Authentic 2 - Bug #31259

présenter les IDP OpenidConnect dans des blocks à part sur la page de login

11 mars 2019 10:16 - Serghei Mihai

Statut:	Fermé	**Début:**	11 mars 2019
Priorité:	Normal	**Echéance:**	
Assigné à:	Serghei Mihai	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

Afin de pouvoir les filtrer en fonction de la provenance de l'usager.

Demandes liées:

Lié à Authentic 2 - Development #30960: Permetttre de faire d'authentic un Id...	**Fermé**	**27 février 2019 31 mai 2019**
Lié à Authentic 2 - Development #28215: permettre le filtrage des frontends d...	**Fermé**	**21 novembre 2018**
Lié à Intégrations graphiques Publik - Development #38063: ne plus utiliser l...	**Fermé**	**28 novembre 2019**
Lié à Authentic 2 - Development #38125: oidc: simplifier la construction des ...	**Fermé**	**02 décembre 2019**
Lié à Intégrations graphiques Publik - Bug #38291: bloc de connexion ozwillo	**Fermé**	**09 décembre 2019**

Révisions associées

Révision 216323c7 - 29 novembre 2019 10:06 - Serghei Mihai

auth: separate OIDC providers in blocks on login page (#31259)

Historique

#1 - 11 mars 2019 10:20 - Serghei Mihai

- Lié à Development #30960: Permetttre de faire d'authentic un IdP proxy transparent ajouté

#2 - 12 mars 2019 10:40 - Serghei Mihai

- Fichier 0001-auth-separate-OIDC-providers-in-blocks-on-login-page.patch ajouté

- Tracker changé de Support à Bug

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Tests en cours d'ajout.
Dépend du renommage des backends dans #14475.

#3 - 12 mars 2019 21:03 - Serghei Mihai

- Fichier 0001-auth-separate-OIDC-providers-in-blocks-on-login-page.patch ajouté

Et avec un simple test.

#4 - 01 avril 2019 17:47 - Paul Marillonnet

- Le tests ne vérifie explicitement pas la présence des deux blocs en même temps.
  J'apprécierais bien un truc un peu plus costaud et plus évocateur, genre un peu de pyquery, au lieu de assert 'machin' in response.
- Un saut à la ligne en trop en à la fin de src/authentic2_auth_oidc/templates/authentic2_auth_oidc/login.html
- Un autre en trop à la fin de la définition de OIDCAuthenticator.instances.
- À la lecture du code, on s'attendrait à ce que cette fonction OIDCAuthenticator.instances soit une propriété de classe, avec un générateur, ce qui n'est pas le cas.
  Je pense que tu peux t'en passer complètement, ça ne changerait pas grand chose de reconstruire le tuple (instance_id, instance) directement dans la vue.
  Ça t'éviterait d'avoir un dictionnaire qui donne l'impression d'être modifié en cours d'itération pendant que chaque itération dépend de son contenu (même si dans les faits ce n'est pas le cas, ce n'est que l'impression que le code donne).

#5 - 01 avril 2019 19:08 - Serghei Mihai

- Fichier 0001-auth-separate-OIDC-providers-in-blocks-on-login-page.patch ajouté

Paul Marillonnet a écrit :

- Le tests ne vérifie explicitement pas la présence des deux blocs en même temps.

test_providers_on_login_page vérifie d'abord qu'il n'y qu'un seul bloc OIDC, puis crée un deuxième IDP OIDC et vérifie que l'onglet associé est créé. Fait avec pyquery.

- Un saut à la ligne en trop en à la fin de src/authentic2_auth_oidc/templates/authentic2_auth_oidc/login.html

Corrigé.

- À la lecture du code, on s'attendrait à ce que cette fonction OIDCAuthenticator.instances soit une propriété de classe, avec un générateur, ce qui n'est pas le cas.

Tant que c'est un itérable: ok pour le générateur.

> Je pense que tu peux t'en passer complètement, ça ne changerait pas grand chose de reconstruire le tuple (instance_id, instance) directement dans la vue.

