

Mon Compte Mobilité

Enregistrement du MAAS dans IDP

# 

# Introduction au document

Ce document décrit les étapes à suivre pour enregistrer une application cliente telle q’un MaaS dans le fournisseur d’identité de MOB.

Seuls les paramètres indiqués sont à saisir (attention à la casse).

Les paramètres non listés sont ceux par défaut.

Au moins un client doit être enregistré manuellement dans l’IDP de MOB par l’intermédiaire de l’équipe MCM. Ils sont de 2 types.

## Client public

Ce type de client OIDC est plutôt à destination des applications Front. Celle-ci doit indiquer à MOB les URL de callback (l'URL de redirection vers laquelle l'internaute est redirigé une fois authentifié) et l’URL de son site.

Il permet le flux « Authorization Code Flow » et de récupérer un jeton de longue durée (si demandé).

A la création, l’équipe MCM renseigne notamment le client ID et le pattern des URLs valides de redirection.

Le client ID est alors communiquée de façon sécurisée par l’équipe MCM.

## Client confidentiel

Ce type de client OIDC est plutôt à destination des applications Backend. Il correspond à un compte de service.

Il permet le flux « Authorization Code Flow », le flux « Client Credentials » et de récupérer un jeton de longue durée (si demandé).

A la création, l’équipe MCM renseigne notamment le client ID et génère un client secret.

Une fois créée, le client ID et le client secret sont alors communiqués de façon sécurisée par l’équipe MCM.

# Client Scopes pour le consentement utilisateur

Dans cette section, on va s’intéresser à la création des Client Scopes à afficher dans la page de consentement présentée à l’utilisateur lorsqu’il demande à lier son compte MOB à son compte MaaS.

Ci-dessous les Client Scopes à ajouter.

## Client Scope « Liste des Aides »

**Name :** incentives\_list  
**Description :** Accès au catalogue d'aides publiques et employeurs  
**Protocol :** openid-connect  
**Display On Consent Screen :** ON  
**Consent Screen Text :** ${consentIncentivesList}  
**Include In Token Scope :** ON  
**GUI order :** 2

## Client Scope « Transmission de Métadonnées de Souscriptions »

**Name :** subscriptions\_metadata  
**Description :** Transmission automatique des justificatifs d'achats de votre APP de mobilité  
**Protocol :** openid-connect  
**Display On Consent Screen :** ON  
**Consent Screen Text :** ${consentSubscriptionsMetadata}  
**Include In Token Scope :** ON  
**GUI order :** 3

## Client Scope « Souscription »

**Name :** subscriptions\_process  
**Description :** Souscription à une demande d'aide  
**Protocol :** openid-connect  
**Display On Consent Screen :** ON  
**Consent Screen Text :** ${consentSubscriptionsProcess}  
**Include In Token Scope :** ON  
**GUI order :** 4

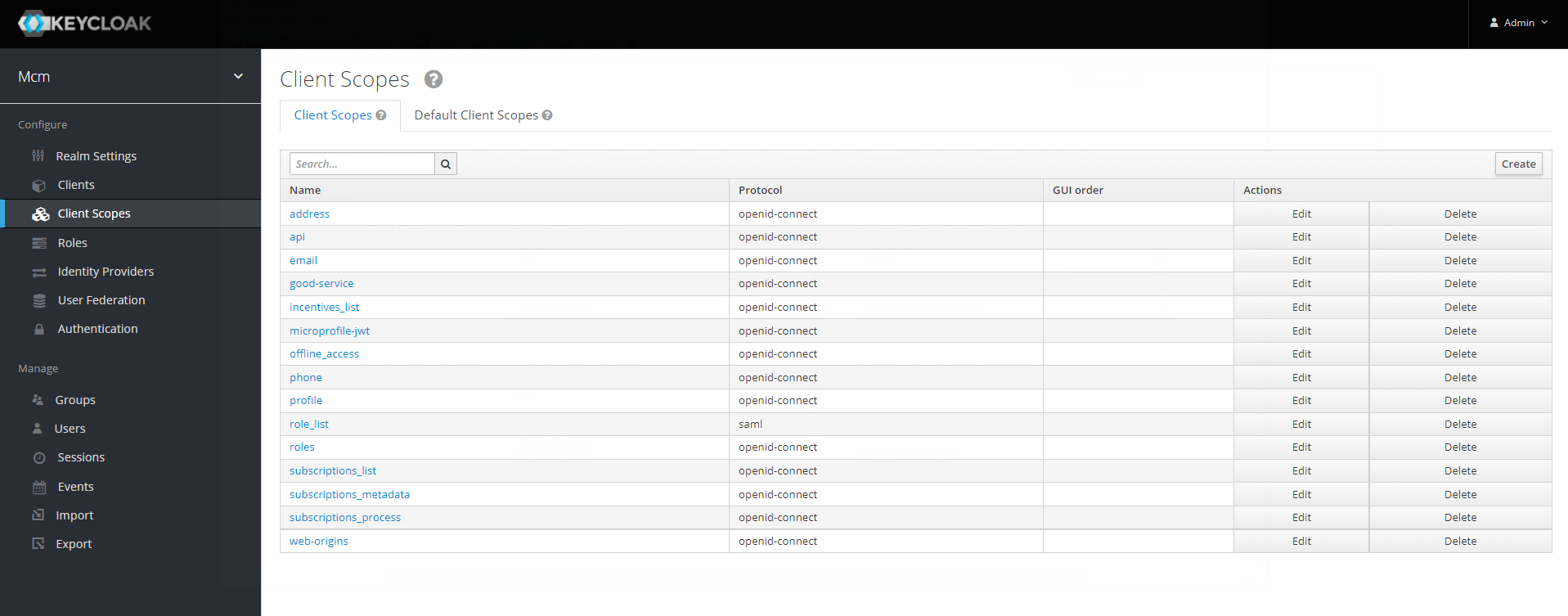
## Client Scope « Liste des Souscriptions »

**Name :** subscriptions\_list  
**Description :** Suivi de l'état de vos demandes réalisées  
**Protocol :** openid-connect  
**Display On Consent Screen :** ON  
**Consent Screen Text :** ${consentSubscriptionsList}  
**Include In Token Scope :** ON  
**GUI order :** 5

