

KEYCLOAK

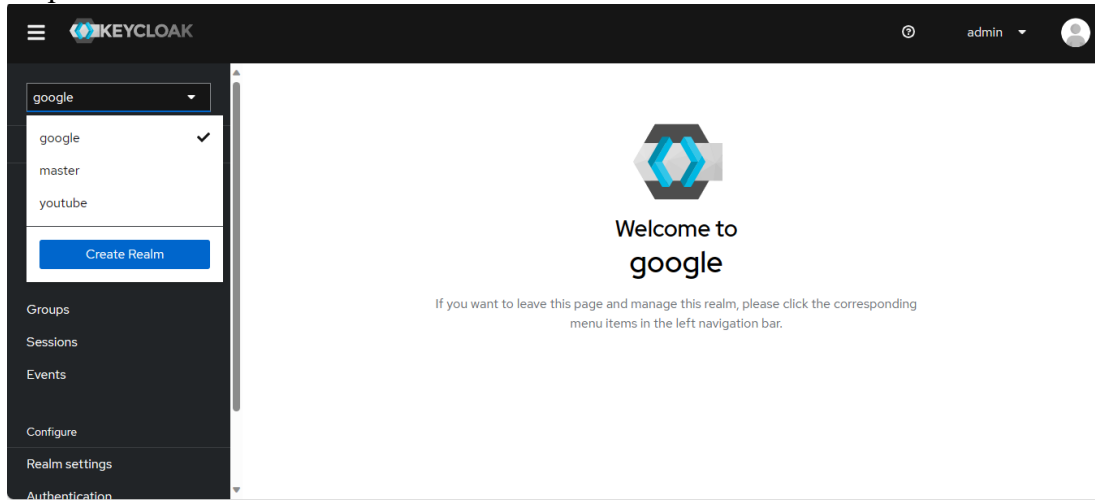
Keycloak 22 Guide by Sakib

Contents

Create Realm	3
Setup Group and Assign Realm Roles	4
Setup Group and Assign Client Roles	10

Create Realm

Step-1



Step-2

Create realm

A realm manages a set of users, credentials, roles, and groups. A user belongs to and logs into a realm. Realms are isolated from one another and can only manage and authenticate the users that they control.

Resource file

Drag a file here or browse to upload

Browse...

