



Design Room ONE

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 integrating with Keycloak.

Table of Contents

<i>Keycloak Server Installation</i>	<i>3</i>
Downloading Keycloak.....	3
Configuring port and hostname.....	3
<i>Keycloak Server Configuration.....</i>	<i>4</i>
Initial Startup of Keycloak Server	4
Creating a Realm Administrator	4
Starting Keycloak Server	4
Importing Previous Configuration.....	5
Configuring the Realm.....	6
<i>Setting up Users and Roles.....</i>	<i>10</i>
Creating Users.....	10
Managing Access with Roles	12
Importing Users from Active Directory	14
<i>Starting the Design Room ONE Server</i>	<i>19</i>
Logging into the Design Room ONE Server with the New User	19
<i>Exporting Models with Design Room ONE Integration Plugin.....</i>	<i>19</i>
Prerequisites:	19
<i>Single Sign-On with Jazz Authorization Server via OpenID</i>	<i>25</i>
Prerequisites	25
Creating a New Identify Provider in Keycloak	25
Setting up JAS Configuration Security.....	27
Creating a Relying Party Application in JAS.....	28
Creating a New Identify Provider in Keycloak Continued	30
Logging in with JAS Authentication	32
<i>Known Limitations.....</i>	<i>33</i>

Keycloak Server Installation

The instructions below use the following variables that need to be replaced with their actual values

DRONE_HOST_NAME e.g. drone.mycomp.any

KEYCLOAK_HOST_NAME e.g. keycloakhost.mycomp.any

KEYCLOAK_IP_ADDRESS e.g. 10.20.30.40

KEYCLOAK_INSTALL_DIR e.g. C:\Install\Keycloak

If Keycloak and Design Room ONE are installed on different machines, these machines should be able to communicate with HTTP/HTTPS requests. This usually means adjusting firewall settings to allow such communications and ensuring all host names can be successfully resolved on every machine.

Downloading Keycloak

Keycloak can be downloaded from the following site:

<https://www.keycloak.org/downloads-archive.html>

Check System Requirements document for supported versions.

Unzip the downloaded file into an installation directory of your choice. We will refer to this installation directory as KEYCLOAK_INSTALL_DIR.

Configuring port and hostname

The script we will use by default is configured with the following file

KEYCLOAK_INSTALL_DIR/standalone/configuration/standalone.xml

```
<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>
```

By default, Keycloak will use 8080 as an http port, 8443 as an https port and localhost as host. If you decided use custom KEYCLOAK_HOST_NAME, it should be specified as shown in bold below.

```
<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="keycloak.mycomp.any">
        <location name="/" handler="welcome-content"/>
        <http-invoker security-realm="ApplicationRealm"/>
    </host>
</server>

```

Keycloak Server Configuration

Initial Startup of Keycloak Server

Run the standalone version of the server in KEYCLOAK_INSTALL_DIR/bin

For Linux

standalone.sh

For Windows

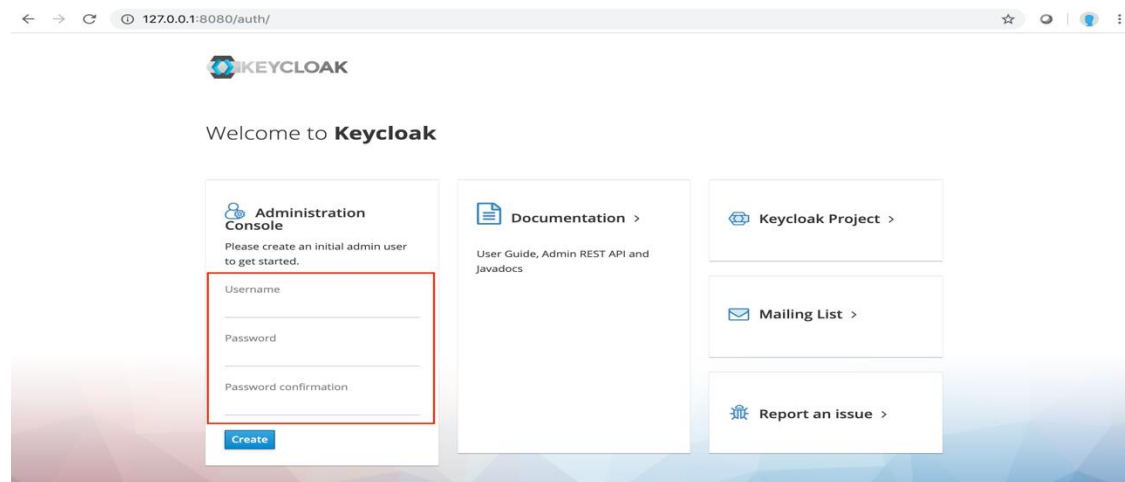
standalone.bat

Creating a Realm Administrator

Note: This step must be performed on a machine running Keycloak.

Open a browser and navigate to https://KEYCLOAK_HOST_NAME:8443/auth (or HTTP endpoint).

Note: the browser may show a warning if e.g. certificate does not match the host name



Specify the admin credentials and press Create.

This admin user is a super user with all full access to Keycloak (realm creation, update, deletion, user creation, update, deletion, etc)

Starting Keycloak Server

Stop the Keycloak server and start it again with -b parameter.

Run the standalone version of the server in KEYCLOAK_INSTALL_DIR/bin

For Linux

```
standalone.sh -b KEYCLOAK_IP_ADDRESS
```

For Windows

```
standalone.bat -b KEYCLOAK_IP_ADDRESS
```

It is possible to use 0.0.0.0 instead KEYCLOAK_IP_ADDRESS to allow connection from any interface, by default only connections from the same machine will be accepted. See Keycloak [documentation](#) for details.

Importing Previous Configuration

If you configured drone realm in the version 2.0 of Design Room ONE you need to perform the following steps to keep your configuration, i.e. users and roles. If you are doing a fresh install of Design Room ONE proceed with the next section.

1. Export your current realm data
 - a. Ensure your keycloak server instance is stopped
 - b. Run the below command to export your current drone Keycloak realm data

Run the command below:

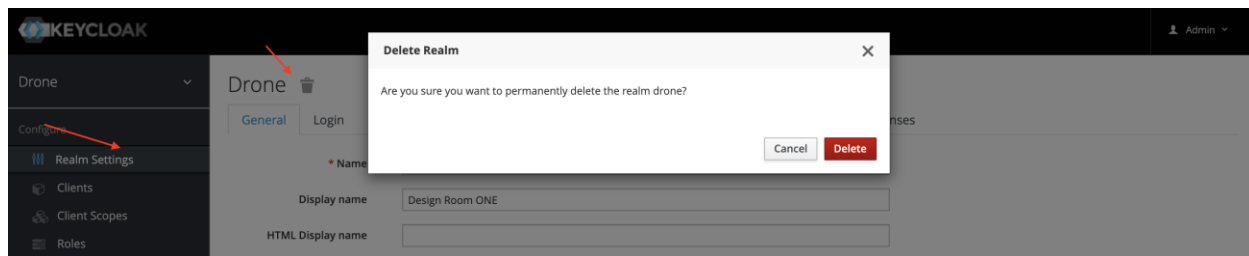
For Linux

```
KEYCLOAK_INSTALL_DIR/bin/standalone.sh -Dkeycloak.migration.action=export  
-Dkeycloak.migration.provider=singleFile  
-Dkeycloak.migration.file=/tmp/myOldRealm.json
```

For Windows

```
KEYCLOAK_INSTALL_DIR\bin\standalone.bat -Dkeycloak.migration.action=export  
-Dkeycloak.migration.provider=singleFile  
-Dkeycloak.migration.file={path_to_temp_folder}\myOldRealm.json
```

2. Delete existing drone realm
 - a. Start your Keycloak server instance
 - b. Delete the DRONE realm as shown below



3. Import Design Room ONE provided sample realm

Run the following command

For Linux

```
KEYCLOAK_INSTALL_DIR/bin/standalone.sh -Dkeycloak.migration.action=import  
-Dkeycloak.migration.provider=singleFile -Dkeycloak.migration.file=  
DR_ONE_INSTALL_DIR/DR_Install/Resources/Keycloak/drone-realm-export.json
```

For Windows

```
KEYCLOAK_INSTALL_DIR\bin\standalone.bat -Dkeycloak.migration.action=import  
-Dkeycloak.migration.provider=singleFile -Dkeycloak.migration.file=  
DR_ONE_INSTALL_DIR\DR_Install\Resources\Keycloak\drone-realm-export.json
```

```
all/Resources/Keycloak/drone-realm-export.json
18:52:06,121 INFO [org.keycloak.exportimport.util.ImportUtils] (ServerService Thread Pool -- 68) Realm 'drone' imported
18:52:06,157 INFO [org.keycloak.services] (ServerService Thread Pool -- 68) KC-SERVICES0032: Import finished successfully
18:52:06,191 INFO [org.jboss.resteasy.resteasy_jaxrs.i18n] (ServerService Thread Pool -- 68) RESTEASY002225: Deploying javax.ws.rs.core.A
tion
```

The Keycloak server will start, you should stop it again.

