

Département Mathématiques et Informatique

Cycle Ingénieur

« Ingénierie Informatique – Big Data et Cloud Computing »

COMPTE-RENDU:

Activité pratique N° 4 - Sécurité des micro services avec Keycloak

OAuth2

- Protocole et Framework de délégation d'autorisations
- Architecture à trois parties
 - Client
 - Resource server
 - Authorization Server

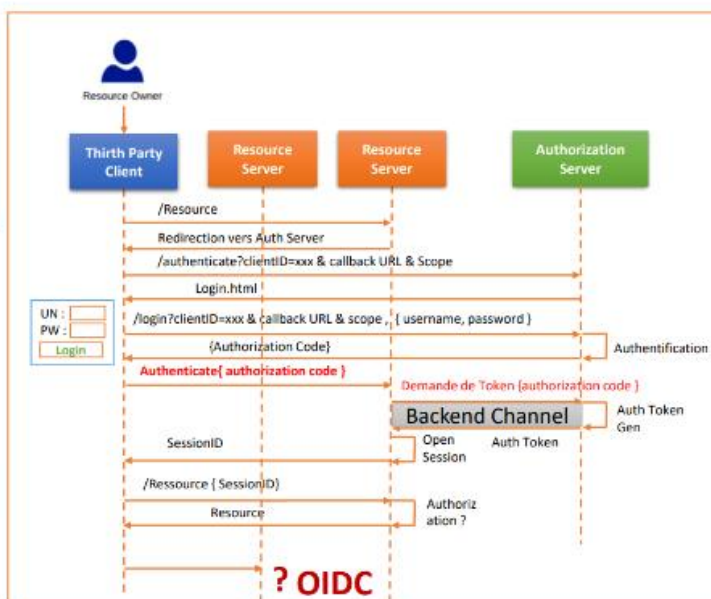
• Implicit Flow :

- Le serveur d'autorisation envoie directement le Token d'authentification pendant la redirection vers le callback

• Authorization-code Flow :

- Le serveur d'autorisation envoie directement un code d'autorisation éphémère pendant la redirection vers le callback (Resource server)
- Resource server envoie le code dans une requête en back channel au serveur d'autorisation pour demander le token

• Authorization-code + PKCE Flow :



Réalisé par: Asmaa ELASRI

Encadré par: Pr. Mohamed YOUSSEFI

Année Universitaire : 2022-2023

Objectifs

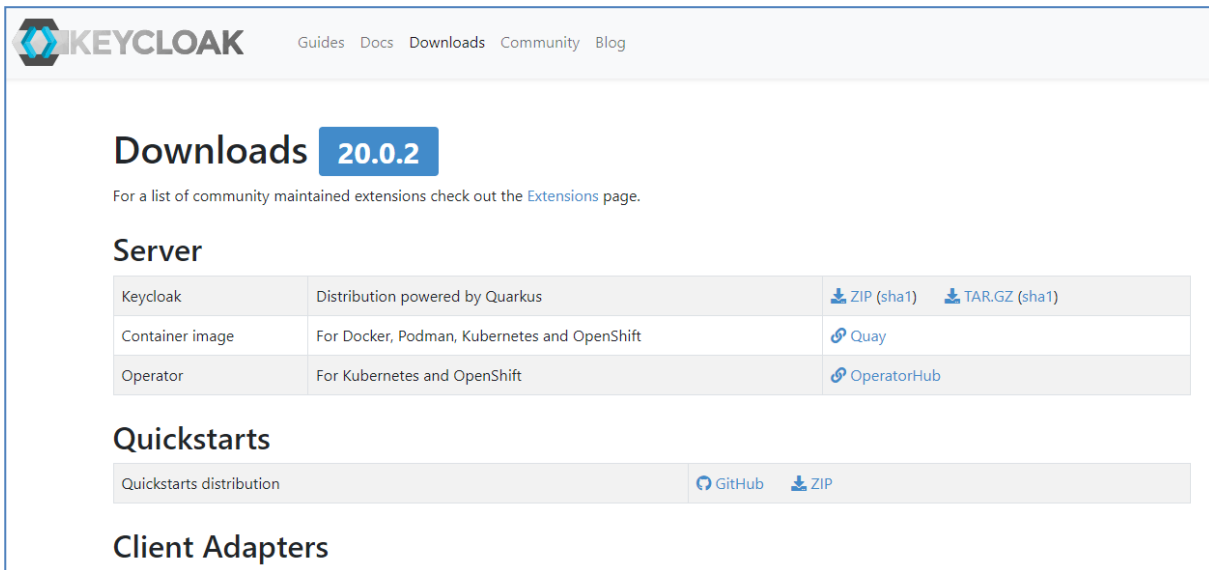
Partie 1 :

- Télécharger Keycloak 19
- Démarrer Keycloak
- Créer un compte Admin
- Créer une Realm
- Créer un client à sécuriser
- Créer des utilisateurs
- Créer des rôles
- Affecter les rôles aux utilisateurs
- Avec PostMan :
 - Tester l'authentification avec le mot de passe
 - Analyser les contenus des deux JWT Access Token et Refresh Token
 - Tester l'authentification avec le Refresh Token
 - Tester l'authentification avec Client ID et Client Secret
 - Changer les paramètres des Tokens Access Token et Refresh Token

Partie 2 : Sécuriser L'architecture Micro services Du projet Customer-service, Inventory-service et Order-service

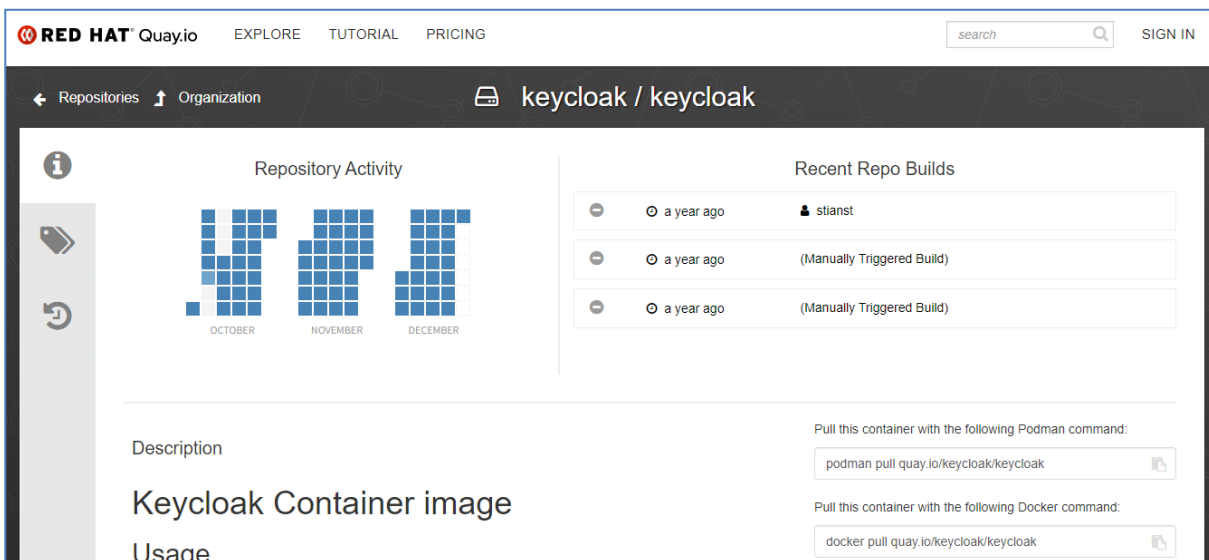
Partie 1 :

