

w.c.s. - Bug #17359

Permettre de resoumettre une demande avec un demandeur différent du demandeur initial

05 juillet 2017 14:09 - Josué Kouka

Statut:	Nouveau	Début:	05 juillet 2017
Priorité:	Normal	Echéance:	
Assigné à:		% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposed:	Non		
Description			
Un des interets serait de pouvoir générer une nouvelle demande pour un traitement backoffice sans que le demandeur initial ne soit notifié.			

Historique

#2 - 05 juillet 2017 14:12 - Frédéric Péters

avec un demandeur différent du demandeur initial

Comment se ferait la sélection du nouveau demandeur ?

#3 - 05 juillet 2017 15:27 - Josué Kouka

Frédéric Péters a écrit :

avec un demandeur différent du demandeur initial

Comment se ferait la sélection du nouveau demandeur ?

En ajoutant un attribut `new_user` en extra. Vu le code de l'api le `new_user` devrait etre un dico avec à priori `{'id': ..., 'email': ..., 'name': }`. Dans le cas ou `new_user = {}` la demande est considérée comme étant anonyme.

#4 - 11 juillet 2017 16:22 - Josué Kouka

Josué Kouka a écrit :

Frédéric Péters a écrit :

avec un demandeur différent du demandeur initial

Comment se ferait la sélection du nouveau demandeur ?

En ajoutant un attribut `new_user` en extra. Vu le code de l'api le `new_user` devrait etre un dico avec à priori `{'id': ..., 'email': ..., 'name': }`. Dans le cas ou `new_user = {}` la demande est considérée comme étant anonyme.

L'attribution d'un nouvel utilisateur lors de la "resoumission" d'une demande n'a pas vraiment d'intérêt.

L'idée serait de permettre une soumission anonymisée via l'api `/submit` quand bien meme un utilisateur est associé. On pourrait passer un paramètre `anonymous` lors de l'appel.

Si ce paramètre est présent dans la query string, on ne lie pas la demande à l'utilisateur.

Vous pensez que c'est une solution adéquate ?

#5 - 11 juillet 2017 18:07 - Thomas Noël

Ma proposition.

Je pense que `/submit` devrait savoir gérer un dictionnaire "à plat", tel que les appels webservice de wcs savent faire.

Selon moi, on a de la place pour cela dans `api.py` :

```
...
def submit(self):
    if 'data' in json_input:
        # the published API expects data in 'data'.
    elif 'fields' in json_input:
        # but the API also supports data in 'fields', to match the json
        # output produced by wf/wscall.py.
    else:
        data = {} <--- ici on peut jouer un autre type d'appel
...

```

dans ce dernier cas je verrais bien ça :

```
data = {k[10:]: v for k,v in json_input.items() if k.startswith('form_var')}
```

et dans ce cas pour les meta :

```
meta = {k[5:]: v for k,v in json_input.items() if k.startswith('meta')}
```