4. Import your exported realm

Run the command below:

For Linux

```
KEYCLOAK_INSTALL_DIR/bin/standalone.sh -Dkeycloak.migration.action=import
-Dkeycloak.migration.provider=singleFile -
Dkeycloak.migration.file=/tmp/myOldRealm.json
```

For Windows

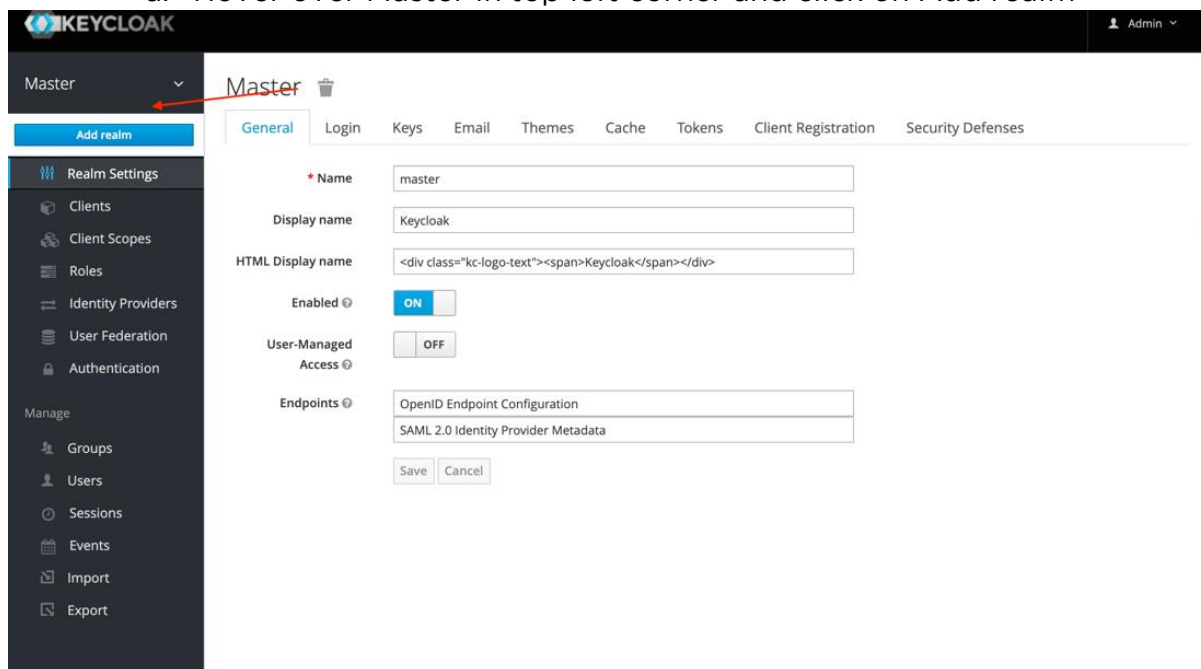
```
KEYCLOAK_INSTALL_DIR\bin\standalone.bat -Dkeycloak.migration.action=import
-Dkeycloak.migration.provider=singleFile -Dkeycloak.migration.file=
{path_to_temp_folder}\myOldRealm.json
```

5. Start the Keycloak server, see [this section](#) for details.
6. Ensure that all previously created users have view-realm role as explained in [creating users section](#)
7. Realm configuration is now complete, you should skip instructions in the next section and proceed with [Setting up Users and Roles](#).

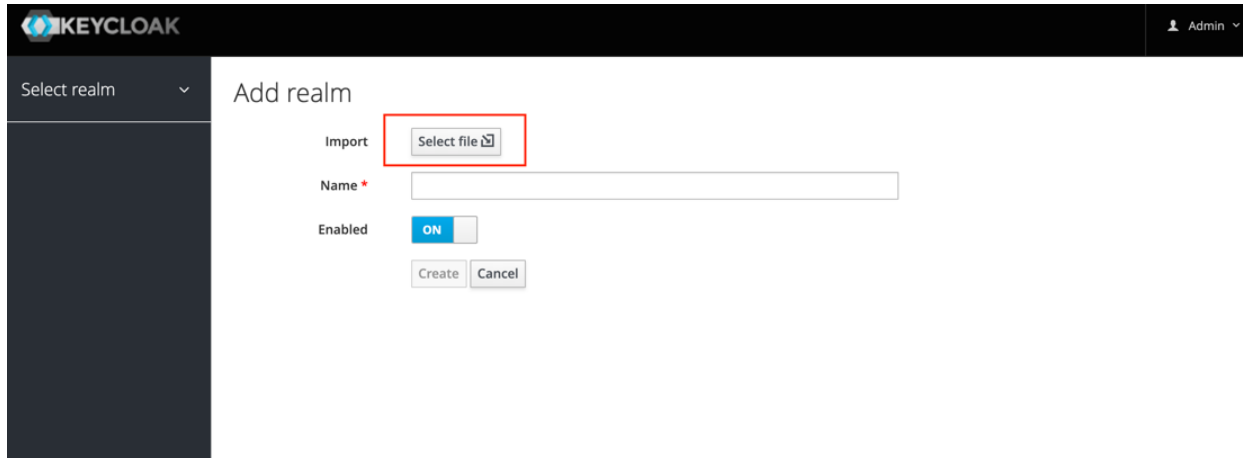
Configuring the Realm

Note: this step is easier to do from the machine where Design Room ONE is installed

1. Import the realm data
 - a. Hover over Master in top left corner and click on Add realm

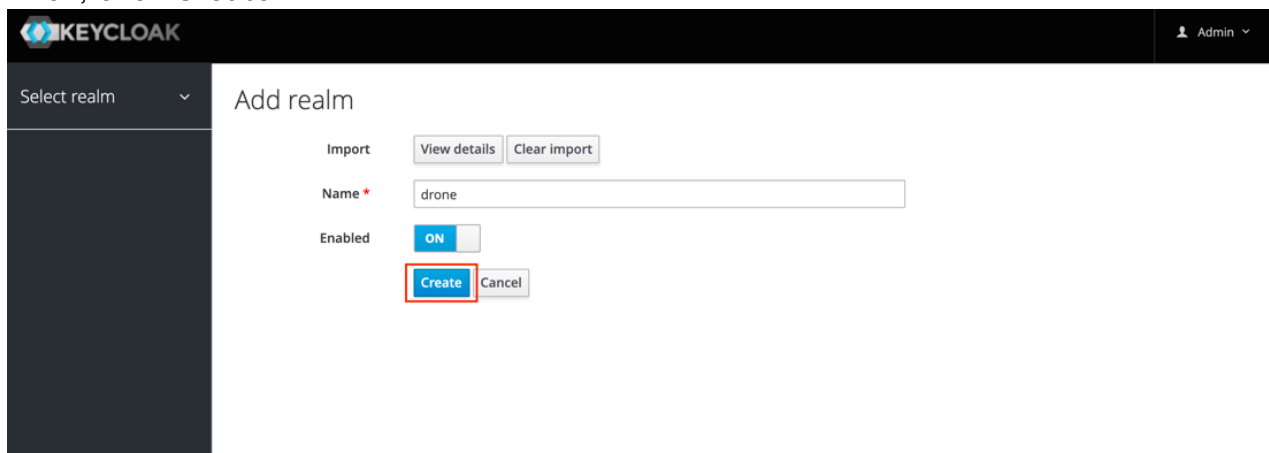


You should see a page similar to the one below.



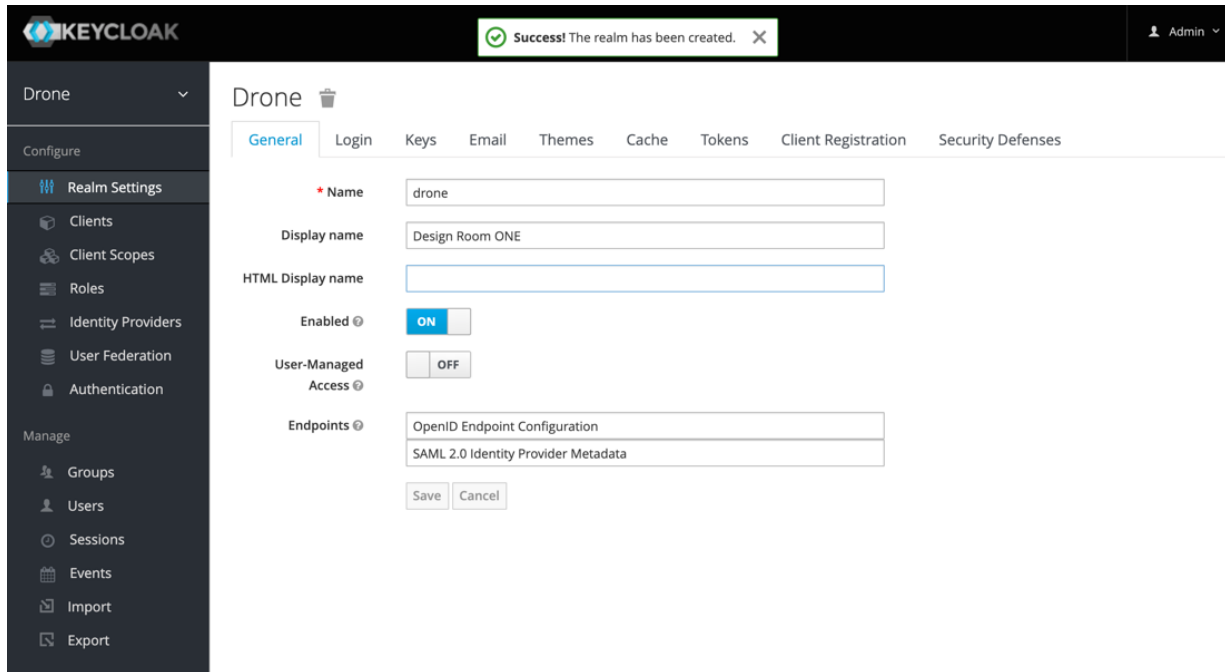
The screenshot shows the Keycloak administration interface. At the top, there is a dark header with the Keycloak logo and the text 'KEYCLOAK' on the left, and a user profile 'Admin' with a dropdown arrow on the right. Below the header, on the left, is a sidebar with a 'Select realm' dropdown menu. The main content area is titled 'Add realm'. It contains an 'Import' section with a 'Select file' button that has a file icon, which is highlighted with a red rectangle. Below this is a 'Name' field with a red asterisk, followed by an 'Enabled' toggle switch set to 'ON'. At the bottom of the form are 'Create' and 'Cancel' buttons.

Click on Select File and point to drone-realm-export.json file. It can be found in machine where Design Room ONE is installed under following path:
DR_ONE_INSTALL_DIR/DR_Install/Resources/Keycloak/drone-realm-export.json
Then, click Create



This screenshot shows the 'Add realm' form after the file has been selected. The 'Import' section now includes 'View details' and 'Clear import' buttons. The 'Name' field is populated with the text 'drone'. The 'Enabled' toggle switch remains 'ON'. The 'Create' button is now highlighted with a red rectangle, indicating it should be clicked to complete the process.