Télécharger Keycloak 19



The image shows the Keycloak Downloads page. At the top, there's a navigation bar with links: Guides, Docs, Downloads, Community, and Blog. The main heading is "Downloads" with a sub-heading "20.0.2". Below this, a note says: "For a list of community maintained extensions check out the [Extensions](#) page." The "Server" section contains a table with download links for ZIP, TAR.GZ, Quay, and OperatorHub. The "Quickstarts" section has a link to the distribution. The "Client Adapters" section is partially visible.

Keycloak	Distribution powered by Quarkus	ZIP (sha1)	TAR.GZ (sha1)
Container image	For Docker, Podman, Kubernetes and OpenShift	Quay	
Operator	For Kubernetes and OpenShift	OperatorHub	



The image shows the Keycloak Container image page on Quay.io. The page has a navigation bar with links: RED HAT Quay.io, EXPLORE, TUTORIAL, and PRICING. The main heading is "keycloak / keycloak". The "Repository Activity" section shows a calendar view for October, November, and December. The "Recent Repo Builds" section shows a list of builds. The "Description" section contains the text "Keycloak Container image" and "Usage". The "Podman" and "Docker" sections provide commands to pull the container image.

```
podman pull quay.io/keycloak/keycloak
```

```
docker pull quay.io/keycloak/keycloak
```

- Exécuter la commande suivante dans votre terminal

`docker pull quay.io/keycloak/keycloak`

```
C:\Users\HP> docker pull quay.io/keycloak/keycloak
Using default tag: latest
latest: Pulling from keycloak/keycloak
3e2a8131eeab: Pull complete
d001b67d2428: Pull complete
5dc4e2096006: Pull complete
Digest: sha256:2e6b1012417b725ffe031eb7cc3d89365c1f4d9e9877b222195c7067a1e8810e
Status: Downloaded newer image for quay.io/keycloak/keycloak:latest
quay.io/keycloak/keycloak:latest
C:\Users\HP>
```

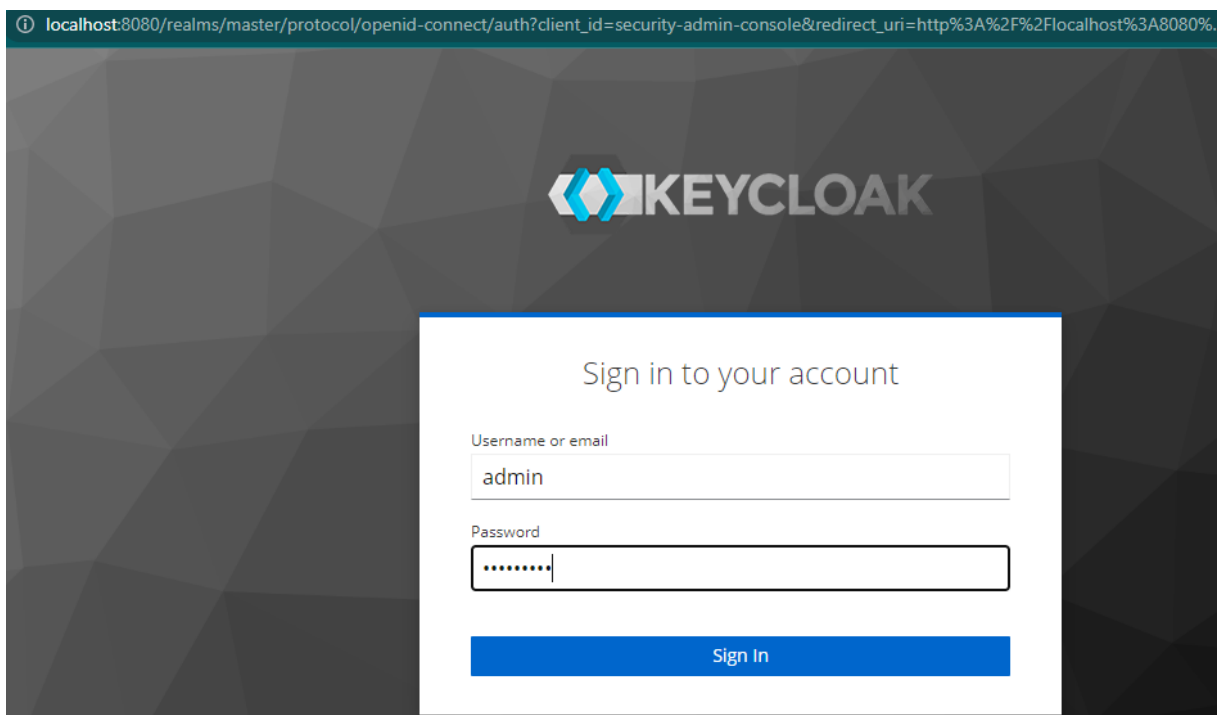
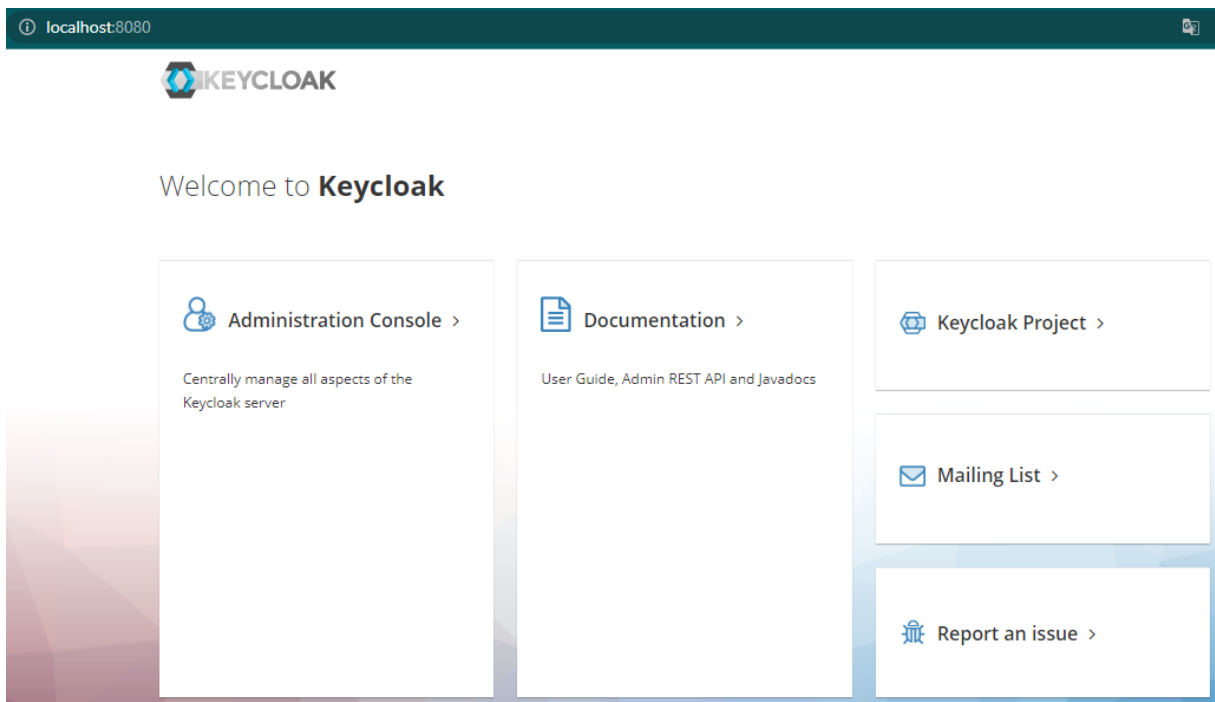
- Démarrer Keycloak avec l'une des commandes suivantes :
- `docker run -p 8080:8080 -e KEYCLOAK_ADMIN=admin -e KEYCLOAK_ADMIN_PASSWORD=change_me quay.io/keycloak/keycloak start-dev`