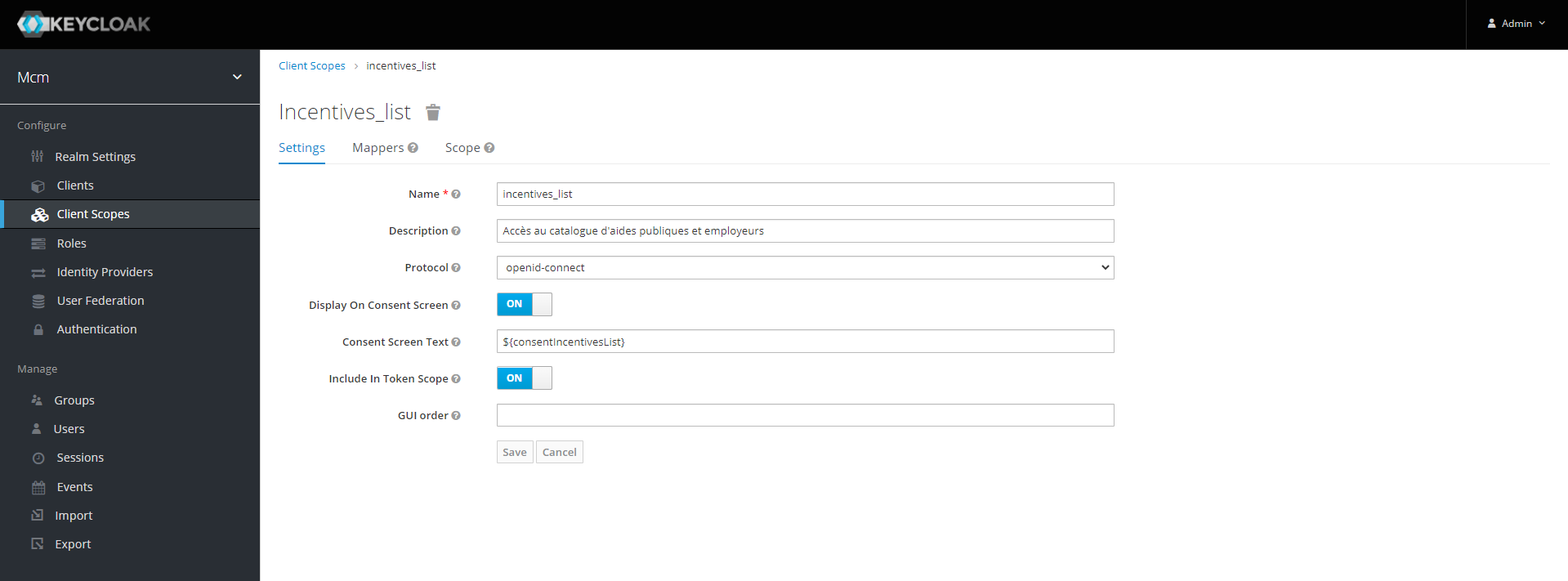
## Etapes

**!!! Les étapes suivantes sont à réaliser une seule fois par environnement, pour chacun des Client Scopes demandés !!!**

1. Aller dans la page Client Scopes via le menu principal puis cliquer sur « Create » pour ajouter un nouveau Client Scope :



1. Complétez les champs par les valeurs requises :



1. Cliquez sur « Save »

# Client(s) MAAS

## Informations requises

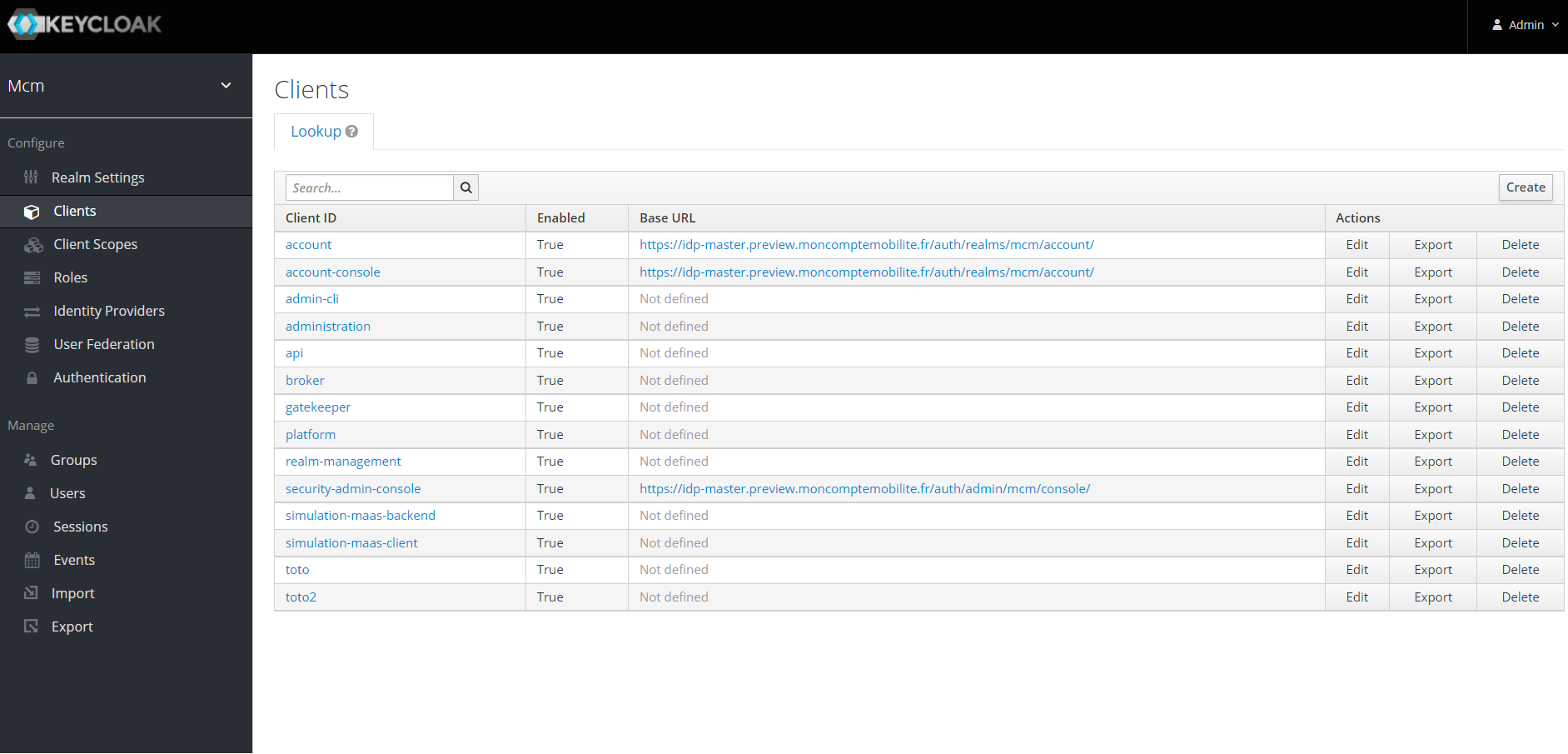
On note pour la suite les variables ci-dessous.