Clear

1

Upload a JSON file

Realm name *

myrealm

Enabled

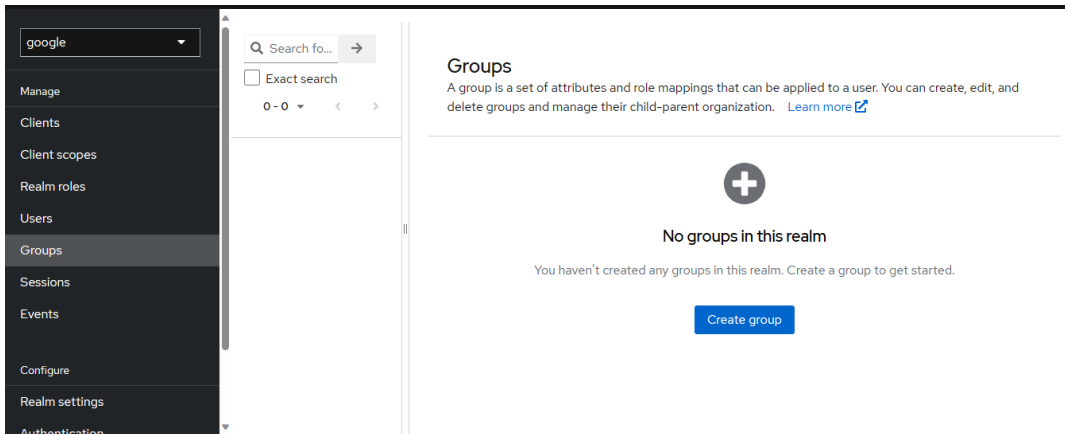
☒ On

Create

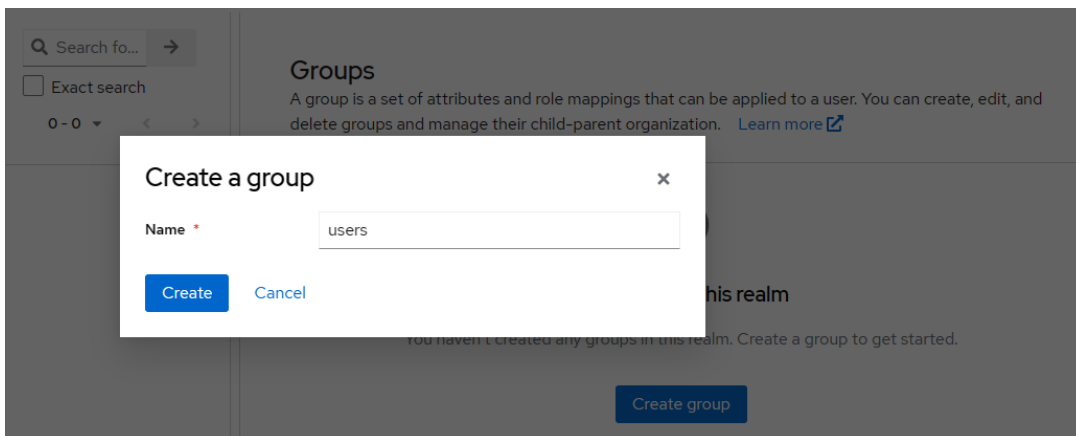
Cancel

Setup Group and Assign Realm Roles

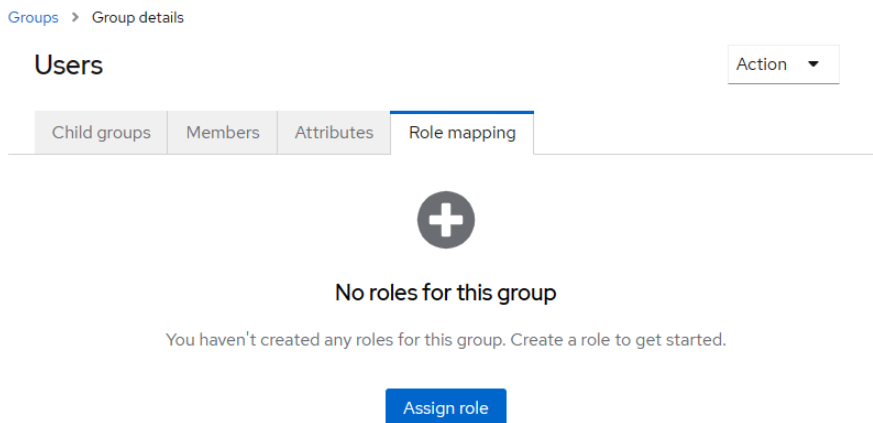
Step:1



Step:2



Step:3 - Assign roles in group



Step:4 – Adding reading post permission for this USERS group

Assign roles to users

Filter by realm roles

Search by role name

→

1 - 4

<

>

<input type="checkbox"/>	Name	Description
<input type="checkbox"/>	default-roles-google	\${role_default-roles}
<input type="checkbox"/>	offline_access	\${role_offline-access}
<input checked="" type="checkbox"/>	postread	This role is for reading post
<input type="checkbox"/>	uma_authorization	\${role_uma_authorization}

1 - 4

Assign

Cancel

Step:5 – Adding this USERS group in the client

- Go to Client>{CLIENT_NAME}>Client Scopes
- i) Go to the google-cli-dedicated (client scope name will depend on client name)

google-cli OpenID Connect Enabled ⓘ Action

Clients are applications and services that can request authentication of a user.