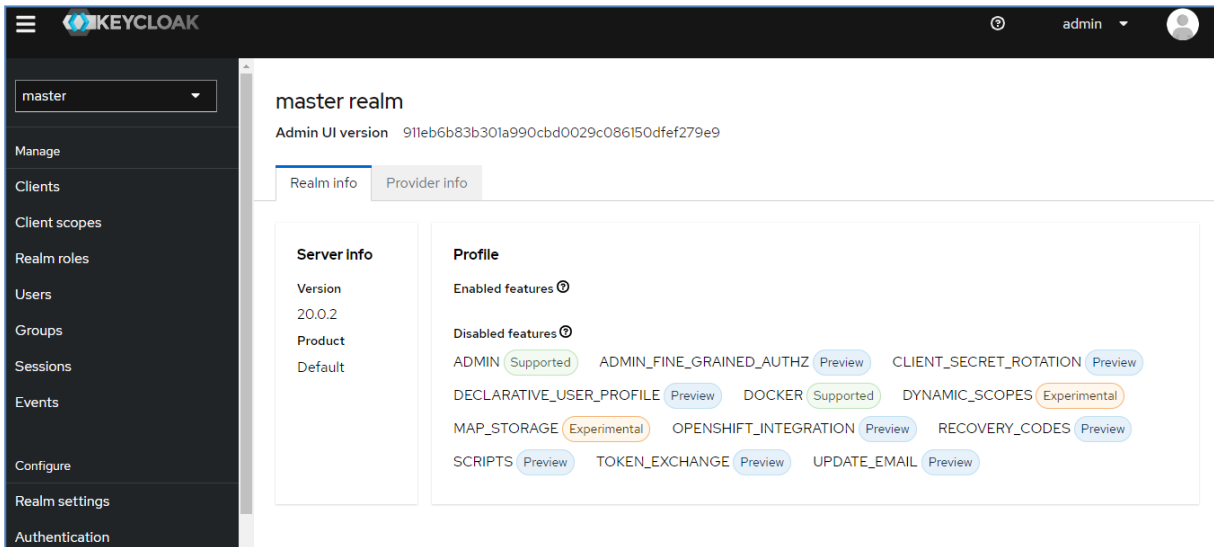
- OU

docker run quay.io/keycloak/keycloak start-dev

```
C:\Users\HP> docker run -p 8088:8088 -e KEYCLOAK_ADMIN=admin -e KEYCLOAK_ADMIN_PASSWORD=change_me quay.io/keycloak/keyc
loak start-dev
Updating the configuration and installing your custom providers, if any. Please wait.
2022-12-26 12:11:41,548 INFO [io.quarkus.deployment.QuarkusAugmentor] (main) Quarkus augmentation completed in 36676ms
2022-12-26 12:11:47,372 INFO [org.keycloak.quarkus.runtime.hostname.DefaultHostnameProvider] (main) Hostname settings:
Base URL: <unset>, Hostname: <request>, Strict HTTPS: false, Path: <request>, Strict BackChannel: false, Admin URL: <uns
et>, Admin: <request>, Port: -1, Proxied: false
2022-12-26 12:11:51,528 WARN [io.quarkus.agroal.runtime.DataSources] (main) Datasource <default> enables XA but transac
tion recovery is not enabled. Please enable transaction recovery by setting quarkus.transaction-manager.enable-recovery=
true, otherwise data may be lost if the application is terminated abruptly
2022-12-26 12:11:54,161 WARN [org.infinispan.PERSISTENCE] (keycloak-cache-init) ISPN000554: jboss-marshalling is deprec
ated and planned for removal
2022-12-26 12:11:54,826 WARN [org.infinispan.CONFIG] (keycloak-cache-init) ISPN000569: Unable to persist Infinispan int
ernal caches as no global state enabled
```