|  |  |  |
| --- | --- | --- |
| Nom variable | Description | Exemple |
| *MAAS* | nom court du MAAS s’interfaçant avec MOB | idfm |
| *MAAS\_LIBELLE* | nom complet du MAAS s’interfaçant avec MOB | Île-de-France Mobilités |
| *FQDN* | Nom de domaine de l’application du MAAS | com.idfm |

## Création des clients Keycloak

### Client public

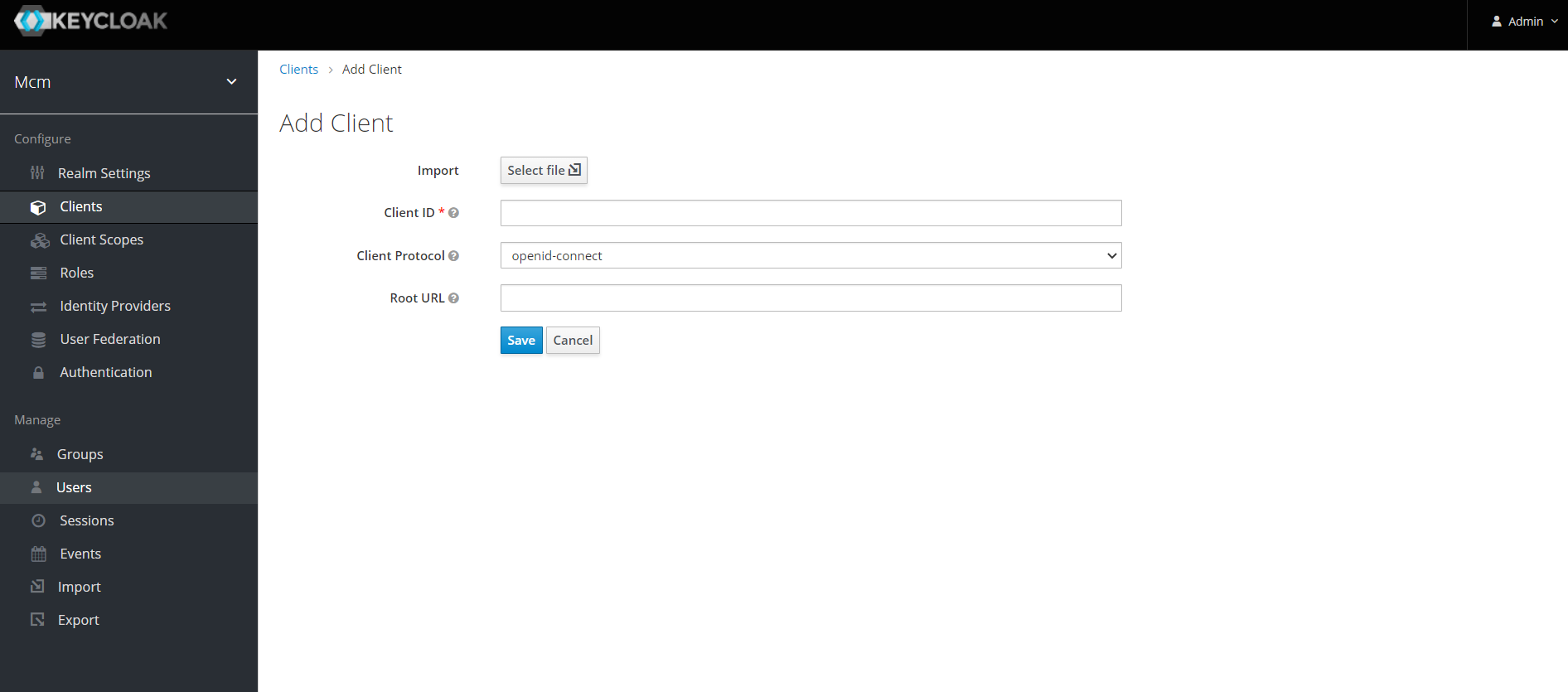
Dans cette section, on va s’intéresser à la création du client keycloak public avec la bonne configuration. Dans l’onglet Clients cliquer sur « Create ».



#### Dans la fenêtre Add Client

**Client ID** : MAAS

**Client Protocol :** openid-connect



Cliquer sur Save.

#### Onglet Settings

**Client Id**: *MAAS*-pub

**Name**: *MAAS\_LIBELLE* (ce champ est obligatoire pour RGPD)

**Enabled** : ON

**Consent Required** : ON (active la demande du consentement utilisateur)

(Ce champ affiche le texte Accès aux données de votre profil dans la page de consentement par défaut)