Your Drone realm should be created successfully as shown below:



KEYCLOAK Success! The realm has been created. Admin

Drone

Configure

Realm Settings

Clients

Client Scopes

Roles

Identity Providers

User Federation

Authentication

Manage

Groups

Users

Sessions

Events

Import

Export

Drone

General Login Keys Email Themes Cache Tokens Client Registration Security Defenses

Name drone

Display name Design Room ONE

HTML Display name

Enabled ON

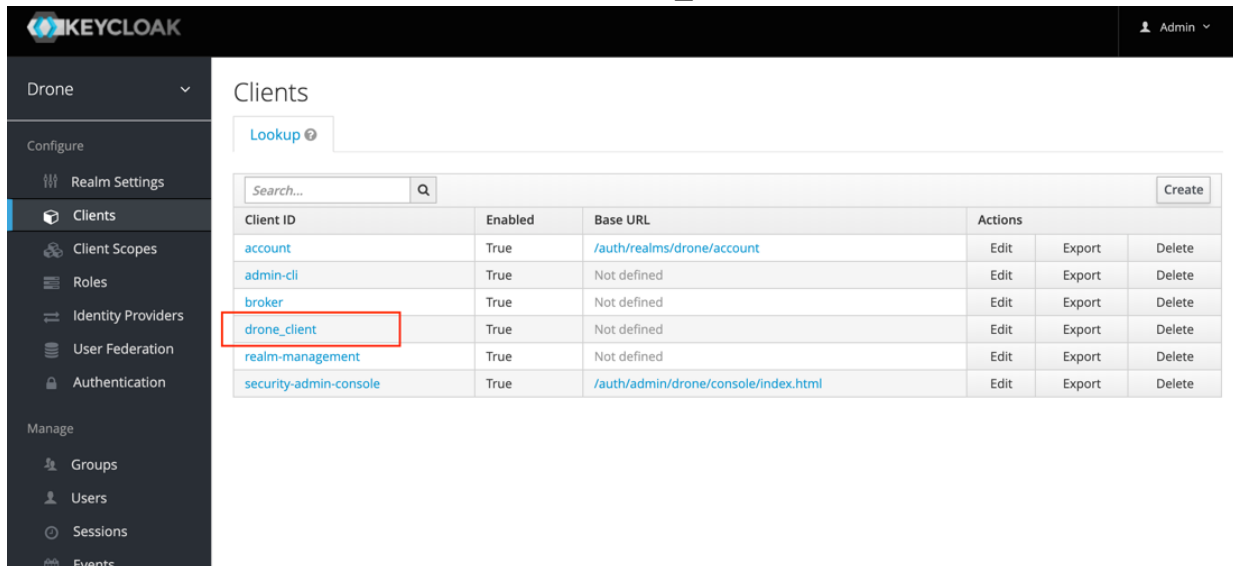
User-Managed Access OFF

Endpoints OpenID Endpoint Configuration SAML 2.0 Identity Provider Metadata

Save Cancel

2. Setup drone_client

- Under Clients click on the drone_client link



KEYCLOAK Admin

Drone

Configure

Realm Settings

Clients

Client Scopes

Roles

Identity Providers

User Federation

Authentication

Manage

Groups

Users

Sessions

Events

Clients

Lookup

Search...

Create

Client ID	Enabled	Base URL	Actions
account	True	/auth/realms/drone/account	Edit Export Delete
admin-cli	True	Not defined	Edit Export Delete
broker	True	Not defined	Edit Export Delete
drone_client	True	Not defined	Edit Export Delete
realm-management	True	Not defined	Edit Export Delete
security-admin-console	True	/auth/admin/drone/console/index.html	Edit Export Delete

- Click on the Settings top menu, scroll down and add DRONE_HOST_NAME (or ip address) accessible by other machines for the server Keycloak as valid redirect URIs as in example below. It is recommended to use lowercase letters. Keep localhost in the list valid URLs since these URLs are used when models being exported to Design Room ONE Server.

Root URL ?	<input type="text" value="https://drone.mycomp.any:10101/dr/web"/>	
* Valid Redirect URIs ?	<input type="text" value="https://drone.mycomp.any:10101/dr/*"/> - <input type="text" value="http://localhost:*"/> - <input type="text" value="https://localhost:*"/> - <input type="text"/> +	
Base URL ?	<input type="text"/>	
Admin URL ?	<input type="text" value="https://drone.mycomp.any:10101/dr/web"/>	
Web Origins ?	<input type="text" value="https://drone.mycomp.any:10101/dr/web"/> - <input type="text"/> +	

- Make sure Root URL, Admin URL, and Web Origins fields are also updated with DRONE_HOST_NAME and hit Save
- Under Installation menu select Keycloak OIDC JSON format

Drone_client

[Settings](#)
[Roles](#)
[Client Scopes ?](#)
[Mappers ?](#)
[Scope ?](#)
[Revocation](#)
[Sessions ?](#)

[Offline Access ?](#)
[Installation ?](#)

Format Option

Download

```

{
  "realm": "drone",
  "auth-server-url": "https://keycloak.mycomp.any:8443/auth/",
  "ssl-required": "external",
  "resource": "drone_client",
  "public-client": true,
  "confidential-port": 0
}

```

Copy and paste auth-server-url and ssl-required properties in the file DR_ONE_INSTALL_DIR/OnPrem_Design_Room/config/server-config.json under dr_keycloak_config object. Also change "dr_auth" attribute value to "keycloak". Then the configuration file would look similar to this.

```
// 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",

//Keycloak configuration
"dr_keycloak_config": {
  "auth-server-url": "https://drone.mycomp.any:8443/auth",
  //Defines security level for Keycloak server
  //"none": HTTPS not required for any IP address
  //"external": Private IP and localhost can access without HTTPS
  //"all": HTTPS required for all IP addresses
  "ssl-required": "external"
}
```

Note: Make sure correct KEYCLOAK_HOST_NAME (or ip-address) specified in auth-server-url attribute and the URL is accessible by machine where Design Room ONE server is installed and from user machines. It is recommended to use lowercase letters in the URLs.

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.

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: Under the Default Roles tab, as shown below, you will notice that drone_user is a default role along with the view-realm client role. Default roles are automatically assigned to newly created users. These roles are required by the Design Room ONE server normal operation.

Drone

Configure

- Realm Settings
- Clients
- Client Scopes
- Roles**
- Identity Providers
- User Federation
- Authentication

Manage

- Groups
- Users
- Sessions

Roles

Realm Roles **Default Roles**

Realm Roles

Available Roles ⓘ

- sample_all_access
- sample_partial_access

Add selected >

Client Roles

realm-management

Available Roles ⓘ

- create-client
- impersonation
- manage-authorization
- manage-clients
- manage-events

Add selected >

Realm Default Roles ⓘ

- drone_user
- offline_access
- uma_authorization

<< Remove selected

Client Default Roles ⓘ

- view-realm

<< Remove selected

1. Create a user

Drone

Configure

- Realm Settings
- Clients
- Client Scopes
- Roles
- Identity Providers
- User Federation
- Authentication

Manage

- Groups
- Users**
- Sessions
- Events
- Import
- Export

Users

Lookup

Search... Q View all users

Unlock users Add user

Please enter a search, or click on view all users

Drone

Configure

- Realm Settings
- Clients
- Client Scopes
- Roles
- Identity Providers
- User Federation
- Authentication

Manage

- Groups
- Users**
- Sessions
- Events
- Import
- Export

Users > Add user

Add user

ID

Created At

Username * john

Email john.doe@example.com

First Name John

Last Name Doe

User Enabled ⓘ ON

Email Verified ⓘ OFF

Required User Actions ⓘ Select an action...

Save Cancel

2. Setup the user password and click Reset Password

Drone

Configure

- Realm Settings
- Clients
- Client Scopes
- Roles
- Identity Providers
- User Federation
- Authentication

Manage

- Groups
- Users**
- Sessions
- Events
- Import
- Export

Users > john

John

Details Attributes **Credentials** Role Mappings Groups Consents Sessions

Manage Password

New Password: [password field]

Password Confirmation: [password field]

Temporary: ☒ ON

Reset Password

Credential Reset

Reset Actions: Select an action...

Expires In: 12 Hours

Reset Actions Email: Send email

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`.

Drone

Configure

- Realm Settings
- Clients
- Client Scopes
- Roles**
- Identity Providers
- User Federation
- Authentication

Manage

- Groups

Roles > sample_partial_access

Sample_partial_access

Details **Attributes** Users in Role

Key	Value	Actions
dr_can_read	traffic*	Delete
dr_can_write	traffic*	Delete
		Add

Save Cancel

It is possible to specify several alternatives by using `|` character, e.g. the following value

`traffic*|*_secretproject|*alice*`

will give read or write access (depending on the attribute it is specified in) to designs, which name either starts with `traffic`, or ends with `_secretproject` or contains the word `alice`. Note that, design names are case sensitive.

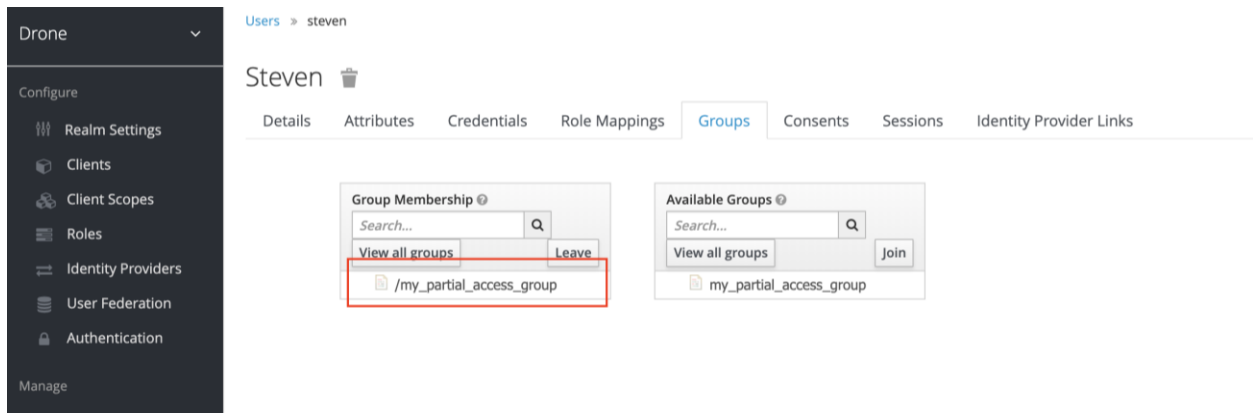
To explicitly assign a role to a user, Role Mapping page under the Users left menu can be used.

