSTALL_DIR/standalone/configuration/standalone.xml` as properties input by default

```
</interfaces>
<socket-binding-group name="standard-sockets" default-interface="public" port-offset="{jboss.socket.binding.port-offset:0}">
  <socket-binding name="management-http" interface="management" port="{jboss.management.http.port:9990}" />
  <socket-binding name="management-https" interface="management" port="{jboss.management.https.port:9993}" />
  <socket-binding name="ajp" port="{jboss.ajp.port:8009}" />
  <socket-binding name="http" port="{jboss.http.port:8080}" />
  <socket-binding name="https" port="{jboss.https.port:8443}" />
  <socket-binding name="txn-recovery-environment" port="4712" />
  <socket-binding name="txn-status-manager" port="4713" />
  <outbound-socket-binding name="mail-smtp">
    <remote-destination host="localhost" port="25" />
  </outbound-socket-binding>
</socket-binding-group>
</server>

<subsystem xmlns="urn:jboss:domain:undertow:9.0" default-server="default-server" default-virtual-host="default-host">
  <buffer-cache name="default" />
  <server name="default-server">
    <http-listener name="default" socket-binding="http" redirect-socket="https" enable-http2="true" />
    <https-listener name="https" socket-binding="https" security-realm="ApplicationRealm" enable-http2="true" />
    <host name="default-host" alias="localhost">
      <location name="/" handler="welcome-content" />
      <http-invoker security-realm="ApplicationRealm" />
    </host>
  </server>
</subsystem>
```

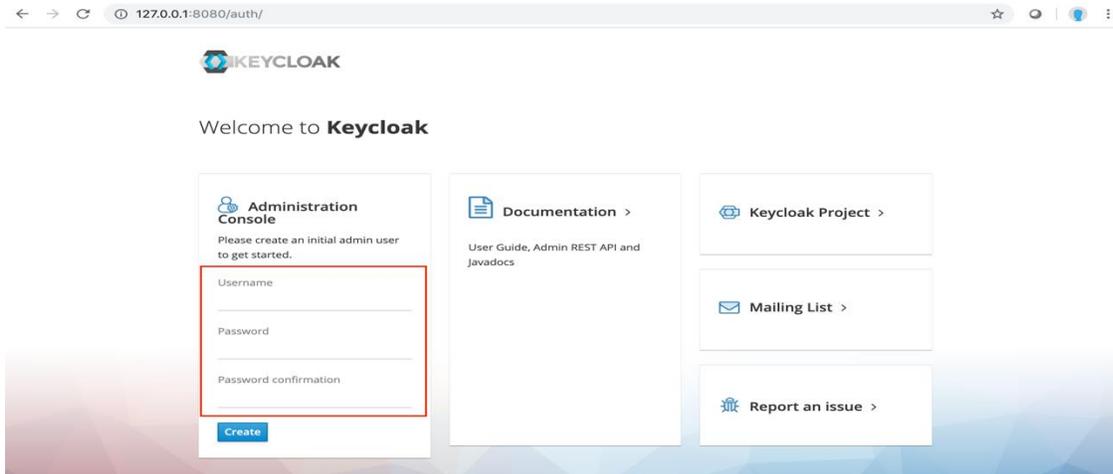
By default, Keycloak will use **8080** as an http port, **8443** as an https port and **localhost** as host. You can leave all these as is and just use the default values or change as you see fit.

Keycloak Server Configuration

Creating a Realm Admin

This admin user can be thought of as a super admin with all access (realm creation, update, deletion, user creation, update, deletion, etc)

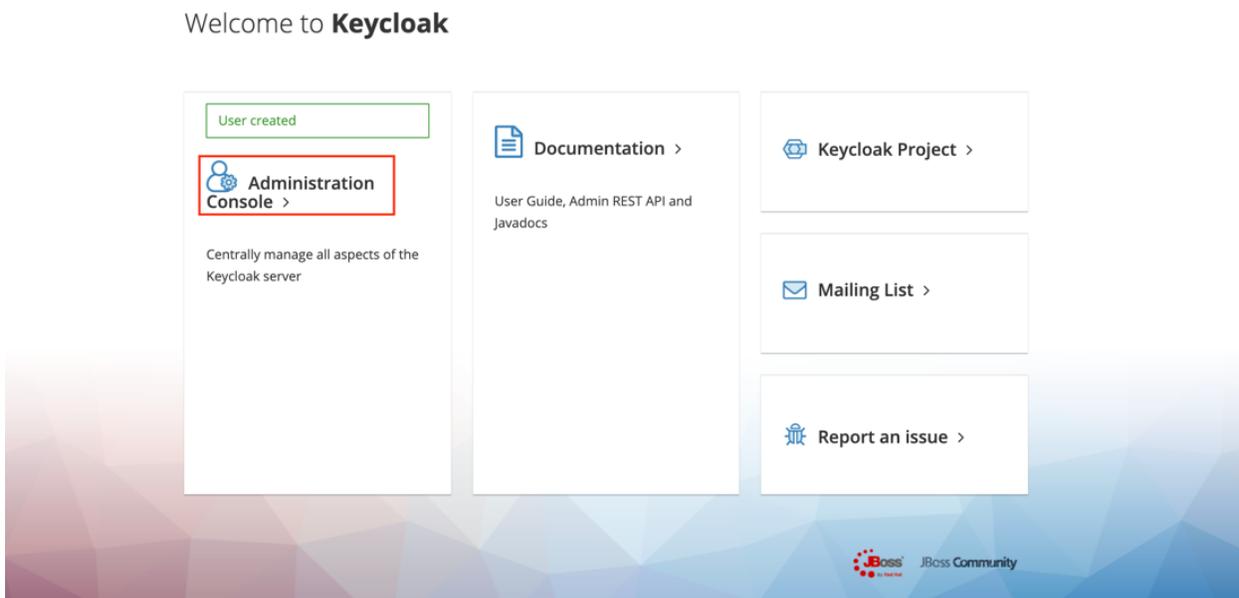
Open a browser and navigate to `http://localhost:8080/auth`



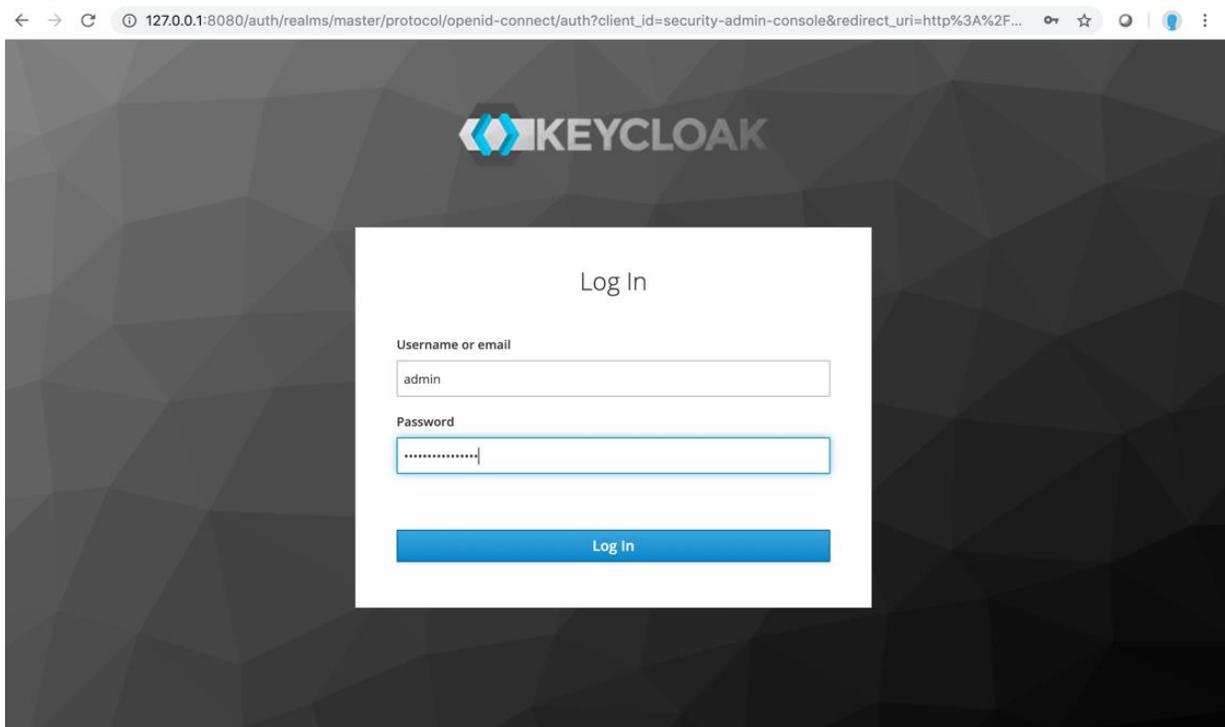
Specify the admin credentials and press create.

Configuring Master Realm

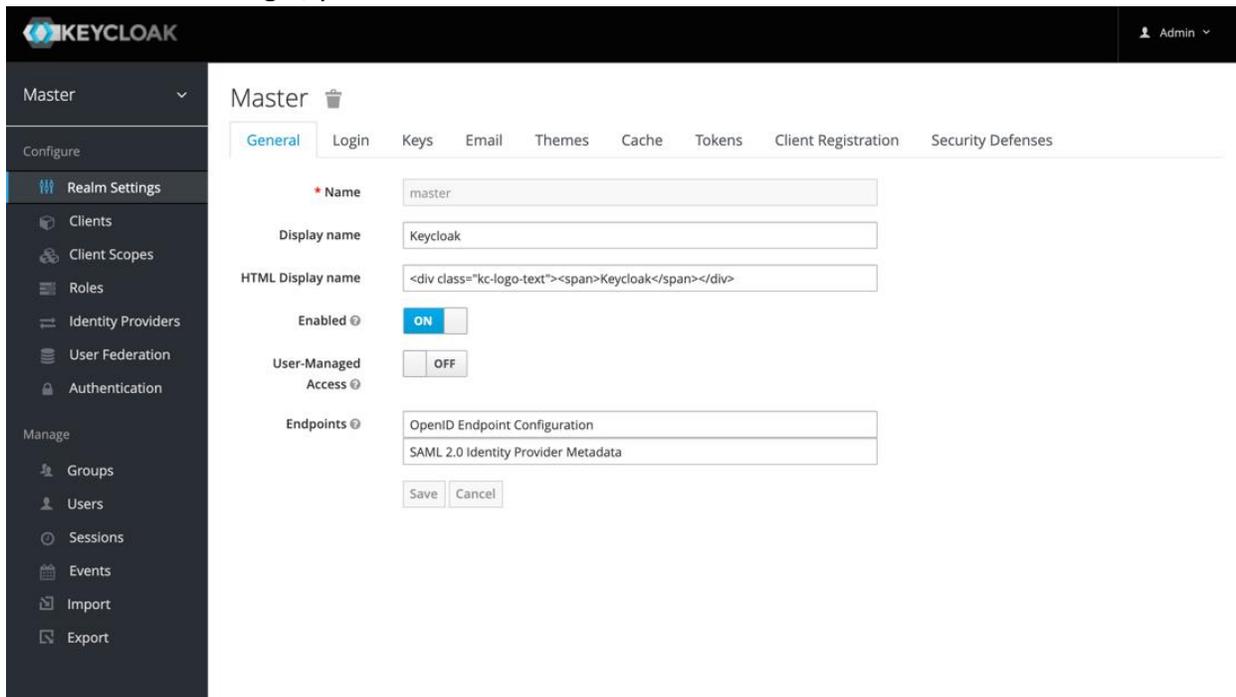
Navigate to the administration console and login by clicking on the **Administration Console** link as illustrated below



You will be directed to

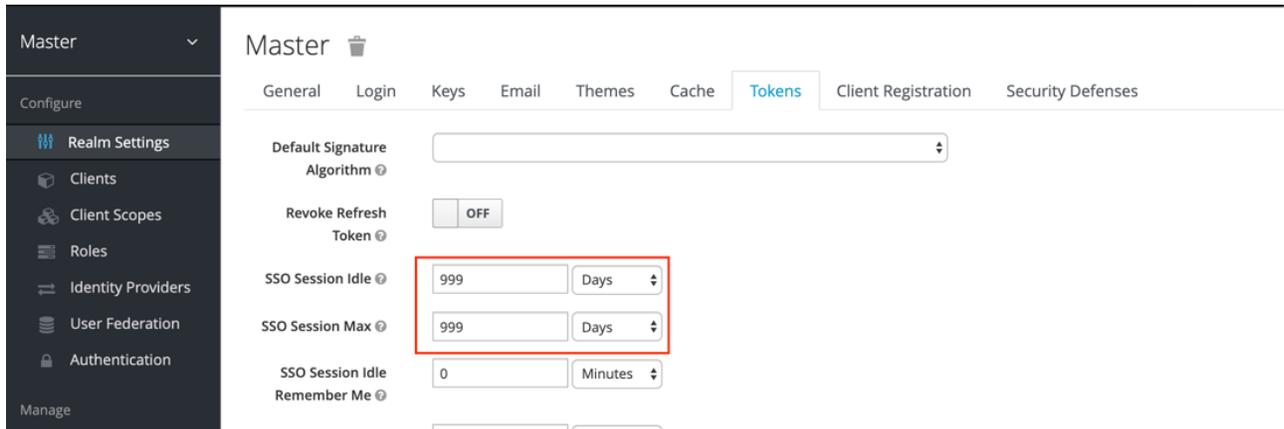


After a successful login, you should see a similar console:



1. Set SSO Session Idle and SSO Session Max to 999 days
 - a. Click on "Tokens" tab

b. Set SSO Session Idle and SSO Session Max values to 999 days.

