

## Authentic 2 - Development #65064

### LDAP user : prendre en compte les attributs générés dans le mapping

10 mai 2022 11:06 - Benjamin Renard

<b>Statut:</b>	Nouveau	<b>Début:</b>	10 mai 2022
<b>Priorité:</b>	Normal	<b>Echéance:</b>	
<b>Assigné à:</b>	Benjamin Renard	<b>% réalisé:</b>	0%
<b>Catégorie:</b>		<b>Temps estimé:</b>	0:00 heure
<b>Version cible:</b>		<b>Planning:</b>	Non
<b>Patch proposed:</b>	Non		
<b>Description</b>			
Dans une de nos intégrations utilisant le backend LDAP, l'adresse mail de l'utilisateur ne peut-être récupéré directement depuis l'annuaire, mais elle est générable et générée via un attribut NG (via une fonction configurée dans <code>ATTRIBUTE_SOURCES</code> ). Seulement, les attributs NG ne sont pas pris en compte dans le mapping des attributs d'un utilisateur LDAP et l'adresse email des utilisateurs dans Authentic n'est donc pas renseignée. Cela nous empêche (entre autres) toutes utilisations de la fonctionnalité de récupération de mot de passe.			

#### Historique

##### #1 - 10 mai 2022 11:08 - Benjamin Renard

- Fichier `0001-LDAP-user-get-attributes-NG-before-mapping-65064.patch` ajouté

Ci-joint, un patch qui implémente une solution à cette problématique.

##### #2 - 10 mai 2022 11:55 - Benjamin Dauvergne

`attributes_ng` n'est pas un truc que je souhaite perpétuer, et donc je souhaite encore moins rendre le backend LDAP dépendant dessus, je serai plus ouvert à une implémentation explicite d'attributs basés sur un template, genre

```
LDAP_SETTINGS = [  
    {  
        ..  
        'generated_attributes': {  
            'email': {'tpl': "{attributes['uid']@[0]}@domain.com", 'mandatory': True}},  
    }  
]
```

avec une gestion de ceci ajoutée dans une méthode `generate_attributes()` appelée par `get_attributes()`, `mandatory=True/False` c'est pour savoir si un échec de la génération doit bloquer le login ou pas par défaut on dirait que oui.

##### #3 - 11 mai 2022 15:49 - Benjamin Renard

Benjamin Dauvergne a écrit :

`attributes_ng` n'est pas un truc que je souhaite perpétuer, et donc je souhaite encore moins rendre le backend LDAP dépendant dessus, je serai plus ouvert à une implémentation explicite d'attributs basés sur un template, genre [...] avec une gestion de ceci ajoutée dans une méthode `generate_attributes()` appelée par `get_attributes()`, `mandatory=True/False` c'est pour savoir si un échec de la génération doit bloquer le login ou pas par défaut on dirait que oui.

Aujourd'hui, un simple template de génération comme le propose ne nous suffirait pas dans un certain nombre de cas que nous avons pu rencontrer. Aujourd'hui, le fait de pouvoir écrire une fonction python pouvant implémenter une logique plus complexe est un vrai plus et ce serait vraiment dommage et pénalisant de perdre cela. En outre, la solution de génération basée sur un template est intéressante et pourrait répondre à bons nombres de cas les plus simples. Il faudrait cependant définir (et documenter) sur quelles informations seraient disponibles pour la génération de ces templates.

##### #4 - 11 mai 2022 18:09 - Benjamin Dauvergne

- Assigné à mis à Benjamin Renard

##### #5 - 12 mai 2022 09:40 - Benjamin Dauvergne

Dans ce cas je vois deux pistes :

- utiliser plutôt du template django si votre besoin se limite à des condition,
- sinon introduire les variables d'`attributes_ng` dans le contexte de formatage, `{attributes_ng['whatever']}['truc']` de ce système (quel genre de

fonction avez-vous exactement par curiosité?)

## #6 - 16 mai 2022 19:27 - Benjamin Renard

Benjamin Dauvergne a écrit :

Dans ce cas je vois deux pistes :

- utiliser plutôt du template django si votre besoin se limite à des condition,

Si ça reste pratique à l'usage pourquoi pas (je vois pas bien à quoi ressemblerait la configuration d'un tel attribut avec l'usage d'un template django).

- sinon introduire les variables d'attributs\_ng dans le contexte de formatage, {attributes\_ng['whatever']['truc']} de ce système (quel genre de fonction avez-vous exactement par curiosité?)

Hum alors là je vois encore moins :)

Un exemple tout bête de cas d'usage que j'ai eut aujourd'hui : générer un attribut *cn* SUPANN depuis *givenName* et *sn*. Selon SUPANN,  $cn = sn + givenName$ , le tout **sans accent**. Ici la méthode se contente de passer le tout à *unidecode* et le tour est joué.

Autre exemple tout aussi simple : générer un email en remplaçant le domaine d'un email récupérer depuis le LDAP.

Après, ayant déjà présenté cette fonctionnalité aux devs d'un client, eux programmat carrément d'aller appeler une API métier à eux pour récupérer des infos de la personne.

**Note** : il faut également savoir gérer les attributs multi-valués, par exemple, pour la génération de *eduPersonScopedAffiliation* depuis *eduPersonAffiliation* : Ici chaque valeur de *eduPersonAffiliation* doit être suffixée avec un domaine.

## Fichiers

---

0001-LDAP-user-get-attributes-NG-before-mapping-65064.patch	2,06 ko	10 mai 2022	Benjamin Renard
---	---------	-------------	-----------------