The screenshot shows the 'Role Mapping' page for user 'John'. The left sidebar is expanded to 'Users'. The breadcrumb is 'Users > john'. The page title is 'John' with a trash icon. The tabs are 'Details', 'Attributes', 'Credentials', 'Role Mappings' (selected), 'Groups', 'Consents', 'Sessions', and 'Identity Provider Links'. The main content area has four panels: 'Realm Roles' (empty), 'Available Roles' (containing 'sample_all_access'), 'Assigned Roles' (containing 'drone_user', 'offline_access', 'sample_partial_access' (highlighted), and 'uma_authorization'), and 'Effective Roles' (containing 'drone_user', 'offline_access', 'sample_partial_access', and 'uma_authorization'). A red arrow points from 'sample_all_access' in 'Available Roles' to 'sample_partial_access' in 'Assigned Roles'. Below 'Available Roles' is an 'Add selected >' button. Below 'Assigned Roles' is a '<< Remove selected' button. At the bottom, there is a 'Client Roles' section with a dropdown menu and the text 'Select client to view roles for client'.

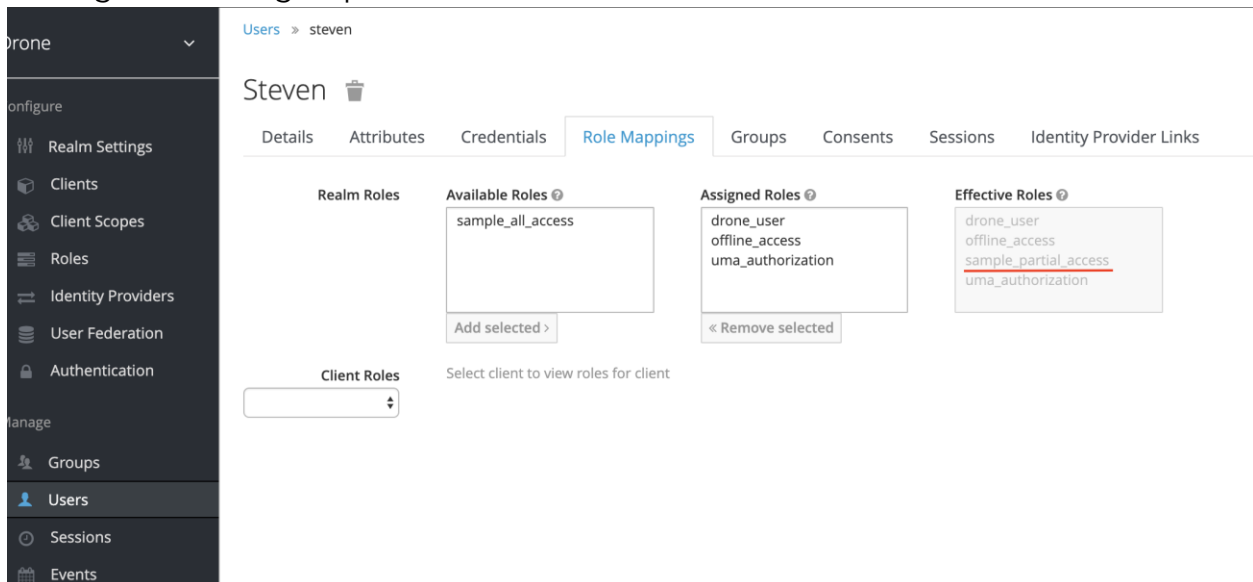
A user can implicitly get role from the groups they belong. To assign a role to a group Role Mapping page under the Groups left menu can be used.

The screenshot shows the 'Role Mapping' page for group 'My_partial_access_group'. The left sidebar is expanded to 'Groups'. The breadcrumb is 'Groups > my_partial_access_group'. The page title is 'My_partial_access_group' with a trash icon. The tabs are 'Settings', 'Attributes', 'Role Mappings' (selected), and 'Members'. The main content area has four panels: 'Realm Roles' (empty), 'Available Roles' (containing 'drone_user', 'offline_access', 'sample_all_access', and 'uma_authorization'), 'Assigned Roles' (containing 'sample_partial_access'), and 'Effective Roles' (containing 'sample_partial_access'). A red arrow points from 'sample_all_access' in 'Available Roles' to 'sample_partial_access' in 'Assigned Roles'. Below 'Available Roles' is an 'Add selected >' button. Below 'Assigned Roles' is a '<< Remove selected' button. At the bottom, there is a 'Client Roles' section with a dropdown menu and the text 'Select client to view roles for client'.

Note: A user will inherit all the roles from the groups they belong. In the example below user Steven belongs to my_partial_access_group



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

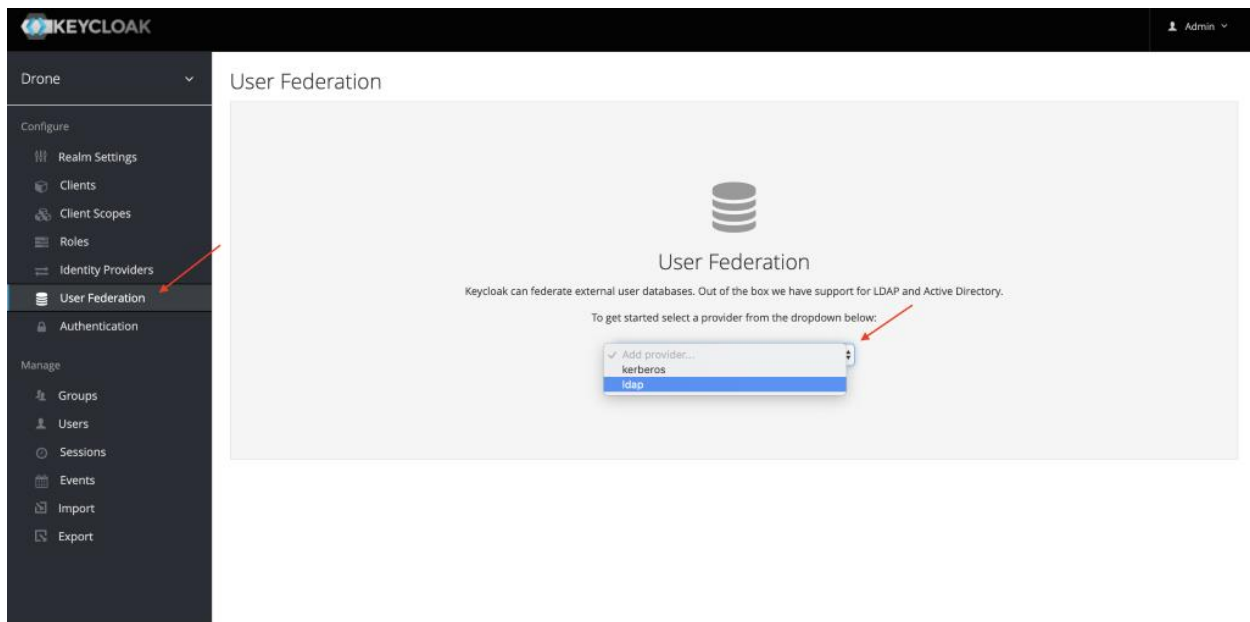


This results any my_partial_access_group member including user Steven being able to read and write all designs with names starting with “traffic” because of the group membership.

Importing Users from Active Directory

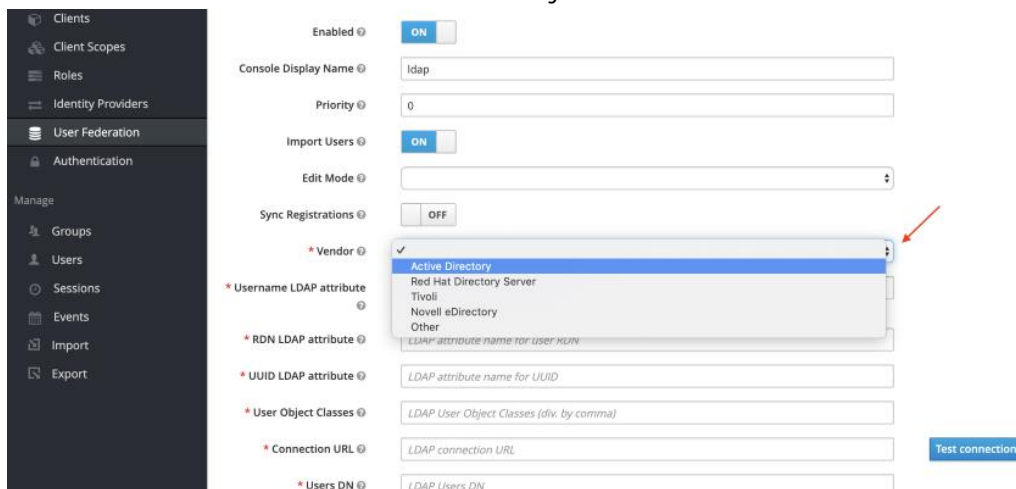
Note: The Active Directory server reference is at the end of this section

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



Note: Basic Active Directory configuration will be covered in this section. Additional configuration options are available in Keycloak with context sensitive help.