Tu parles de

```
block['id'] = instance_id
```

?

Je suis bien obligé car sinon le code de la vue ne fait pas de distinction entre les blocs "oidc". Branche à jour.

#6 - 26 août 2019 16:58 - Paul Marillonnet

- Statut changé de Solution proposée à Information nécessaire

Désolé, j'ai laissé passer ça. Cette partie du code ayant un peu bougé entre temps, peux-tu proposer un patch rebasé sur master stp ?

#7 - 02 octobre 2019 12:27 - Serghei Mihai

Branche à jour par rapport au master.

#8 - 02 octobre 2019 18:33 - Paul Marillonnet

Quelques modifs qui me semblent pertinentes (dont un bogue déjà présent sur l'argument de querystring next dans le gabarit du bloc d'authentification) :

```
diff --git a/src/authentic2/views.py b/src/authentic2/views.py
index eb1fb8a4f..063a42c7b 100644
--- a/src/authentic2/views.py
+++ b/src/authentic2/views.py
@@ -325,7 +325,7 @@ def login(request, template_name='authentic2/login.html',
                           'context': context}
             # check if the authenticator has multiple instances
             if hasattr(authenticator, 'instances'):
-                for instance_id, instance in authenticator.instances(**parameters):
+                for instance_id, instance in authenticator.instances:
                     parameters['instance'] = instance
                     block = utils.get_authenticator_method(authenticator, 'login', parameters)
                     # update block id in order to separate instances
diff --git a/src/authentic2_auth_oidc/authenticators.py b/src/authentic2_auth_oidc/authenticators.py
index 84ee27a87..47b106336 100644
--- a/src/authentic2_auth_oidc/authenticators.py
+++ b/src/authentic2_auth_oidc/authenticators.py
@@ -30,7 +30,8 @@ class OIDCAuthenticator(object):
     def id(self):
         return 'oidc'

-    def instances(self, request, *args, **kwargs):
```

```
+    @property
+    def instances(self):
         for p in utils.get_providers(shown=True):
             yield (p.slug, p)
```

```
diff --git a/src/authentic2_auth_oidc/templates/authentic2_auth_oidc/login.html b/src/authentic2_auth_oidc/tem
plates/authentic2_auth_oidc/login.html
index bd3056883..c2aeab906 100644
--- a/src/authentic2_auth_oidc/templates/authentic2_auth_oidc/login.html
+++ b/src/authentic2_auth_oidc/templates/authentic2_auth_oidc/login.html
@@ -1,4 +1,4 @@
 <p id="oidc-p-{% firstof provider.slug provider.name|slugify %}">
   <a id="oidc-a-{% firstof provider.slug  provider.name|slugify %}"
-     href="{% url "oidc-login" pk=provider.pk %}?{{ request.GET.urlencode }}">{{ provider.name }}</a>
+     href="{% url "oidc-login" pk=provider.pk %}?{{ request.GET.next|urlencode }}">{{ provider.name }}</a>
 </p>
diff --git a/tests/test_auth_oidc.py b/tests/test_auth_oidc.py
index 5910cf275..00a92204d 100644
--- a/tests/test_auth_oidc.py
+++ b/tests/test_auth_oidc.py
@@ -309,8 +309,6 @@ def test_providers_on_login_page(oidc_provider, app):
     assert response.pyquery('p#oidc-p-oidcidp-2')

-
-
 def test_sso(app, caplog, code, oidc_provider, oidc_provider_jwkset, login_url, login_callback_url, hooks):
     OU = get_ou_model()
     cassis = OU.objects.create(name='Cassis', slug='cassis')
```

Une dernière chose encore, qui sera peut-être l'objet d'un autre ticket, et qui est le comportement de ce bout de code dans authentic2.views.login :

```
            # If a login frontend method returns an HttpResponse with a status code != 200
            # this response is returned.
            for block in auth_blocks:
                if block:
                    if block['status_code'] != 200:
                        return block['response']
```

-> Ce n'est peut-être plus ce que l'on veut, maintenant qu'on a potentiellement des blocs issus d'instances multiples d'un même authentificateur (cas dans lequel on pourrait peut-être, au lieu de remonter l'erreur, détecter si au moins un bloc pour une des instances d'un même authentificateur renvoie bien un 200, auquel cas on ne retiendrait pas les autres en erreur).