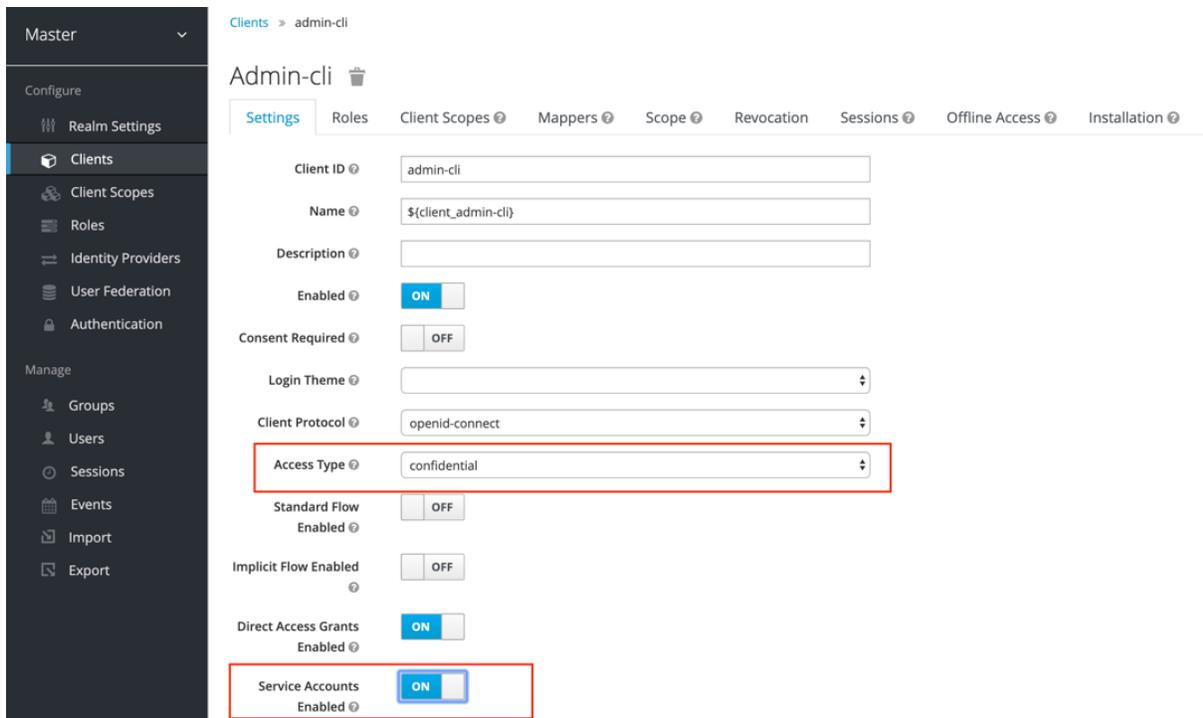


2. Ensure **Access Type** is set to “confidential”

a. In the left menu, click on “Clients”, then click on “admin-cli” client

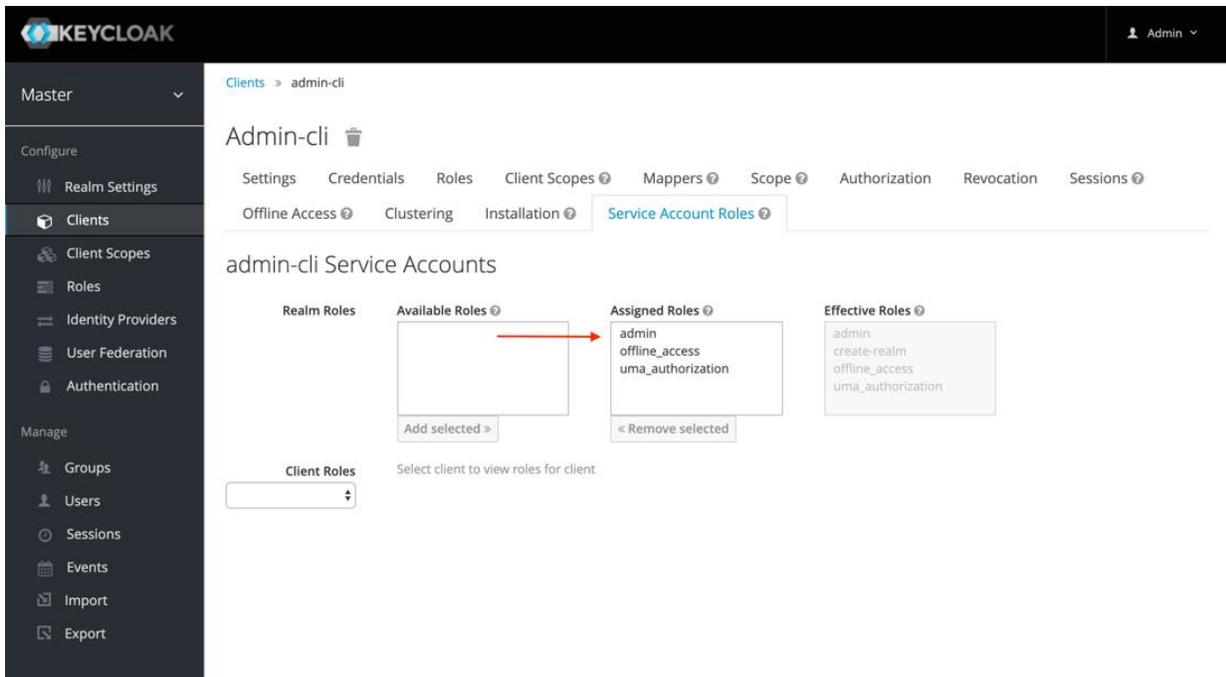
3. Activate **Service Accounts Enabled** switch