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 Test connection

* Users DN ? LDAP Users DN

* Bind Type ? simple

- 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 Test connection

* Users DN ? LDAP Users DN

* Bind Type ? simple

Enable StartTLS ? OFF

- Set the active directory users database here

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

* Bind Type ? simple

Enable StartTLS ? OFF

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

* Connection URL ? ldap://steven.ad Success! LDAP authentication successful. X Test connection

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

* Bind Type ? simple

Enable StartTLS ? OFF

* Bind DN ? cn=admin,dc=hcl,dc=com

* Bind Credential ? ***** Test authentication

Custom User LDAP Filter ? LDAP Filter

6. 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 Success! Sync of users finished successfully. 1 imported users, 0 updated users

Read Timeout

Pagination ☒ ON

Kerberos Integration

Allow Kerberos authentication ☐ OFF

Use Kerberos For Password Authentication ☐ OFF

Sync Settings

Batch Size

Periodic Full Sync ☐ OFF

Periodic Changed Users Sync ☐ OFF

Cache Settings

Cache Policy

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

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

KEYCLOAK Admin

Drone

Configure

- Realm Settings
- Clients
- Client Scopes
- Roles
- Identity Providers
- User Federation
- Authentication

Manage

- Groups
- Users**
- Sessions
- Events
- Import
- Export

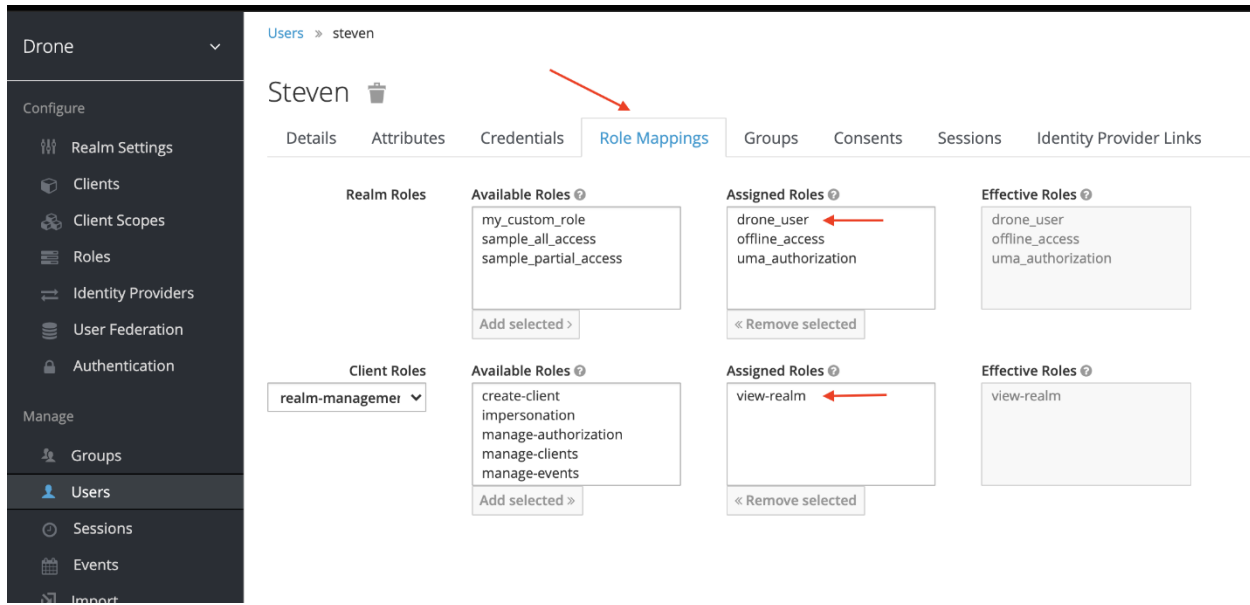
Users

Lookup

Search... View all users

ID	Username	Email	Last Name	First Name	Actions
05c2cd34-6e92-4be9-a2...	john	john.doe@example.com	Doe	John	Edit Impersonate Delete
d6937788-d841-4210-8d...	steven		S		Edit Impersonate Delete