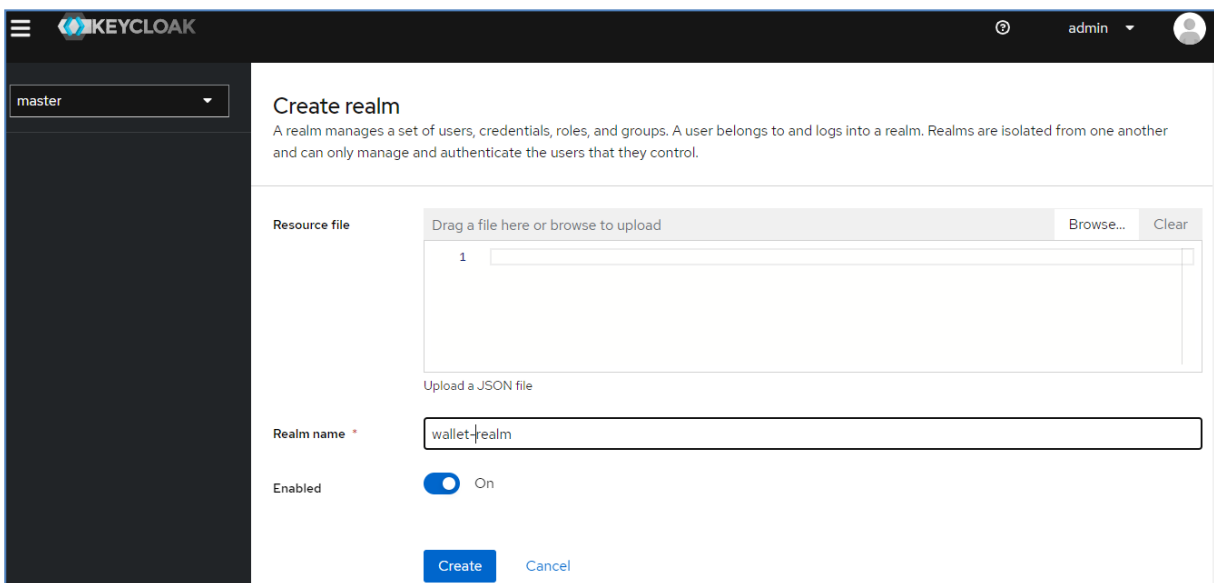
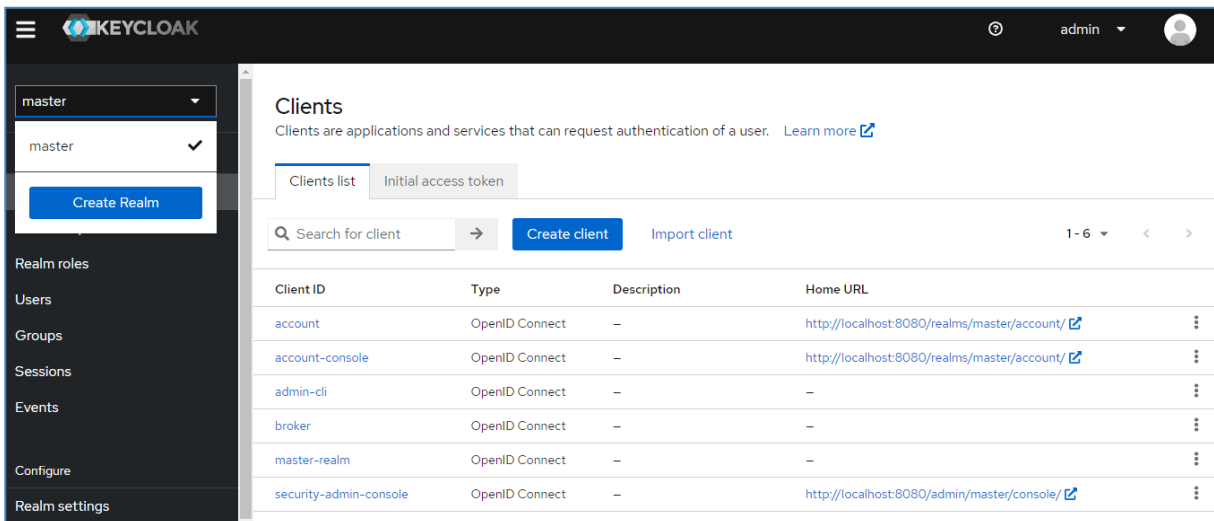
Console d'administration Keycloak





1.3 Créer un realm

un realm : le territoire à sécuriser (concerne un ensemble d'apps à sécuriser)



1.4 Créer un client à sécuriser

KEYCLOAK

admin

wallet-realm

Manage

Clients

Client scopes

Realm roles

Users

Groups

Sessions

Events

Configure

Realm settings

Clients > Create client

Create client

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

1 General Settings

Client type

OpenID Connect

Client ID *

wallet-client

Name

Description

Always display in console

Off

KEYCLOAK

admin

wallet-realm

Manage

Clients

Client scopes

Realm roles

Users

Groups

Sessions

Events

Configure

Realm settings

Authentication

Identity providers

Clients > Create client

Create client

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

1 General Settings

2 Capability config

Client authentication

Off

Authorization

Off

Authentication flow

☒ Standard flow

☒ Direct access grants

☐ Implicit flow

☐ Service accounts roles

☐ OAuth 2.0 Device Authorization Grant

☐ OIDC CIBA Grant

Save

Back

Cancel

KEYCLOAK

admin

wallet-realm

Manage

Clients

Client scopes

Realm roles

Users

Groups

Sessions

Events

Configure

Realm settings

Authentication

Identity providers

Access settings

Root URL

Home URL

http://localhost:4200/

Valid redirect URIs

http://localhost:4200/*

+ Add valid redirect URIs

Valid post logout redirect URIs

http://localhost:4200/

+ Add valid post logout redirect URIs

Web origins

+ Add web origins

Admin URL

Save

Revert

Jump to section

General Settings

Access settings

Capability config

Login settings

Logout settings

Client ID	Type	Description	Home URL
account	OpenID Connect	—	http://localhost:8080/realms/wallet-realm/account/
account-console	OpenID Connect	—	http://localhost:8080/realms/wallet-realm/account/
admin-cli	OpenID Connect	—	—
broker	OpenID Connect	—	—
realm-management	OpenID Connect	—	—
security-admin-console	OpenID Connect	—	http://localhost:8080/admin/wallet-realm/console/
wallet-client	OpenID Connect	—	http://localhost:4200/

1.5 Créer des utilisateurs

- User 1 :

No users found

Change your search criteria or add a user

[Create new user](#)

Users > Create user

Create user Enabled Action

Username *

Email

Email verified ☐ Off

First name

Last name

Required user actions

Groups [Join Groups](#)

KEYCLOAK

admin

wallet-realm

Manage

Clients

Client scopes

Realm roles

Users

Groups

Sessions

Events

Configure

Realm settings

Authentication

user1

Enabled

Action

Details

Attributes

Credentials

Role mapping

Groups

Consents

Identity provider links

Sessions

ID *

f2b79bb4-53ef-463a-a084-cf90e5243802

Created at *

12/26/2022, 10:19:23 PM

Username *

user1

Email

user1@gmail.com

Email verified ⓘ

Off

First name

user1FirstName

Last name

user1LastName

KEYCLOAK

admin

wallet-realm

Manage

Clients

Client scopes

Realm roles

Users

Groups

Sessions

Events

Configure

Realm settings

Authentication

Users > User details

user1

Enabled

Action

Details

Attributes

Credentials

Role mapping

Groups

Consents

Identity provider links

Sessions

+

No credentials

This user does not have any credentials. You can set password for this user.

Set password

Credential Reset

KEYCLOAK

admin

wallet-realm

Manage

Clients

Client scopes

Realm roles

Users

Groups

Sessions

Events

Configure

Realm settings

Authentication

Users > User details

user1

Enabled

Action

Details

Attributes

Credentials

Role mapping

Groups

Consents

Identity provider links

Sessions

Set password for user1

✕

Password *

.....

👁

Password confirmation *

.....