4. Press **Save**



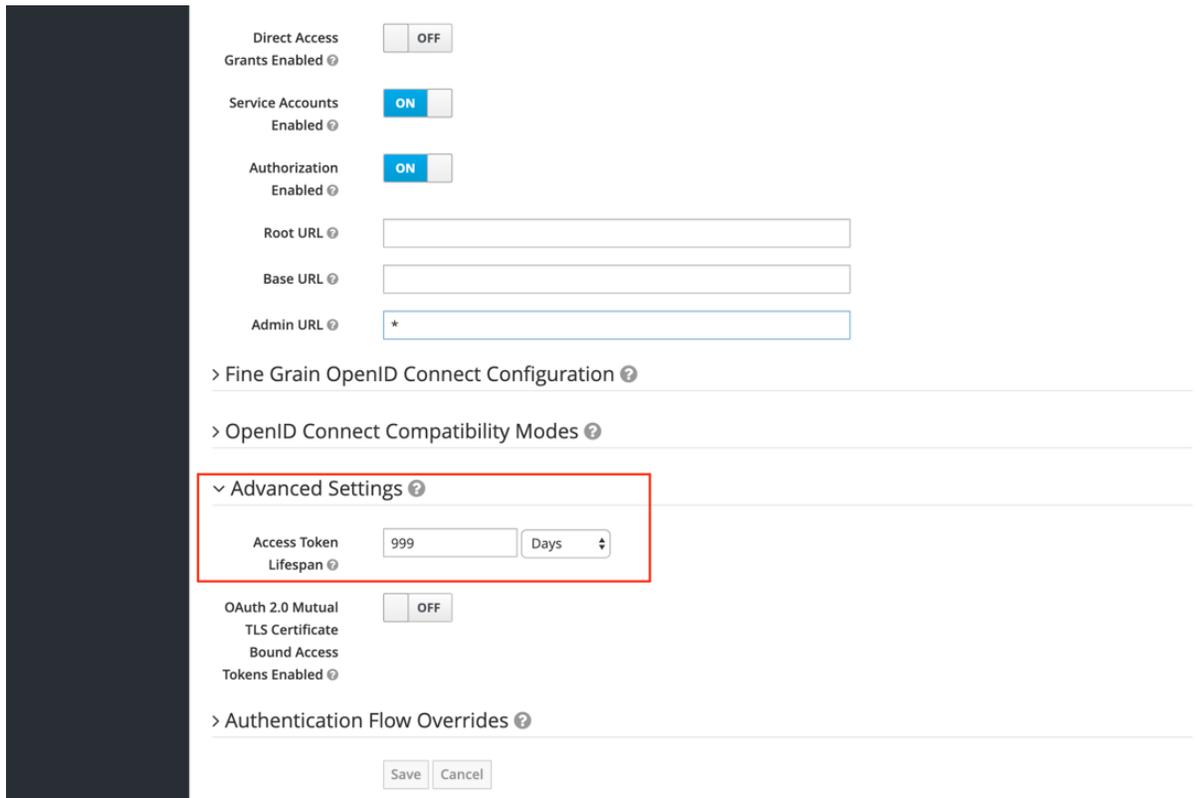
5. Under the **Service Account Roles**

a. Add the admin role to the master service account



6. Click on the **Settings** tab, scroll down and set token to 999 days under **Advanced Settings**

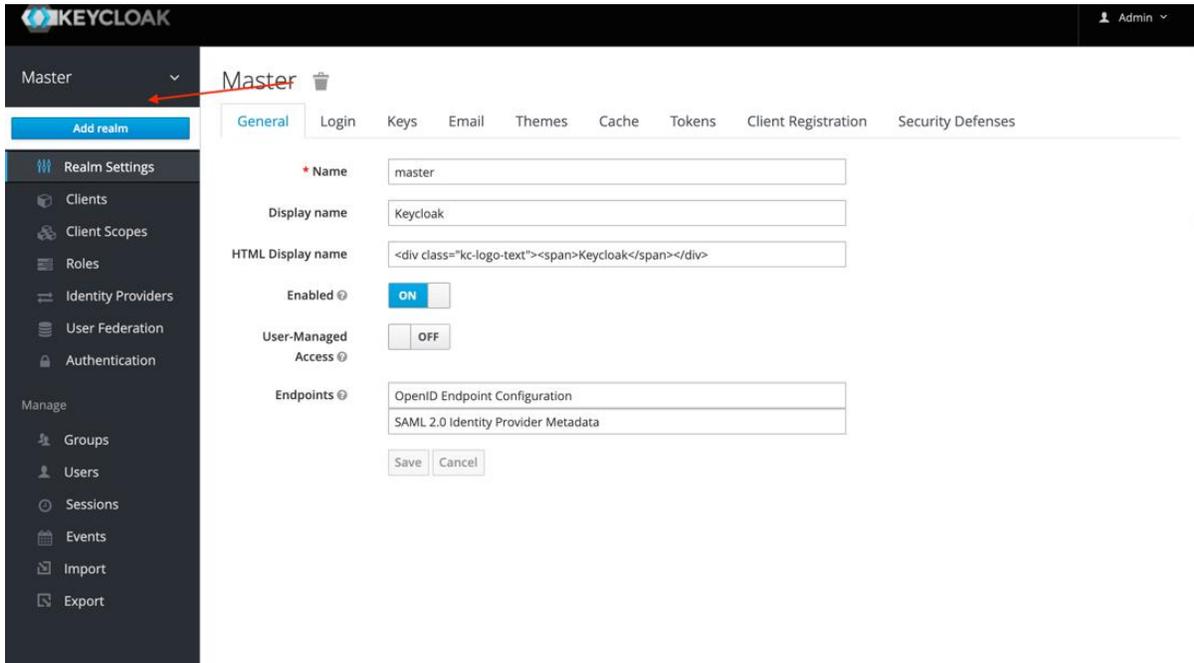
7. Click **Save**



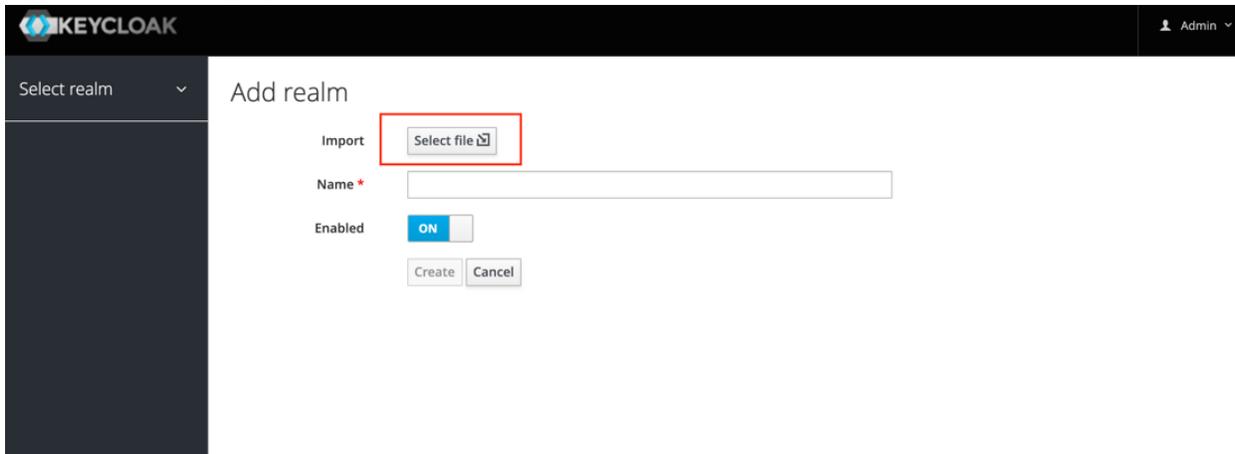
Configuring Drone Realm

1. Import the realm data

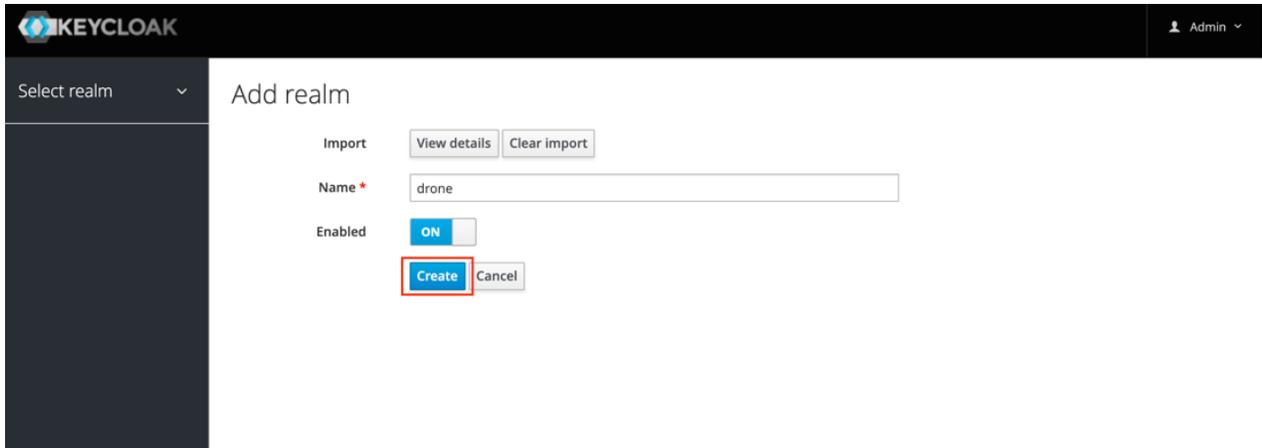
- a. Hover over **Master** in top left corner and click on **Add realm**



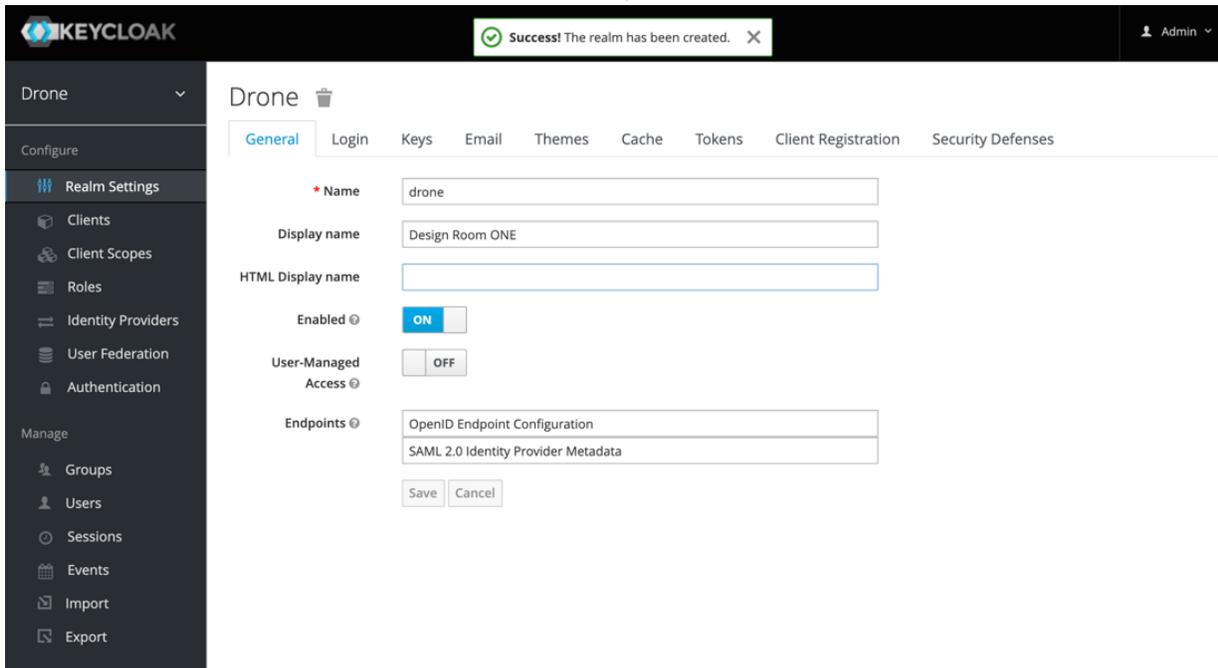
You should see a page similar to the one below.



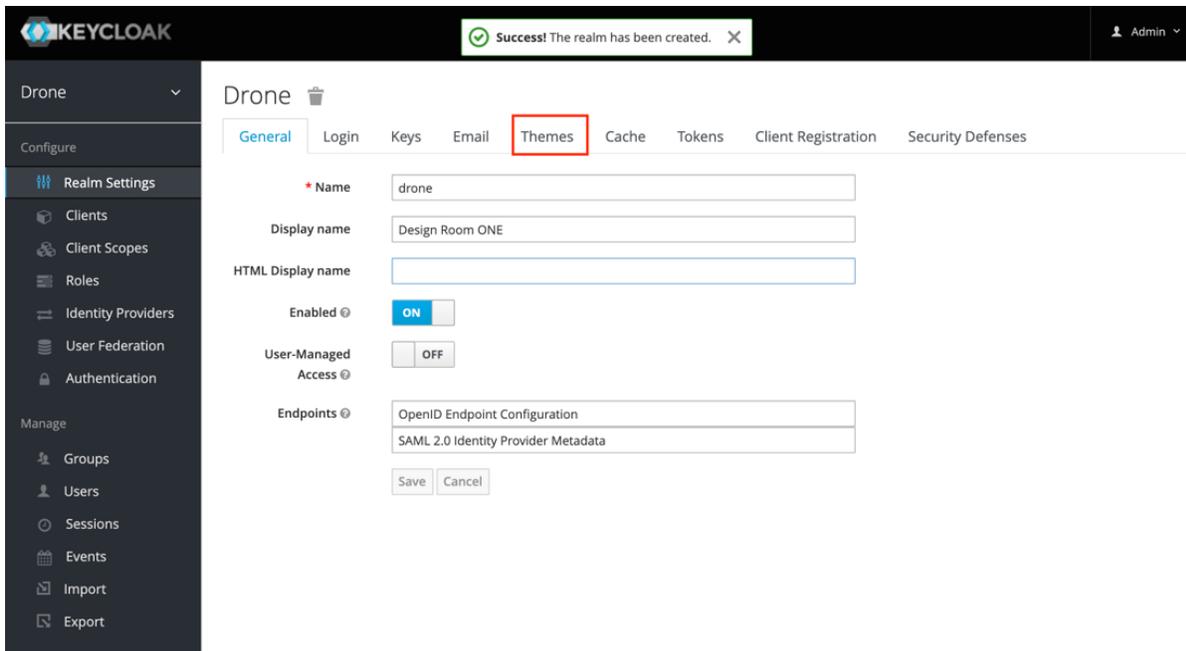
Click on **Select File** and select import file at
DR_ONE_INSTALL_DIR/DR_Install/Resources/Keycloak/drone-realm-export.json
Then, click **Create**



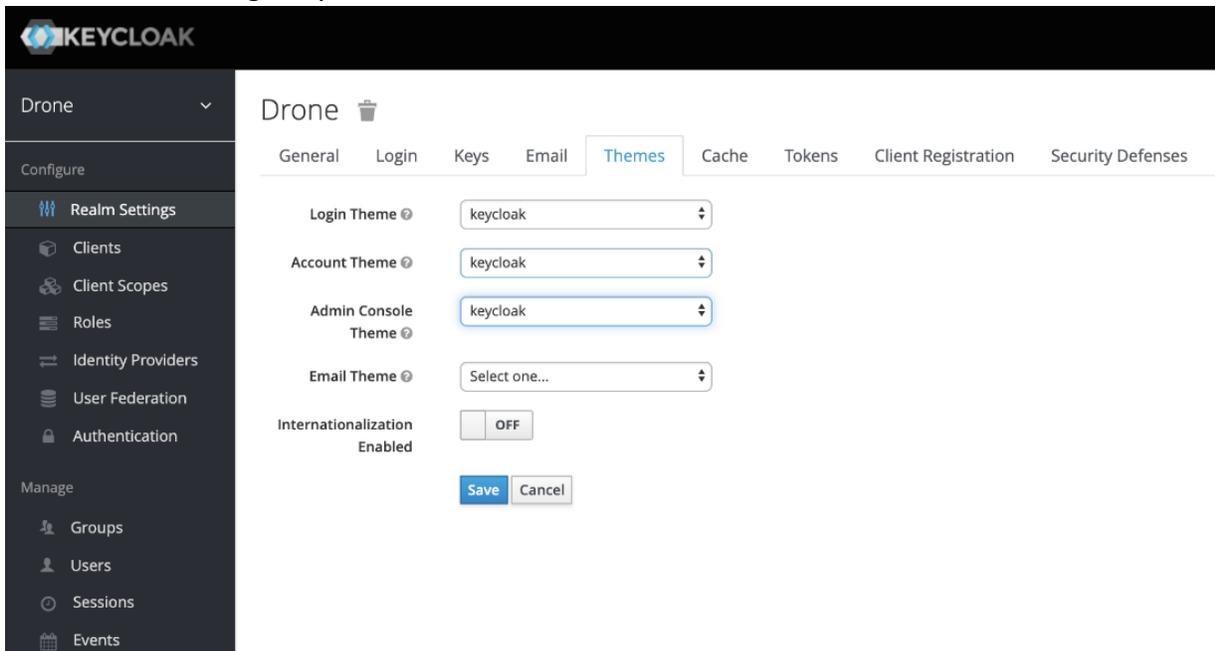
Your **Drone** realm should be created successfully as shown below:



2. Assign the proper theme
 - a. Click on the **Themes** tab

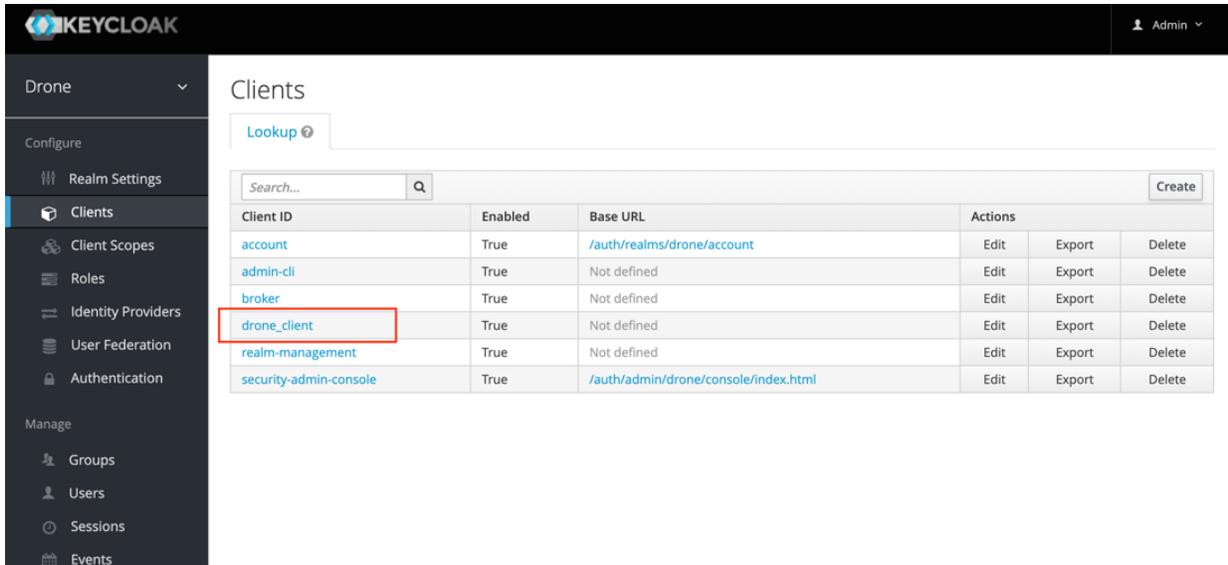


b. Assign keycloak themes as shown below and **Save**.