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



9. 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

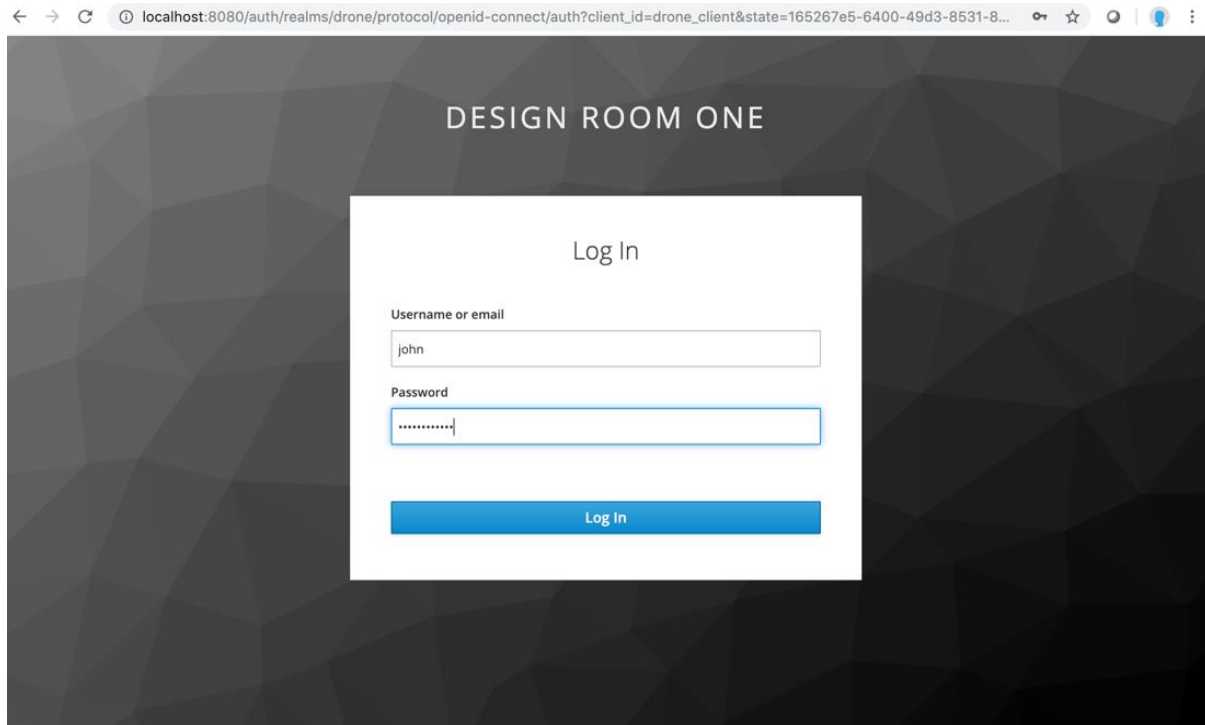
# numResponses: 5
# numEntries: 4
```

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

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.

Logging into the Design Room ONE Server with the New User

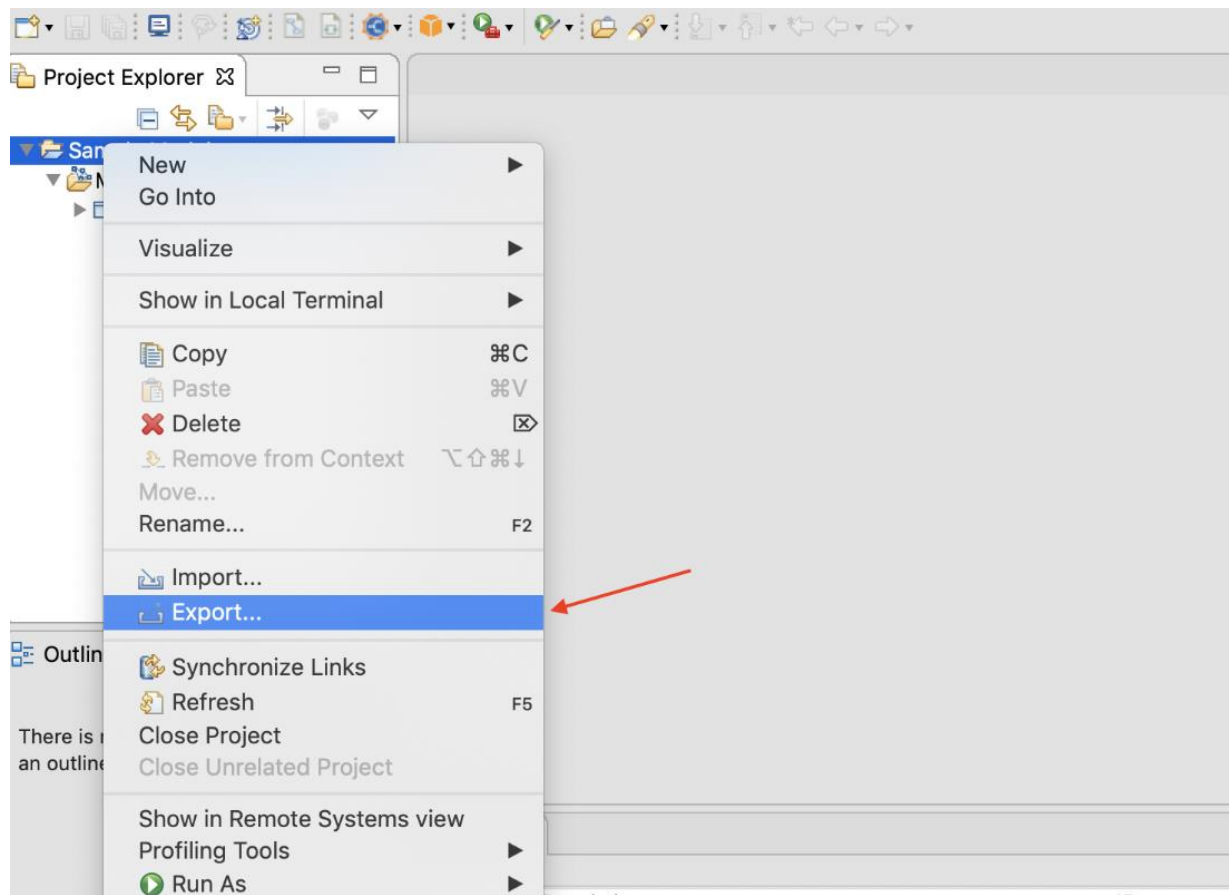


Exporting Models with Design Room ONE Integration Plugin

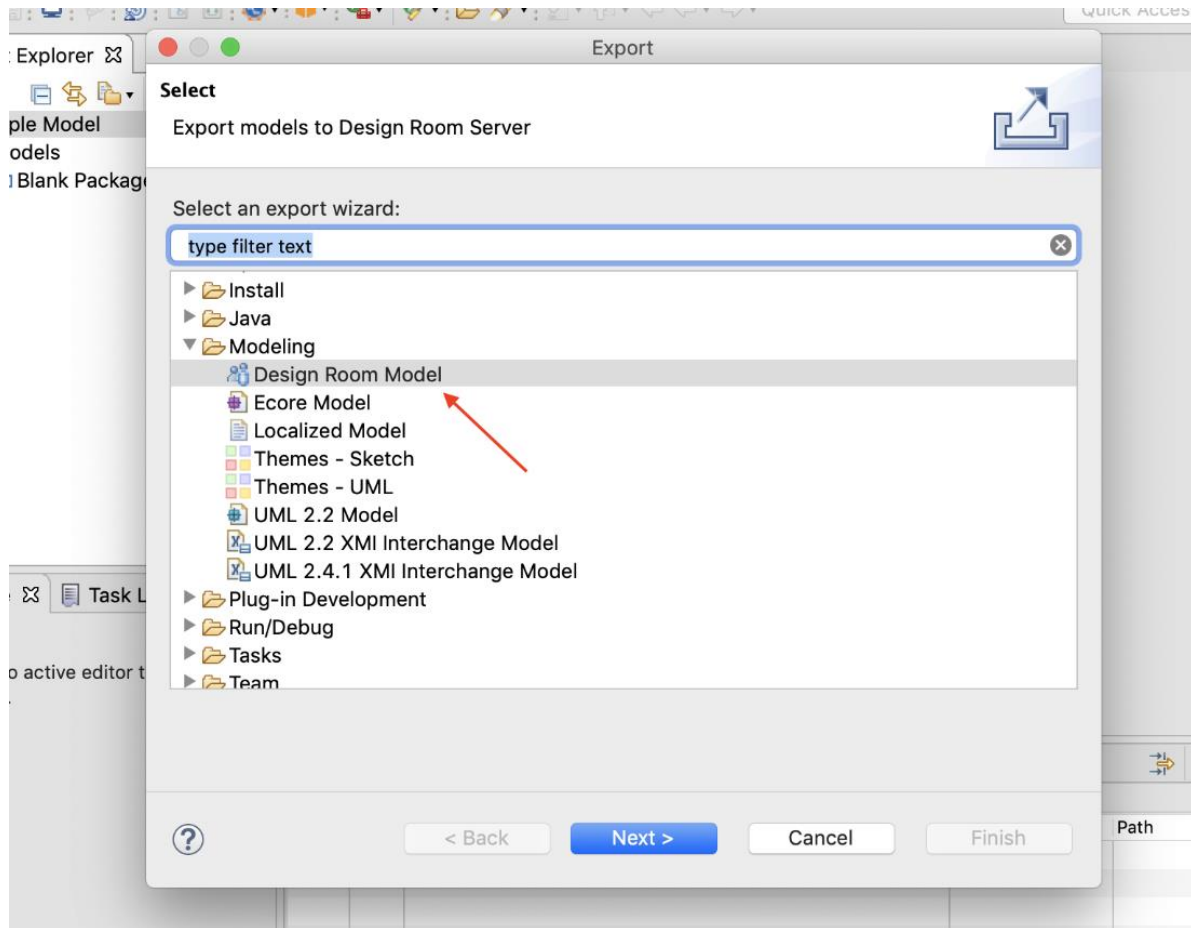
Below, we will highlight the steps to login in the eclipse client once Keycloak is enabled on the Design Room ONE server.

Prerequisites:

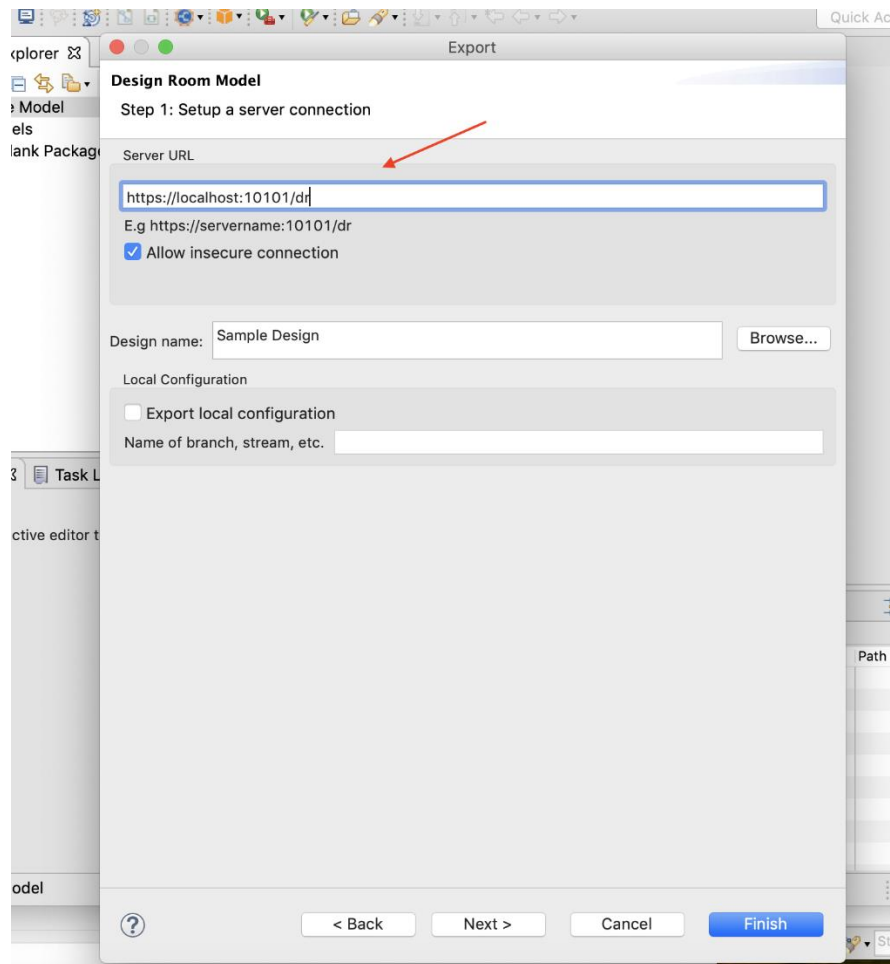
1. Install the Design Room ONE Integration feature in your modeling software by following the steps in the Design Room ONE installation document
 2. Then ensure that the Design Room ONE server is started successfully.
- Right click on a project you want to export and select Export



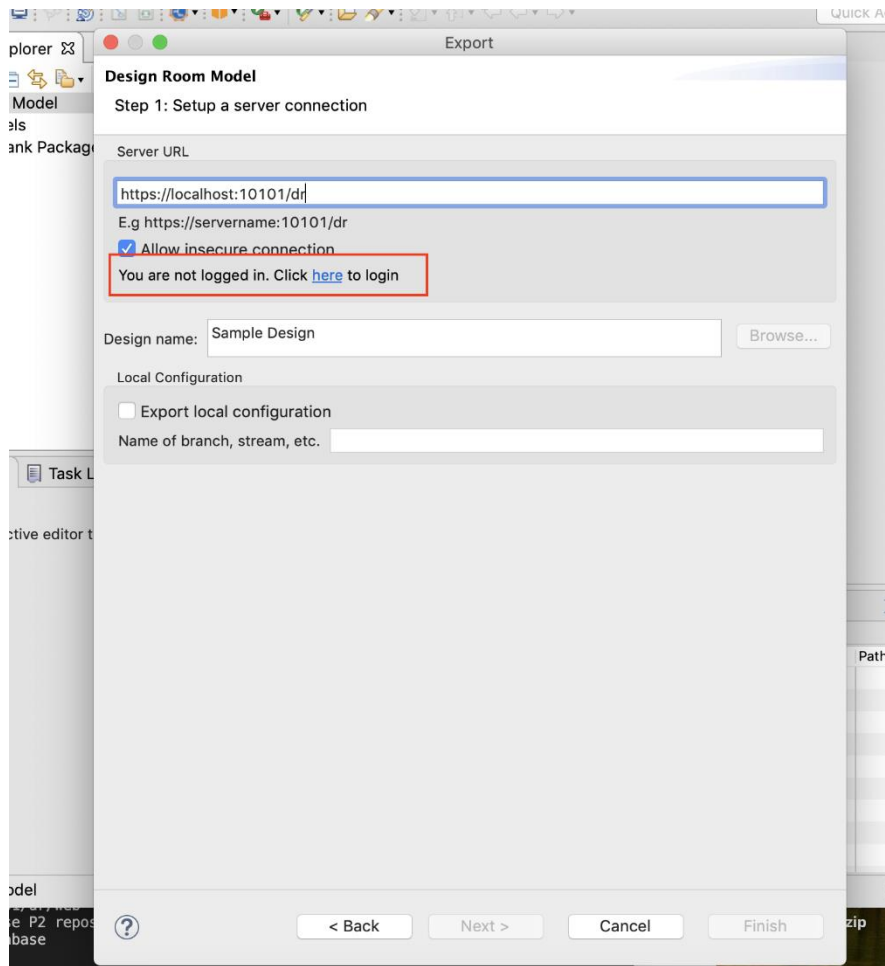
Under the Modeling folder, select Design Room Model



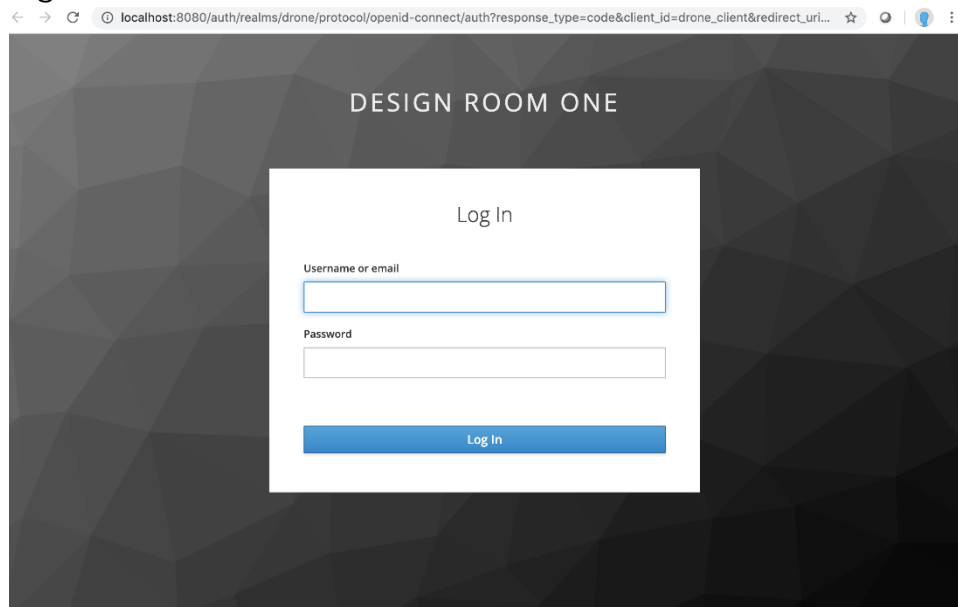
Then enter the server URL for your Design Room ONE server.



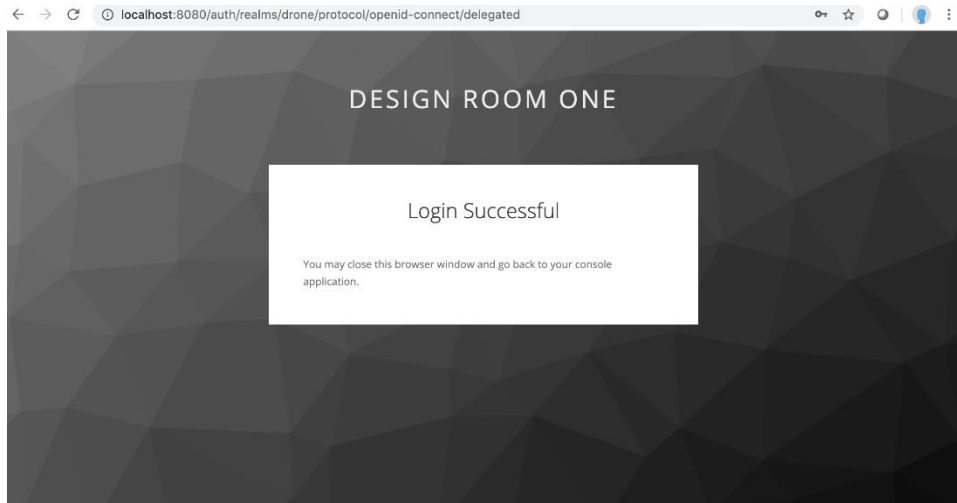
If authentication is enabled on the server and the URL is valid, you will see the message with a link to login.



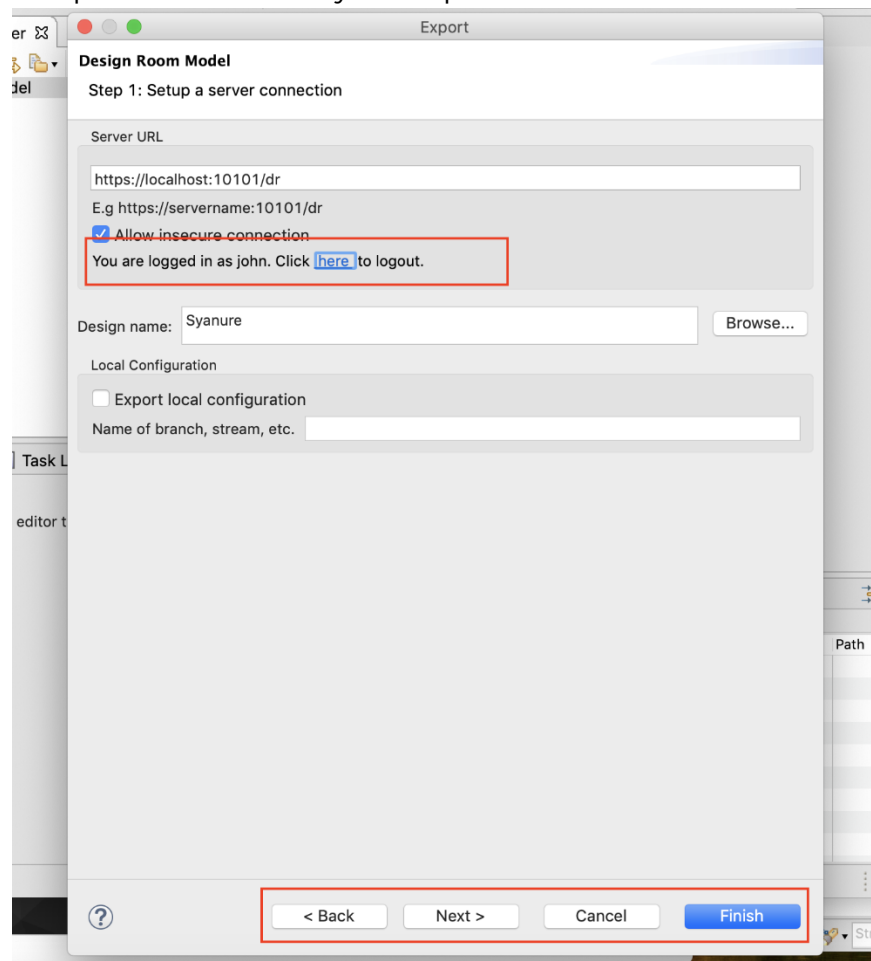
Once you click on hyperlink here, you will be redirected to the native browser to login



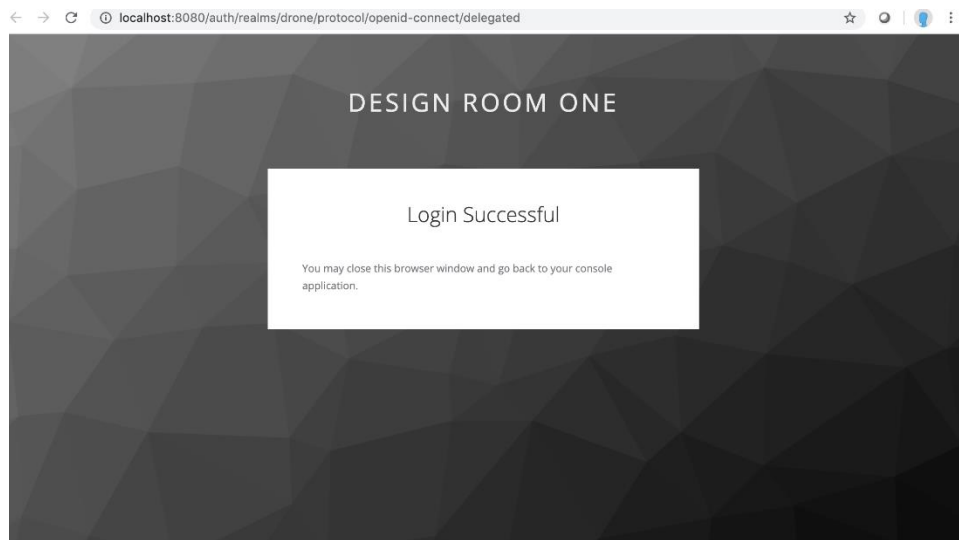
After a successful login, you will see the message



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.



Single Sign-On with Jazz Authorization Server via OpenID

Prerequisites

1. [Jazz Authorization Server](#) (JAS) is installed
2. Keycloak Server is installed

Creating a New Identify Provider in Keycloak

1. Login into Keycloak server as admin
2. Click on DRONE realm on the top left
