



Keycloak

IDENTITY AND ACCESS MANAGEMENT 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.



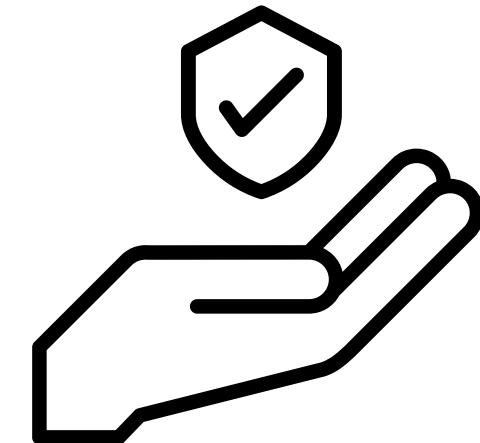
- Développé par **Red Hat** en **septembre 2014** en utilisant **Java**.
- Plus de 6 000 forks et 18 700 étoiles dans le dépôt officiel sur GitHub.



- Facile à commencer à l'utiliser en utilisant 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'authentification
(OpenID Connect, OAuth 2.0, SAML)

Gère les tokens (JWT) et les sessions

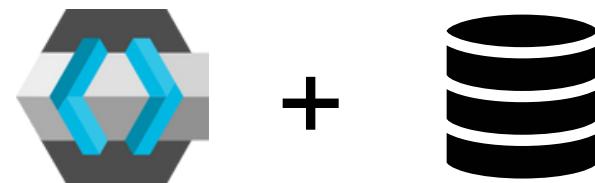
Multi-factor-authentication (MFA)

Single sign-on (SSO)

Gestion des rôles et groupes des utilisateurs

User Federation

Utilisateurs enregistrés par keycloak | interne



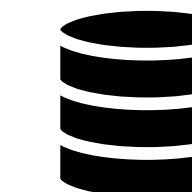
Keycloak

Base de données RDBMS /
NoSql de Keycloak

Autres sources d'utilisateurs | externe



LDAP Active directory



Base de données
RDBMS / NoSql



Réseaux sociaux

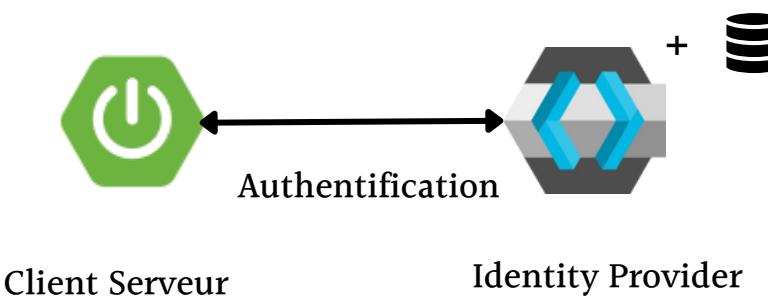
A screenshot of the Keycloak management interface. The sidebar menu includes: master, Manage, Clients, Client scopes, Realm roles, Users, Groups, Sessions, Events, Configure, Realm settings, Authentication, Identity providers, and User federation.