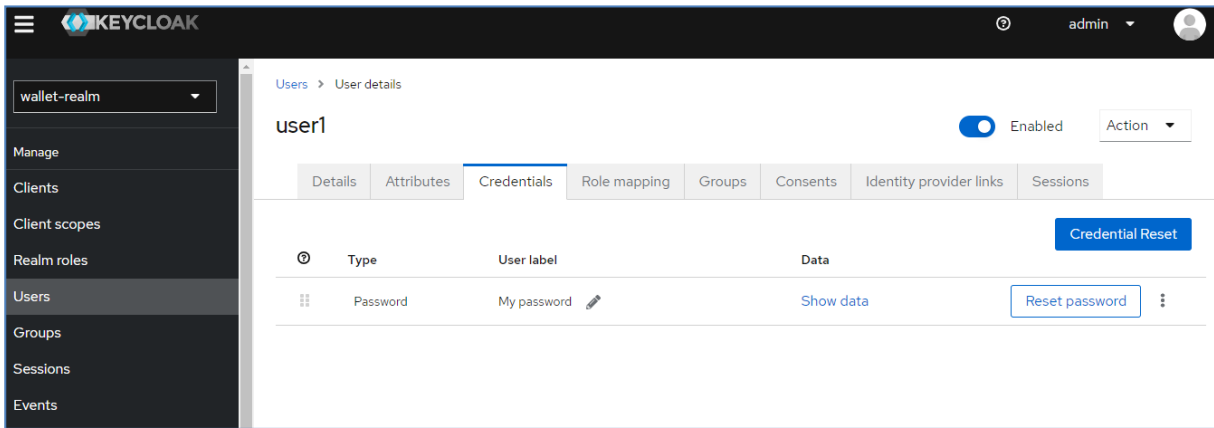
👁

Temporary ⓘ

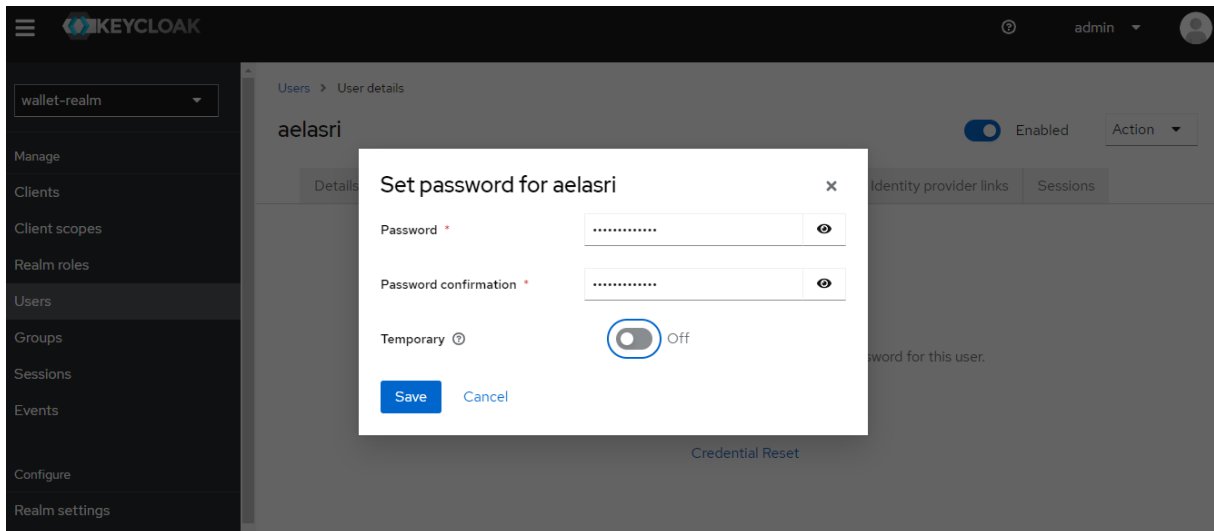
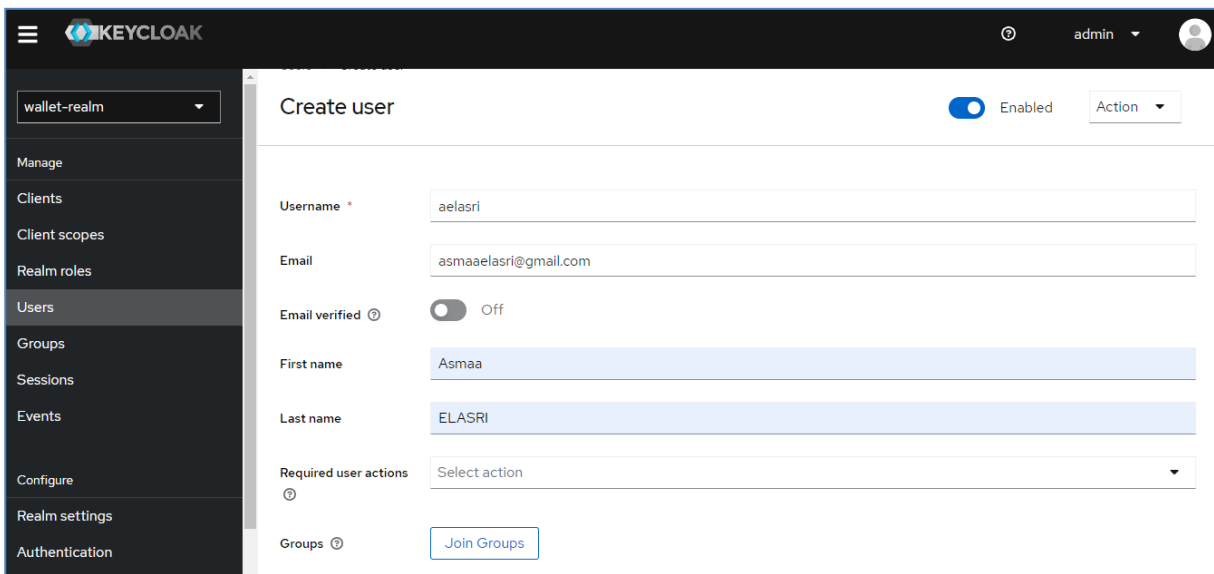
Off

Save

Cancel



- User 2 :



Users
Users are the users in the current realm. [Learn more](#)

User list Permissions

Search user → Add user Delete user 1-2 < >

Username	Email	Last name	First name	Status
aelasri	asmaaelasri@gmail.com	ELASRI	Asmaa	—
user1	user1@gmail.com	user1LastName	user1FirstName	—