Settings Keys Credentials Roles Client scopes Sessions Advanced

Setup Evaluate

Name

Search by name

→

Add client scope

Change type to

⋮

1 - 10

<

>

<input type="checkbox"/>	Assigned client scope	Assigned type	Description
<input type="checkbox"/>	google-cli-dedicated	none	Dedicated scope and mappers for this client
<input type="checkbox"/>	acr	Default	OpenID Connect scope for add acr (authentication context class reference) to the token
<input type="checkbox"/>	address	Optional	OpenID Connect built-in scope: address
<input type="checkbox"/>	email	Default	OpenID Connect built-in scope: email

- ii) Click on Configure a new mapper

google-cli

This is a client scope which includes the dedicated mappers and scope

Mappers

Scope



No mappers

If you want to add mappers, please click the button below to add some predefined mappers or to configure a new mapper.

[Add predefined mapper](#)[Configure a new mapper](#)

iii) Click on Group Membership

Configure a new mapper



Choose any of the mappings from this table

Name	Description
Allowed Web Origins	Adds all allowed web origins to the 'allowed-origins' claim in the token
Audience	Add specified audience to the audience (aud) field of token
Audience Resolve	Adds all client_ids of "allowed" clients to the audience field of the token. Allowed client means the client for which user has at least one client role
Authentication Context Class Reference (ACR)	Maps the achieved LoA (Level of Authentication) to the 'acr' claim of the token
Claims parameter Token	Claims specified by Claims parameter are put into tokens.
Claims parameter with value ID Token	Claims specified by Claims parameter with value are put into an ID token.
<u>Group Membership</u>	Map user group membership
Hardcoded claim	Hardcode a claim into the token.

iv) Click on Save

Group Membership

046d100c-a4bb-426c-b269-1dfb6b817e47

Mapper type Group Membership

Name * group

Token Claim Name group

Full group path Off

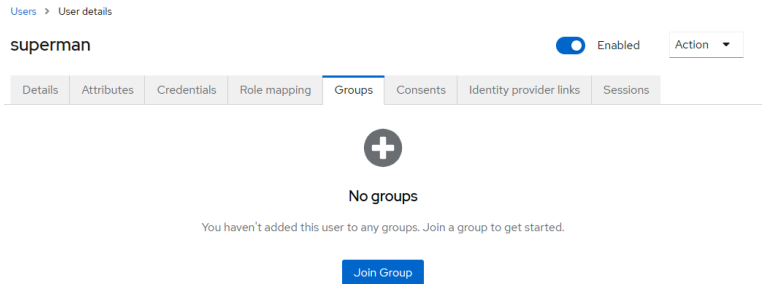
Add to ID token On

Add to access token On

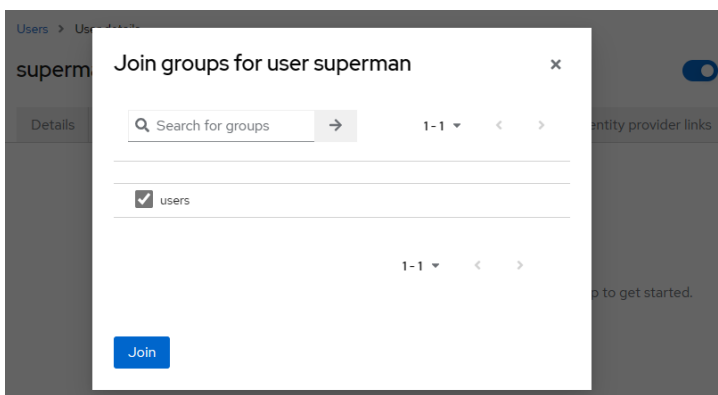
Add to userinfo On

[Save](#)[Cancel](#)

