Keycloak API to create users returns a 403 Forbidden

Asked 3 years, 5 months ago Modified 3 years, 5 months ago Viewed 5k times

4

Experimenting with Keycloak as an Identity Provider. I'm running it by using the `./standalone.sh` script.

So, I obtain the `access_token` like this:

```
curl --request POST \
  --url http://localhost:8080/auth/realms/master/protocol/openid-connect/token \
  --header 'Content-Type: application/x-www-form-urlencoded' \
  --data grant_type=client_credentials \
  --data client_id=admin-cli \
  --data client_secret=<the-client-secret-under-master-realm-for-admin-cli-client>
```

Response:

```
{
  "access_token": "the-access-token",
  "expires_in": 60,
  "refresh_expires_in": 0,
  "token_type": "Bearer",
  "not-before-policy": 0,
  "scope": "profile email"
}
```

And then quickly, under my `test-realm` I try to create a user as follows:

```
curl --request POST \
  --url http://localhost:8080/auth/admin/realms/test-realm/users \
  --header 'Authorization: Bearer the-access-token' \
  --header 'Content-Type: application/json' \
  --data '{
    "firstName": "Sergey",
    "lastName": "Kargopolov",
    "email": "test@test.com",
    "enabled": "true",
    "username": "app-user"
}'
```

And I get hit with a `403`:

```
< HTTP/1.1 403 Forbidden
< X-XSS-Protection: 1; mode=block
< X-Frame-Options: SAMEORIGIN
< Referrer-Policy: no-referrer
< Date: Thu, 28 Jan 2021 23:43:57 GMT
< Connection: keep-alive
< Strict-Transport-Security: max-age=31536000; includeSubDomains
< X-Content-Type-Options: nosniff
< Content-Type: application/json
< Content-Length: 25
```

Is there something I'm missing? I'm following this tutorial and I'm doing everything exactly as described!

Edit: I tried the Password Grant way to obtain the Bearer Token and that worked, but NOT the client secret way. I obviously prefer the client secret way (which is where I'm stuck currently). What could be the issue here?

keycloak keycloak-services keycloak-rest-api

Share Edit Follow

edited Jan 29, 2021 at 7:05
dreamcrash
50.2k ● 26 ● 102 ● 126

asked Jan 28, 2021 at 23:45
Saturnian
1,868 ● 8 ● 45 ● 72

1 Answer

Sorted by: Highest score (default)

2

To create the user using the Keycloak Rest API, one just need to request from the *admin-cli* client a token on behalf of the admin user by providing its name and password, for instance as follows:

```
TOKEN=$(curl -k -sS     -d "client_id=admin-cli" \
                        -d "username=$ADMIN_NAME" \
                        -d "password=$ADMIN_PASSWORD" \
                        -d "grant_type=password" \
                        http://$KEYCLOAK_IP/auth/realms/master/protocol/openid-
connect/token)
```

from the $TOKEN object extract the access token (let us named `$ACCESS_TOKEN`).

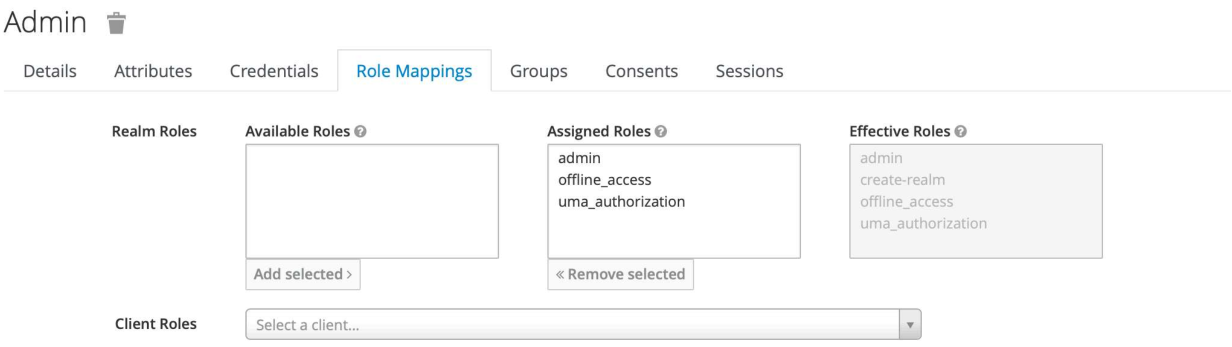
And then create the user as follows:

```
curl -k -sS -X POST https://$KEYCLOAK_IP/auth/admin/realms/$REALM_NAME/users \
        -H "Content-Type: application/json" \
        -H "Authorization: Bearer $ACCESS_TOKEN" \
        -d "$USER_JSON_DATA"
```

`$USER_JSON_DATA` will be the json data representation of the user to be created. There is no need to add the role *admin* to the *master* admin deployed with Keycloak by default.

If setup normally, you would just need to know (as I already described) the *admin's* name and password, which is configured in the initial setup anyway.

If you click on the *admin* user > roles, you would see the following:



Admin

Realm Roles

Available Roles

Assigned Roles: admin, offline_access, uma_authorization

Effective Roles: admin, create-realm, offline_access, uma_authorization

Client Roles: Select a client...

