NEXT GENERATION IT

Keycloak

IDENTITY AND ACCESS MANAGMENT SOLUTION

Presenté par Hjaiej Mohamed

Keycloak : Générale

- Keycloak est une solution de connexion unique (single sign-on) pour les applications web et les services web RESTful.

- Keycloak est un produit logiciel open source.

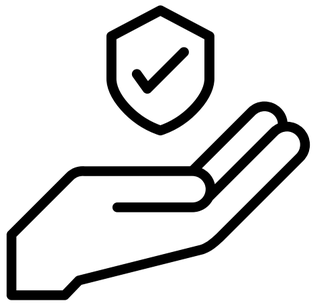
- Développer par Red Hat en septembre 2014 en utilisant Java.

- Plus de 6 000 forks et 18 700 étoiles dans le dépôt officiel sur GitHub.

- Facile à commencer à l'utiliser en utlisant docker.

```
docker run -p 8080:8080 -e KEYCLOAK_ADMIN=admin -e KEYCLOAK_ADMIN_PASSWORD=admin quay.io/keycloak/keycloak:23.0.4 start-dev
```

Keycloak : Serveur d'authentification ? Oui



Gère tout!

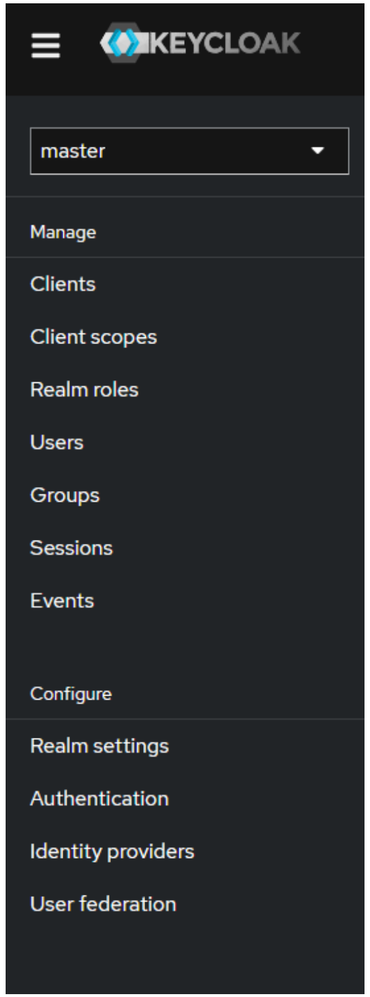
Support des protocoles standard d'authentifaction (OpenID Connect, OAuth 2.0, SAML)

Gère les tokens (JWT) et les sessions

Multi-factor-authentification (MFA)

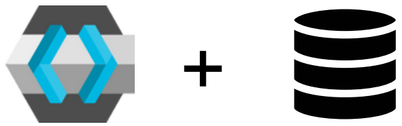
Single sign-on (SSO)

Gestion des rôles et groupes des utilisateurs



User Federation

Utilisateurs enregistrés par keycloak | interne



Keycloak + Base de données RDBMS / NoSql de Keycloak

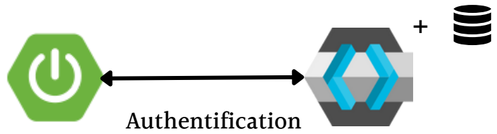
Autres sources d'utilisateurs | externe

LDAP | Active directory | Base de données RDBMS / NoSql | Réseaux sociaux

Keycloak : Identity provider ou Identity broker? les deux

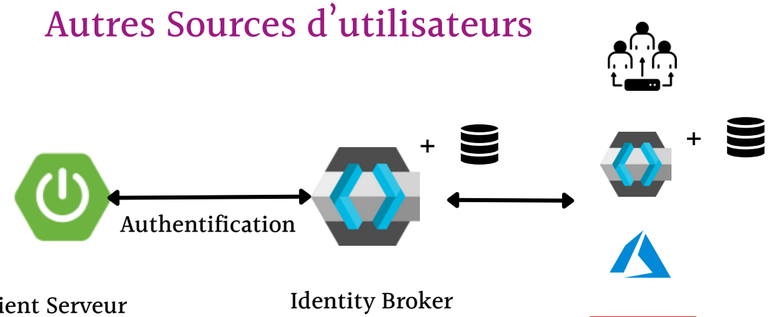
Keycloak peut jouer les deux roles:

Utilisateur dans la base de données de keycloak

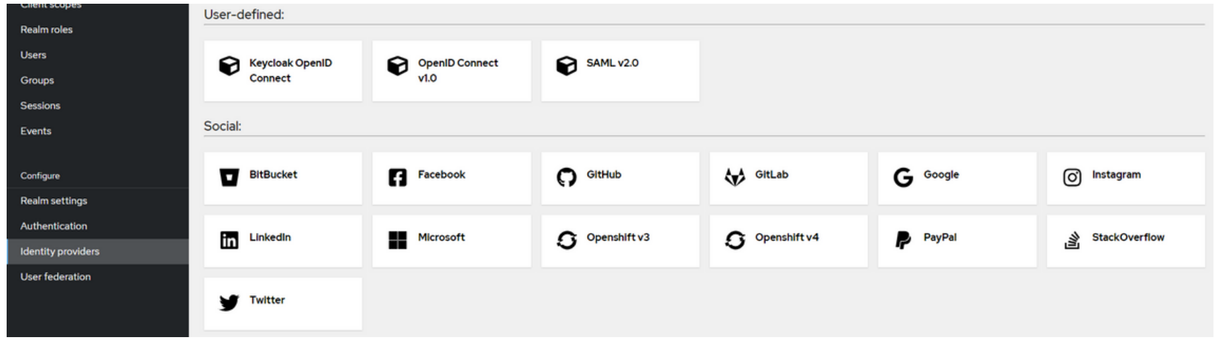


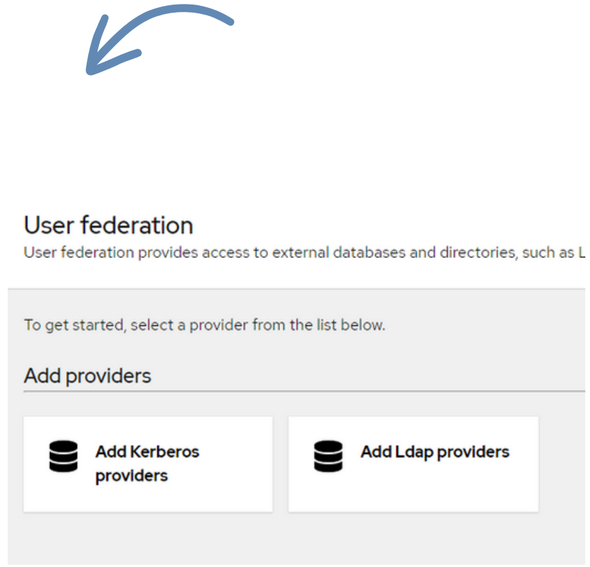
Client Serveur Identity Provider

Autres Sources d'utilisateurs



Client Serveur Identity Broker Identity Provider



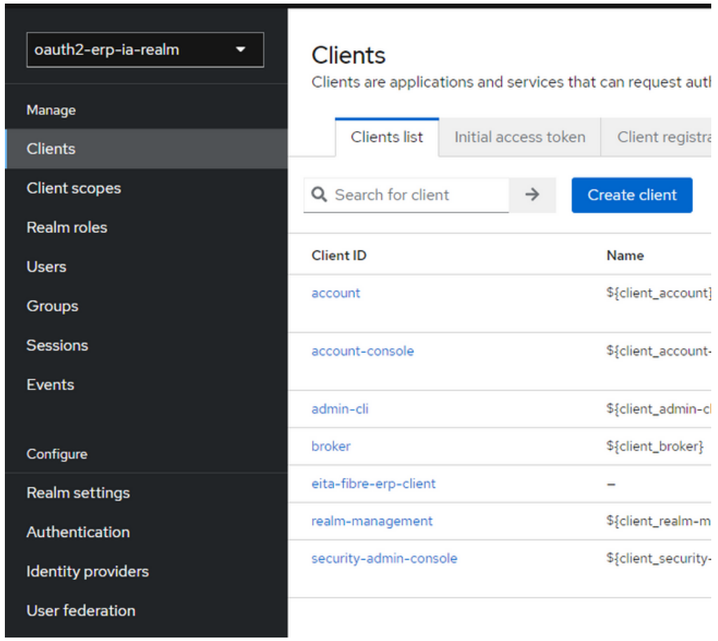


Keycloak + Code? Avec les adaptateurs

Les adaptateurs Keycloak sont des bibliothèques ou des modules qui facilitent l'intégration de Keycloak avec différentes technologies et plates-formes (Java, JavaScript, Node.js, Spring, etc.).