Keycloak : Identity provider ou Identity broker? les deux

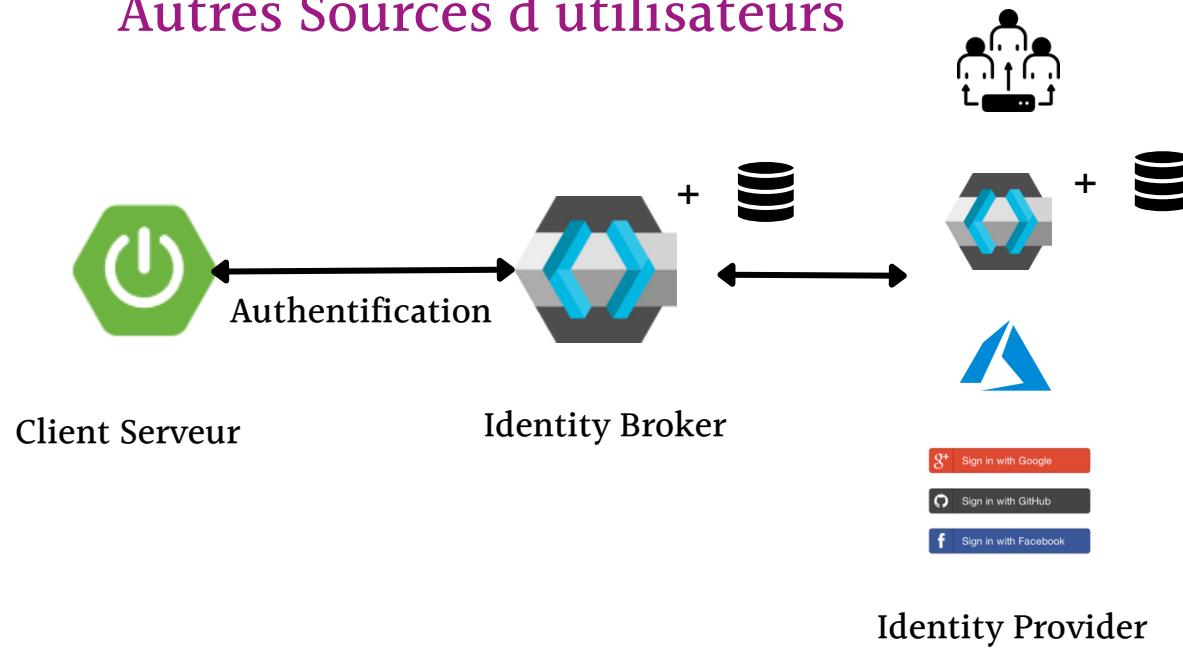


Keycloak peut jouer les deux rôles:

Utilisateur dans la base de données de keycloak



Autres Sources d'utilisateurs



A screenshot of the Keycloak configuration interface. On the left, a sidebar menu includes "Client scopes", "Realm roles", "Users", "Groups", "Sessions", "Events", "Configure", "Realm settings", "Authentication", "Identity providers" (which is currently selected), and "User federation". The main panel shows sections for "User-defined:" and "Social:". Under "User-defined:", there are three boxes: "Keycloak OpenID Connect", "OpenID Connect v1.0", and "SAML v2.0". Under "Social:", there are six boxes: "BitBucket", "Facebook", "GitHub", "GitLab", "Google", "Instagram", "LinkedIn", "Microsoft", "Openshift v3", "Openshift v4", "PayPal", and "StackOverflow". At the bottom, there is a "Twitter" box.

User federation

User federation provides access to external databases and directories, such as LDAP, Kerberos, and SAML.

To get started, select a provider from the list below.

Add providers

Add Kerberos providers

Add Ldap providers

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**.

The screenshot shows the Keycloak administration interface under the 'Manage' tab. On the left, a sidebar lists 'Clients' as the selected item, along with other options like 'Client scopes', 'Realm roles', 'Users', 'Groups', 'Sessions', 'Events', 'Configure', 'Realm settings', 'Authentication', 'Identity providers', and 'User federation'. The main area is titled 'Clients' with the subtitle 'Clients are applications and services that can request auth'. It includes tabs for 'Clients list', 'Initial access token', and 'Client registration'. A search bar 'Search for client' and a 'Create client' button are at the top. Below is a table listing clients:

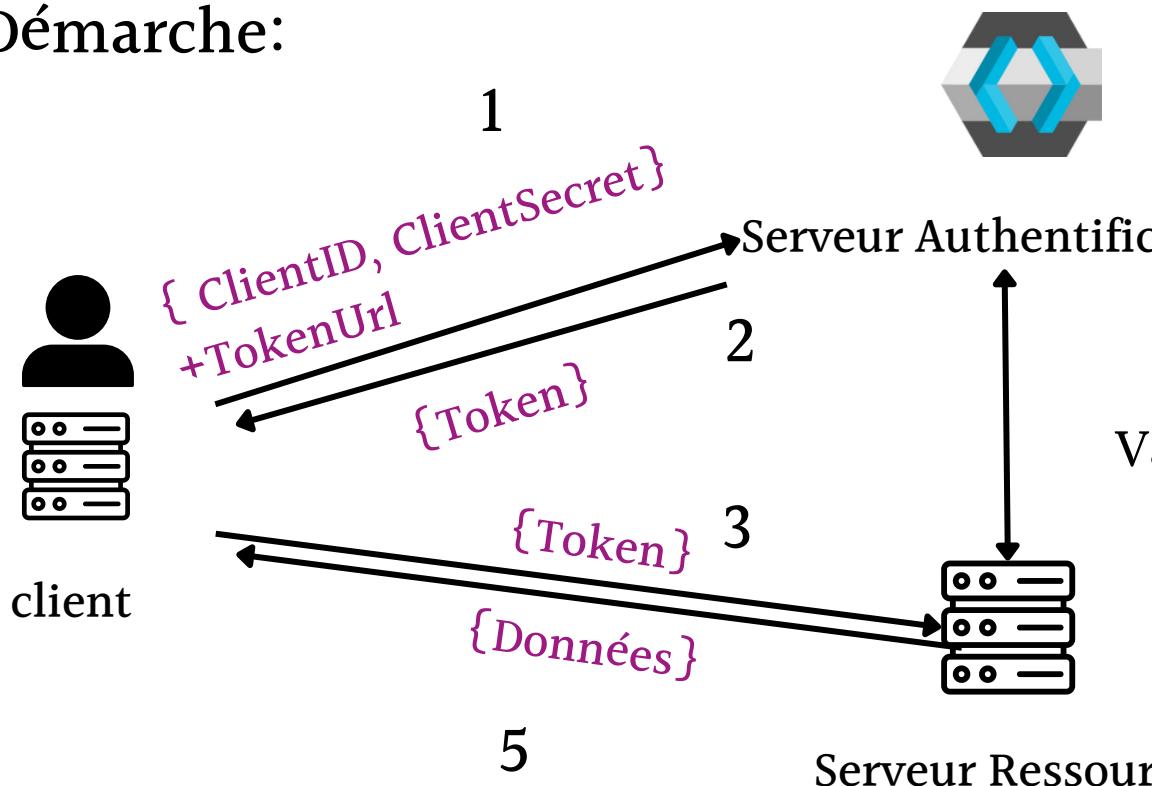
Client ID	Name
account	\${client_account}
account-console	\${client_account-}
admin-cli	\${client_admin-cl}
broker	\${client_broker}
eita-fibre-erp-client	-
realm-management	\${client_realm-m}
security-admin-console	\${client_security-}

Keycloak : OpenID Connect (OIDC)



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

Démarche:



Simulation de la phase 1:

POST http://10.10.10.100:8080/realm/erp-ia-realm/protocol/openid-connect/token

Body (x-www-form-urlencoded)

Key	Value
grant_type	client_credentials
client_id	elita-fibre-erp-client
client_secret	yDbfAVkKAKiZQbZEyuLbqFrZ4M0bDI33

Status: 200 OK Time: 244 ms Size: 2 KB

```
access_token: eyJhbGciOiJSUzI1NiIiSInR5cCIgOiAiSlldUiwiia2lkIiA6ICJrxUhJiVmRrbGZRUHFsD0JEX1prVTN3clpKcFM00WFsVDFIOW91T1B5TkNrIn0.eyJleHAiOjE3MDUzODkzODIsImhdCI6MTcwNTM40TA4MiwianRpIjoiOWFizGQ4NzAtNDEwYy00MjA3LTl1intQtYzk4NTI0OWViNWEOIiwiiaXNzIjoi aHR0cDovLzEwljEwljEwMdo4MDgwL3J1Ywxtcy9YXv0adItzXjwLwlhX3l1YwxtiwiYXvkiijo1YWNjb3VudCisIn1Yi16imViN2I5GVmLTwm ZDItNGVhOC04NWFmLTBiMTVkNTk4NGI4YiIsInR5cCI6Ikj1YXJlciiIsImF6cCI6ImVpdGEtZmlcmUtZXJwLwnsaWvdCisImFjci6IjEiLCjhbgxv d2VklW9yawdpbni0lsilyoixSwicmVhbg1fYWNjZXNzIjp7InJvbGVzIjpbim9mZmpbmVfYWNjXNzIiwiGVmYXVsdC1yb2xlcylvYXV0aDltZXJw LwlhXJ1YwxtIiwidW1hX2F1dGhcm16YXRp24ixX0sInJlc291cmNlx2FjY2VzcI6eyJhY2NvdW50Ijp7InJvbGVzIjpbIm1hbmFnZS1hY2NvdW50 IiwbWFuYwdlILWfjY291bnqtbhglua3MilcJ2aW3LXByb2ZpbGUixX9LCJzY29wZSI6ImVtYwlsIHBByb2ZpbGUilCJjbG1bnRIB3N0Ijo1TAUMTAu MTAUMTU0IiwiZw1haWxfdmVyaWZpZWQi0mZhbHNlLCJwcmVmZKJyZWRfdXNlcm5hbWUi0iJzXJ2aWnlWfjY291bnQtZw10YS1maWjyZS1lcnAtY2xp ZW50IiwiY2xpZW50QWRkmVzcyI6IjEwljE1NCiIsImNsawVudF9pZCI6ImVpdGEtZmlcmUtZXJwLwnsaWvdCj9. Ju11fk4rSy2e8IvsA0ezJ3zDdt12gK2Zzwmfz9PfirZe0X1hQleJ-vbTE0-bn870zDM54mQw5Te6iQ-9VVVKf0ysYjsTK10P3IKZo4dE_zcCGxe9nQz WtMAcoKCWVSTfxV7-L9Xq6AuAAJPJTETVqb8JusKh6TUMetz-FoaCT4N8N15TWh5tsJcp1YNVgYZLp0o5rBGfldBeazoNC16eksikrFL7Do8zLYXhvVv 7zhjbL120cBf5yZ2M7m9cWl_oqcwkvb1qbxxVGCK4h5AktfSpfxg16ftGvwxxwuy7w9jbPZhK3Mdabd0xVfZGngwfQ9YF0txYoPG-t38kykzg", "expires_in":300, "refresh_expires_in":0, "token_type": "Bearer", "not-before-policy":0, "scope": "email profile"}
```