#9 - 03 octobre 2019 17:59 - Serghei Mihai

Tes remarques prises en compte.

Pour le status code, comme c'est général pour tous les authentificateurs on devrait gérer cela séparément.

#10 - 03 octobre 2019 17:59 - Serghei Mihai

- Fichier 0002-auth-separate-OIDC-providers-in-blocks-on-login-page.patch ajouté

- Fichier 0001-oidc-fix-next-url-encoding-on-login-page.patch ajouté

- Statut changé de Information nécessaire à Solution proposée

#11 - 03 octobre 2019 18:14 - Paul Marillonnet

- Statut changé de Solution proposée à Solution validée

(Il reste encore un quadruple saut de ligne à la fin du test, à corriger) sinon c'est bon pour moi. Ack.

#12 - 03 octobre 2019 19:01 - Benjamin Dauvergne

- Statut changé de Solution validée à En cours

Les tests ne passent pas.

#13 - 03 octobre 2019 19:25 - Paul Marillonnet

(Trop fatigué pour comparer avec le build précédent et voir pourquoi ça ne passe pas maintenant. Je regarde demain.)

#14 - 03 octobre 2019 20:02 - Frédéric Péters

> (Trop fatigué pour comparer avec le build précédent et voir pourquoi ça ne passe pas maintenant. Je regarde demain.)

Ce ne serait pas la suggestion de faire :

```
-     href="{% url "oidc-login" pk=provider.pk %}?{{ request.GET.urlencode }}">{{ provider.name }}</a>
+     href="{% url "oidc-login" pk=provider.pk %}?{{ request.GET.next|urlencode }}">{{ provider.name }}</a>
```

qui va faire ?whatever plutôt que ?next=whatever ?

(je ne comprends pas de quel bug tu pouvais parler dans le commentaire précédent)

#15 - 08 octobre 2019 11:56 - Serghei Mihai

- Fichier 0002-auth-separate-OIDC-providers-in-blocks-on-login-page.patch ajouté

- Fichier 0001-oidc-fix-next-url-encoding-on-login-page.patch ajouté

- Statut changé de En cours à Solution proposée

Bien vu: next rajouté.

#16 - 08 octobre 2019 12:15 - Benjamin Dauvergne

Tant qu'à bosser sur une fonctionnalité pas super passionnante j'aimerai beaucoup qu'on en profite pour ne plus construire d'URL dans un template (je plaide coupable); tout ticket doit être l'occasion d'améliorer un peu le code quand on passe dessus (ici utiliser authentic2.utils.make_url qui est fait pour gérer toutes les manipulations d'URL).

#17 - 17 octobre 2019 18:09 - Serghei Mihai

- Fichier 0001-auth-separate-OIDC-providers-in-blocks-on-login-page.patch ajouté

Oké. En un seul patch.

#18 - 17 octobre 2019 18:13 - Benjamin Dauvergne

Merci, je relis.

#19 - 26 novembre 2019 15:41 - Serghei Mihai

- Lié à Development #28215: permettre le filtrage des frontends d'authentification à exposer à l'usager ajouté

#20 - 28 novembre 2019 12:42 - Benjamin Dauvergne

- Statut changé de Solution proposée à Solution validée

Benjamin Dauvergne a écrit :

> Merci, je relis.

Garder l'id initial dans les blocks, genre block['id'] = block['id'] + '_' + instance_id sinon on a rien qui permette d'éviter des collisions.

J'ai cherché il n'y a rien qui utilise is_hidden tu peux virer ce code, i.e. mettre is_hidden à False tout le temps et commentaire FIXME pour le virer des templates plus tard.