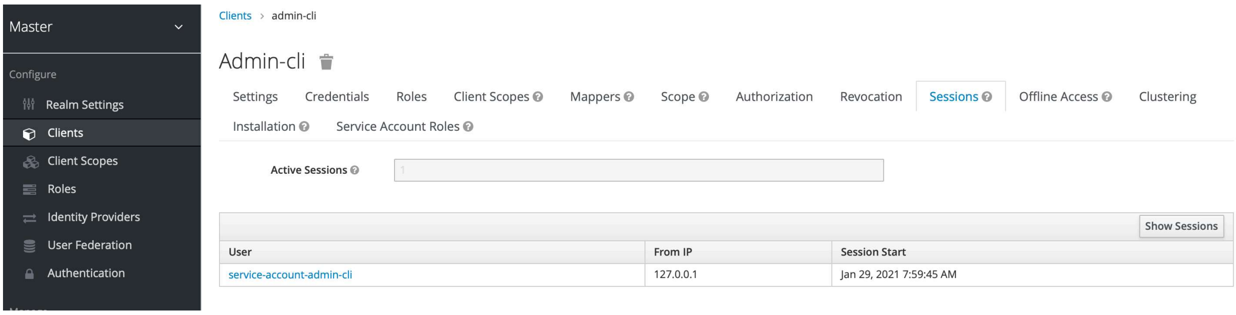
The *admin* user, has already the *admin* role.

> Edit: I tried the Password Grant way to obtain the Bearer Token and that worked, but NOT the client secret way. I obviously prefer the client secret way (which is where I'm stuck currently). What could be the issue here?

Now if you change the *admin_cli* configuration exactly as you did then you need to add to the `Service-account-admin-cli` user the role admin.

Now the problem is that `Service-account-admin-cli` user is *hidden* in the *User* section. Nonetheless, you can do the following:

1. Request again the admin token with your setup;
2. Go to Master Realm > Clients > admin-cli > Session > Click on [Show Session]:

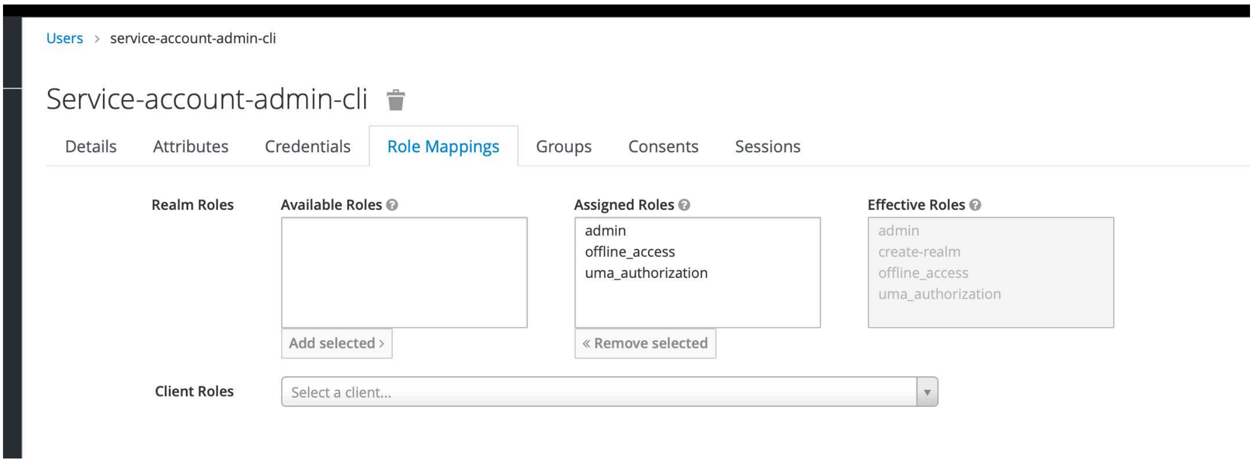


Admin-cli

Active Sessions: 1

User	From IP	Session Start
service-account-admin-cli	127.0.0.1	Jan 29, 2021 7:59:45 AM

3. click on the user `service-account-admin-cli`;
4. Go to Role Mappings;
5. Assign the `admin` role;



Users > service-account-admin-cli

Service-account-admin-cli

Realm Roles

Available Roles

Assigned Roles: admin, offline_access, uma_authorization

Effective Roles: admin, create-realm, offline_access, uma_authorization

Client Roles: Select a client...

Since the `service-account-admin-cli` user has now the `admin` role, a token request on that user's behalf will contain the necessary privileges to create the users.

If the aforementioned does not work, then do the following go to:

- Realm Master;
- Clients > admin-cli;
- Go to Mappers;
- Click on [Create];
- As Mapper Type select "Hardcoded Role";
- Click on Select Role and selection "admin";
- Click [Save].

Share Edit Follow

edited Feb 5, 2021 at 12:49

answered Jan 29, 2021 at 6:52

dreamcrash

50.2k ● 26 ● 102 ● 126