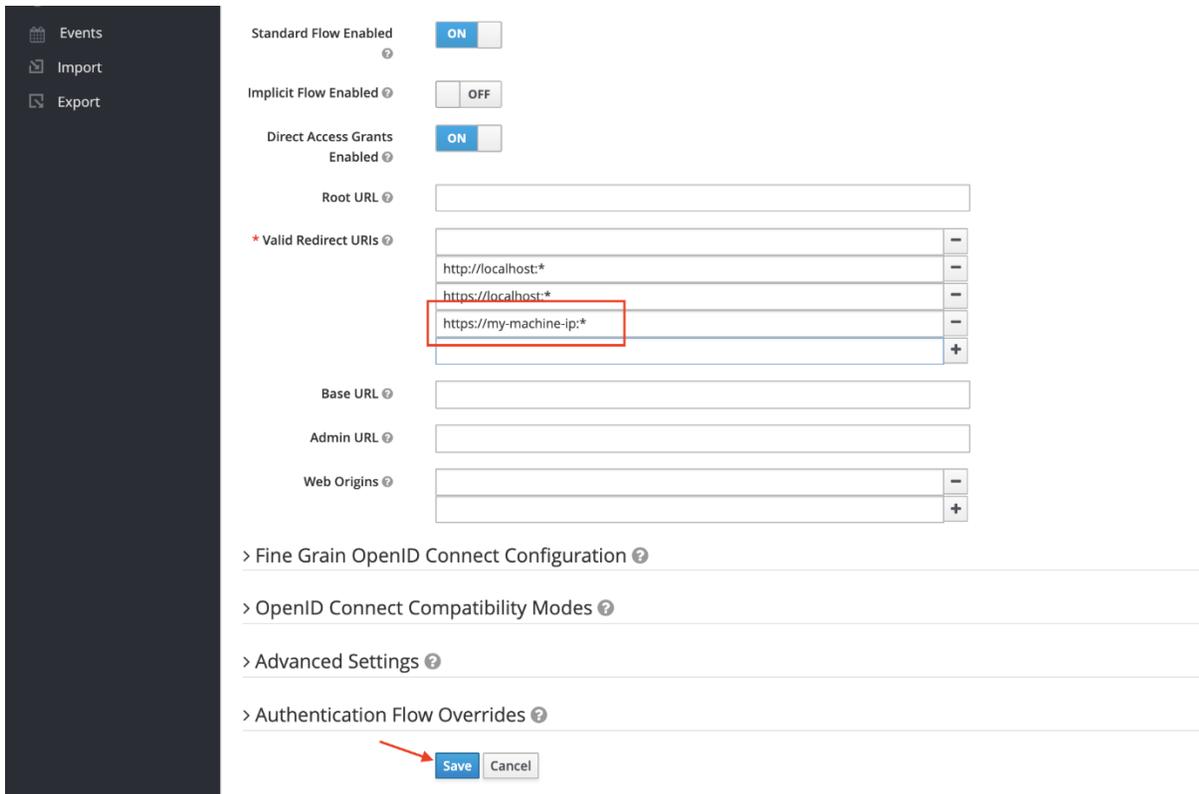


3. Setup drone_client

a. Under **Clients** click on the drone_client link



- b. Click on the **Settings** top menu, scroll down and add hostname or ip address accessible by other machines for the server Keycloak as valid redirect URIs as shown below. It is recommended to use lowercase letters.



- c. Under **Installation** menu, download the config file. The file will be downloaded as keycloak.json. Do not change the name and move the file to the DR_ONE_INSTALL_DIR\OnPrem_Design_Room\config folder

The screenshot shows the Keycloak Admin Console interface for configuring a client. The breadcrumb navigation is 'Clients > drone_client'. The main heading is 'Drone_client'. Below the heading are several tabs: Settings, Roles, Client Scopes, Mappers, Scope, Revocation, and Sessions. The 'Client Scopes' tab is active, and the 'Installation' sub-tab is highlighted with a red box. Below the tabs, there is a 'Format Option' dropdown menu set to 'Keycloak OIDC JSON'. A 'Download' button is visible above a text area containing the following JSON configuration:

```
{
  "realm": "drone",
  "auth-server-url": "http://localhost:8080/auth",
  "ssl-required": "external",
  "resource": "drone_client",
  "public-client": true,
  "confidential-port": 0
}
```

Important: After the keycloak.json file is moved, open ensure you have a valid hostname or ip-address in your **auth-server-url** attribute which other machines can use to access the server keycloak is deployed on. It is recommended to use lowercase letters.

```

1  {
2    "realm": "drone",
3    "auth-server-url": "http://my-machine-ip:8080/auth",
4    "ssl-required": "external",
5    "resource": "drone_client",
6    "public-client": true,
7    "confidential-port": 0
8  }
9
10
11
```

Setting up Users and Roles

Creating Users

Before we create a user, you can notice that some roles were created by default in Keycloak notably the ones arrowed.

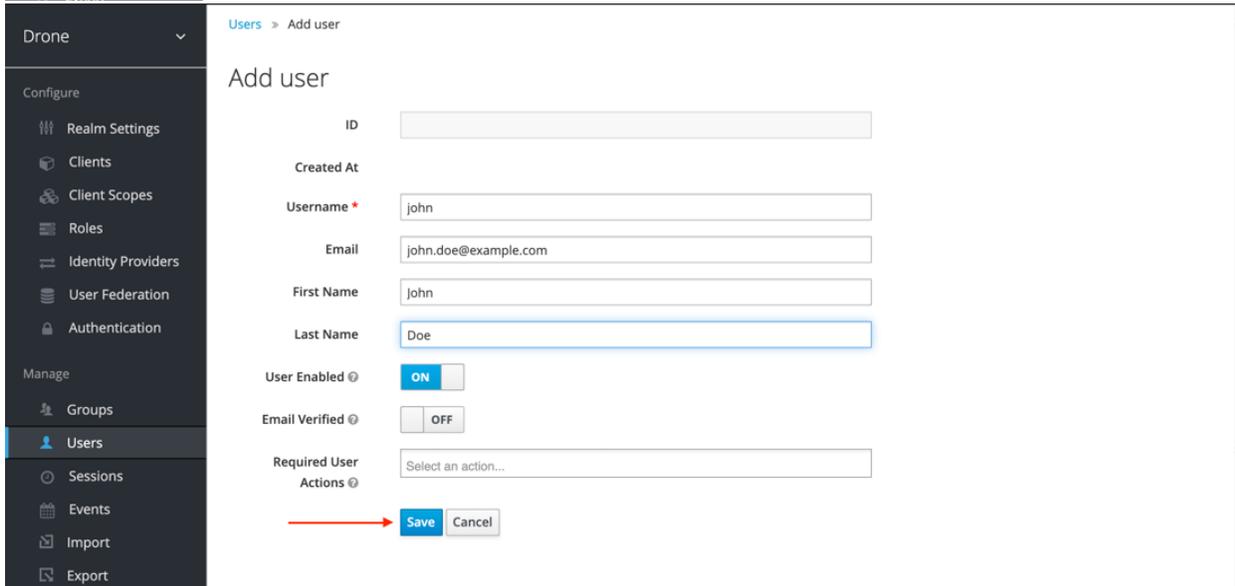
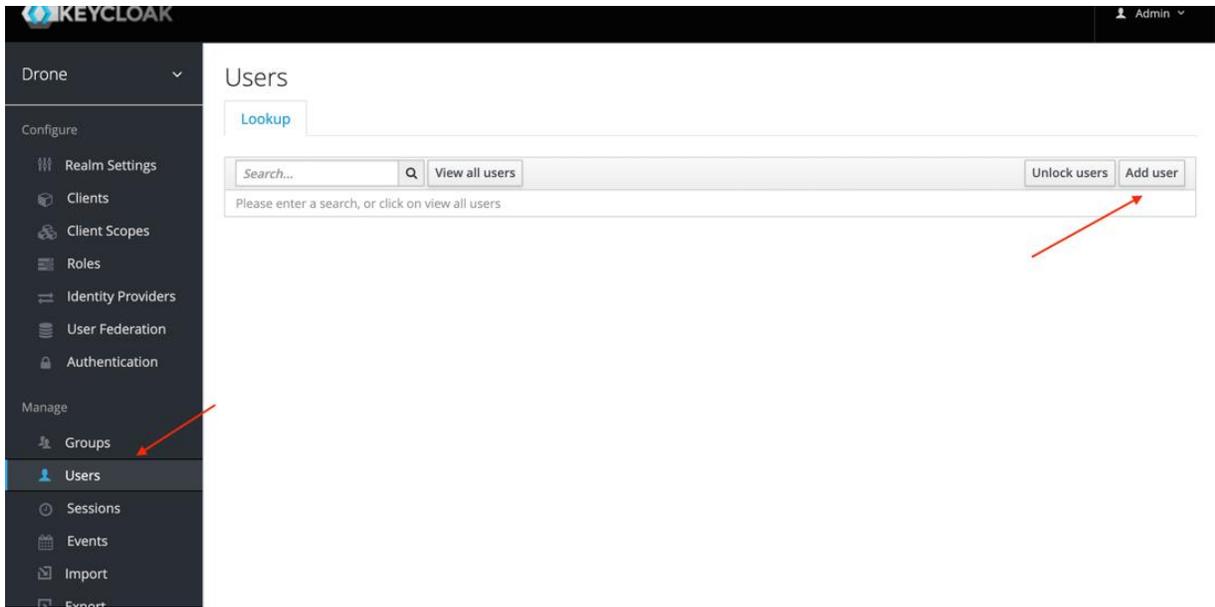
The screenshot shows the Keycloak administration interface for the 'Drone' realm. The 'Roles' page is active, displaying a table of 'Realm Roles'. The table has columns for Role Name, Composite, Description, and Actions. The roles listed are:

Role Name	Composite	Description	Actions
drone_user	False	A user in DRONE must have this role to login	Edit Delete
offline_access	False	\$(role_offline-access)	Edit Delete
sample_all_access	False	This role gives read and write access to all designs.	Edit Delete
sample_partial_access	False	This role will give read access to all design starting with traffic	Edit Delete
uma_authorization	False	\$(role_uma_authorization)	Edit Delete

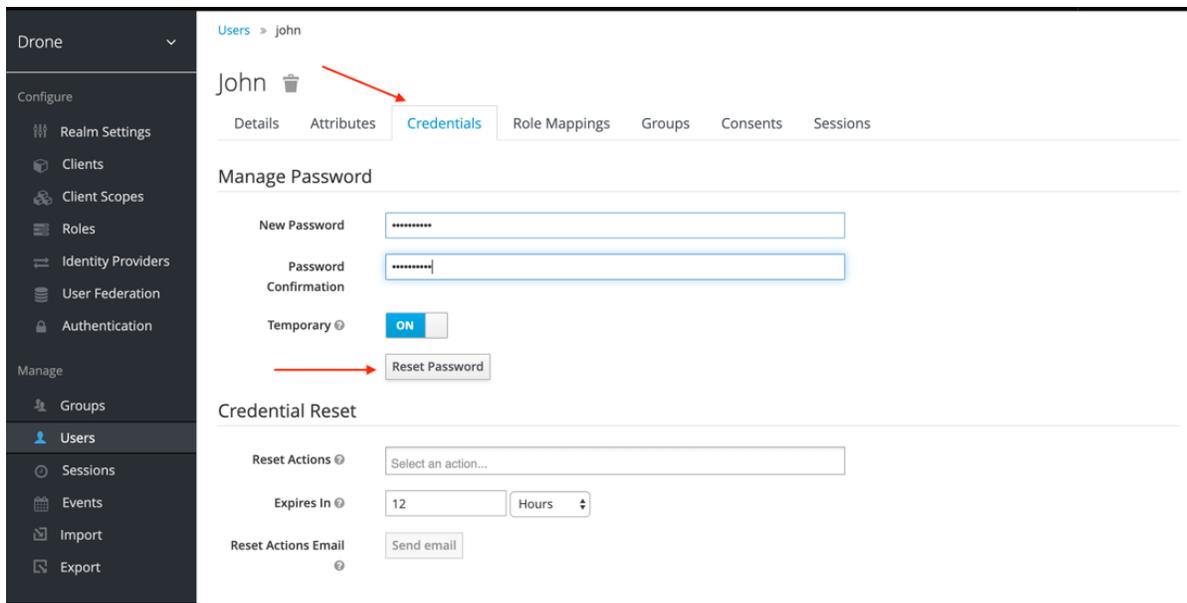
Note: **drone_user** is also a default role which means that every new user will by default inherit this role as shown below

The screenshot shows the 'Default Roles' configuration page in Keycloak. It displays three panels: 'Realm Roles', 'Available Roles', and 'Realm Default Roles'. The 'Available Roles' panel lists 'sample_all_access' and 'sample_partial_access'. The 'Realm Default Roles' panel lists 'drone_user', 'offline_access', and 'uma_authorization'. A red arrow points to 'drone_user' in the 'Realm Default Roles' panel, indicating it is a default role.