Keycloak : Creation d'un client



Démarche:

1 : Definir clientId

Create client

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

1 General settings

Client type: OpenID Connect

Client ID: application-externe1

Name:

Description:

Always display in UI: Off

2 Definir type d'authentification

Create client

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

1 General settings

Client authentication: On

2 Capability config

Authorization: Off

3 Login settings

Authentication flow:

- Standard flow
- Implicit flow
- OAuth 2.0 Device Authorization Grant
- Service accounts roles
- OIDC CIBA Grant

3

Create client

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

- 1 General settings
- 2 Capability config
- 3 Login settings

Root URL:

Home URL:

4 : fin creation

application-externe1 (OpenID Connect)

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

Settings Keys **Credentials** Roles Client scopes Service accounts roles Sessions Advanced

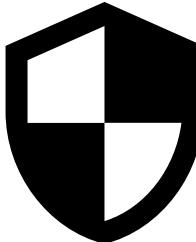
Client Authenticator: Client Id and Secret

Save

Client Secret: Regenerate

Registration access token: Regenerate

Problématique :

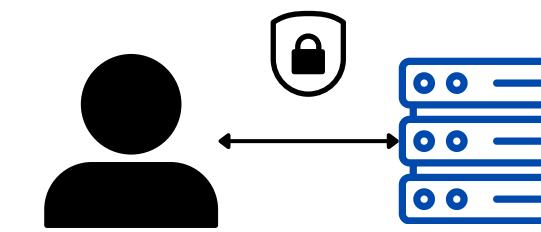


Authentification / Autorisation ?

Client: Utilisateur

Accès limité (n fois) à une interface | API

avec quoi?



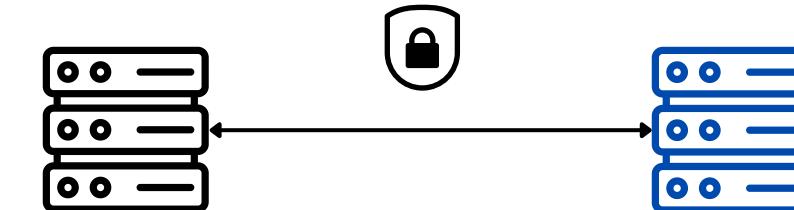
Serveur :

Ressource / Service (IA)

Client: Application / Serveur

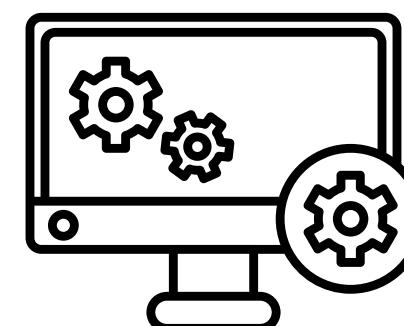
Accès illimité / longue durée

avec quoi?