Step-6: Adding user to the group



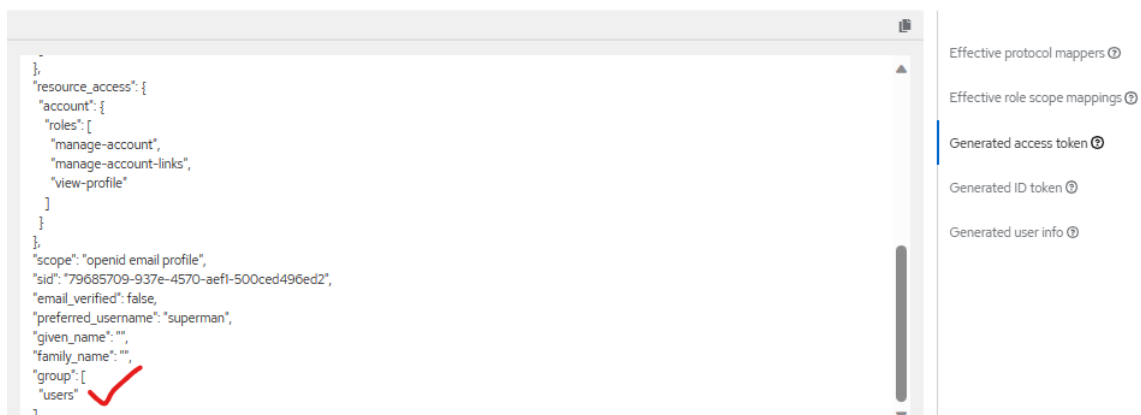
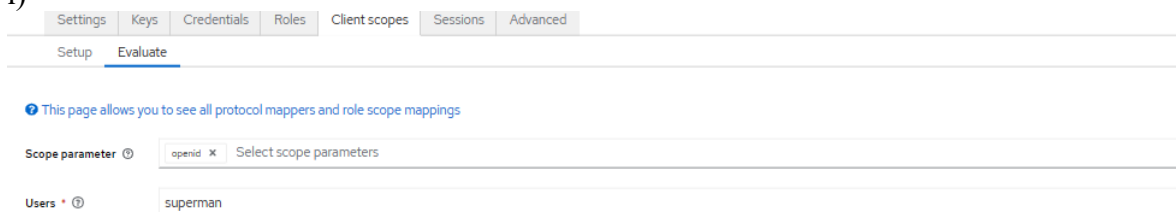
Add the user to the USERS group. Or we can add the user from group directly.



Step-7 - Let's check that user's access token

Go to Client > {CLIENT_NAME} > Client Scopes. Then Evaluate tab

i)



```

session_state : 2225425a-cc69-484c-act6-06/a0d0a3bac ,
"acr": "1",
"allowed-origins": [
  "http://localhost:3000"
],
"realm_access": {
  "roles": [
    "default-roles-google",
    "offline_access",
    "post.read",
    "uma_authorization"
  ]
},
"resource_access": {
  "account": {
    "roles": [
      "manage-account",
      "manage-account-links",
      "view-profile"
    ]
  }
},
"scope": "openid email profile",
"sid": "2225425a-cc69-484c-acf8-687a6d0a3bac",
"email_verified": false,
"groups": [
  "/users"
],

```

Adding custom property for roles in access token

This can be done by adding a new mapper in client scope

Configure a new mapper

×

Choose any of the mappings from this table

sha-256 hash. See OpenID Connect specification for more info about pairwise subject identifiers.

Role Name Mapper	Map an assigned role to a new name or position in the token.
User Address	Maps user address attributes (street, locality, region, postal_code, and country) to the OpenID Connect 'address' claim.
User Attribute	Map a custom user attribute to a token claim.
User Client Role	Map a user client role to a token claim.
User Property	Map a built in user property (email, firstName, lastName) to a token claim.
User Realm Role	Map a user realm role to a token claim.
User Session Note	Map a custom user session note to a token claim.
User's full name	Maps the user's first and last name to the OpenID Connect 'name' claim. Format is <first> + ' ' + <last>

Add mapper

If you want more fine-grain control, you can create protocol mapper on this client

Mapper type	User Realm Role
Name * ?	realm_roles
Realm Role prefix ?	
Multivalued ?	<input checked="" type="checkbox"/> On
Token Claim Name ?	realm_roles
Claim JSON Type ?	String
Add to ID token ?	<input checked="" type="checkbox"/> On
Add to access token ?	<input checked="" type="checkbox"/> On
Add to userinfo ?	<input checked="" type="checkbox"/> On
<div>Save Cancel</div>	

Users * ?

```
{
  "view-profile": [
    {
      "scope": "openid email profile",
      "sid": "723645a9-4c02-4ed1-9d05-eeb8694fc6f4",
      "realm_roles": [
        "default-roles-google",
        "offline_access",
        "postread",
        "uma_authorization"
      ],
      "email_verified": false,
      "groups": [
        "/users"
      ],
      "preferred_username": "superman",
      "given_name": "",
      "family_name": ""
    }
  ]
}
```

Effective protocol mappers ?