**Login Theme** : mcm\_template

**Client protocol** : openid-connect

**Access type** : public

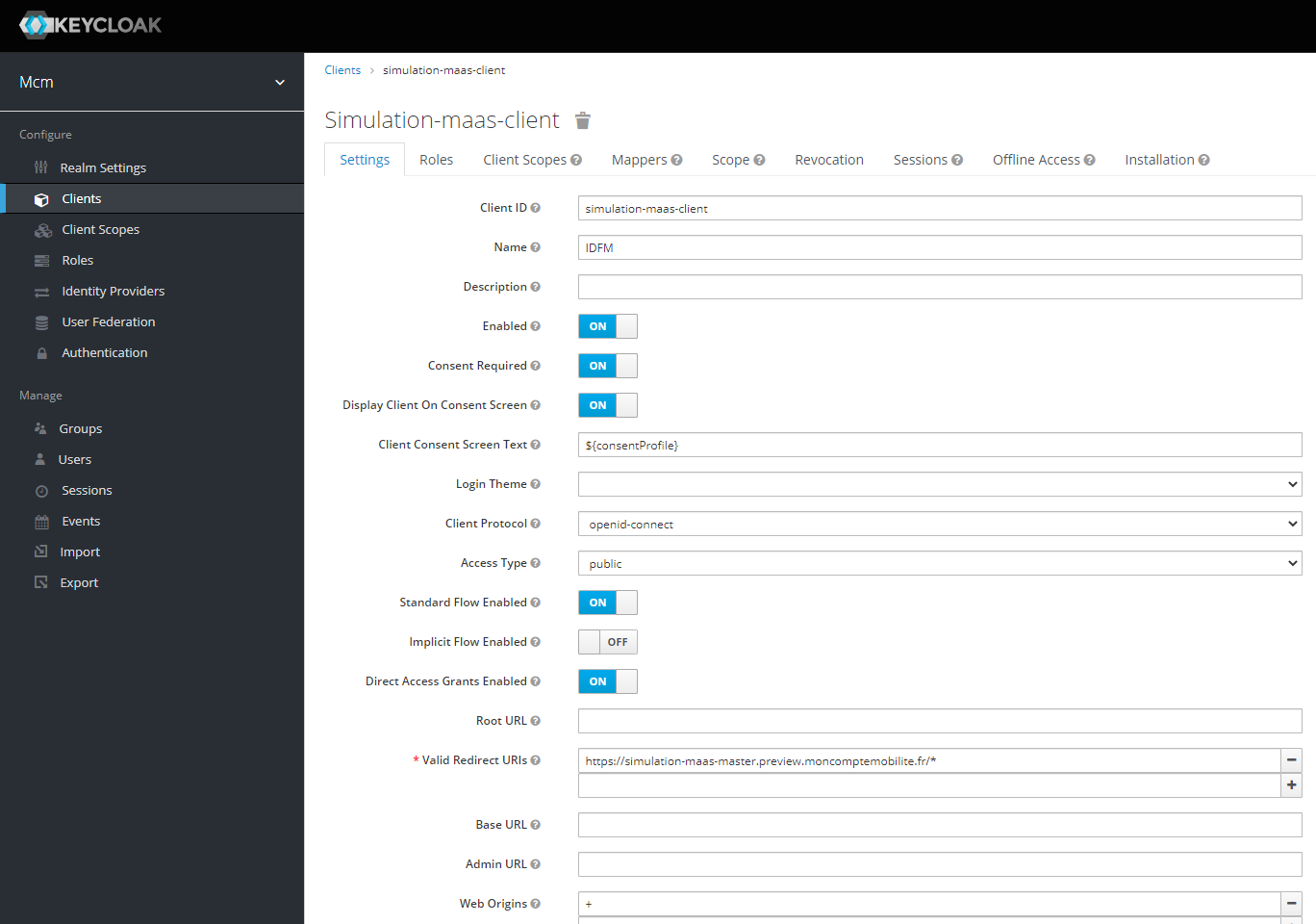
**Valid Redirect UrIs** : https://*FQDN*/\*

(Ce champ devrait être prérempli, mais vous pouvez rajouter des Urls de redirection valides autres, à adapter selon les besoins du MaaS)

**Web origins** : +

(Ce champ devrait être prérempli, mais vous pouvez rajouter des Urls d'origines valides ou simplement rajouté \* après avoir cliqué sur +)

**Backchannel Logout Session Required** : OFF

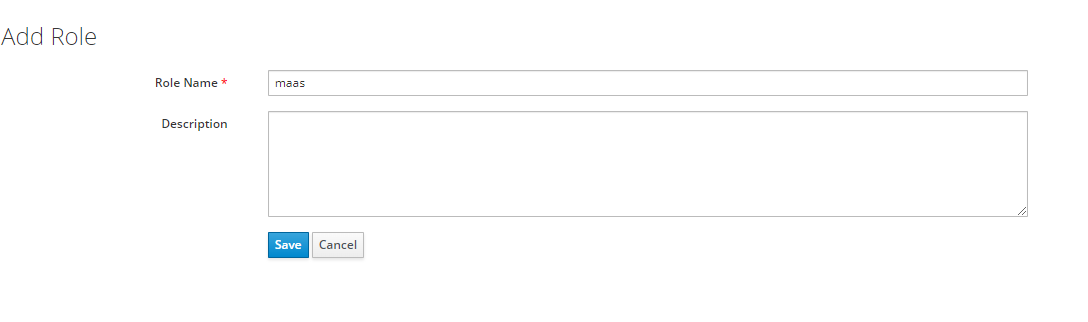
  
PS : Ne pas oublier de saisir le champ Name.

Cliquer sur Save.

#### Onglet Roles

Dans cet onglet, il faut ajouter un rôle avec le nom « maas » (en minuscules).

Cliquer sur Add Role.

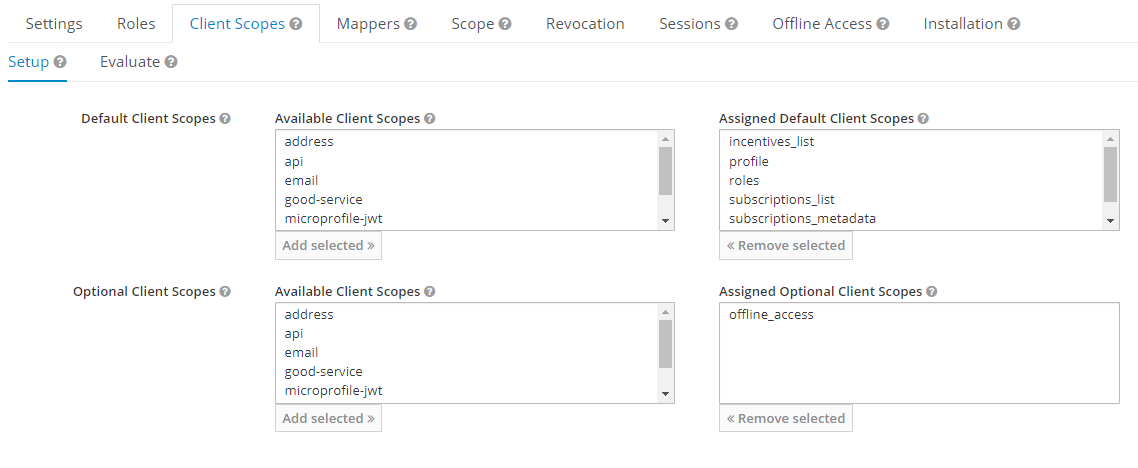


Cliquer sur Save.