#21 - 28 novembre 2019 15:52 - Serghei Mihai

- Fichier 0001-auth-separate-OIDC-providers-in-blocks-on-login-page.patch ajouté

- Statut changé de Solution validée à Solution proposée

Yes, le nettoyage de ce is_hidden et visible_blocks_count à coordonner avec le nettoyage dans publik-base-theme.

#22 - 28 novembre 2019 16:33 - Thomas Noël

Moi je suis surpris par l'identation qui change pour ce qui concerne l'ancien code

```
        if hasattr(authenticator, 'is_hidden'):
            blocks[-1]['is_hidden'] = authenticator.is_hidden(request)
        else:
```

```
            blocks[-1]['is_hidden'] = False
```

qui n'était pas compris dans le "else: # New frontends API" comme il semble l'être dans ce patch.

#23 - 28 novembre 2019 17:20 - Benjamin Dauvergne

Effectivement le is_hidden va manquer aux blocks legacy.

#24 - 28 novembre 2019 17:43 - Serghei Mihai

- Fichier 0001-auth-separate-OIDC-providers-in-blocks-on-login-page.patch ajouté

Prochaine étape est de se débarasser de ce "is_hidden"

#25 - 28 novembre 2019 17:47 - Serghei Mihai

- Lié à Development #38063: ne plus utiliser la variable "visible_blocks_count" pour le nombre de blocks sur la page d'authentification ajouté

#26 - 28 novembre 2019 18:01 - Benjamin Dauvergne

- Statut changé de Solution proposée à Solution validée

#27 - 29 novembre 2019 10:23 - Serghei Mihai

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

```
commit 216323c7ad560b8417535dcc5fdca694b3713d7e (origin/master, origin/HEAD)
Author: Serghei Mihai <smihai@entrouvert.com>
Date:   Fri Mar 8 14:57:45 2019 +0100

    auth: separate OIDC providers in blocks on login page (#31259)
```

#28 - 02 décembre 2019 13:32 - Serghei Mihai

- Lié à Development #38125: oidc: simplifier la construction des urls de connexion ajouté

#29 - 04 décembre 2019 18:15 - Frédéric Péters

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

#31 - 09 décembre 2019 09:48 - Frédéric Péters

(pour info ça a cassé toodego) (je m'en occupe, pour continuer à contourner)

#32 - 09 décembre 2019 10:05 - Frédéric Péters

- Lié à Bug #38291: bloc de connexion ozwillo ajouté

Fichiers

0001-auth-separate-OIDC-providers-in-blocks-on-login-page.patch	4,24 ko	12 mars 2019	Serghei Mihai
0001-auth-separate-OIDC-providers-in-blocks-on-login-page.patch	5,76 ko	12 mars 2019	Serghei Mihai
0001-auth-separate-OIDC-providers-in-blocks-on-login-page.patch	5,76 ko	01 avril 2019	Serghei Mihai
0002-auth-separate-OIDC-providers-in-blocks-on-login-page.patch	5,76 ko	03 octobre 2019	Serghei Mihai
0001-oidc-fix-next-url-encoding-on-login-page.patch	1,08 ko	03 octobre 2019	Serghei Mihai
0002-auth-separate-OIDC-providers-in-blocks-on-login-page.patch	5,8 ko	08 octobre 2019	Serghei Mihai
0001-oidc-fix-next-url-encoding-on-login-page.patch	1,09 ko	08 octobre 2019	Serghei Mihai
0001-auth-separate-OIDC-providers-in-blocks-on-login-page.patch	6,31 ko	17 octobre 2019	Serghei Mihai
0001-auth-separate-OIDC-providers-in-blocks-on-login-page.patch	5,73 ko	28 novembre 2019	Serghei Mihai
0001-auth-separate-OIDC-providers-in-blocks-on-login-page.patch	6,28 ko	28 novembre 2019	Serghei Mihai