1. Create a user



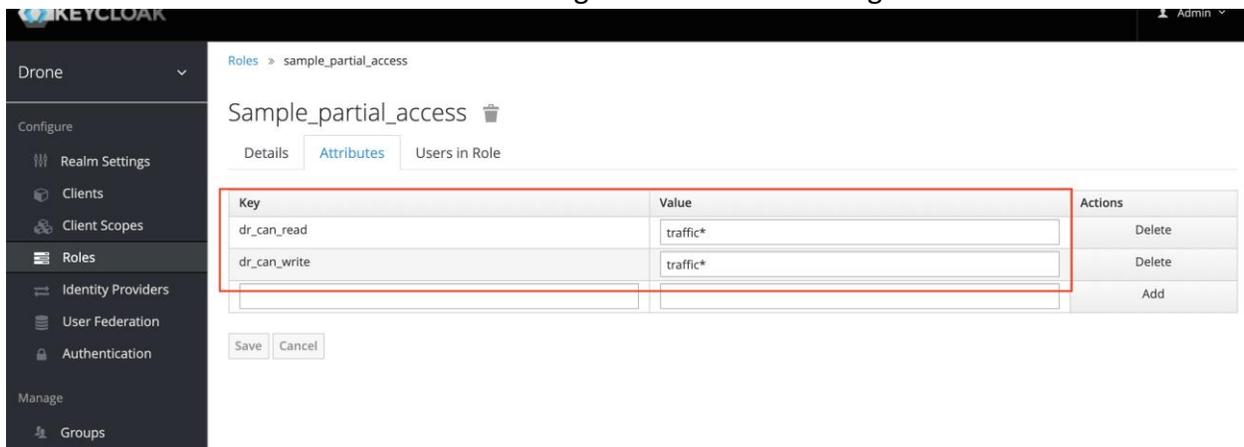
2. Setup the user password and click **Reset Password**



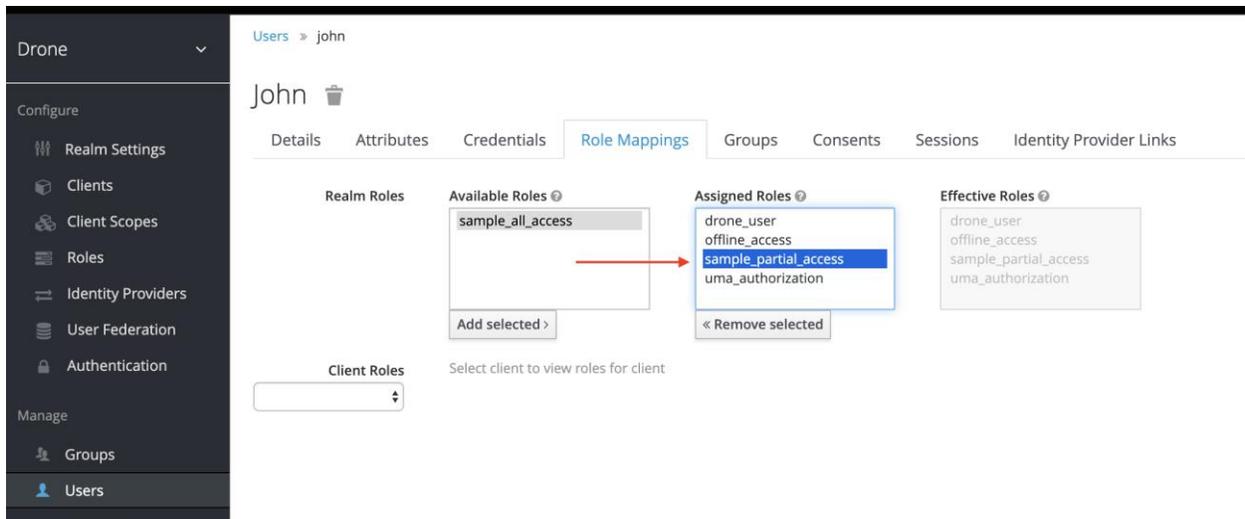
Managing Access with Roles

Access to designs in Design Room ONE is controlled with special attributes that can be specified for a role in Keycloak. The **dr_can_read** and **dr_can_write** attributes of roles give users respectively read and write access to specific designs. These attributes support wildcard as shown in the picture below.

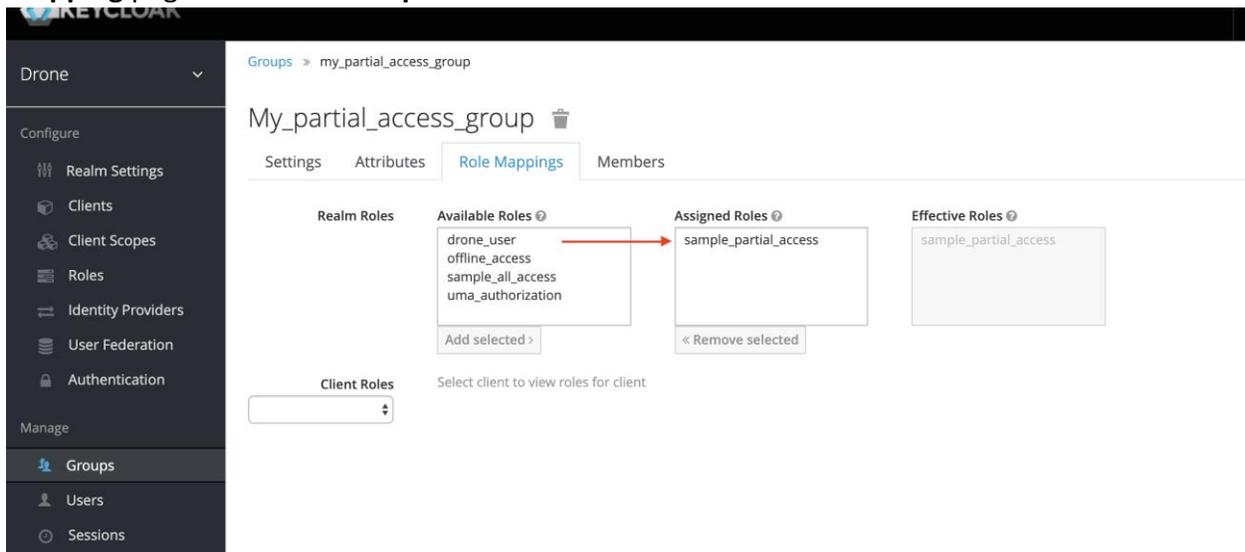
Under **Roles > sample_partial_access > Attributes** you can see that the **dr_can_read** and **dr_can_write** attributes are both set to "traffic*" which means that users or groups with this role will be able to read and write to all designs with names starting with "traffic".



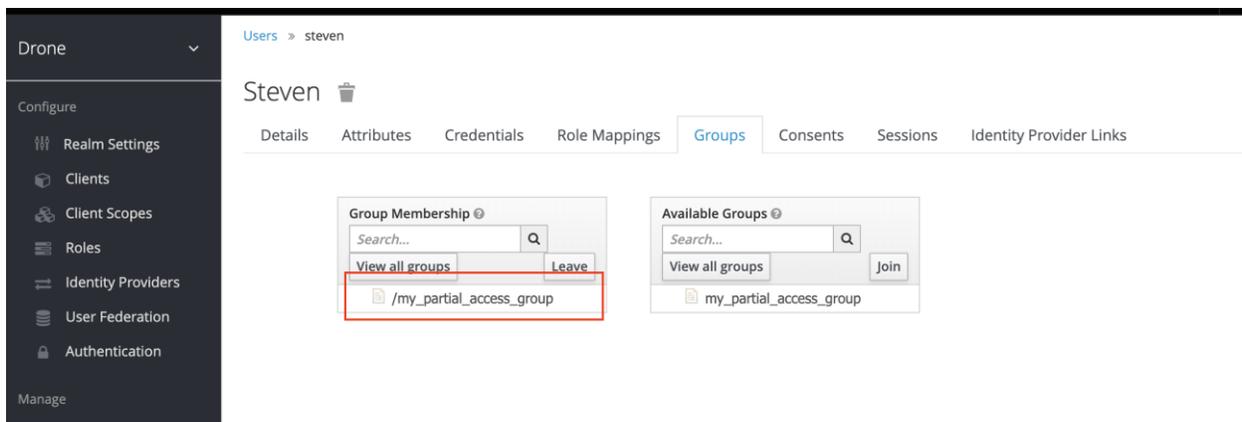
To explicitly assign a role to a user, **Role Mapping** page under the **Users** left menu can be used.



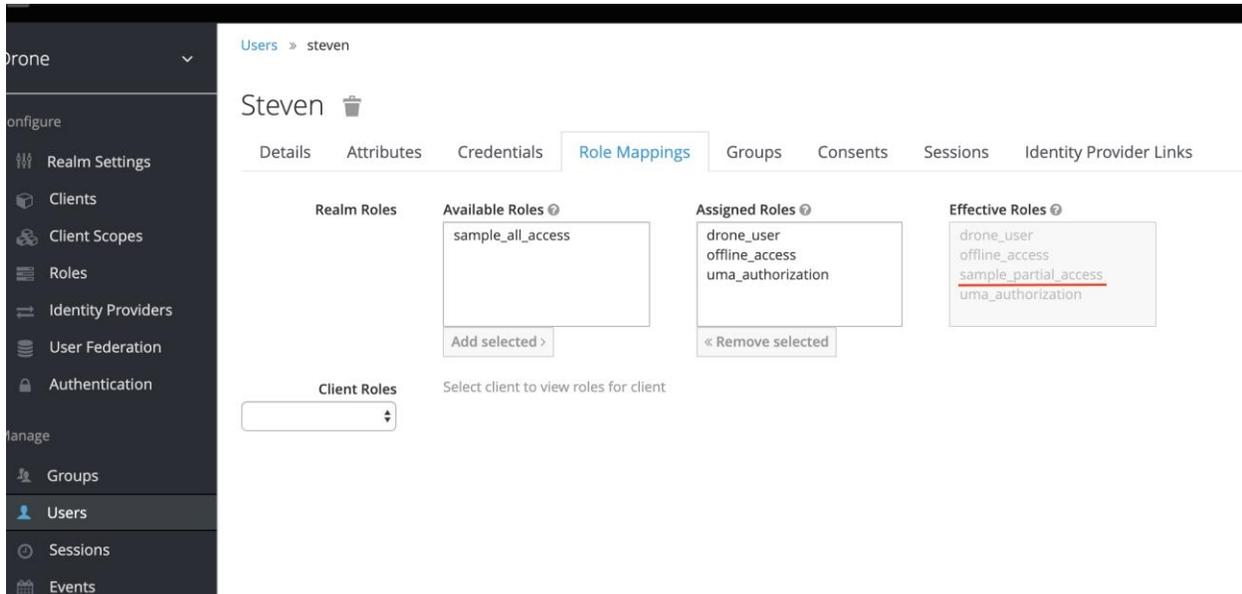
A user can implicitly get role from the groups they belong. To assign a role to a group **Role Mapping** page under the **Groups** left menu can be used.



Note: A user will inherit all the roles from the group the belong
In the example below user **Steven** belongs to **my_partial_access_group**



This means user **Steven** automatically inherits **sample_partial_access** role as it is assigned to the group.