#### Onglet Client Scopes

Dans cet onglet, il faut retirer les rôles non nécessaires. Le client doit avoir les scopes suivants :

* Default client scopes
  + profile
  + roles
  + selon les droits du client MaaS :
    - subscriptions\_metadata **(Transmission automatique des justificatifs d'achats de votre APP de mobilité)**
    - incentives\_list **(Accès au catalogue d'aides publiques et employeurs)**
    - subscriptions\_list **(Suivi de l'état de vos demandes réalisées)**
    - subscriptions\_process **(Souscription à une demande d'aide)**
* Assigned optional scopes
  + offline\_access



#### Onglet Mappers

Dans cet onglet, il faut ajouter 2 mappers avec la configuration ci-dessous.

##### Maas\_name

Cliquer sur Create.

**Name** : maas\_name (ce nom peut-être mis en dur car il n’appartient qu’au MAAS-pub)

**Mapper type** : Hardcoded claim

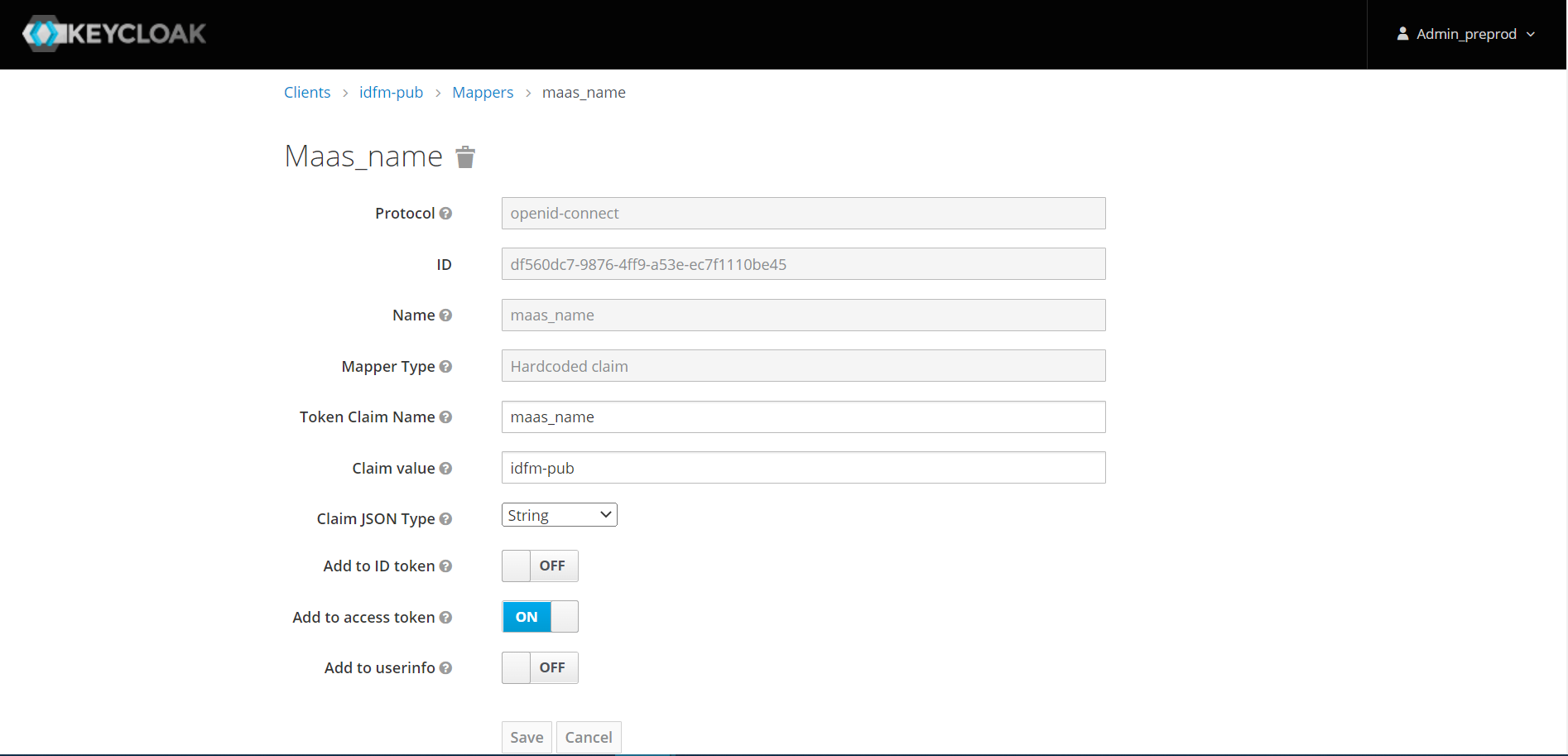
