

Keycloak Admin Client in Spring Boot

wallyson

May 6

I'm having some trouble to use keycloak-admin-client in spring boot. I'm getting some errors.

If I try with this code I get 401 (unauthorized):

```
public Keycloak getKeycloakInstance() {  
    var keycloak = KeycloakBuilder.builder()  
        .serverUrl(SERVER_URL)  
        .realm(REALM)  
        .username(USERNAME)  
        .password(PASSWORD)  
        .clientId(CLIENT_ID)  
        .build();  
    return keycloak;  
}
```

Also, if I put `.resteasyClient(...)` and `.clientSecret(...)` in the code above I get badrequest.

In the client roles I created a new composite role and gave all realm-management roles to it, maybe I configured something wrong?

Where can I find some documentation on how to use this Admin Client Dependency?

```
<dependency>  
    <groupId>org.keycloak</groupId>  
    <artifactId>keycloak-admin-client</artifactId>  
    <version>10.0.0</version>  
</dependency>
```

May 6

1 / 4

May 6

May 8

zonaut

May 7

You seem to be mixing things up a bit, I will provide a couple of examples on how you could use the keycloak admin client.

Personally I would choose example 2, creating a dedicated service account client as we are communicating service to service.

Example 1 -> Using a user

- Create new client under your desired realm -> keycloak-admin
- Select public client with only direct access grant enabled
- Create new role, enable composite roles
 - type realm-managment into client roles under composite roles
 - add available roles that you need
- Select a user and open role mappings tab
 - type keycloak-admin in client roles and add needed roles

Code:

```
Keycloak keycloak = KeycloakBuilder.builder()  
    .serverUrl("http://localhost:8080/auth")  
    .grantType(OAuth2Constants.PASSWORD)  
    .realm("realm-name")  
    .clientId("keycloak-admin")  
    .username("username")  
    .password("password")  
    .resteasyClient(  
        new ResteasyClientBuilder()
```

```
        .connectionPoolSize(10).build()
    ).build();
```

```
keycloak.tokenManager().getAccessToken();
RealmResource realmResource = keycloak.realm("realm-name");
```

Example 2 -> Using a confidential service account

- Create new client under your desired realm -> keycloak-admin
- Select confidential client with only service account enabled
- Select tab service account roles
 - type realm-management into client roles
 - add available roles that you need

Code:

```
Keycloak keycloak = KeycloakBuilder.builder()
    .serverUrl("http://localhost:8080/auth")
    .grantType(OAuth2Constants.CLIENT_CREDENTIALS)
    .realm("realm-name")
    .clientId("keycloak-admin")
    .clientSecret("1c7e2815-c4dc-401c-af2f-ebddad3b4a79")
    .resteasyClient(
        new ResteasyClientBuilder()
            .connectionPoolSize(10).build()
    ).build();

keycloak.tokenManager().getAccessToken();
RealmResource realmResource = keycloak.realm("realm-name");
```

Example 3 -> Using admin account

You could also use the admin user with the password grant and use the existing admin-cli client.

```
Keycloak keycloak = KeycloakBuilder.builder()
    .serverUrl("http://localhost:8080/auth")
    .grantType(OAuth2Constants.PASSWORD)
    .realm("master")
    .clientId("admin-cli")
    .username("admin")
    .password("password")
    .resteasyClient(
        new ResteasyClientBuilder()
            .connectionPoolSize(10).build()
    ).build();

keycloak.tokenManager().getAccessToken();
RealmResource realmResource = keycloak.realm("realm-name");
```

[🔗 Spring Boot Service Accounts](#)

[🔗 Purge unused realms](#)

[🔗 \[java confidential client\] is it necessary to manually instantiate client to access admin api?](#)

wallyson

May 7

@zonaut , thanks very much for your reply, it worked. I'm still pretty new to keycloak (OAuth and OIDC too for that matter), but I'm getting the hang of it.
I just found it weird to have so little information on the web about using the keycloak-admin-client.
Maybe is not so common to use it?

zonaut

May 8

Most probably don't need more advanced workflows or don't know the client exists at all.

And lot's of others are just reading the API docs and use their favorite tools their used too.

I like this client though, versioned and stable.