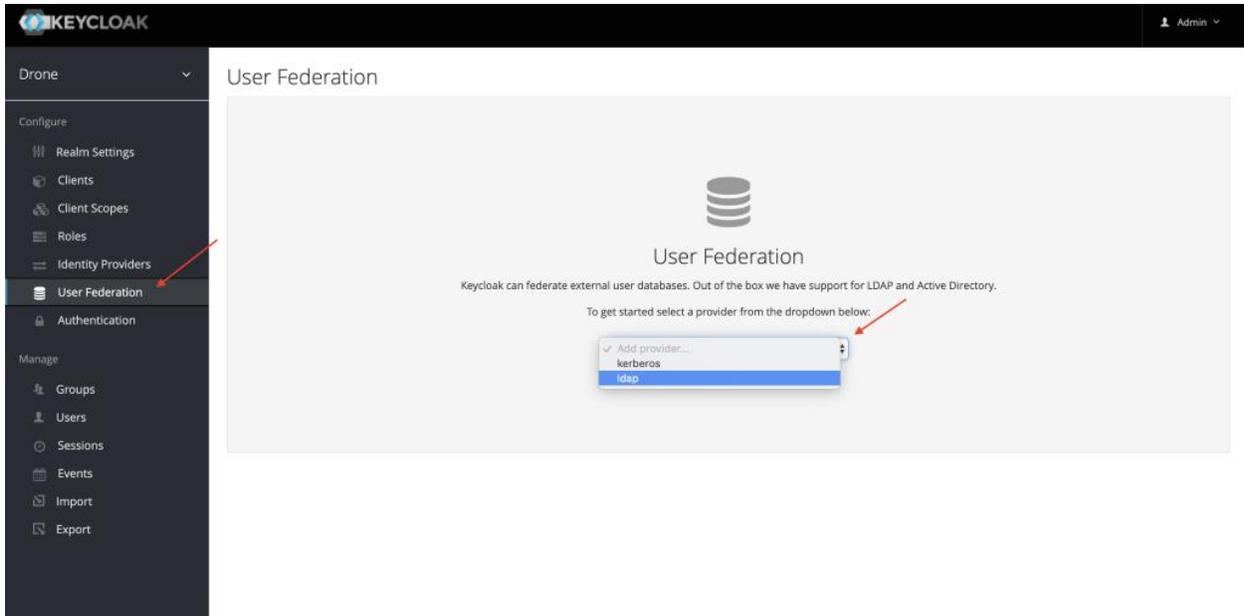


This results any **my_partial_access_group** member including user **Steven** being able to read and write all designs with names starting with “traffic” because he belongs to the group membership.

Importing Users from Active Directory

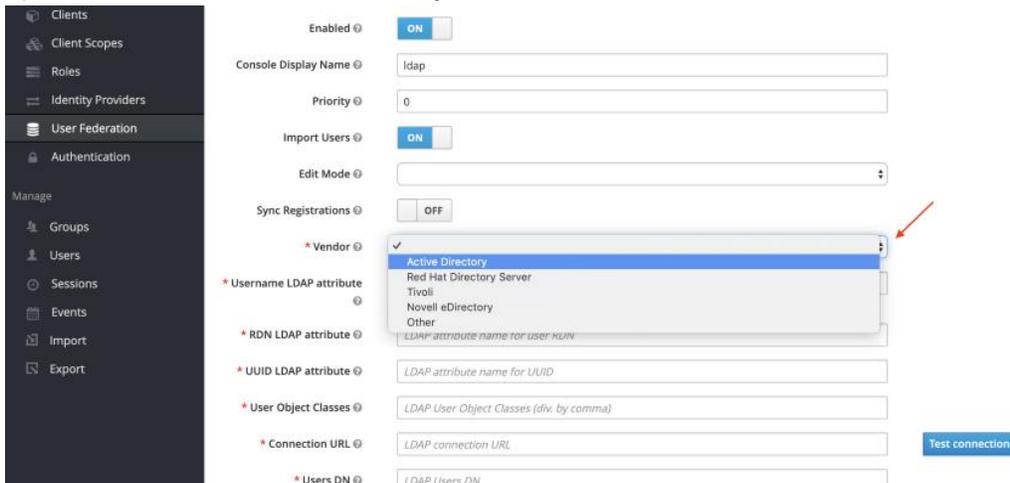
Note: The Active directory server reference is at the end of the file.

- 1) Ensure you are in the **Drone** realm and click on **User Federation** left menu item
Then select **Idap** from the dropdown



Note: We will go with basic active directory setup here. All the page options are configurable with tooltip information to allow for more advanced setups if one chooses to.

2) Set Vendor to Active Directory



You should see some fields pre-filled with default values as shown below:

* Vendor ? Active Directory

* Username LDAP attribute ? cn

* RDN LDAP attribute ? cn

* UUID LDAP attribute ? objectGUID

* User Object Classes ? person, organizationalPerson, user

* Connection URL ? LDAP connection URL

* Users DN ? LDAP Users DN

* Bind Type ? simple

Test connection

2) Set the connection url and test the connection via the **Test connection** button as shown below:

User Federation

Authentication

Import Users ON

Success! LDAP connection successful. X

Edit Mode ?

Sync Registrations OFF

* Vendor ? Active Directory

* Username LDAP attribute ? cn

* RDN LDAP attribute ? cn

* UUID LDAP attribute ? objectGUID

* User Object Classes ? organizationalPerson

* Connection URL ? ldap://steven.ad

* Users DN ? LDAP Users DN

* Bind Type ? simple

Enable StartTLS OFF

Test connection

3) Set the active directory users database here

* Users DN ? ou=users,dc=hcl,dc=com

* Bind Type ? simple

Enable StartTLS OFF

4) Set the Active Directory (AD) admin credentials and test authentication to the AD server as shown below:

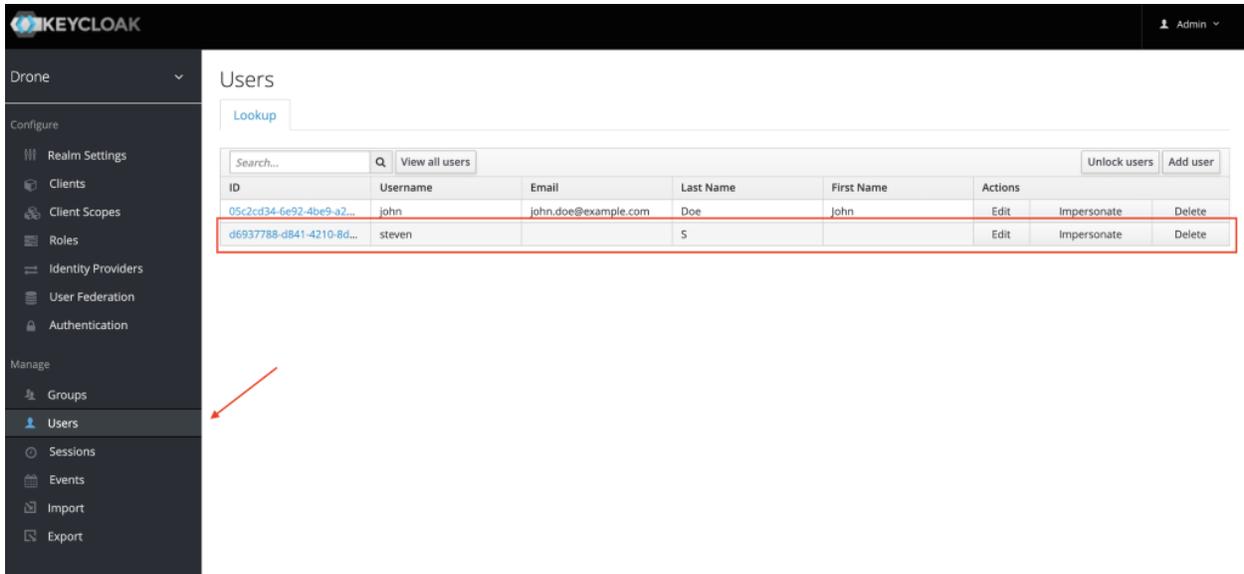
* Connection URL Success! LDAP authentication successful. X Test connection
 * Users DN
 * Bind Type
 Enable StartTLS
 * Bind DN Test authentication
 * Bind Credential
 Custom User LDAP Filter

5) Leave the rest as is and scroll down and click **Save**, then click on **Synchronize all users** to import all existing users from AD to Keycloak

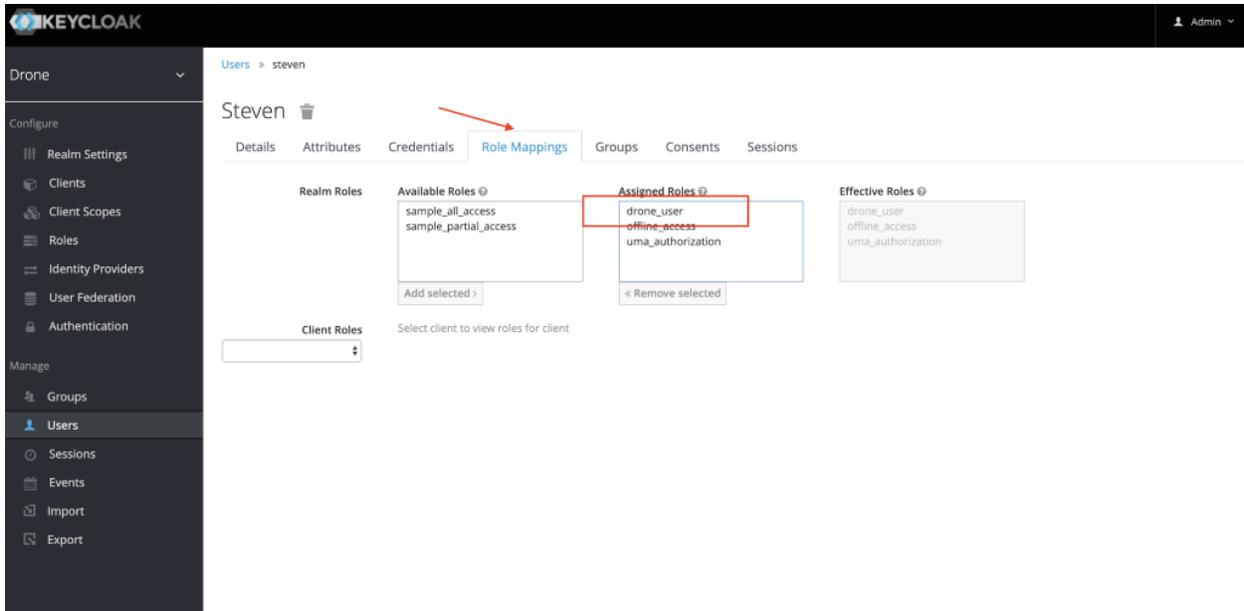
Connection Timeout
 Read Timeout
 Pagination ON
Kerberos Integration
 Allow Kerberos authentication
 Use Kerberos For Password Authentication
Sync Settings
 Batch Size
 Periodic Full Sync
 Periodic Changed Users Sync
Cache Settings
 Cache Policy
Save Cancel Synchronize changed users Synchronize all users remove imported Unlink users

In our case 1 user was imported from our active directory server.

6) If we now click on the Users menu, we see our imported user.



7) Click on the user and observe the role mappings. The user should inherit all default roles (including drone_user) to have access to the Design Room ONE server



8) You can now login into Design Room ONE with the newly created user.

LDAP Directory Information Tree data reference

```
#
# LDAPv3
# base <dc=hcl,dc=com> (default) with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#
# hcl.com
dn: dc=hcl,dc=com
objectClass: top
objectClass: dcObject
objectClass: organization
o: hcl
dc: hcl

# admin, hcl.com
dn: cn=admin,dc=hcl,dc=com
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator

# users, hcl.com
dn: ou=users,dc=hcl,dc=com
objectClass: organizationalUnit
ou: users

# steven, users, hcl.com
dn: cn=steven,ou=users,dc=hcl,dc=com
cn: steven
sn: US
objectClass: organizationalPerson

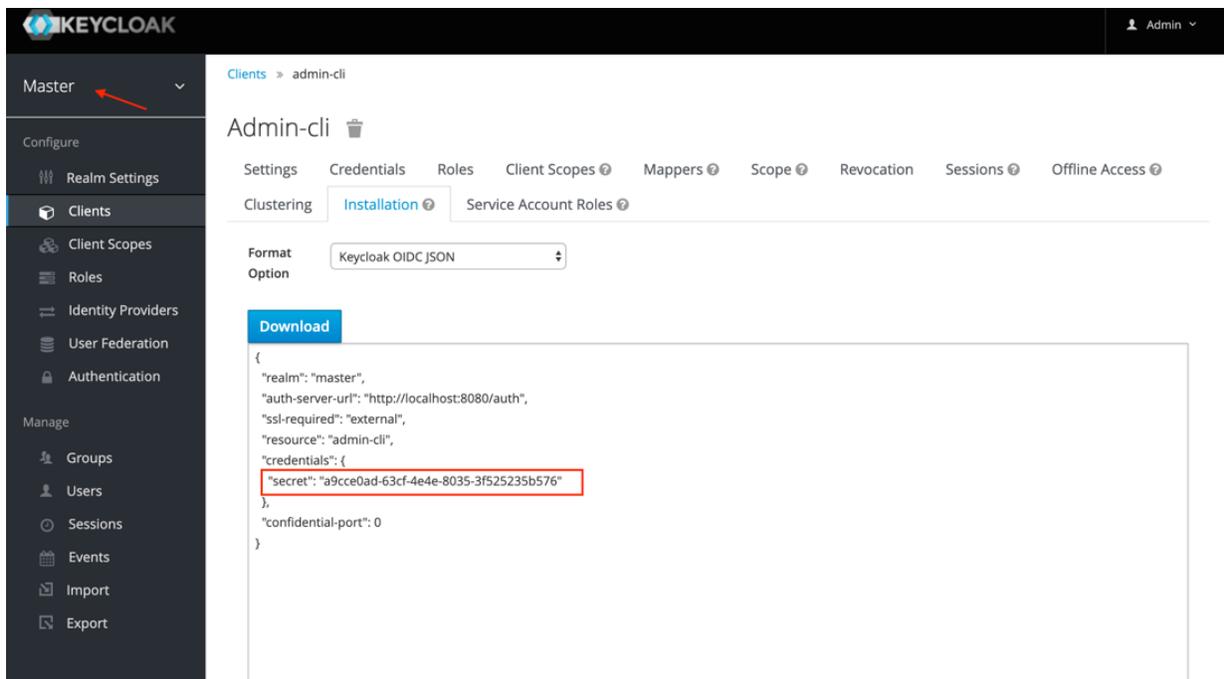
# search result
search: 2
result: 0 Success

# numResponses: 5
# numEntries: 4
```