**Token claim name** : maas\_name

**Claim value** : *MAAS*-pub

**Claim json type** : string

**Add to ID token** : OFF

**Add to userinfo** : OFF



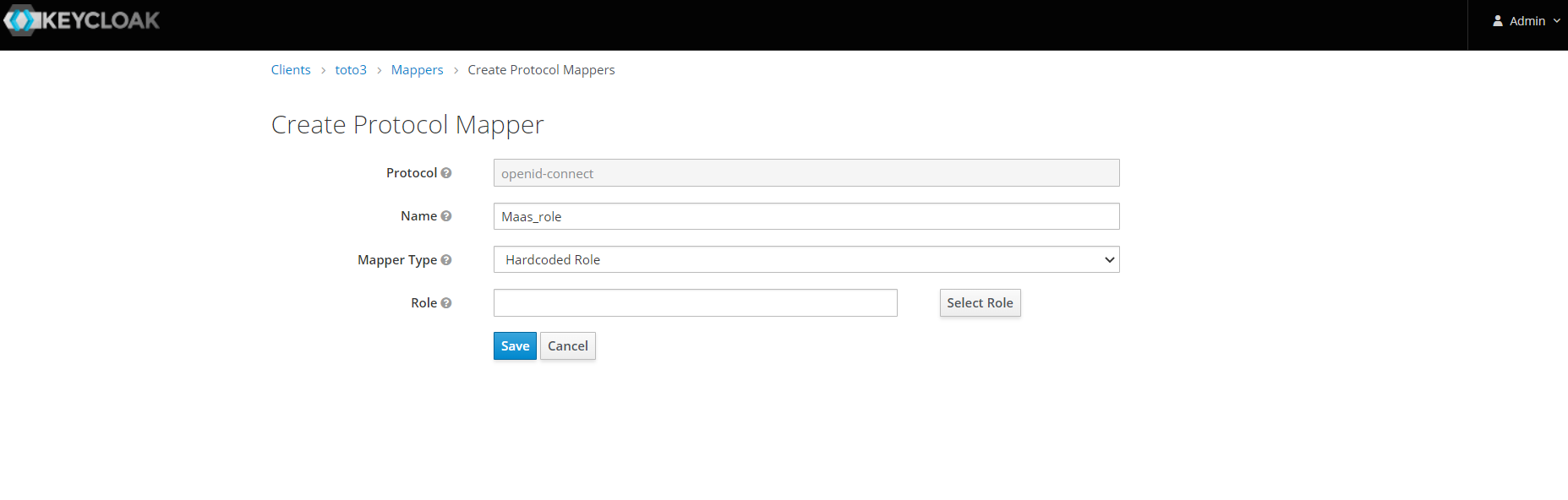
##### Maas\_role

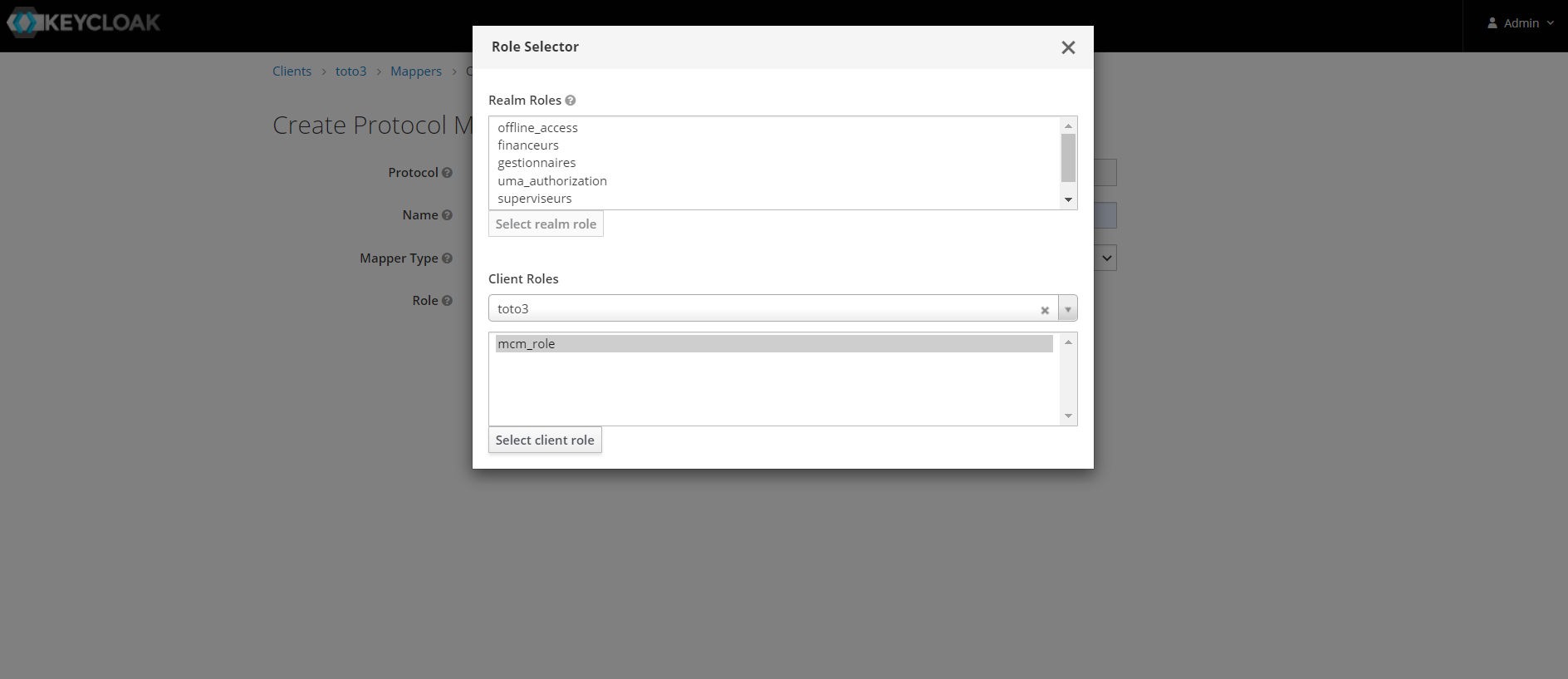
Cliquer sur Create.

**Name** : maas\_role (Ce nom peut-être mis en dur car il n’appartient qu’au MAAS-pub)

**Mapper type** : hardcoded role

**Rôle** : sélectionner le client rôle *MAAS*-pub, et sélectionner le rôle maas.





Cliquer sur Save.

##### Identity Provider

Cliquer sur Create.

**Name** : Identity Provider

**Mapper type** : User Session Note

**User Session Note** : identity\_provider

**Token Claim name** : identity\_provider

**Claim json type** : string

**Add to ID token** : ON

**Add to access token** : ON

### Client confidentiel

Dans cette section, on va s’intéresser à la création du client keycloak confidentiel avec la bonne configuration. La procédure est à réaliser depuis la même interface que pour le client public.