Keycloak : Notion de client?

Un client dans Keycloak représente une application ou un service qui a besoin d'authentifier ses utilisateurs et d'accéder à des ressources protégées.



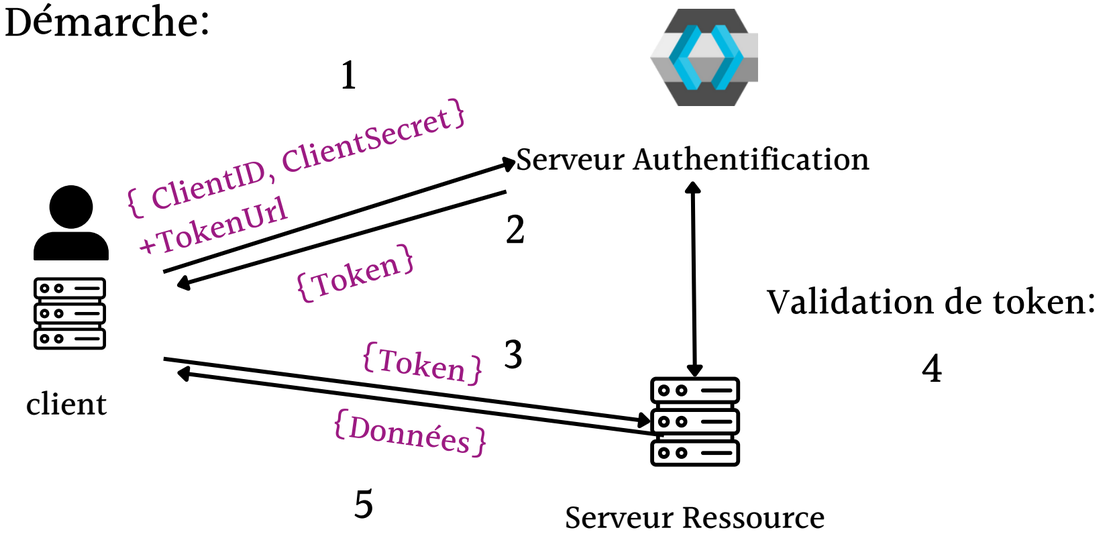
Les clients peuvent être des applications web, des applications mobiles, des services back-end, etc.

Les clients utilisent Keycloak comme serveur d'authentification et d'autorisation. Ils sont configurés pour interagir avec Keycloak en utilisant des protocoles tels que OAuth 2.0 et OpenID Connect.