Summary: Hcl.com(organization)->users(organization unit or OU)->steven(Member of OU)

Setting Up Design Room ONE Server

Under the **Master Realm**> Clients> Admin-cli>Installation, copy secret key



Uncomment and paste the value for **kc_admin_secret** in your Design Room ONE server configuration file (DR_ONE_INSTALL_DIR > OnPrem_Design_Room > config > server-config.json) as shown below:

```
// Number of milliseconds to wait for an ongoing server request to complete before shutting
// down the server.
"dr_shutdown_timeout": 5000,

// Authentication (for accessing information stored in Design Room ONE)
// "none": Do not use any authentication. Everyone can access all designs.
// "jazz": Use Jazz authentication. User needs to be logged in to Jazz to access designs.
// "keycloak": Use Keycloak authentication. User needs to be logged via keycloak to access designs.
"dr_auth": "keycloak",

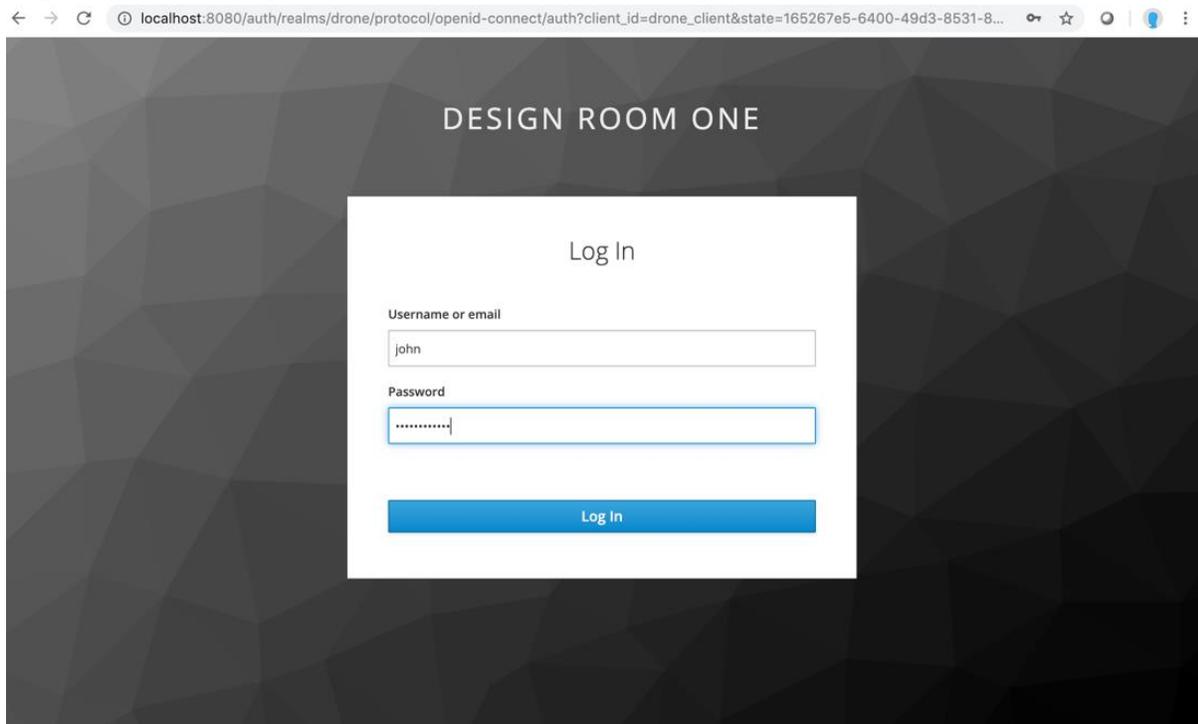
//this secret key MUST be set when using keycloak
"kc_admin_secret": "a9cce0ad-63cf-4e42-8035-3f525235b576",
```

Ensure that you use **“keycloak”** as value for **“dr_auth”** to use keycloak for authentication. You have now successfully configured Design Room ONE.

Starting the Design Room ONE server.

Ensure Design Room ONE server is started. Navigate to your Design Room ONE installation and launch the dr-deploy.js script.

Login into the Design Room ONE server with the new user



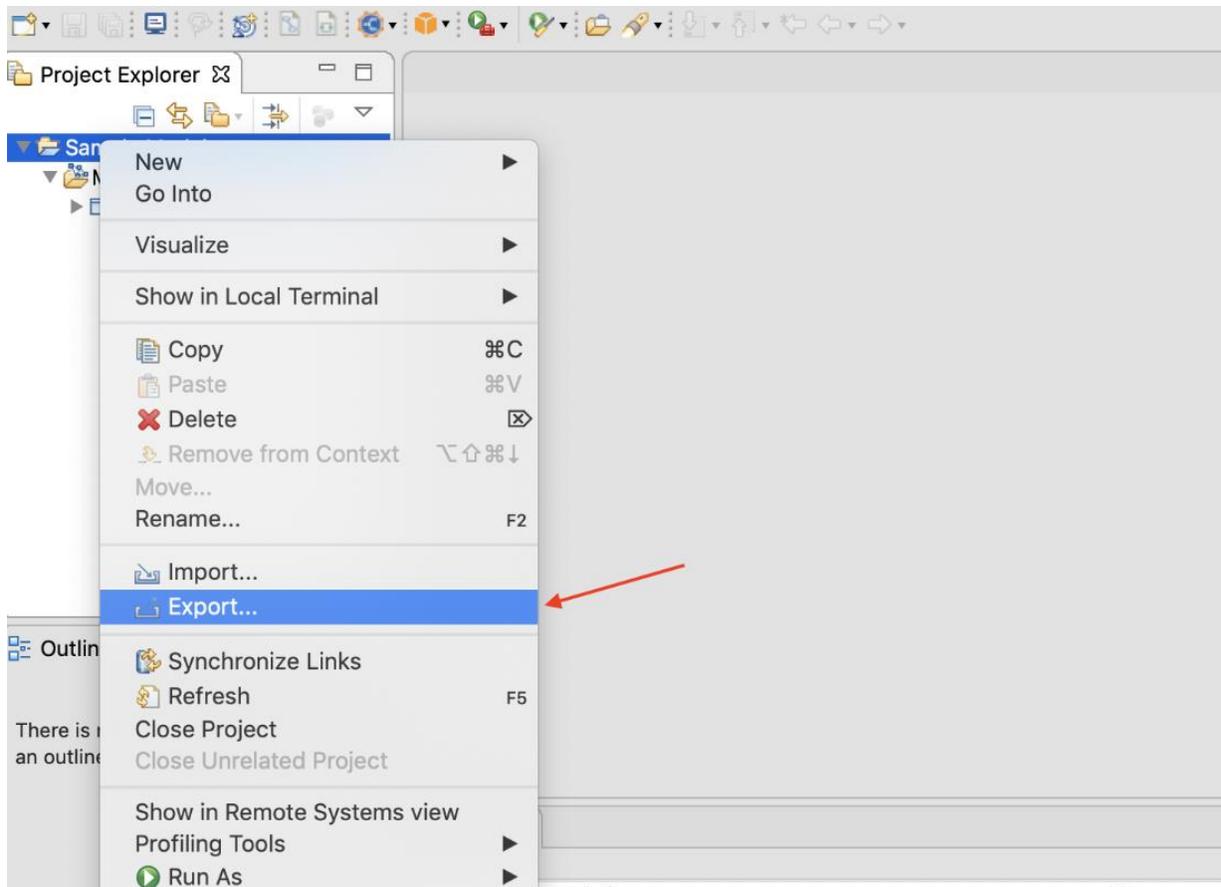
Exporting Models with Design Room ONE Integration Plugin

Below, we will highlight the steps to login in the eclipse client once Keycloak is enabled on the Design Room ONE server.

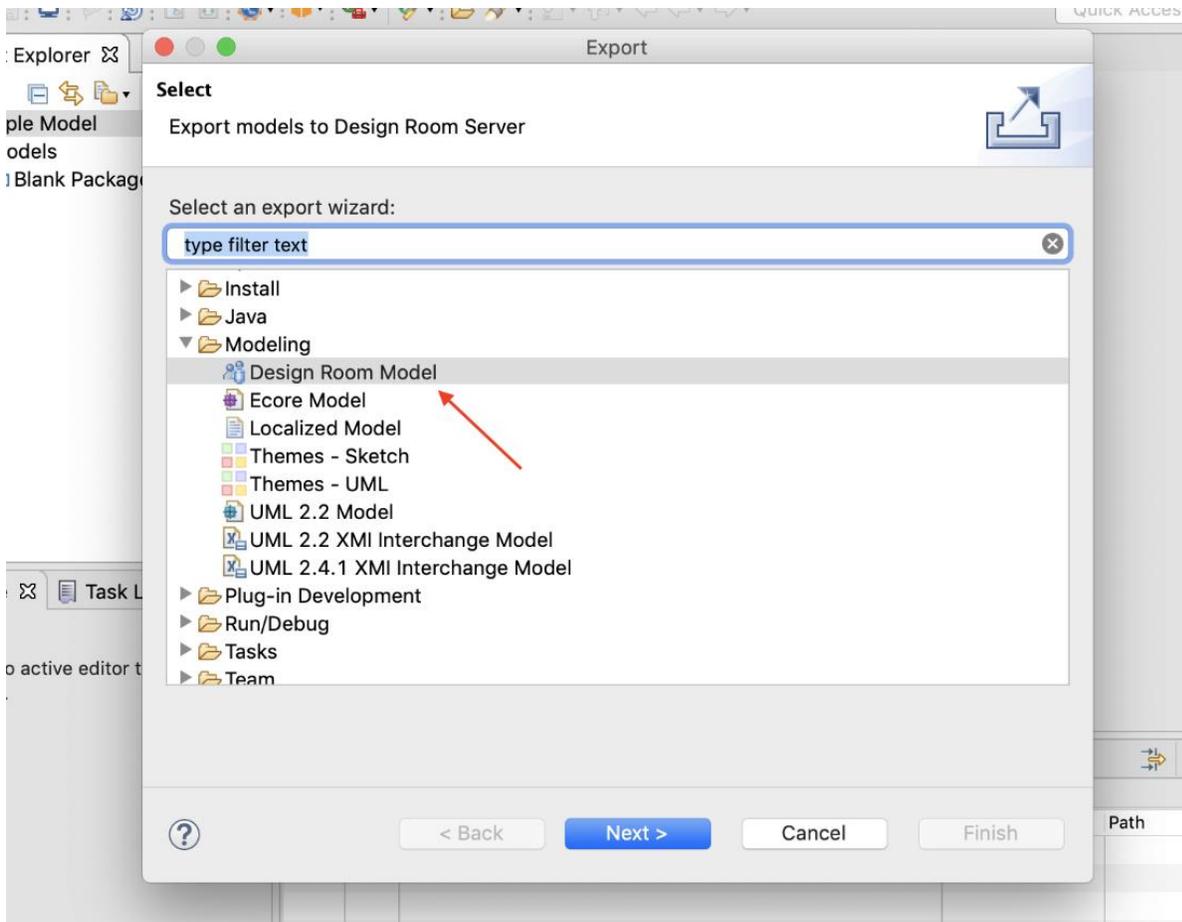
Prerequisites:

1. Install the Design Room ONE Integration feature in your modeling software by following the steps in the Design Room ONE installation document
2. Then ensure that the Design Room ONE server is started successfully.