#### Onglet Settings

**Client Id** : MAAS-backend

**Name** : MAAS-backend (ce champ est facultatif)

**Enabled** : ON

**Login Theme** : mcm\_template

**Client protocol** : openid-connect

**Access type** : confidential

**Direct access grants enabled** : OFF

**Service account enabled** : ON

**Valid Redirect UrIs** : \*

**Backchannel Logout Session Required** : OFF

**Web origins** : +

#### Onglet Credentials

Il faut récupérer le secret et le transmettre au MaaS pour qu’ils puissent se connecter à notre fournisseur d’identité.

#### Onglet Roles

Dans cet onglet, il faut ajouter un rôle avec le nom *maas* et un autre avec le nom *service\_maas* (attention à la casse).

#### Onglet Client Scopes

Dans cet onglet, il ne faut garder que le scope *roles* dans les scopes par défaut, et dans les scopes optionnels, il ne faut garder que *offline\_access*.

#### Onglet Mappers

Dans cet onglet, il faut ajouter 2 mappers avec la configuration ci-dessous.

##### Maas\_name

Cliquer sur Create.

**Name** : maas\_name

**Mapper type** : hardcoded claim

**Token claim name** : maas\_name

**Claim value** : MAAS-backend

**Claim json type** : string

**Add to ID token** : OFF

**Add to userinfo** : OFF

Cliquer sur Save.

##### Maas\_role

Cliquer sur Create.

**Name** : maas\_role

**Mapper type** : hardcoded role

**Rôle** : sélectionner le client rôle *MAAS-backend*, et sélectionner le rôle *service\_maas*.

Cliquer sur Save.

##### groups

Cliquer sur Create.

