Authentic 2 - Development #37884

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

22 novembre 2019 12:43 - 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

Comme #37871 mais dans les attributs fournis par Authentic quand il est IdP OIDC.

Demandes liées:

Lié à Authentic 2 - Development #37871: permettre d'utiliser un gabarit djang...	**Fermé**	**22 novembre 2019**

Révisions associées

Révision 307987ce - 17 mars 2020 16:23 - Paul Marillonnet

forms: add a datalist textinput widget (#37884)

Révision 139e9f0e - 17 mars 2020 16:23 - Paul Marillonnet

idp_oidc: use custom widget in client admin page (#37884)

Révision 6803bcd2 - 17 mars 2020 16:23 - Paul Marillonnet

add a template module, copycatting wcs.qommon.template (#37884)

Révision afa8e45a - 17 mars 2020 16:23 - Paul Marillonnet

idp_oidc: render templated claims in user-info-creation utilities (#37884)

Historique

#1 - 22 novembre 2019 12:47 - Paul Marillonnet

- Lié à Development #37871: permettre d'utiliser un gabarit django pour la correspondance attribut oidc / attribut authentic (côté réception) ajouté

#2 - 22 novembre 2019 12:48 - Paul Marillonnet

- Assigné à mis à Paul Marillonnet

#4 - 17 février 2020 17:45 - Paul Marillonnet

Une vague idée de plan :

- avoir une classe de champ ou de formulaire permettant, à partir d'un contexte donné, de rendre du langage de gabarit django inline, par exemple en s'inspirant de ce qui a déjà été fait dans w.c.s.
- une courte modification du modèle de revendications OIDC pour gérer les cas où les claims sont en langage de gabarit, par exemple comme ça.
- ajouter, quand nécessaire, quelques lignes d'appel au code de rendu de langage de gabarit au moment de l'évaluation des revendications dans authentic_idp_oidc.utils.create_user_info

#5 - 20 février 2020 13:46 - Paul Marillonnet

Après réflexion, je me demande côté UI d'administration s'il ne serait pas préférable que les revendications dont la valeur est exprimée en langage de gabarit à partir des attributs du profil bénéficient de leur propre écran de configuration (/admin/authentic2_idp_oidc/oidcclaim/). Ça me paraît difficilement réalisable que de se baser sur le bout d'écran déjà existant en fin de page /admin/authentic2_idp_oidc/oidcclient/{id}/change/.

#6 - 20 février 2020 16:53 - Paul Marillonnet

- Fichier 0001-idp_oidc-add-is_templated-option-to-claim-model-3788.patch ajouté

- Fichier 0002-idp_oidc-templated-claim-model-migration-file-37884.patch ajouté

- Fichier 0003-template-initial-class-derived-from-wcs.qommon.templ.patch ajouté

- Fichier 0004-idp_oidc-render-templated-claims-in-user-info-creati.patch ajouté

- Fichier 0005-idp_oidc-add-claim-admin-page-37884.patch ajouté

- Fichier 0006-idp_oidc-test-templated-claims-37884.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Voilà ce à quoi ça pourrait ressembler. Je veux bien qu'on me confirme que c'est une direction souhaitable, pendant que j'écris les tests du module nouvellement introduit authentic2.utils.template.

#7 - 20 février 2020 17:19 - Emmanuel Cazenave

Je dirai que si le rendering du template foire il faudrait fallback sur le comportement sans template (je n'ai pas l'impression que ce soit le cas).

L'objet Template me parait un peu lourd, le code appelant ne se sert pas des variations de comportement proposées par raises et autoescape (et donc genre une simple fonction me semble faire l'affaire).

#8 - 20 février 2020 17:58 - Paul Marillonnet

Emmanuel Cazenave a écrit :

> Je dirai que si le rendering du template foire il faudrait fallback sur le comportement sans template (je n'ai pas l'impression que ce soit le cas).

Si oui c'est bien le cas. C'est du code qui est très largement inspiré (RIP Larry Tesler) de ce qui est déjà dans w.c.s.

> L'objet Template me parait un peu lourd, le code appelant ne se sert pas des variations de comportement proposées par raises et autoescape (et donc genre une simple fonction me semble faire l'affaire).

J'ai volontairement laissé la généricité comme dans w.c.s. J'aime bien l'idée d'un objet qu'on initialise avec un bout de gabarit inline, pour ensuite n'effectuer le rendu que dans un second temps, avec un dictionnaire de contexte particulier. Mais si ça choque je simplifie et j'en fais un fonction, pas de souci.

#9 - 20 février 2020 18:27 - Emmanuel Cazenave

Paul Marillonnet a écrit :

> Emmanuel Cazenave a écrit :
>
> > Je dirai que si le rendering du template foire il faudrait fallback sur le comportement sans template (je n'ai pas l'impression que ce soit le cas).
>
> Si oui c'est bien le cas.

Je n'arrive toujours pas à le voir. Soit tu te trompes, soit je ne le le vois pas parce que ce Template est vraiment trop tordu, soit je n'ai pas les yeux en face des trous. Je passe, on verra ce que dira le prochain.