1-2 < >

1.6 Créer des rôles

Realm roles > Create role

Create role

Role name * USER

Description

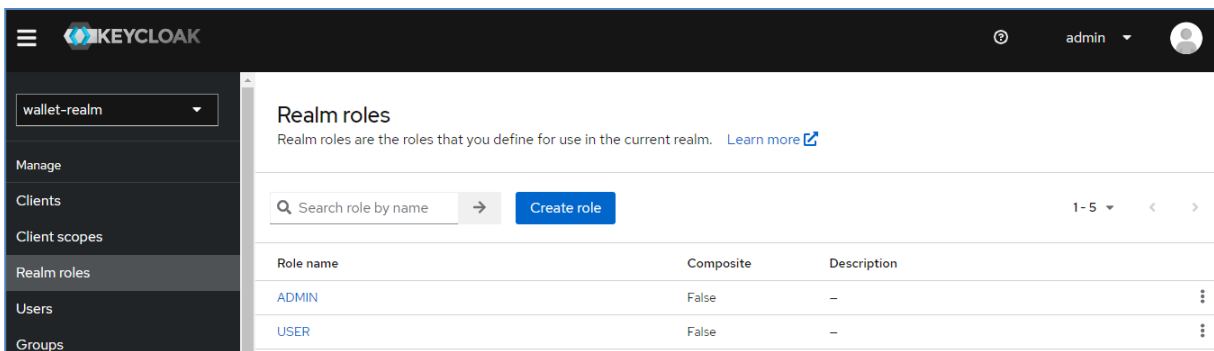
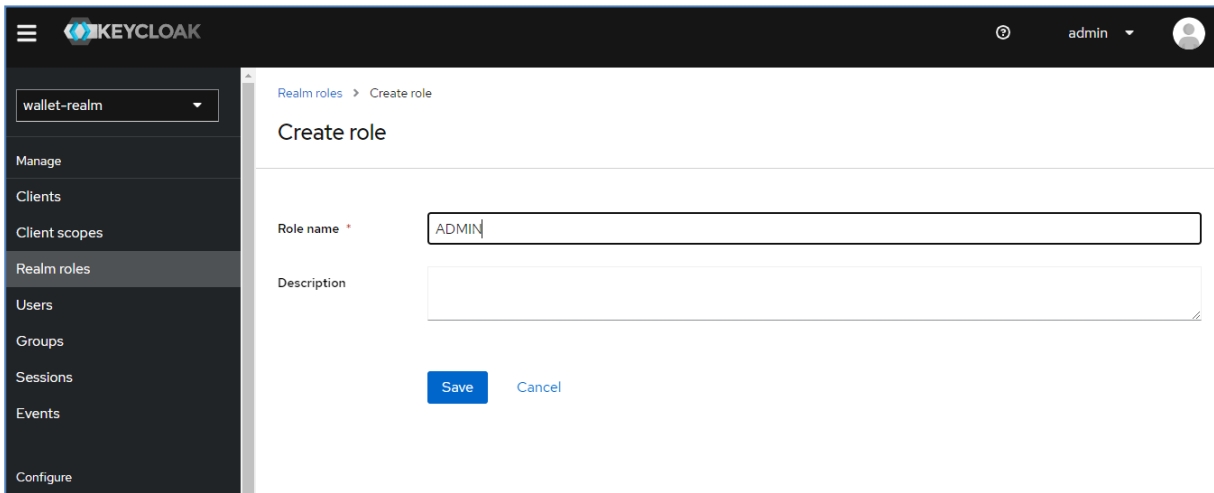
Save Cancel

Realm roles
Realm roles are the roles that you define for use in the current realm. [Learn more](#)

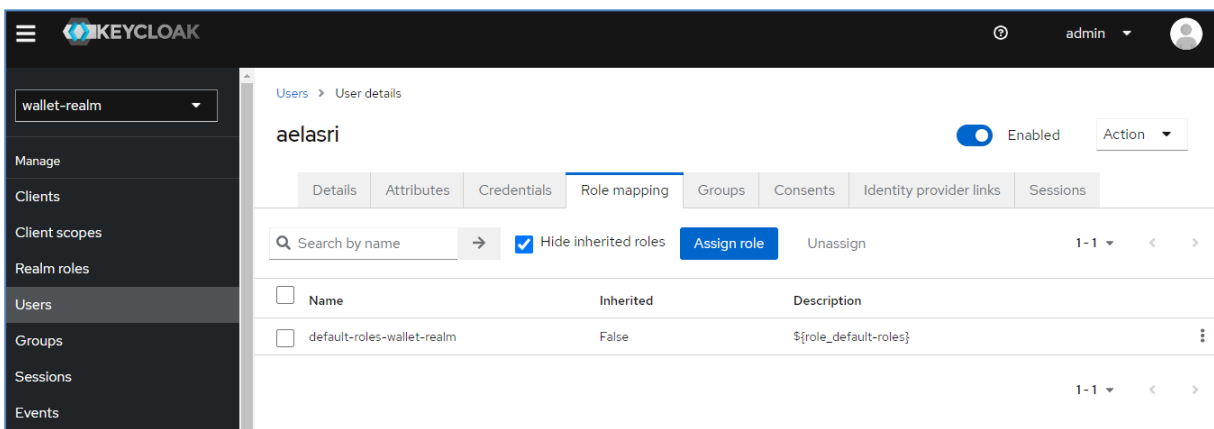
Search role by name → Create role 1-4 < >

Role name	Composite	Description
USER	False	—
default-roles-wallet-realm	True	\${role_default-roles}
offline_access	False	\${role_offline-access}
uma_authorization	False	\${role_uma_authorization}

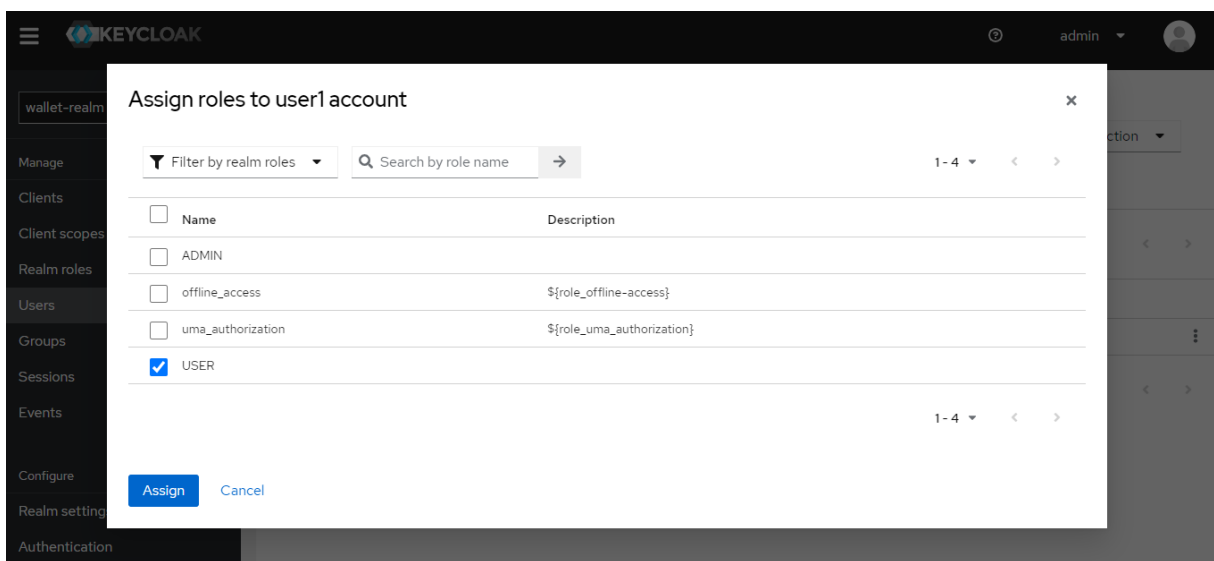
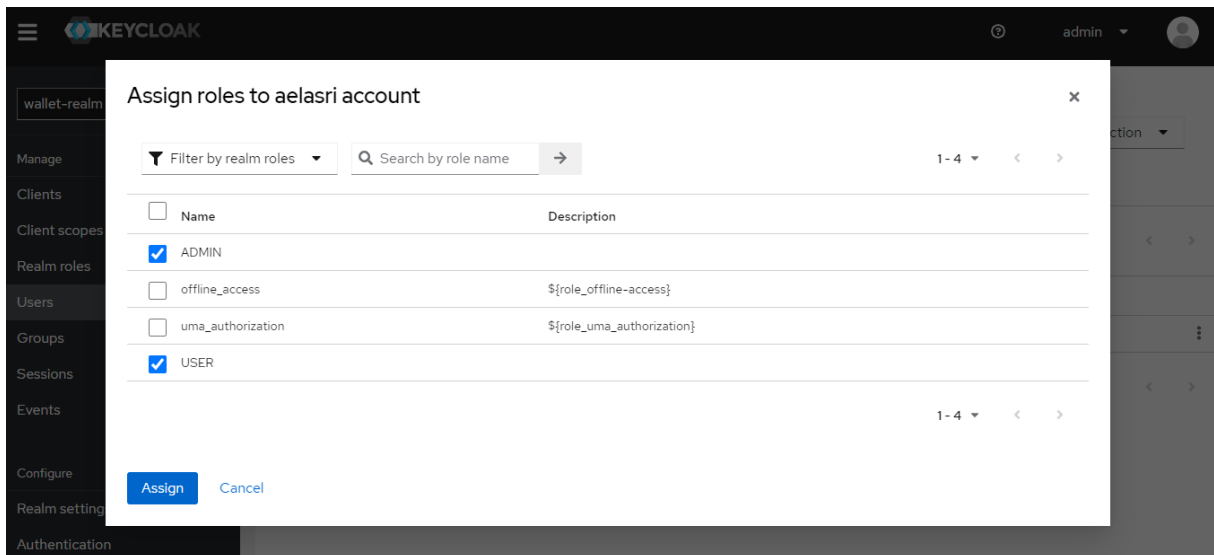
1-4 < >



