

w.c.s. - Development #22389

accès direct aux attributs d'un utilisateur

08 mars 2018 20:02 - Frédéric Péters

Statut:	Fermé	Début:	08 mars 2018
Priorité:	Normal	Echéance:	
Assigné à:		% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposed:	Oui		
Description			
La classe User expose juste un nom, qui dans une configuration Publik classique est assemblé en mettant "first name" + "last name".			
Pour compatibilité avec des templates django qui pourraient utiliser user.last_name plutôt que user.get_full_name, on devrait chercher aussi dans ces attributs définis via des champs.			

Révisions associées

Révision 653a2b8e - 09 mars 2018 16:10 - Frédéric Péters

users: look for attributes in extra fields (#22389)

Historique

#1 - 08 mars 2018 20:02 - Frédéric Péters

- Fichier 0001-users-look-for-attributes-in-extra-fields-22389.patch ajouté

- Statut changé de Nouveau à En cours

- Patch proposed changé de Non à Oui

#3 - 09 mars 2018 10:52 - Thomas Noël

Je suis un peu réticent à l'idée de les mettre directement dans l'espace de nommage "user", avec le risque qu'un jour quelqu'un nomme mal une variable, genre "get_formdef", et boum, conflit...

Par ailleurs, avoir juste la valeur mais ne pas savoir le label du champ, ça ne simplifie pas l'écriture de templates génériques.

Quid d'un "user.fields" ? Quitte même à ce que chaque fields contienne label et value...

```
user.fields.last_name.label  
user.fields.last_name.value
```

avec la possibilité de faire un "for field in user.fields" pour parcourir les champs dans l'ordre de définition du formdef, ce genre d'outils.

Ça ne va pas dans le sens d'une simplification, mais plutôt d'une rationalisation ; ça serait un principe générique pour les "champs formdef" des objets user, formdata, etc... L'idée serait aussi que "value" serait un type de donnée propre (je pense surtout aux dates qui seraient des datetime).

#4 - 09 mars 2018 10:53 - Frédéric Péters

Je suis un peu réticent à l'idée de les mettre directement dans l'espace de nommage "user", avec le risque qu'un jour quelqu'un nomme mal une variable, genre "get_formdef", et boum, conflit...

L'objectif est d'être compatible avec l'«API» exposée par Django, qui a user.first_name/user.last_name.

#5 - 09 mars 2018 11:06 - Thomas Noël

Frédéric Péters a écrit :

L'objectif est d'être compatible avec l'«API» exposée par Django, qui a user.first_name/user.last_name.

Ok, pigé. Et je viens de faire le test, les attributs existants sont bien préservés, tout va bien. Ack.

Au cas où, j'ai même testé mon inquiétude

```

diff --git a/tests/test_users.py b/tests/test_users.py
index f7c857ea..e2bcb09c 100644
--- a/tests/test_users.py
+++ b/tests/test_users.py
@@ -61,14 +61,19 @@ def test_get_users_with_role():
     def test_user_formdef_getattr():
         from wcs.admin.settings import UserFieldsFormDef
         formdef = UserFieldsFormDef(pub)
-        formdef.fields = [fields.StringField(id='3', label='test', type='string', varname='plop')]
+        formdef.fields = [
+            fields.StringField(id='3', label='test', type='string', varname='plop'),
+            fields.StringField(id='9', label='noop', type='string', varname='get_formdef'),
+        ]
         formdef.store()

         user = pub.user_class()
         assert user.plop is None
+        assert user.get_formdef() # get_formdef is not overridden by varname "get_formdef"

-        user.form_data = {'3': 'Bar'}
+        user.form_data = {'3': 'Bar', '9': 'Foo'}
         assert user.plop == 'Bar'
+        assert user.get_formdef()

         with pytest.raises(AttributeError):
             user.xxx

```

#6 - 09 mars 2018 16:13 - Frédéric Péters

- Statut changé de *En cours* à *Résolu* (à déployer)

J'ai intégré tes tests, en modifiant le dernier assert pour faire `assert user.get_firmdef != 'Foo'`, que je trouvais plus explicite (que planter sur 'Foo' qui n'aurait pas été callable).

```

commit 653a2b8efc6cf1034b8f9245a92f6b87a6ecf7e2
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Thu Mar 8 20:02:15 2018 +0100

```

```

users: look for attributes in extra fields (#22389)

```

#7 - 23 décembre 2018 14:42 - Frédéric Péters

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

Fichiers

0001-users-look-for-attributes-in-extra-fields-22389.patch	2,19 ko	08 mars 2018	Frédéric Péters
--	---------	--------------	-----------------