#10 - 21 février 2020 09:18 - Frédéric Péters

> Après réflexion, je me demande côté UI d'administration s'il ne serait pas préférable que les revendications dont la valeur est exprimée en langage de gabarit à partir des attributs du profil bénéficient de leur propre écran de configuration (/admin/authentic2_idp_oidc/oidcclaim/). Ça me paraît difficilement réalisable que de se baser sur le bout d'écran déjà existant en fin de page /admin/authentic2_idp_oidc/oidcclient/{id}/change/.

Perso je ne capte pas le problème qu'il y a avec le bas de page actuel; pour reprendre le cas de Serghei, on écrirait le petit bout gender : {% if title "Monsieur" }male{ elif title "Madame %}female{ endif %}; alors oui ça déborde peut-être un petit peu mais la présentation pourra être améliorée quand ces pages passeront dans /manage/

#11 - 21 février 2020 14:12 - Paul Marillonnet

- Statut changé de Solution proposée à En cours

Emmanuel Cazenave a écrit :

> Soit tu te trompes, soit je ne le le vois pas parce que ce Template est vraiment trop tordu[…].

Motif suffisant pour que ça mérite une modification. Je corrige ça.

#12 - 21 février 2020 15:03 - Paul Marillonnet

Frédéric Péters a écrit :

> Perso je ne capte pas le problème qu'il y a avec le bas de page actuel; pour reprendre le cas de Serghei, on écrirait le petit bout gender : {% if title "Monsieur" }male{ elif title "Madame %}female{ endif %}; alors oui ça déborde peut-être un petit peu mais la présentation pourra être améliorée quand ces pages passeront dans /manage/

En fait, je ne voulais pas modifier le comportement du formulaire inline dans le bas de page, qui est de présenter une liste déroulante pour sélectionner un attribut. Je ne sais pas trop comment on peut avoir un comportement double, qui conserverait cette liste mais qui offrirait aussi la possibilité de saisir un bout de langage de gabarit inline. Peut-être avec un bout de JS et un peu de django-floppyforms ?

Je pense que ça rendrait possible quelque chose comme de conserver la présentation de liste déroulante de sélection de l'attribut à associer à la revendication, mais d'y ajouter une entrée "Expression de gabarit" dont la sélection par l'admin afficherait un champ texte supplémentaire.

#13 - 21 février 2020 15:14 - Paul Marillonnet

Paul Marillonnet a écrit :

> En fait, je ne voulais pas modifier le comportement du formulaire inline dans le bas de page,

Pour être plus clair, ce que je veux éviter est que pour prendre en charge le cas rare où saisir du langage de gabarit est nécessaire, on perde l'affichage de la liste déroulante qui elle est d'une grande utilité dans tous les autres cas.

#14 - 21 février 2020 15:18 - Frédéric Péters

Je propose de ne rien modifier, simplement que là où il y avait aujourd'hui du texte, si jamais un gabarit est présent à la place, il est interprété.

#15 - 21 février 2020 15:25 - Paul Marillonnet

- Fichier configuration_apparairage.png ajouté

Frédéric Péters a écrit :

> Je propose de ne rien modifier, simplement que là où il y avait aujourd'hui du texte, si jamais un gabarit est présent à la place, il est interprété.

Il n'y pas pas de texte aujourd'hui mais bien une liste déroulante (cf capture).

#16 - 21 février 2020 15:41 - Frédéric Péters

(my bad, j'étais encore sur l'écran côté SP...)

#17 - 24 février 2020 15:44 - Paul Marillonnet

Bon, ce qui me paraît encore le plus simple est d'écrire un widget custom ayant recours à la balise HTML5 datalist pour le rendu d'un champ en saisie libre. Sauf avis contraire je vais partir là-dessus.

#18 - 25 février 2020 12:40 - Paul Marillonnet

- Fichier 0001-forms-add-a-datalist-textinput-widget-37884.patch ajouté

- Fichier 0002-idp_oidc-use-custom-widget-in-client-admin-page-3788.patch ajouté

- Fichier 0003-add-a-template-module-copycatting-wcs.qommon.templat.patch ajouté

- Fichier 0004-idp_oidc-render-templated-claims-in-user-info-creati.patch ajouté

- Statut changé de En cours à Solution proposée

Et ça donne quelque chose comme ça. Rien de compliqué.

#19 - 25 février 2020 14:04 - Frédéric Péters

Perso 0003 n'est pas un bout que j'apprécie, on se trouve à chercher 'def render(' et ce bout de code n'apparait pas. Dans le même, je ne pense pas qu'il soit utile de renommer les exceptions.

#20 - 27 février 2020 14:51 - Paul Marillonnet

- Fichier 0003-add-a-template-module-copycatting-wcs.qommon.templat.patch ajouté

Frédéric Péters a écrit :

> Perso 0003 n'est pas un bout que j'apprécie, on se trouve à chercher 'def render(' et ce bout de code n'apparait pas. Dans le même, je ne pense

pas qu'il soit utile de renommer les exceptions.