Effective role scope mappings ?

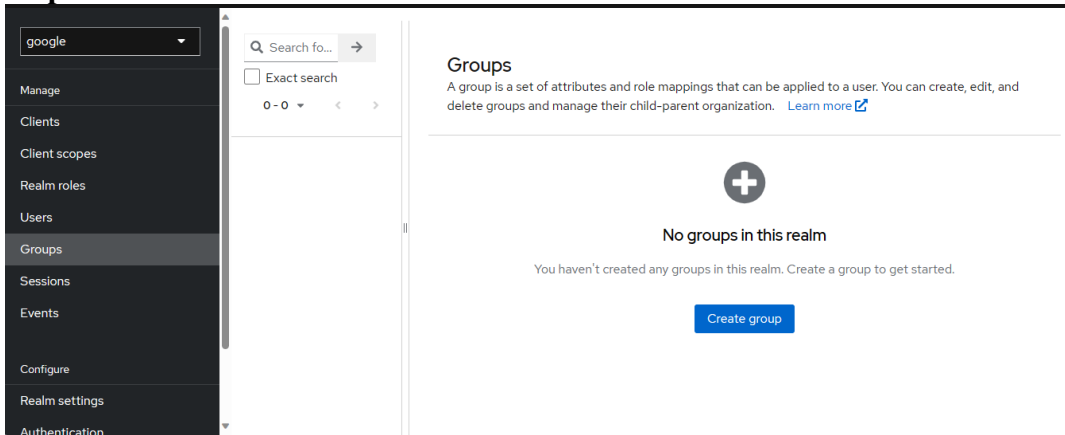
Generated access token ?

Generated ID token ?

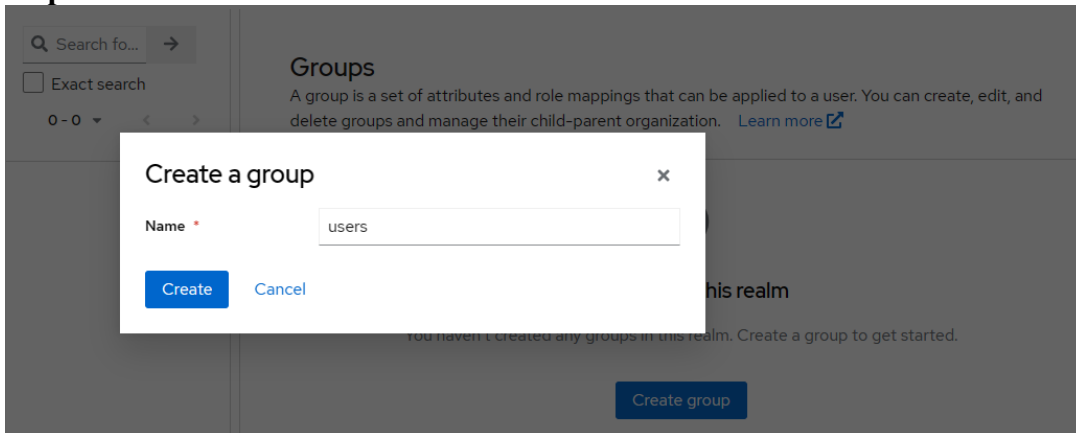
Generated user info ?