1.7 Affecter les rôles aux utilisateurs



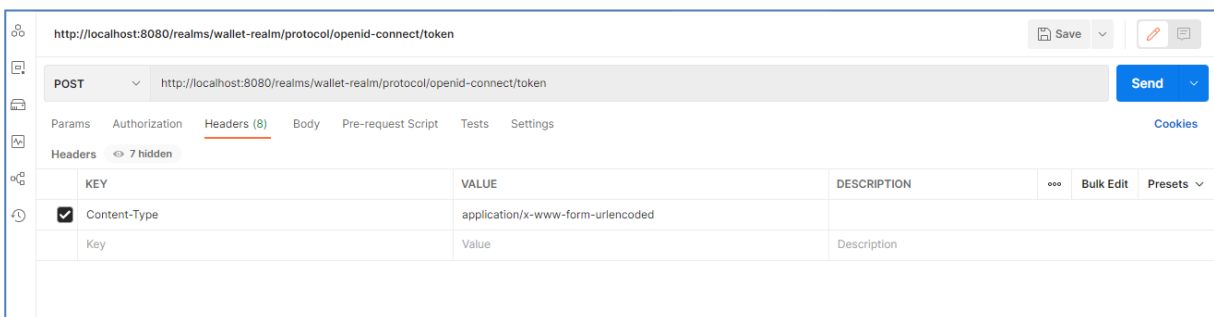
Assign Role



1.8 Avec PostMan :

Il y a trois manières de s'authentifier via mot de passe , refresh Token (précisez le token) et l'authentification avec Client ID et Client Secret

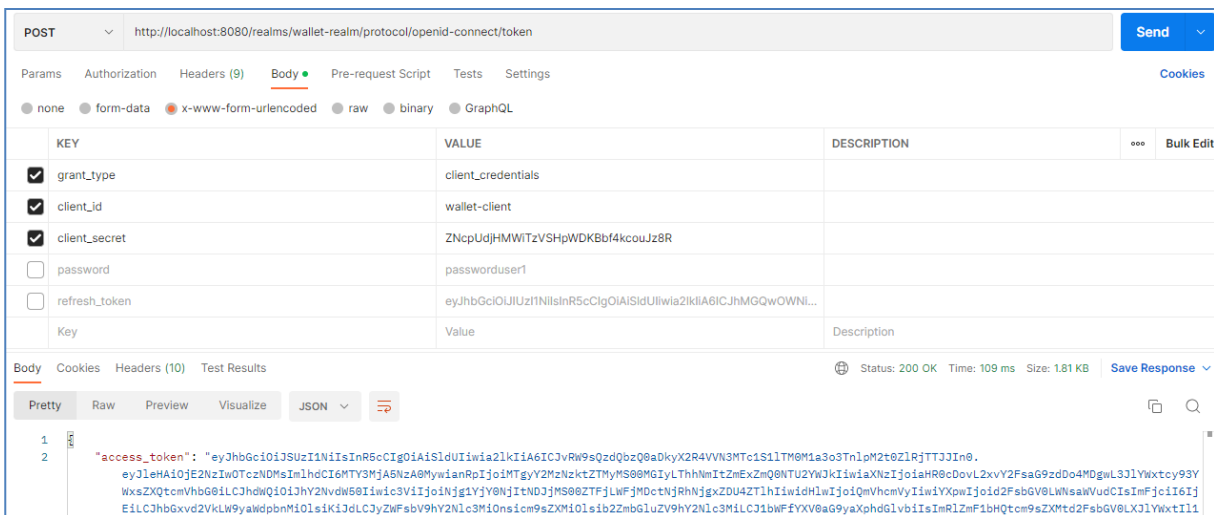
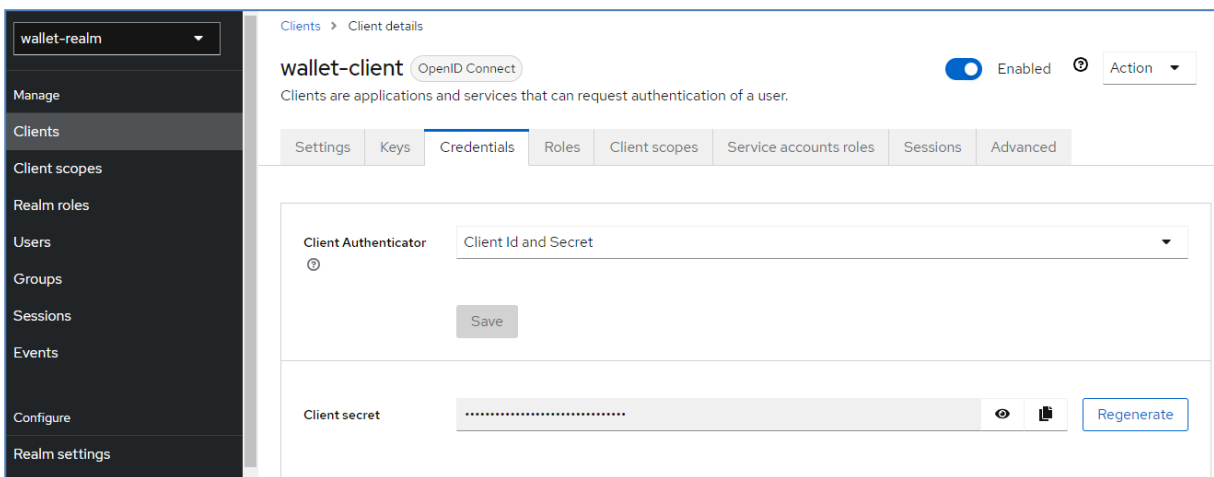
- Tester l'authentification avec le mot de passe