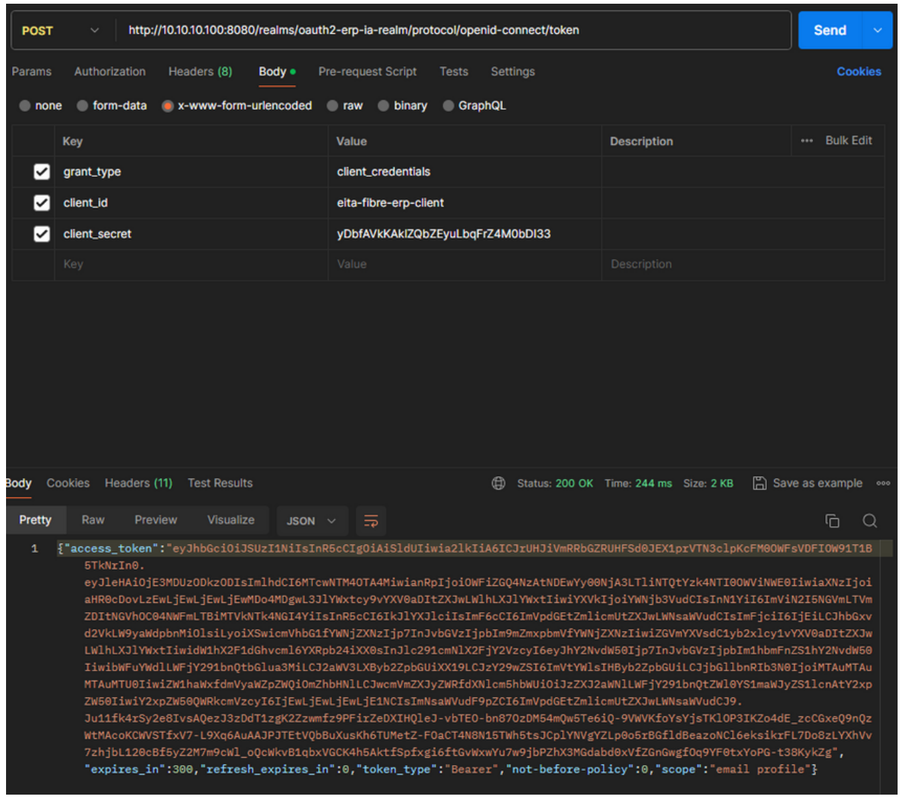
Keycloak : OpenID Connect (OIDC)

OpenID Connect (OIDC) est un protocole d'authentification basé sur le standard OAuth 2.0 a travers un ClientID et ClientSecret.

Démarche:



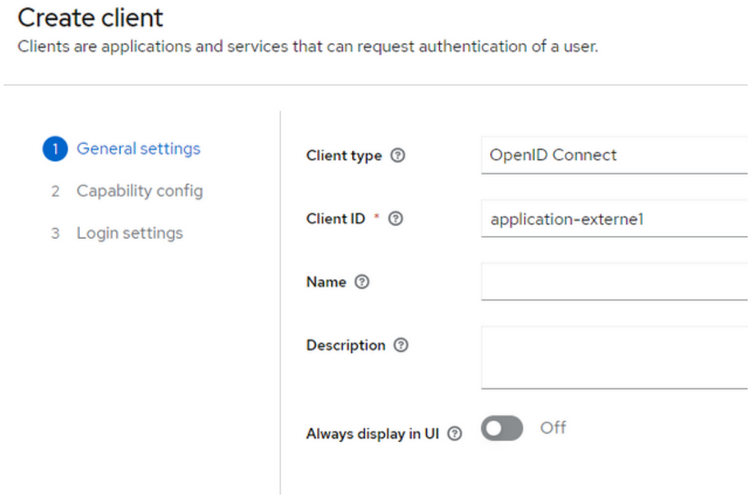
Simulation de la phase 1:



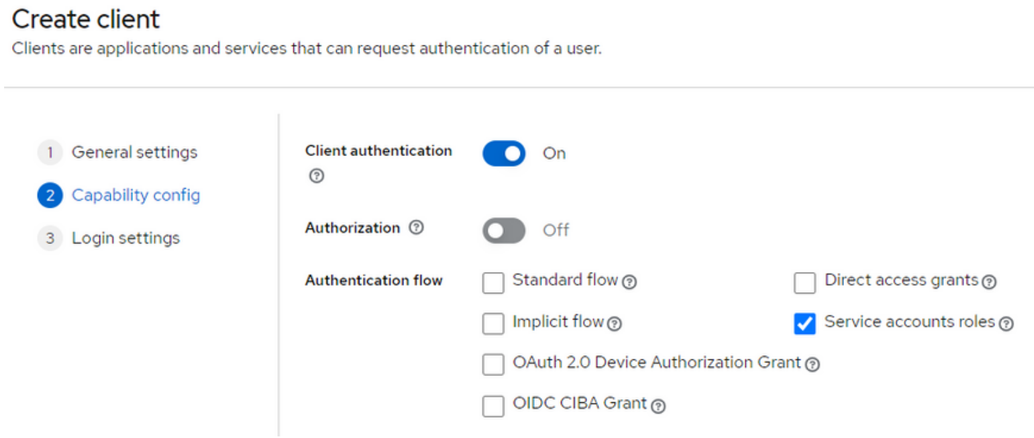
Keycloak : Creation d’un client

Démarche:

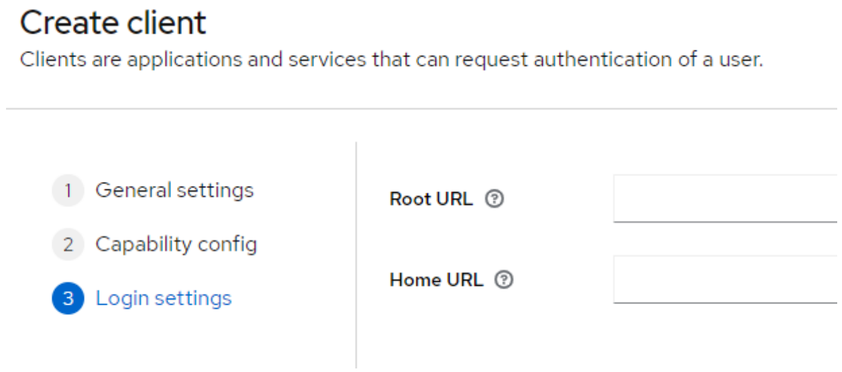
1 : Definir clientID



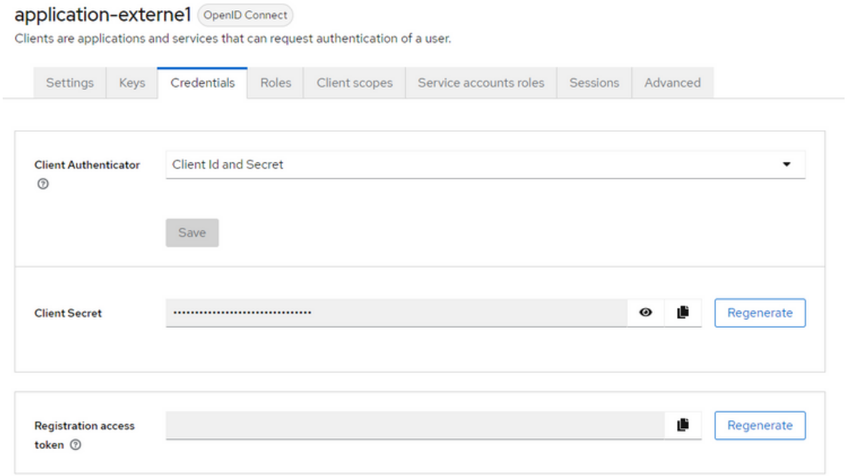
2 Definir type d’authentification



3



4 : fin creation



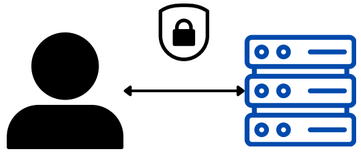
Problématique :

Authentification / Autorisation ?

avec quoi?

Client: Utilisateur

Accès limité (n fois) a une interface | API



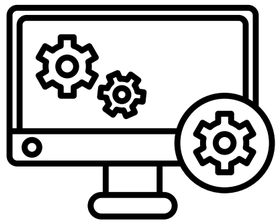
Serveur :
Ressource / Service (IA)

avec quoi?

