

## w.c.s. - Development #29342

### Connaitre le rôle de l'utilisateur connecté (filtre has\_role)

25 décembre 2018 18:32 - Laurent Séguin

<b>Statut:</b>	Fermé	<b>Début:</b>	25 décembre 2018
<b>Priorité:</b>	Normal	<b>Echéance:</b>	
<b>Assigné à:</b>	Thomas Noël	<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>	Oui		
<b>Description</b>			
User Story : En tant que créateur d'une démarche, je veux pouvoir afficher un champ de formulaire dédié en fonction de rôle/s de l'utilisateur connecté sans qu'un utilisateur n'ayant pas ce/s rôle/s puisse afficher le champ, afin de pouvoir utiliser un même formulaire qui s'adaptera légèrement selon le/s rôle/s attribué à l'utilisateur connecté.  Il a été évoqué de créer une variable : <code>session_user has_role:"&lt;string&gt;"</code>			

#### Révisions associées

##### Révision e22d2b52 - 30 octobre 2019 14:56 - Thomas Noël

misc: add has\_role filter (#29342)

#### Historique

##### #1 - 26 décembre 2018 00:43 - Thomas Noël

Si on passe par une condition d'affichage du champ, il faut savoir que techniquement le champ peut exister sur la page : il est juste caché. Un simple "view-source" permettrait à un usager de quand même voir des informations auxquelles il n'est pas prévu qu'il ait accès -- c'est j'imagine ce qu'on veut faire ici. Je ne suis donc pas chaud à mettre cette possibilité en avant.

(note à Frédéric, qui objectera que c'est pas vrai si la condition qui n'utilise pas de "form\_var\_truc" : oui, d'accord. Justement j'imagine une condition qui ferait un "form\_var\_truc = "chose" and session\_user|has\_role:"bidule", et boum)

##### #2 - 26 décembre 2018 08:25 - Frédéric Péters

on passe par une condition d'affichage du champ, il faut savoir que techniquement le champ peut exister sur la page : il est juste caché.

Je pense utile de rappeler ça.

un usager de quand même voir des informations auxquelles il n'est pas prévu qu'il ait accès

Mais on parle d'un champ, normalement ça ne dévoile pas d'informations. (et d'accord, oui, de loin, on pourrait imaginer un champ commentaire affiché avec des instructions "confidentielles" pour les agents).

##### #3 - 26 décembre 2018 12:27 - Laurent Séguin

Il est effectivement bon de rappeler qu'il est possible de savoir que le champ existe par une lecture du source de la page.

Dans mon user story, le « sans qu'un utilisateur n'ayant pas ce/s rôle/s puisse afficher le champ » faisait plutôt référence au fait que l'on pourra pas tester/remplir l'ensemble des champs du formulaire si on dispose pas du/des/ rôle/s de la condition.

Si l'on souhaite vraiment une confidentialité forte du/des champ/s, il sera probablement préférable de faire appel au mécanisme des pages conditionnelles, avec la même condition sur le/s rôle/s de l'utilisateur connecté, voir carrément de faire un autre formulaire qui sera réservé à un rôle.

##### #4 - 09 janvier 2019 15:53 - Frédéric Péters

Côté combo on a eu |has\_role:... via [#15847](#).

##### #6 - 29 octobre 2019 15:14 - Thomas Noël

Une solution serait quelque chose comme :

```
--- a/wcs/users.py
```

```

+++ b/wcs/users.py
@@ -90,6 +90,14 @@ class User(StorableObject):
     def get_roles(self):
         return (self.roles or [])

+     def get_roles_names(self):
+         from .roles import Role
+         for role_id in self.get_roles():
+             try:
+                 yield Role.get(role_id).name
+             except KeyError: # role has been deleted
+                 pass
+
     def set_attributes_from_formdata(self, formdata):
         users_cfg = get_cfg('users', {})

```

qui permet de faire une condition comme

```
"Agent" in session_user.get_roles_names
```

ou plus génériquement avec une fonction "truc" dans le workflow :

```
form_role_truc_name in session_user.get_roles_names
```

Le session\_user fait que ça va marcher lors de la soumission, mais ces conditions ne sont plus correctes par la suite quand la demande est analysée par un autre utilisateur.

(Je reste vraiment pas intéressé par proposer ce hack sauf aux personnes de bonne volonté qui comprendront les tenants et aboutissants)

#### #7 - 29 octobre 2019 15:18 - Frédéric Péters

Mais pourquoi pas |has\_role:"...", par unité avec Combo ?

#### #8 - 29 octobre 2019 15:34 - Thomas Noël

Frédéric Péters a écrit :

Mais pourquoi pas |has\_role:"...", par unité avec Combo ?

Parce que je faignandise et que j'ai vraiment pas envie de faire de la pub à cette feature que je vois déjà utilisée de travers. Mais ok, je vais la faire, et on verra bien...

#### #9 - 29 octobre 2019 17:13 - Thomas Noël

- Statut changé de Nouveau à En cours

- Assigné à mis à Thomas Noël

#### #10 - 29 octobre 2019 22:05 - Thomas Noël

- Fichier 0001-misc-add-has\_role-filter-29342.patch ajouté

- Sujet changé de Connaitre le rôle de l'utilisateur connecté à Connaitre le rôle de l'utilisateur connecté (filtre has\_role)

- Statut changé de En cours à Solution proposée

- Patch proposed changé de Non à Oui

Voilà donc has\_role.

Note : xxx|has\_role:r renvoie faux si xxx n'est pas un objet user. L'idée est de ne pas cracher avec form\_user pour un formulaire sans utilisateur (ou session\_user sans session, dans un cron), ce genre de choses.

#### #11 - 30 octobre 2019 14:44 - Frédéric Péters

- Statut changé de Solution proposée à Solution validée

Ok.

#### #12 - 30 octobre 2019 14:58 - Thomas Noël

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

```
commit e22d2b52aeeb3c526bc9e49ef0de5500e618b59f
```

Author: Thomas NOEL <tnoel@entrouvert.com>

Date: Tue Oct 29 17:12:41 2019 +0100

misc: add has\_role filter (#29342)

### #13 - 06 novembre 2019 19:15 - Frédéric Péters

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

#### Fichiers

---

0001-misc-add-has\_role-filter-29342.patch

3,18 ko 29 octobre 2019

Thomas Noël