**Name** : groups

**Mapper type** : Group Membership

**Token claim name** : membership

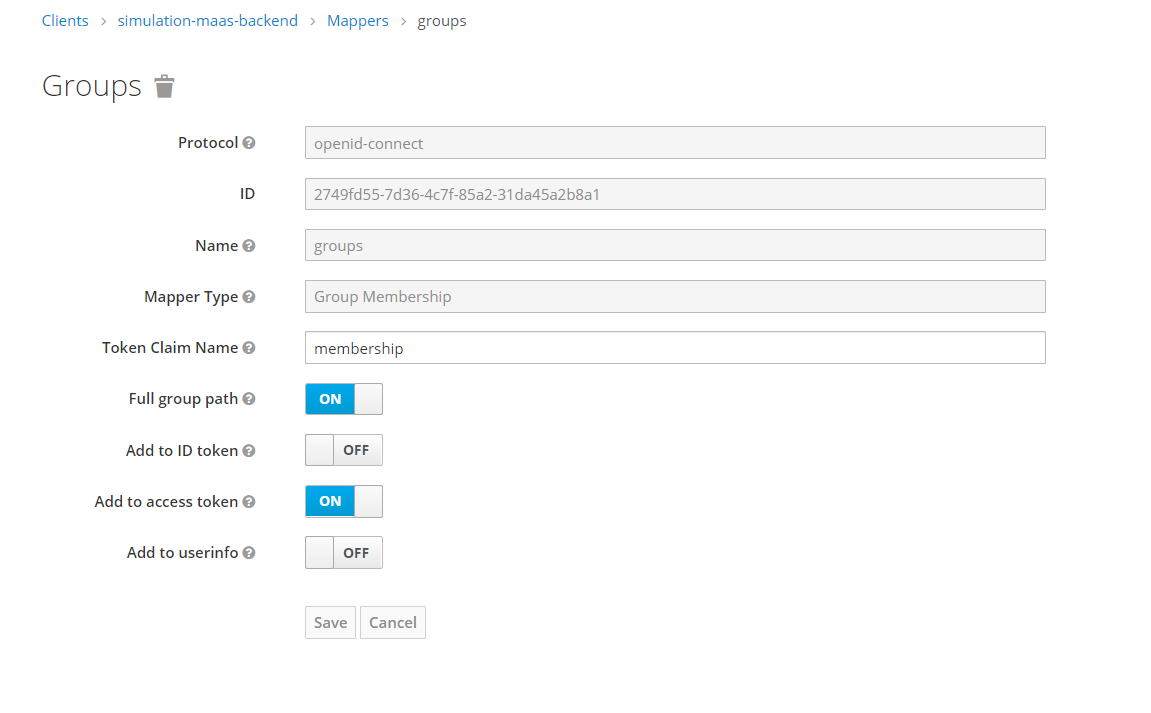
**Full group path** : ON

**Add to ID token** : OFF

**Add to access token** : ON

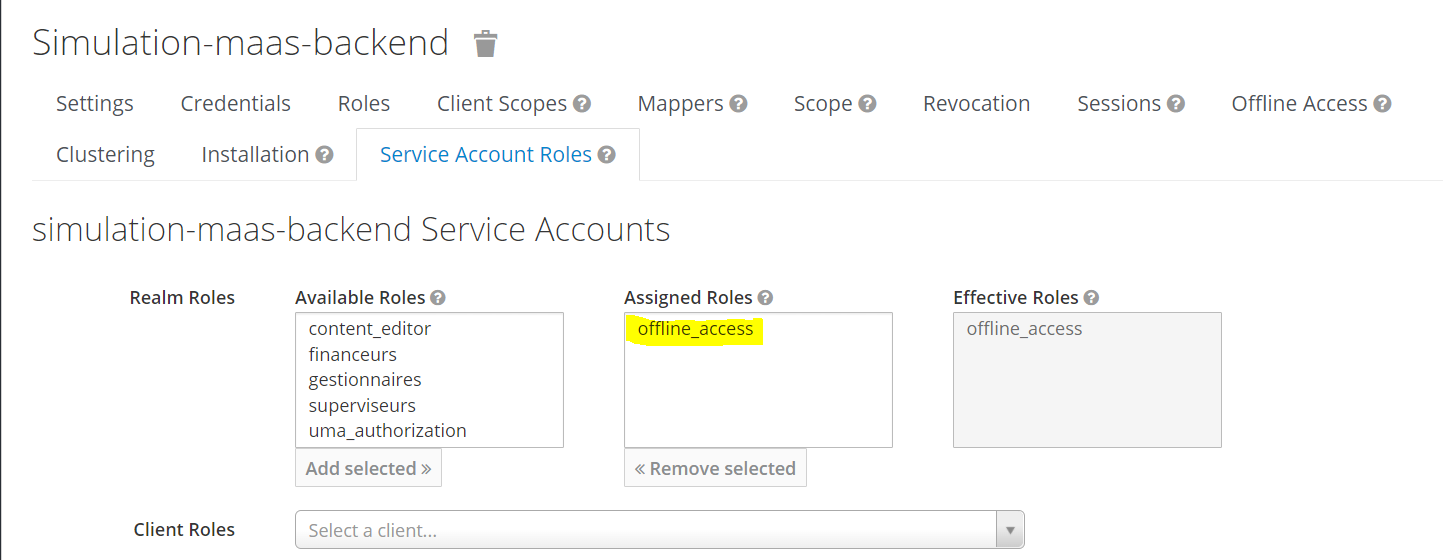
**Add to userinfo** : OFF

Cliquer sur Save.

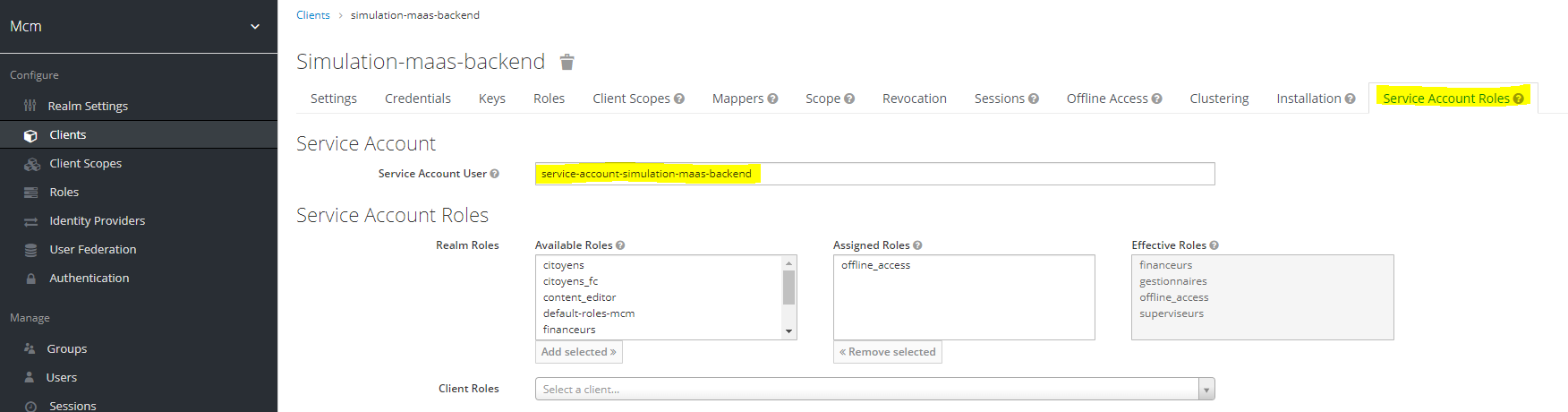


#### Onglet Service Account Roles

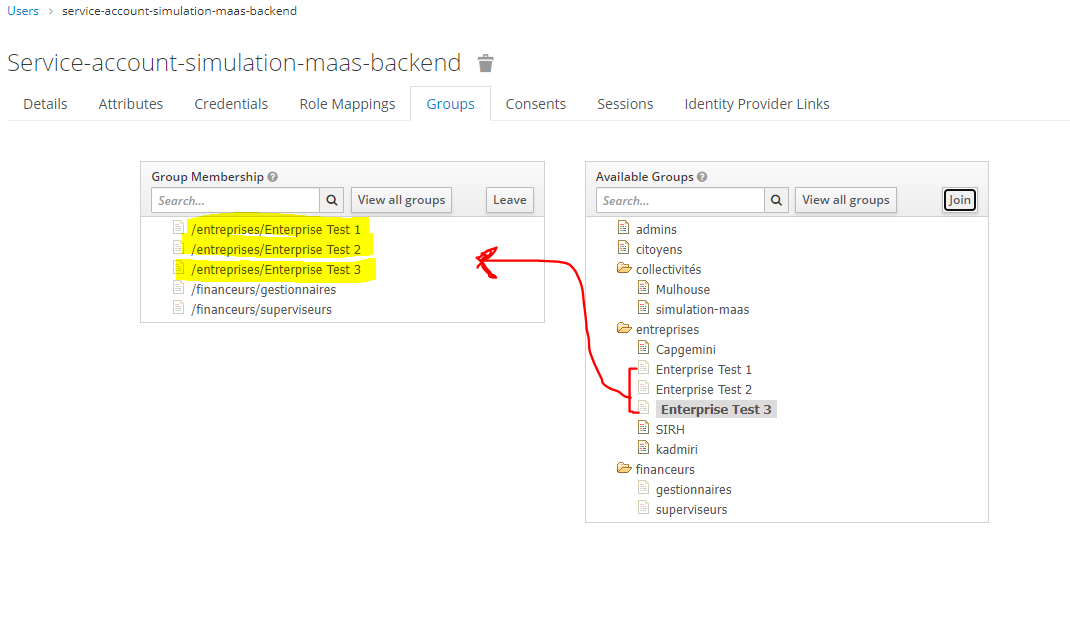
Dans cet onglet, il ne faut garder que le rôle offline\_access dans les assigned roles.



Si le MaaS est aussi financeur, cliquer sur le service account : **service-account-MAAS-backend**



Keycloak redirige vers une nouvelle page afin d’ajouter les **groups correspondant aux enterprises pour lesquelles ce client sera utilisé**. Pour chacune, la slectionner et cliquer sur Join.



# Utilisation du jeton pour invoquer l’API

Le serveur d’identification permet de donner un JWT, contenant un token de rafraichissement et un token d’accès.

Ce token d’accès doit être ajouter dans le header de tous les appels API.

Si ce token d’accès est expiré, il faudra utiliser le token de rafraichissement pour obtenir un nouveau token d’accès.

Si le mode offline\_access est activé, alors le token de rafraichissement n’aura pas de date de fin de validité, et il sera toujours valable pour demander un nouveau token d’accès.