3. Click on Identity Providers left menu open
4. Under the dropdown, select OpenID Connect v1.0 as shown below

The screenshot shows the Keycloak Admin Console interface. On the left, the 'Identity Providers' menu item is highlighted with a red box. The main area displays the 'Identity Providers' configuration page. A table lists the existing providers, and a dropdown menu for adding new providers is open, with 'OpenID Connect v1.0' selected and highlighted by a red box.

Name	Provider	Enabled	Hidden	Link only	GUI order
jAserver	oidc	True	False	False	

Add provider...

- Add provider...
- User-defined
- SAML v2.0
- OpenID Connect v1.0**
- Keycloak OpenID Connect
- Social**
- GitHub
- Twitter
- Facebook
- OpenShift v3
- Google
- GitLab
- LinkedIn
- Instagram
- Microsoft
- BitBucket
- PayPal
- StackOverflow

5. Create the alias name of your choice and use the redirect URI below to proceed in the next steps

Drone

Identity Providers > Add identity provider

Add identity provider

Redirect URI

* Alias

Display Name

Enabled ☒

Store Tokens ☐

Stored Tokens Readable ☐

Trust Email ☐

Account Linking Only ☐

Hide on Login Page ☐

GUI order

First Login Flow

Post Login Flow

OpenID Connect Config

* Authorization URL

Pass login_hint ☐

Pass current locale ☐

* Token URL

Logout URL

Backchannel Logout ☐

Disable User Info ☐

User Info URL

* Client ID

* Client Secret

6. You will notice that you are missing a few required fields such as “Authorization URL”, “Token URL”, “Client ID” and “Client Secret”. We will get these values from the jazz authorization server and complete the form later. Make note of the “Redirect URI” as we will need it in the next step.

Setting up JAS Configuration Security

If the JAS server is started with a self-signed certificate or no certificate at all, modify the `oauthProvider` element in the `appConfig.xml` file as shown below. Change the `httpsRequired` attribute to `false`.