Serveur :

Ressource / Service (IA)



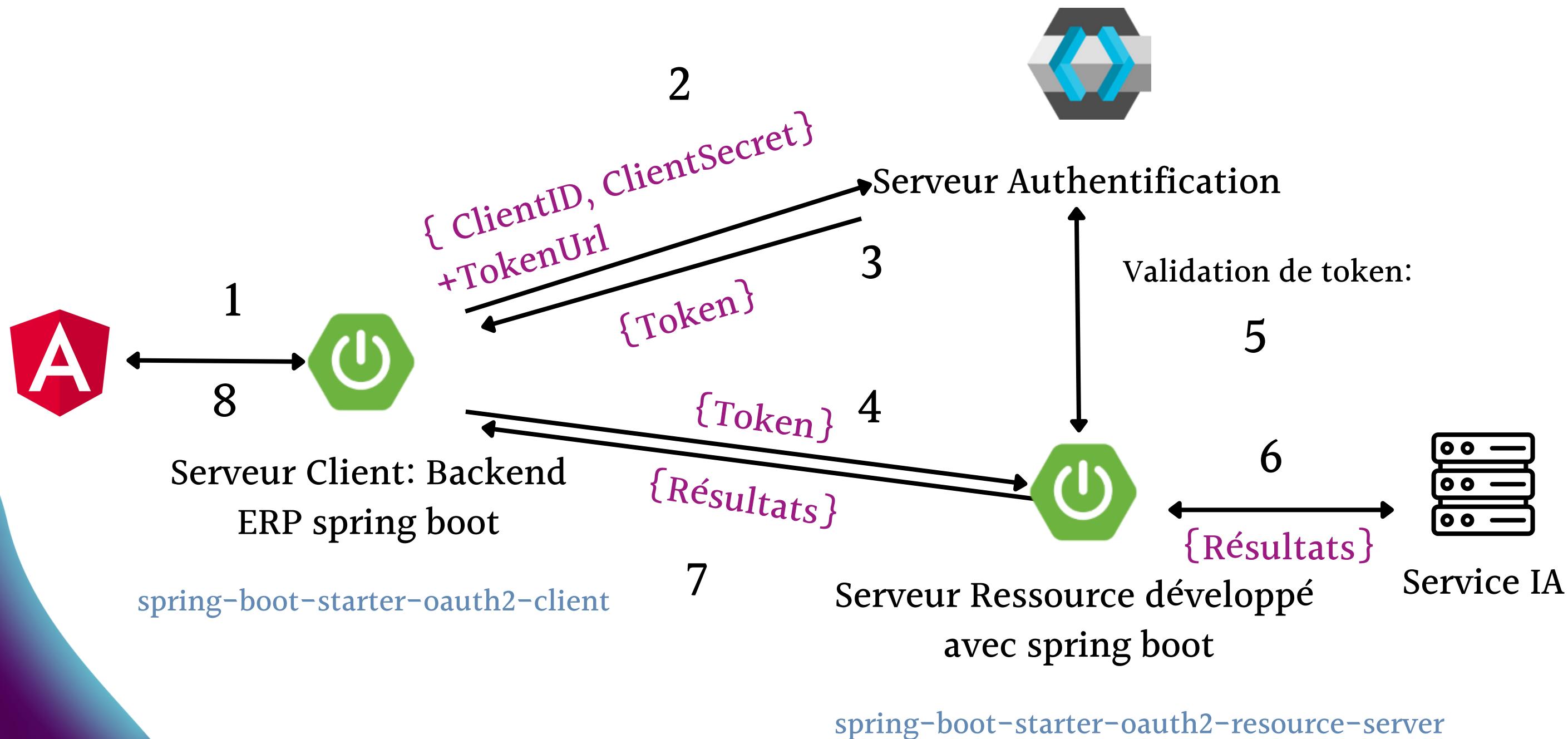
Interface administration pour créer des clients?





Keycloak : Architecture proposée en ERP:

Démarche:





!

Keycloak : Configuration

serveur client spring:
application.properties

```
61 ----- Auth server config-----
62 erp.ia.security.client.clientId=eita-fibre-erp-client
63 erp.ia.security.client.clientSecret=yDbfAVkKAkLZQbZEyuLbqFrZ4M0bDI33
64 erp.ia.security.client.tokenUri=https://api.eitafibre.fr/token
65
66 ----- IA Ressource Server -----
67 erp.ia.security.resource.amende.uri=https://api.eitafibre.fr/api/amende/extract
68 erp.ia.security.resource.fps.uri=https://api.eitafibre.fr/api/fps/extract
69
70
```

serveur ressource spring:
application.properties

```
spring.security.oauth2.resourceserver.jwt.jwk-set-uri=http://10.10.10.100:8080/realm/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?