Client: Application / Serveur

Accès illimité / longue durée

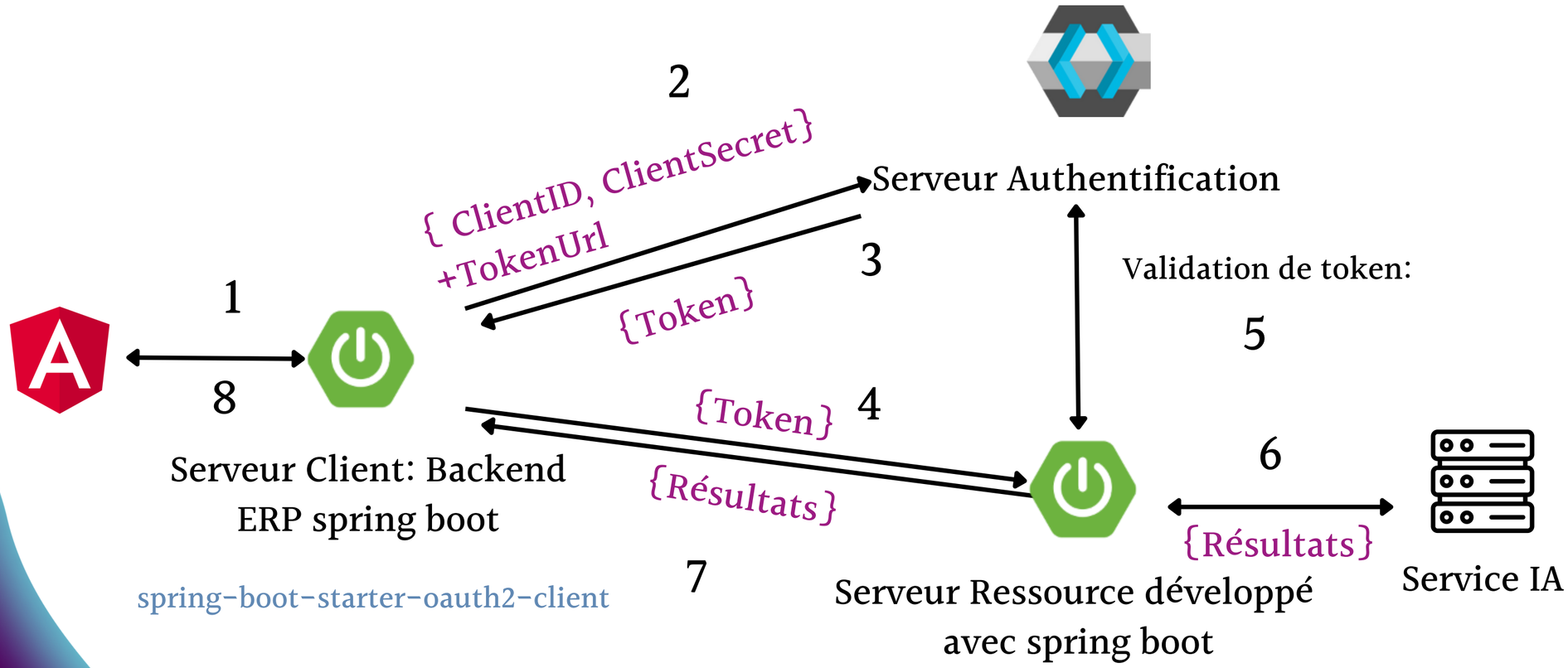
Serveur :
Ressource / Service (IA)



Interface administration pour créer des clients?

Keycloak : Architecture proposée en ERP:

Démarche:



Keycloak : Configuration

serveur client spring:
application.properties

```
#------------------ Auth server config------------------
erp.ia.security.client.clientId=eita-fibre-erp-client
erp.ia.security.client.clientSecret=yDbfAVkKAklZQbZEyuLbqFrZ4M0bDI33
erp.ia.security.client.tokenUri=https://api.eitafibre.fr/token

#--------------- IA Ressource Server ------------------
erp.ia.security.resource.amende.uri=https://api.eitafibre.fr/api/amende/extract
erp.ia.security.resource.fps.uri=https://api.eitafibre.fr/api/fps/extract
```

serveur ressource spring:
application.properties

```
spring.security.oauth2.resourceserver.jwt.jwk-set-uri=http://10.10.10.100:8080/realms/oauth2-erp-ia-realm/protocol/openid-connect/certs
service.ia.amende.url= http://10.10.10.100:8888/amende/extract
service.ia.fps.url= http://10.10.10.100:8888//fps/extract
```

Realm: En Keycloak realm regroupe un ensemble d'utilisateurs, clients, de rôles . Chaque realm ses politiques de sécurité.

Dans notre cas : Le client “eita-fibre-erp-client” appartient a le realm “oauth2-erp-ia-realm”

Les clients d'autres realms dans le même serveur d'authentification (notre Keycloak) ne peuvent pas accéder a ce serveur de ressources. Autrement dit, jusqu'à présent, chaque serveur de ressources avait un realm unique dans Keycloak.

Questions?