Authentic 2 - Development #37871

permettre d'utiliser un gabarit django pour la correspondance attribut oidc / attribut authentic (côté réception)

22 novembre 2019 09:15 - Frédéric Péters

Statut:	Fermé	**Début:**	22 novembre 2019
Priorité:	Normal	**Echéance:**	
Assigné à:	Paul Marillonnet	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

L'idée est Authentic branché à un fournisseur d'identités externe, en OIDC, quand il obtient les attributs(/claims), pouvoir les passer à travers un gabarit; cas pratique, on obtient par OIDC deux attributs "nom de naissance" et "nom d'usage", et on voudrait mettre dans nom {% firstof nom d'usage nom de naissance %}.

Demandes liées:

Lié à Authentic 2 - Development #37884: permettre d'utiliser un gabarit djan...	**Fermé**	**22 novembre 2019**

Révisions associées

Révision dccf9a57 - 02 avril 2020 16:33 - Paul Marillonnet

auth_oidc: extend mapping claim max length (#37871)

Révision c4636a41 - 02 avril 2020 16:33 - Paul Marillonnet

auth_oidc: use custom widget in claim mapping admin form (#37871)

Révision 7991c486 - 02 avril 2020 16:33 - Paul Marillonnet

auth_oidc: select from existing attributes in admin provider page (#37871)

Révision 556f3e16 - 02 avril 2020 16:33 - Paul Marillonnet

auth_oidc: add id token 'as_dict' method (#37871)

Révision 2b46edf4 - 02 avril 2020 16:33 - Paul Marillonnet

auth_oidc: render templated claim values during authn (#37871)

Historique

#2 - 22 novembre 2019 10:22 - Paul Marillonnet

Si on décide d'aller dans cette direction, le plus difficile je crois va être de proposer une interface intuitive tout en assurant la rétrocompatibilité (ne pas casser les configs OIDC existantes).
On pourrait peut-être, dans le champ attribut de la partie de définition de l'appairage lors de la configuration d'un fournisseur, accepter du langage de gabarit inline, en faisant en sorte que le contexte de rendu du bout de gabarit contienne les revendications (claims) servies par le fournisseur.

Je ne me souviens plus comment est cette partie du code A2 de la gestion de l'appairage entre revendications OIDC et attributs du profil, mais il va falloir étudier le cas où l'appairage fait intervenir en entrée des revendications du jeton ID et d'autres à aller chercher auprès de l'endpoint UserInfo (et sans doute, pour faire les choses bien, proposer un mécanisme de rendu lazy du gabarit).

#3 - 22 novembre 2019 10:30 - Paul Marillonnet

- Assigné à mis à Paul Marillonnet

En tout cas oui chouette idée :)

#4 - 22 novembre 2019 11:02 - Benjamin Dauvergne

On parle de l'IdP ou du SP OIDC ici ? Parce qu'on peut faire ça à la réception ou au départ.

#5 - 22 novembre 2019 11:04 - Frédéric Péters

"quand il obtient les attributs", j'étais sur faire ça à la réception.

#6 - 22 novembre 2019 11:31 - Paul Marillonnet

Oui c'est ce que j'avais compris aussi ("Authentic branché à un fournisseur d'identités externe").

#7 - 22 novembre 2019 11:40 - Benjamin Dauvergne

De manière à limiter les efforts coté SP (pour GLC notamment) et éviter les discussions "pourquoi on fait ça sur Toodego et pas partout" je trouverai mieux de faire ça au départ et d'étendre ainsi le profil GLC en précisant qu'il y a deux nouveaux attributs prenom / nom qui sera soit le nom de naissance soit le nom d'usage (idem pour le prénom) les deux étant toujours disponibles séparément. Le seul truc chiant c'est les noms baroques qu'on a adopté coté GLC :

- prénom / nom de naissance : first_name / last_name
- prénom / nom d'usage : preferred_givenname / preferred_username

Et donc maintenant :

- prénom / nom effectifs : prenom / nom

Après je ne sais pas comment seront gérés les formulaires avec verrouillage des champs, on ne pourra pas utiliser prenom / nom pour ceux-ci à moins d'ajouter aussi une expression pour définir le statut vérifié (en pseudo code verified = False if nom_usage else nom_de_naissance.verified)

#8 - 22 novembre 2019 12:34 - Benjamin Dauvergne

Je comprends très bien que les deux ont leur utilité, mais avec ma casquette GLC je dis que je souhaiterai que ce soit fait coté IdP.

#9 - 22 novembre 2019 12:44 - Frédéric Péters

- Sujet changé de permettre d'utiliser un gabarit django pour la correspondance attribut oidc / attribut authentic à permettre d'utiliser un gabarit django pour la correspondance attribut oidc / attribut authentic (côté réception)

Je précise donc dans le titre de celui-ci que je parlais côté SP, et j'ai créé un autre pour le côté IdP (#37884).