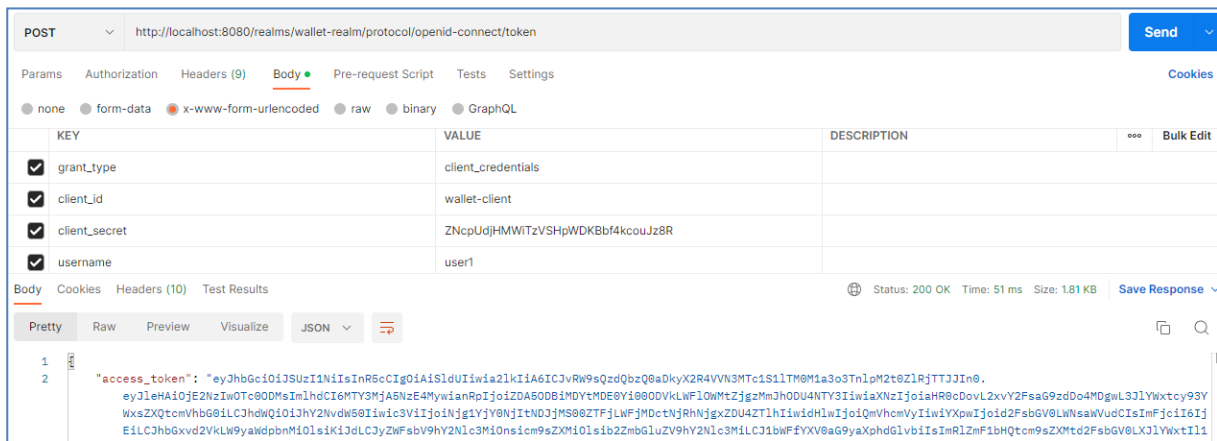


- **Tester l'authentification avec Client ID et Client Secret**

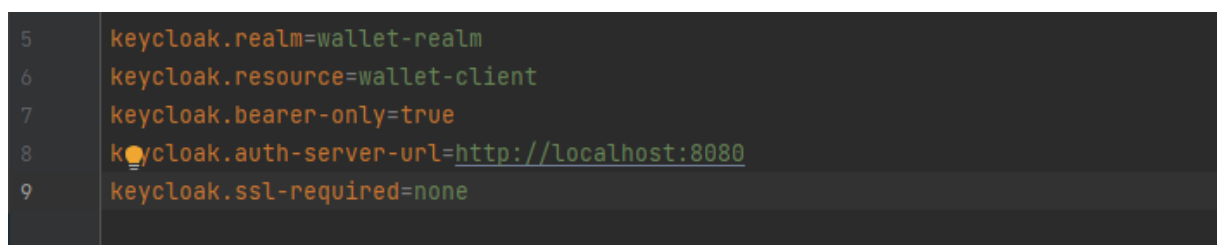
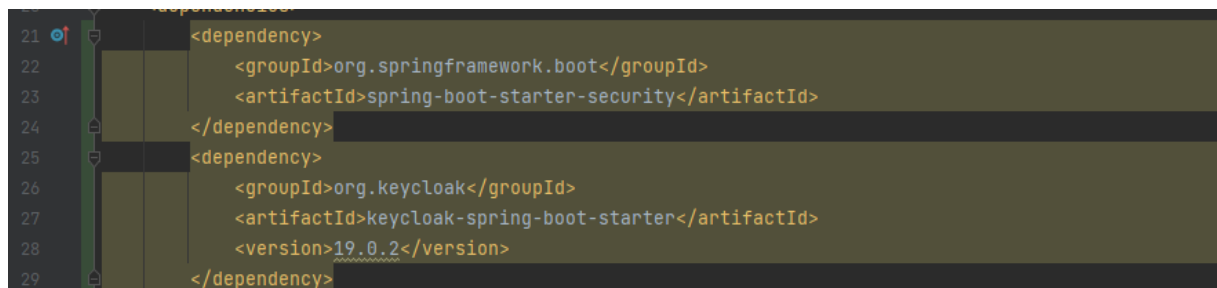
> wallet-client

> Activer Client authentication : je dois authentifier l'app qui m'envoie l'auth





Partie 2 : Sécuriser L'architecture Micro services




```

15  @KeycloakConfiguration
16  @EnableGlobalMethodSecurity(prePostEnabled = true)
17  public class KeycloakConfig extends KeycloakWebSecurityConfigurerAdapter {
18      no usages new *
19      @Override
20      protected SessionAuthenticationStrategy sessionAuthenticationStrategy() {
21          return new RegisterSessionAuthenticationStrategy(new SessionRegistryImpl());
22      }
23
24      new *
25      @Override
26      protected void configure(AuthenticationManagerBuilder auth) throws Exception {
27          auth.authenticationProvider(KeycloakAuthenticationProvider());
28      }
29
30      new *
31      @Override
32      protected void configure(HttpSecurity http) throws Exception {
33          super.configure(http);
34          http.csrf().disable();
35          //http.authorizeRequests().anyRequest().permitAll();
36          http.authorizeRequests().antMatchers( ...antPatterns: "/h2-console/**").permitAll();
37          http.headers().frameOptions().disable();
38          http.authorizeRequests().anyRequest().authenticated();
39      }
40  }

```

- The screenshot shows an IDE with a REST client interface. The top pane displays a REST client request:

```

1 POST http://localhost:8080/realms/wallet-realm/protocol/openid-connect/token
2 Accept: application/json
3
4 grant_type=password&username=user1&password=passworduser1&client_id=wallet-client
5
6 2022-12-27T011814.200.json
7 ###
  
```

The bottom pane shows the response body as a JSON object:

```

{
  "access_token": "eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2kiOiA6ICJvRW9sQzdQbzQ0aDkyX2R4VWV3MTc1S1lTM0M1a3o3TnlpM2t0ZjRjTTJj",
  "expires_in": 300,
  "refresh_expires_in": 1800,
  "refresh_token": "eyJhbGciOiJIUzI1NiIsInR5cCIgOiAiSldUIiwia2kiOiA6ICJhMGQwOWNiYS0yOWJkLTQ5YjctOTQxNy0yHWE4YWYzMWU4ZWU1fQ.eyJj",
  "token_type": "Bearer",
  "not-before-policy": 0,
}
  
```

- Récupérer tous les Customers