Ok, le patch 0003 modifié en conséquence, les autres restent inchangés.

#21 - 05 mars 2020 12:19 - Benjamin Dauvergne

0002: il y a un bug bien caché (qui était déjà dans le commit initial de Josué) :

```
        data = dict(get_service_attributes(self.instance.client_id)).keys()
```

c'est self.instance.client qu'il faut ; je verrai bien une refonte de ce formulaire de toute façon. Il y a une partie Meta à ModelForm qui permet d'indiquer le widget souhaité pour un champ : https://docs.djangoproject.com/fr/1.11/topics/forms/modelforms/#overriding-the-default-fields, ça nécessitera quand même de manipuler data dans *init* mais c'est plus Django-esque.

#22 - 09 mars 2020 11:41 - Paul Marillonnet

- Fichier 0004-idp_oidc-render-templated-claims-in-user-info-creati.patch ajouté

- Fichier 0003-add-a-template-module-copycatting-wcs.qommon.templat.patch ajouté

- Fichier 0002-idp_oidc-use-custom-widget-in-client-admin-page-3788.patch ajouté

- Fichier 0001-forms-add-a-datalist-textinput-widget-37884.patch ajouté

Avec un getattr pour tenter d'obtenir le client lié à la revendication, car le formulaire inline est prérempli avec des revendications associées à aucun client.

#23 - 17 mars 2020 16:21 - Benjamin Dauvergne

- Statut changé de Solution proposée à Solution validée

C'est moche comment marchent les widgets (mais bon c'est Django). Devoir passer explicitement l'id pour la liste n'est pas terrible. Ok quand même.

#24 - 17 mars 2020 17:00 - Paul Marillonnet

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

```
commit afa8e45afd26ee8403bfdd604f02fb7118afe514
Author: Paul Marillonnet <pmarillonnet@entrouvert.com>
Date:   Mon Feb 17 17:14:34 2020 +0100

    idp_oidc: render templated claims in user-info-creation utilities (#37884)

commit 6803bcd2f01d872b12f174793153750dd4f56a78
Author: Paul Marillonnet <pmarillonnet@entrouvert.com>
Date:   Thu Feb 20 11:15:03 2020 +0100

    add a template module, copycatting wcs.qommon.template (#37884)

commit 139e9f0e3b29dadb307446f404ac712883bed5cd
Author: Paul Marillonnet <pmarillonnet@entrouvert.com>
Date:   Thu Feb 20 15:29:10 2020 +0100

    idp_oidc: use custom widget in client admin page (#37884)

commit 307987ce5506b4e168a87b4fff42d852c9169546
Author: Paul Marillonnet <pmarillonnet@entrouvert.com>
Date:   Mon Feb 24 16:12:37 2020 +0100

    forms: add a datalist textinput widget (#37884)
```

#25 - 19 mars 2020 22:14 - Frédéric Péters

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

Fichiers

0001-idp_oidc-add-is_templated-option-to-claim-model-3788.patch	1,04 ko	20 février 2020	Paul Marillonnet
0002-idp_oidc-templated-claim-model-migration-file-37884.patch	1,73 ko	20 février 2020	Paul Marillonnet
0003-template-initial-class-derived-from-wcs.qommon.templ.patch	2,84 ko	20 février 2020	Paul Marillonnet
0004-idp_oidc-render-templated-claims-in-user-info-creati.patch	1,54 ko	20 février 2020	Paul Marillonnet
0005-idp_oidc-add-claim-admin-page-37884.patch	1,24 ko	20 février 2020	Paul Marillonnet

0006-idp_oidc-test-templated-claims-37884.patch	3,67 ko	20 février 2020	Paul Marillonnet
configuration_apparairage.png	42,1 ko	21 février 2020	Paul Marillonnet
0001-forms-add-a-datalist-textinput-widget-37884.patch	3,96 ko	25 février 2020	Paul Marillonnet
0002-idp_oidc-use-custom-widget-in-client-admin-page-3788.patch	1,36 ko	25 février 2020	Paul Marillonnet
0003-add-a-template-module-copycatting-wcs.qommon.templat.patch	6,78 ko	25 février 2020	Paul Marillonnet
0004-idp_oidc-render-templated-claims-in-user-info-creati.patch	4,92 ko	25 février 2020	Paul Marillonnet
0003-add-a-template-module-copycatting-wcs.qommon.templat.patch	6,76 ko	27 février 2020	Paul Marillonnet
0003-add-a-template-module-copycatting-wcs.qommon.templat.patch	6,76 ko	09 mars 2020	Paul Marillonnet
0004-idp_oidc-render-templated-claims-in-user-info-creati.patch	4,94 ko	09 mars 2020	Paul Marillonnet
0002-idp_oidc-use-custom-widget-in-client-admin-page-3788.patch	1,64 ko	09 mars 2020	Paul Marillonnet
0001-forms-add-a-datalist-textinput-widget-37884.patch	3,96 ko	09 mars 2020	Paul Marillonnet