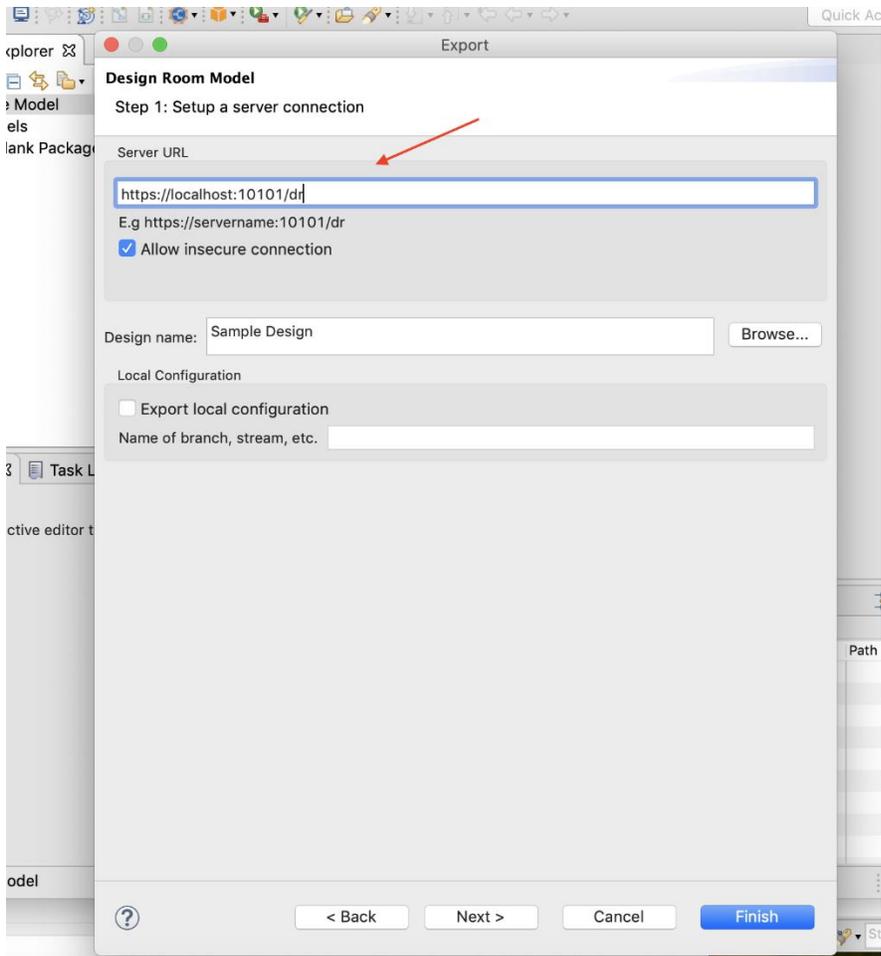
Right click on a project you want to export and select **Export**



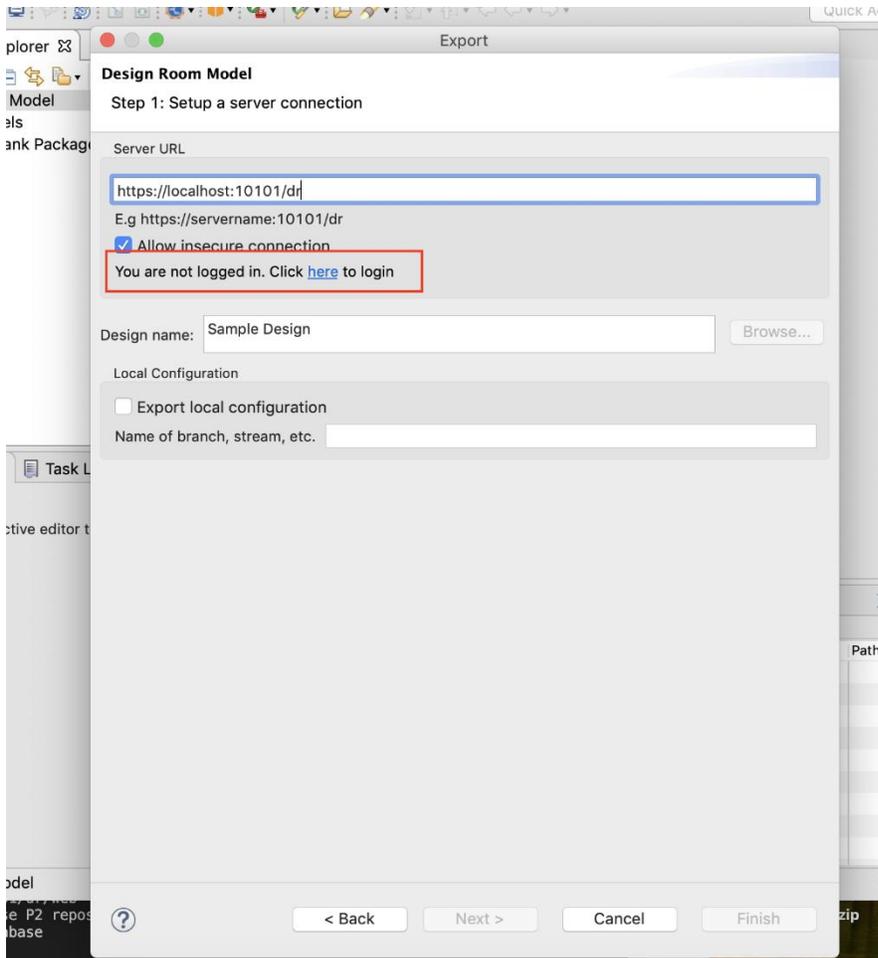
Under the **Modeling** folder, select **Design Room Model**



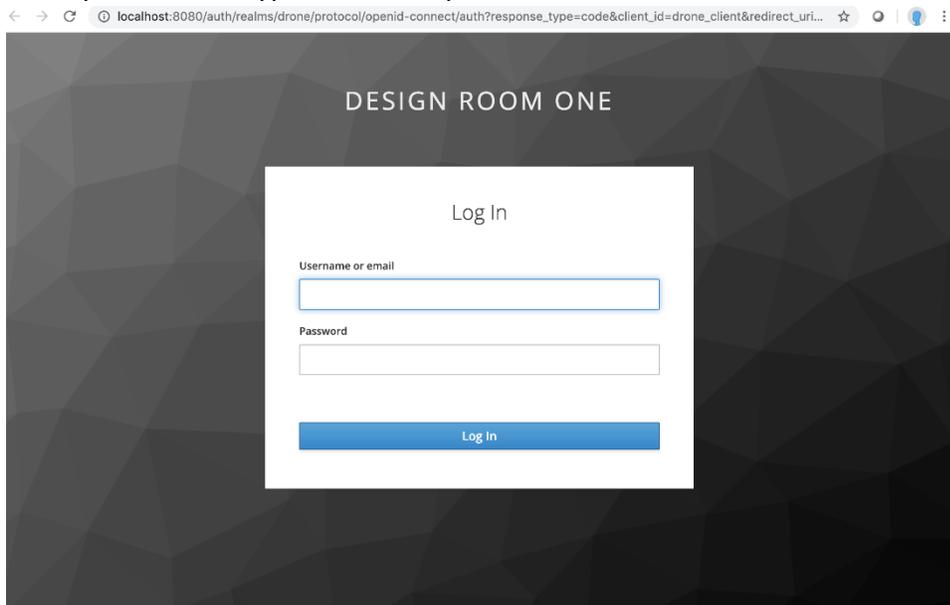
Then enter the server URL for your Design Room ONE server.



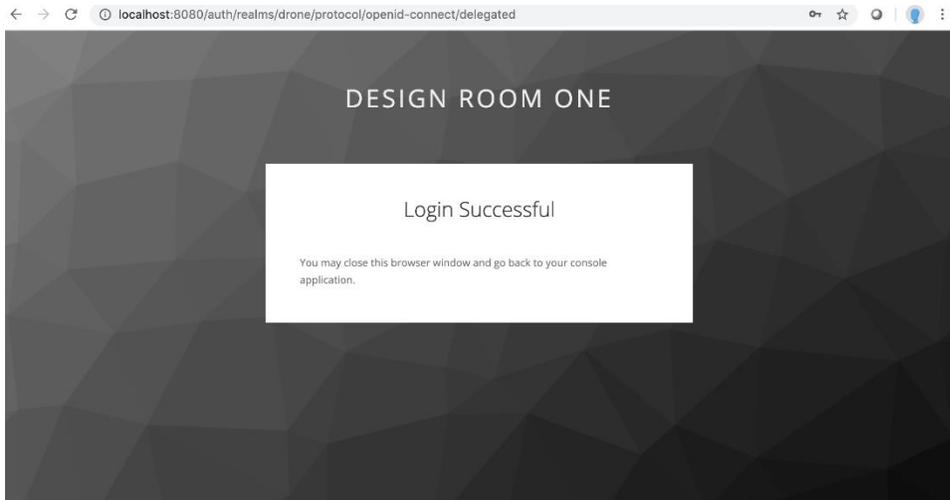
If authentication is enabled on the server and the URL is valid, you will see the message with the a link to login.



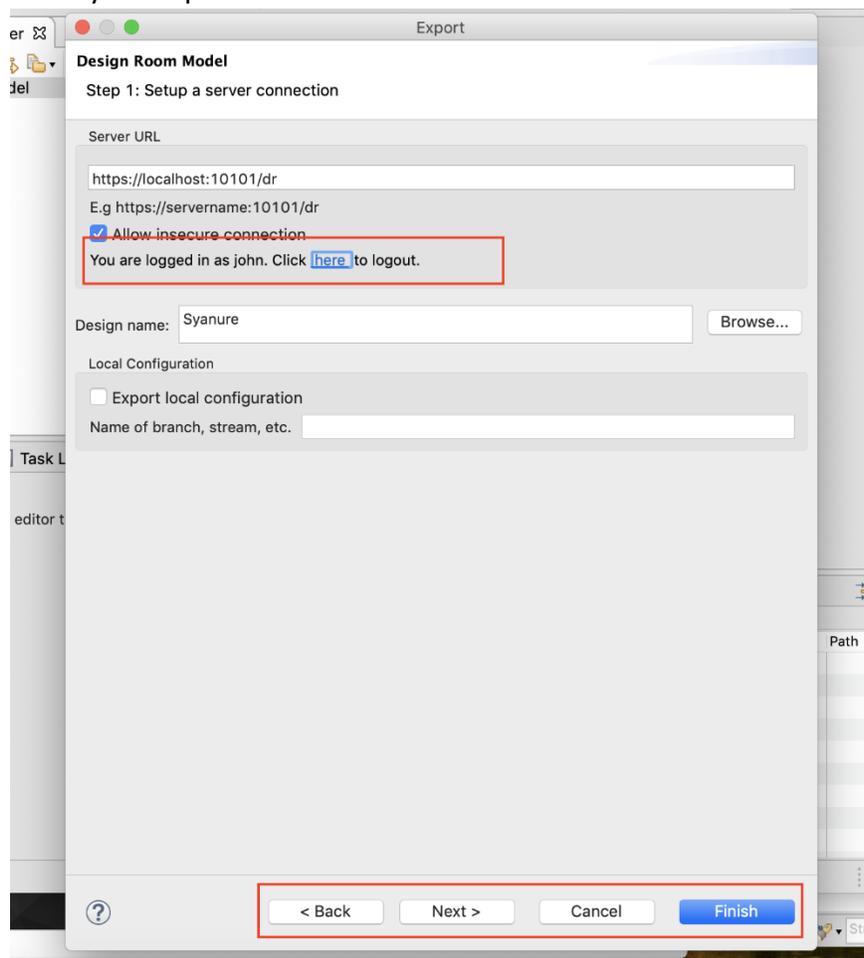
Once you click on hyperlink **here**, you will be redirected to the native browser to login



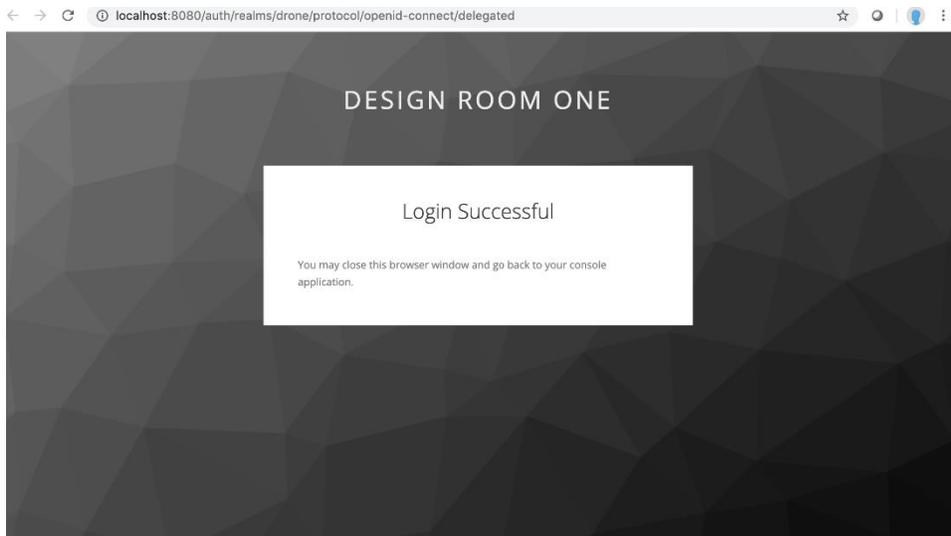
After a successful login, you will see the message



Once you go back to your modeling tool, you will see that you are logged in and can proceed to make your export



To logout, you can click on the **here** link again and the window below will open in your native browser.



Known Limitations

1. Dockerized setup instructions to be provided in a later delivery
2. Automated export scenarios do not support authentication yet.
3. After clicking on **logout** link in the exporting wizard in a modeling tool a web browser with a message "Login Successful" appears, the message should say "Logout Successful"