Setup Group and Assign Client Roles

Step:1



Step:2



Step:3 – Create Client Role

Go to Client>{CLIENT_NAME}>Roles

[Clients](#) > [Client details](#) > Create role

Create role

Role name *	<input type="text" value="postread"/>
Description	<input type="text" value="This role is reading post"/>
<div><input type="button" value="Save"/> <input type="button" value="Cancel"/></div>	

Step:4 - Assign roles in group

Users

Action ▾

Child groups

Members

Attributes

Role mapping



No roles for this group

You haven't created any roles for this group. Create a role to get started.

Assign role

Step:5 – Adding reading post permission for this USERS group

Assign roles to users

Filter by clients ▾

Search by role name



1 - 10 ▾



<input type="checkbox"/>	Name	Description
<input type="checkbox"/>	account delete-account	`\${role_delete-account}`
<input type="checkbox"/>	account manage-account	`\${role_manage-account}`
<input type="checkbox"/>	account manage-account-links	`\${role_manage-account-links}`
<input type="checkbox"/>	account manage-consent	`\${role_manage-consent}`
<input type="checkbox"/>	account view-applications	`\${role_view-applications}`
<input type="checkbox"/>	account view-consent	`\${role_view-consent}`
<input type="checkbox"/>	account view-groups	`\${role_view-groups}`
<input type="checkbox"/>	account view-profile	`\${role_view-profile}`
<input type="checkbox"/>	broker read-token	`\${role_read-token}`
<input checked="" type="checkbox"/>	google-cli post.read	This role is for reading post

1 - 10 ▾



Assign

Cancel

Step:6 – Adding this USERS group in the client

- i) Go to Client>{CLIENT_NAME}>Client Scopes

google-cli OpenID Connect Enabled Action

Clients are applications and services that can request authentication of a user.

Settings Keys Credentials Roles **Client scopes** Sessions Advanced

Setup Evaluate

▼ Name 🔍 Search by name → Add client scope Change type to ⋮

1 - 10 < >

<input type="checkbox"/> Assigned client scope	Assigned type	Description
<input checked="" type="checkbox"/> <u>google-cli-dedicated</u>	none	Dedicated scope and mappers for this client
<input type="checkbox"/> acr	Default	OpenID Connect scope for add acr (authentication context class reference) to the token
<input type="checkbox"/> address	Optional	OpenID Connect built-in scope: address
<input type="checkbox"/> email	Default	OpenID Connect built-in scope: email

ii) Go to the google-cli-dedicated (client scope name will depend on client name)

Clients > Client details > Dedicated scopes

google-cli
This is a client scope which includes the dedicated mappers and scope

Mappers Scope

+

No mappers

If you want to add mappers, please click the button below to add some predefined mappers or to configure a new mapper.

Add predefined mapper Configure a new mapper

iii) If one MAPPER already exist , click on By configuration

Clients > Client details > Dedicated scopes

google-cli
This is a client scope which includes the dedicated mappers and scope

Mappers Scope

🔍 Search for mapper → Add mapper ⋮

1 - 1 < >

Name	Category	Priority
groups	Token mappers	0

From predefined mappers
By configuration ✓

iv) Click on User Client Role

Configure a new mapper

Choose any of the mappings from this table

Group Membership	Map user group membership
Hardcoded claim	Hardcode a claim into the token.
Hardcoded Role	Hardcode a role into the access token.
Pairwise subject identifier	Calculates a pairwise subject identifier using a salted sha-256 hash. See OpenID Connect specification for more info about pairwise subject identifiers.
Role Name Mapper	Map an assigned role to a new name or position in the token.
User Address	Maps user address attributes (street, locality, region, postal_code, and country) to the OpenID Connect 'address' claim.
User Attribute	Map a custom user attribute to a token claim.
User Client Role	Map a user client role to a token claim.
User Property	Map a built in user property (email, firstName, lastName) to a token claim.

v) Click on Save

[Clients](#) > [Client details](#) > [Dedicated scopes](#) > [Mapper details](#)

Add mapper

If you want more fine-grain control, you can create protocol mapper on this client