#10 - 22 novembre 2019 12:47 - Paul Marillonnet

- Lié à Development #37884: permettre d'utiliser un gabarit django pour la correspondance attribut oidc / attribut authentic (côté IdP) ajouté

#11 - 22 novembre 2019 12:59 - Benjamin Dauvergne

Frédéric Péters a écrit :

> Je précise donc dans le titre de celui-ci que je parlais côté SP, et j'ai créé un autre pour le côté IdP (#37884).

Ok donc je demande à ce qu'on traite le coté IdP d'abord.

#12 - 25 février 2020 12:57 - Paul Marillonnet

- Statut changé de Nouveau à En cours

#13 - 31 mars 2020 14:00 - Paul Marillonnet

- Fichier 0005-auth_oidc-render-templated-claim-values-during-authn.patch ajouté

- Fichier 0004-auth_oidc-add-id-token-as_dict-method-37871.patch ajouté

- Fichier 0003-auth_oidc-select-from-existing-attributes-in-admin-p.patch ajouté

- Fichier 0002-auth_oidc-use-custom-widget-in-claim-mapping-admin-f.patch ajouté

- Fichier 0001-auth_oidc-extend-mapping-claim-max-length-37871.patch ajouté

- Statut changé de En cours à Solution proposée

- Patch proposed changé de Non à Oui

Allez, une première proposition. Qui ne tient pas compte de la réquisition (OIDCClaimMapping.required) et de la vérification (OIDCClaimMapping.verified) lorsque la revendication est exprimée en langage de gabarit, parce que ça me paraît arbitraire (l'attribut devient vérifié si toutes les revendications dans l'expression de gabarit le sont ? ou une seule ? même question pour l'état de réquisition).
Si le cas d'usage à l'origine du ticket impose le support de ces deux concepts, alors je reverrai ma copie (sans doute par ajout de code d'inspection de la chaîne de gabarit, d'extraction des variables et de comparaison avec le contenu du jeton ID et du payload UserInfo) une fois qu'on se sera mis d'accord sur le côté arbitraire dont je parle ici.

#14 - 31 mars 2020 14:05 - Paul Marillonnet

(Typo à la toute dernière ligne du patch 0005 — dans les tests — corrigé à l'instant dans la branche)

#15 - 31 mars 2020 15:27 - Benjamin Dauvergne

Le patch 4 si provider=None, parsed n'est pas défini; provider n'est pas optionnel je pense.

#16 - 31 mars 2020 15:53 - Paul Marillonnet

- Fichier 0004-auth_oidc-add-id-token-as_dict-method-37871.patch ajouté

Benjamin Dauvergne a écrit :

> Le patch 4 si provider=None, parsed n'est pas défini; provider n'est pas optionnel je pense.

Ouch oui en effet. Corrigé dans la branche et ici.

#17 - 31 mars 2020 16:28 - Benjamin Dauvergne

Paul Marillonnet a écrit :

> Benjamin Dauvergne a écrit :
>
> > Le patch 4 si provider=None, parsed n'est pas défini; provider n'est pas optionnel je pense.
>
> Ouch oui en effet. Corrigé dans la branche et ici.

C'est pareil, pourquoi if provider ?

#18 - 31 mars 2020 18:04 - Paul Marillonnet

- Fichier 0004-auth_oidc-add-id-token-as_dict-method-37871.patch ajouté

Benjamin Dauvergne a écrit :

> C'est pareil

C'est rouge mais cette fois-ci j'y suis pour rien :)
On dirait un souci de dépendance sur jenkins, je reproduis pas en local cette erreur de syntaxe (code python3 dans un venv python2) :

```
Traceback (most recent call last):
    […]
    app_config = AppConfig.create(entry)
  File "/tmp/jenkins-authentic-wip-wip%2F37871-auth-oidc-template-mapping-19/tox-jenkins/authentic/py27-covera
ge-dj111-authentic-pg-oldldap/lib/python2.7/site-packages/django/apps/config.py", line 94, in create
    module = import_module(entry)
  File "/usr/lib/python2.7/importlib/__init__.py", line 37, in import_module
    __import__(name)
  File "/var/lib/jenkins/workspace/37871-auth-oidc-template-mapping/src/authentic2/manager/__init__.py", line
18, in <module>
    import django_select2.conf  # noqa: F401
  File "/tmp/jenkins-authentic-wip-wip%2F37871-auth-oidc-template-mapping-19/tox-jenkins/authentic/py27-covera
ge-dj111-authentic-pg-oldldap/lib/python2.7/site-packages/django_select2/conf.py", line 5, in <module>
    from appconf import AppConf
  File "/tmp/jenkins-authentic-wip-wip%2F37871-auth-oidc-template-mapping-19/tox-jenkins/authentic/py27-covera
ge-dj111-authentic-pg-oldldap/lib/python2.7/site-packages/appconf/__init__.py", line 1, in <module>
    from .base import AppConf  # noqa
  File "/tmp/jenkins-authentic-wip-wip%2F37871-auth-oidc-template-mapping-19/tox-jenkins/authentic/py27-covera
ge-dj111-authentic-pg-oldldap/lib/python2.7/site-packages/appconf/base.py", line 107
    class AppConf(metaclass=AppConfMetaClass):
                          ^
SyntaxError: invalid syntax
```