```
<oauthProvider id="JazzOP"
  httpsRequired="false"
  autoAuthorize="true"
  customLoginURL="/jazzop/form/login"
  accessTokenLifetime="7201"
  authorizationGrantLifetime="604801">
  <autoAuthorizeClient>client01</autoAuthorizeClient>
  <databaseStore dataSourceRef="OAuthFvtDataSource" />
</oauthProvider>
```

Creating a Relying Party Application in JAS

1. Create a body.json file as shown below and make sure that you also add the above Keycloak redirect URI to the list of redirect_uris

```
{
  "token_endpoint_auth_method":"client_secret_basic",
  "scope":"openid profile email general",
  "grant_types":[
    "authorization_code",
    "client_credentials",
    "implicit",
    "refresh_token",
    "urn:ietf:params:oauth:grant-type:jwt-bearer"
  ],
  "response_types":[
    "code",
    "token",
    "id_token token"
  ],
  "application_type":"web",
  "subject_type":"public",
  "preauthorized_scope":"openid profile email general",
  "introspect_tokens":true,
  "trusted_uri_prefixes":[
    "https://keycloak.mycomp.any:*"
  ],
  "redirect_uris":[
    "http://keycloak.mycomp.any:8080/auth/realms/drone/broker/oidc/endpoint",
    "https://keycloak.mycomp.any:8443/auth/realms/drone/broker/oidc/endpoint"
  ]
}
```

Open command line and run the below command to create an application in the JAS server

```
curl --insecure --user admin:password --data @"./body.json"
http://jas.mycomp.any:9280/oidc/endpoint/jazzop/registration --header
"Content-Type: application/json"
```

2. If successful, the JAS server will respond with a response as shown below

```
{
  "client_id_issued_at":1583359157,
  "registration_client_uri":"https://localhost:9643/oidc/endpoint/jazzop/registration/d5852705ab4f4204ae29812373537277",
  "client_secret_expires_at":0,
  "token_endpoint_auth_method":"client_secret_basic",
  "scope":"openid profile email general",
  "grant_types":[
    "authorization_code",
    "client_credentials",
    "implicit",
    "refresh_token",
    "urn:ietf:params:oauth:grant-type:jwt-bearer"
  ],
  "response_types":[
    "code",
    "token",
    "id_token token"
  ],
  "application_type":"web",
  "subject_type":"public",
  "preauthorized_scope":"openid profile email general",
  "introspect_tokens":true,
  "trusted_uri_prefixes":[
    "https://localhost:*/"
  ],
  "resource_ids":[

  ],
  "client_id":"d5852705ab4f4204ae29812373537277",
  "client_secret":"p7Da1WpAxs0ujm80mFn9pN2HDVXXtDEWXJy3pIa792TNgfUyGbU7tyKXPGSd",
  "client_name":"d5852705ab4f4204ae29812373537277",
  "redirect_uris":[
    "http://localhost:8080/auth/realms/drone/broker/oidc/endpoint",
    "https://localhost:8443/auth/realms/drone/broker/oidc/endpoint"
  ],
  "allow_regexp_redirects":false
}
```

3. Note that we now have a valid client_id and client_secret to complete the creation of our identity provider application in Keycloak. Please make a note of that client_id and client_secret.
4. Browsing to the public endpoint <http://jas.mycomp.any:9280/oidc/endpoint/jazzop/.well-known/openid-configuration> will provide the remaining Authorization and token URLs of the JAS server assuming the JAS http server is started locally and on port 9280. A sample response is shown below.

```
{
  "introspection_endpoint":"http://localhost:9280/oidc/endpoint/jazzop/introspect",
  "coverage_map_endpoint":"http://localhost:9280/oidc/endpoint/jazzop/coverage_map",
  "issuer":"http://localhost:9280/oidc/endpoint/jazzop",
  "authorization_endpoint":"http://localhost:9280/oidc/endpoint/jazzop/authorize",
  "token_endpoint":"http://localhost:9280/oidc/endpoint/jazzop/token",
  "jwks_uri":"http://localhost:9280/oidc/endpoint/jazzop/jwk",
  "response_types_supported":[
    "code",
    "token",
    "id_token token"
  ],
  "subject_types_supported":[
    "public"
  ],
  "id_token_signing_alg_values_supported":[
    "HS256"
  ],
  "userinfo_endpoint":"http://localhost:9280/oidc/endpoint/jazzop/userinfo",
  "registration_endpoint":"http://localhost:9280/oidc/endpoint/jazzop/registration",
  "scopes_supported":[
    "openid",
    "general",
    "profile",
    "email",
    "address",
    "phone"
  ],
  "claims_supported":[
    "sub",
    "groupIds",
    "name",
    "preferred_username",
    "picture",
    "locale",

```

5. We can now proceed back to Keycloak admin web page and finalize the identity provider form.

Creating a New Identify Provider in Keycloak Continued

6. Fill in the Client ID, Client Secret and other fields as shown in the picture below.

OpenID Connect Config ?

* Authorization URL ?	<input type="text" value="http://jas.mycomp.any:9280/oidc/endpoint/jazzop/authorize"/>
Pass login_hint ?	<input type="checkbox"/> OFF
Pass current locale ?	<input type="checkbox"/> OFF
* Token URL ?	<input type="text" value="http://jas.mycomp.any:9280/oidc/endpoint/jazzop/token"/>
Logout URL ?	<input type="text" value="http://jas.mycomp.any:9280/oidc/endpoint/jazzop/end_session"/>
Backchannel Logout ?	<input type="checkbox"/> OFF
Disable User Info ?	<input type="checkbox"/> OFF
User Info URL ?	<input type="text"/>
* Client Authentication ?	<input type="text" value="Client secret sent as post"/> ▼
* Client ID ?	<input type="text" value="4d94c7c76fd541ab8d6248a276e40400"/>
* Client Secret ?	<input type="password" value="....."/> 👁
Issuer ?	<input type="text"/>
Default Scopes ?	<input type="text"/>
Prompt ?	<input type="text" value="unspecified"/> ▼
Accepts prompt=none forward from client ?	<input type="checkbox"/> OFF
Validate Signatures ?	<input type="checkbox"/> OFF

- Make sure Backchannel Logout setting is off. After all required fields are completed, click the Save button.
- Copy Logout URL and add it as the value "clm_end_session" property of DR_ONE_INSTALL_DIR/OnPrem_Design_Room/config/server-config.json. The property should be placed inside "dr_keycloak_config" object.

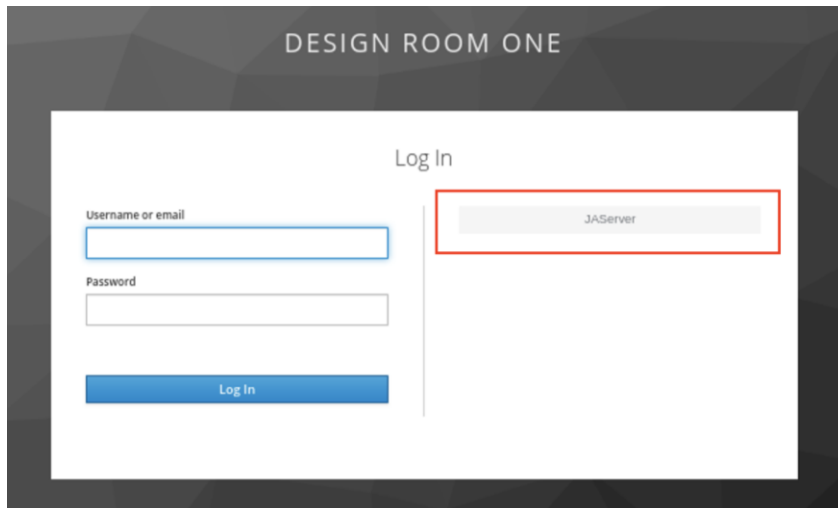
```
//Keycloak configuration
"dr_keycloak_config": {
  "auth-server-url": "https://localhost:8443/auth",
  //Defines security level for Keycloak server
  //"none": HTTPS not required for any IP address
  //"external": Private IP and localhost can access without HTTPS
  //"all" : HTTPS required for all IP addresses
  "ssl-required": "external",
  "clm_end_session": "https://jas.mycomp.any:9280/oidc/endpoint/jazzop/end_session"
```

},

9. Restart Design Room ONE server to apply the configuration changes.

Logging in with JAS Authentication

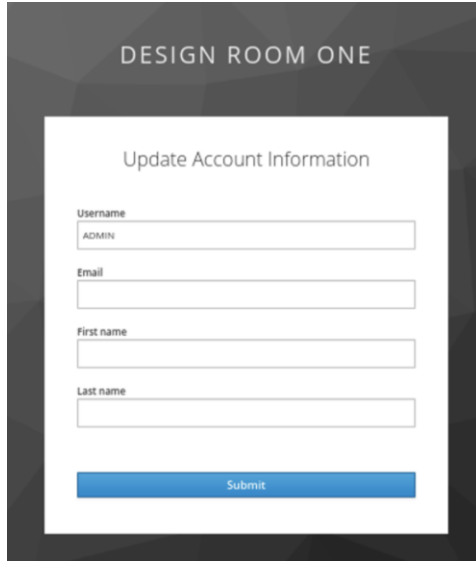
1. Ensure that Keycloak is enabled in your Design Room ONE server (i.e. server-config.json) file. Start your Design Room ONE server and navigate to some URL e.g. <https://drone.mycomp.any:10101/dr/web>. You should see the new identify provider option (JAServer) as shown here.



2. Click on JAServer option and you will be redirected to the JAS server to login



3. After a successful login you will be redirected back to the Keycloak server to add additional details for the user.



The screenshot shows a web interface titled "DESIGN ROOM ONE" at the top. Below the title is a form titled "Update Account Information". The form contains four input fields: "Username" (with the value "ADMIN" entered), "Email", "First name", and "Last name". At the bottom of the form is a blue "Submit" button.

After a successful completion of the form, you will be directed back to the Design Room ONE page you navigated.

Known Limitations

1. 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".