Mapper type

User Client Role

Name *

client_roles

Client ID

google-cli

Client Role prefix

Multivalued

☒ On

Token Claim Name

client_roles

Claim JSON Type

String

Add to ID token

☒ On

Add to access token

☒ On

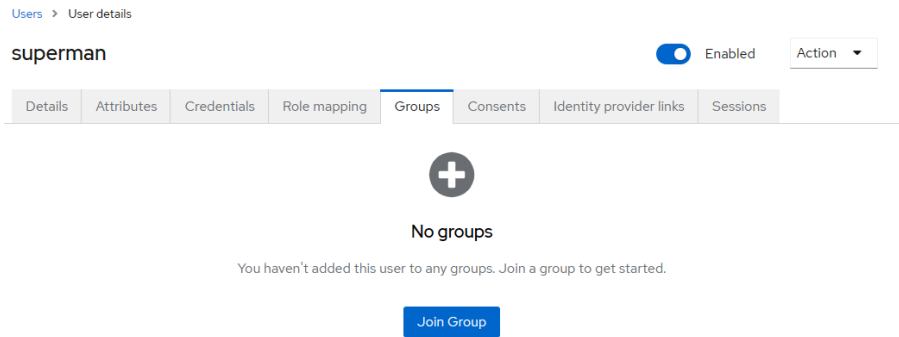
Add to userinfo

☒ On

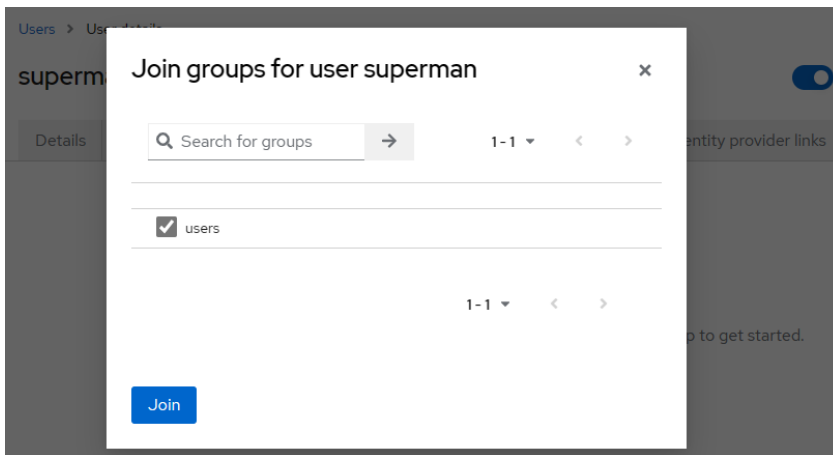
Save

Cancel

Step-6: Add the user to the USERS group. Or we can add the user from group directly.



Add group



Step-7 - Let's check that user's access token
Go to Client>{CLIENT_NAME}>Client Scopes. Then Evaluate Tab

Clients > Client details

google-cli OpenID Connect

Clients are applications and services that can request authentication of a user.

Settings Keys Credentials Roles Client scopes Sessions Advanced

Setup Evaluate

This page allows you to see all protocol mappers and role scope mappings

Scope parameter ⓘ openid ✕ Select scope parameters

Users ⌵ ⓘ superman

```
{
  "exp": 1694811946,
  "iat": 1694775946,
  "auth_time": 0,
  "jti": "62d76aed-18cc-4894-a57a-1bdfc63b114b",
  "iss": "http://localhost:8180/realm/google",
  "aud": "google-cli",
  "sub": "3228b610-5e00-4591-a662-e2b6c8efb64a",
  "typ": "ID",
  "azp": "google-cli",
  "session_state": "42f36f55-7a34-42be-a63a-917cf92b057b",
  "acr": "1",
  "sid": "42f36f55-7a34-42be-a63a-917cf92b057b",
  "client_roles": [
    "postread" ✓
  ],
  "email_verified": false,
  "groups": [
    "/users" ✓
  ],
  "preferred_username": "superman",
}
```

Effective protocol mappers ⓘ

Effective role scope mappings ⓘ

Generated access token ⓘ

Generated ID token ⓘ

Generated user info ⓘ