> pourquoi if provider ?

Parce que je ne me relis pas assez :/
Corrigé ici.

#19 - 31 mars 2020 18:08 - Paul Marillonnet

- Statut changé de Solution proposée à En cours

Paul Marillonnet a écrit :

> Parce que je ne me relis pas assez :/
> Corrigé ici.

Ah oui, si quand même, c'est parce que dans parse_id_token si provider n'a pas d'attribut jwkset ça va lever une AttributeError. Mais c'est de sa faute :)
Je vais corriger.

#20 - 31 mars 2020 18:29 - Paul Marillonnet

- Fichier 0004-auth_oidc-add-id-token-as_dict-method-37871.patch ajouté

- Statut changé de En cours à Solution proposée

Quelque chose comme ça me paraît plus logique, et notamment plus raccord avec la docstring de parse_id_token.

#21 - 02 avril 2020 15:15 - Benjamin Dauvergne

- Statut changé de Solution proposée à Solution validée

Ça ne sert à rien de lever des exceptions pour des trucs qui ne doivent jamais arriver.

#22 - 02 avril 2020 16:37 - Paul Marillonnet

- Statut changé de Solution validée à Résolu (à déployer)

Ok j'ai viré ce bout de code.

```
commit 2b46edf4c578c124a29a18cc95ec9ca7a6f1b869
Author: Paul Marillonnet <pmarillonnet@entrouvert.com>
Date:   Fri Mar 13 16:21:03 2020 +0100

    auth_oidc: render templated claim values during authn (#37871)

commit 556f3e169e28da0f30b890c056593f60a7f713a8
Author: Paul Marillonnet <pmarillonnet@entrouvert.com>
Date:   Mon Mar 30 11:46:14 2020 +0200

    auth_oidc: add id token 'as_dict' method (#37871)

commit 7991c4869ed74a5754bd321f58c54d85875af906
Author: Paul Marillonnet <pmarillonnet@entrouvert.com>
Date:   Fri Mar 27 17:07:24 2020 +0100

    auth_oidc: select from existing attributes in admin provider page (#37871)

commit c4636a41cef5987204d5c2f9e7ef765537ad12aa
Author: Paul Marillonnet <pmarillonnet@entrouvert.com>
Date:   Fri Mar 13 15:01:12 2020 +0100

    auth_oidc: use custom widget in claim mapping admin form (#37871)

commit dccf9a571f1fdfe519ed7e7bb08bcd57151852c3
Author: Paul Marillonnet <pmarillonnet@entrouvert.com>
Date:   Tue Mar 17 17:32:46 2020 +0100

    auth_oidc: extend mapping claim max length (#37871)
```

#23 - 07 avril 2020 13:12 - Paul Marillonnet

- Statut changé de Résolu (à déployer) à Solution déployée

Fichiers

0004-auth_oidc-add-id-token-as_dict-method-37871.patch	1,1 ko	31 mars 2020	Paul Marillonnet
0005-auth_oidc-render-templated-claim-values-during-authn.patch	7,98 ko	31 mars 2020	Paul Marillonnet
0003-auth_oidc-select-from-existing-attributes-in-admin-p.patch	1,43 ko	31 mars 2020	Paul Marillonnet
0002-auth_oidc-use-custom-widget-in-claim-mapping-admin-f.patch	2,65 ko	31 mars 2020	Paul Marillonnet
0001-auth_oidc-extend-mapping-claim-max-length-37871.patch	2,09 ko	31 mars 2020	Paul Marillonnet
0004-auth_oidc-add-id-token-as_dict-method-37871.patch	1,1 ko	31 mars 2020	Paul Marillonnet

0004-auth_oidc-add-id-token-as_dict-method-37871.patch	1,05 ko	31 mars 2020	Paul Marillonnet
0004-auth_oidc-add-id-token-as_dict-method-37871.patch	1,37 ko	31 mars 2020	Paul Marillonnet