w.c.s. - Bug #37095

form_var_user_* n'est pas initialisé dans les sources de donnée quand on reprend un formulaire via code de suivi

21 octobre 2019 11:59 - Benjamin Dauvergne

Statut:	Rejeté	**Début:**	21 octobre 2019
Priorité:	Normal	**Echéance:**	
Assigné à:	Nicolas Roche	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Oui

Description

1. faire une source de donnée dépendant d'un élément du profil utilisateur ~~dans son URL ou un paramètre~~ (ça marche aussi avec une source "expression Python")
2. avoir un ItemField lié à cette source sur la page 1
3. se connecter et commencer une demande, aller en page 2, attendre l'autosave ou conserver un brouillon et noter le code de suivi
4. se déconnecter et reprendre le formulaire avec le code de suivi
5. soumettre la demande

Constater que le l'ItemField ne contient plus que la valeur _raw (ou id) et pas les valeurs étendues _structured ou _display ("text").

Demandes liées:

Lié à w.c.s. - Bug #29218: Un numero de suivi généré depuis un brouillon de d...	**Fermé**	**19 décembre 2018**

Historique

#2 - 23 octobre 2019 11:35 - Nicolas Roche

- Assigné à mis à Nicolas Roche

#3 - 23 octobre 2019 13:55 - Nicolas Roche

- Fichier form-commande-resto.wcs ajouté

- Fichier menu-vegan.ods ajouté

Wunderbar !
Je reproduis avec ma source de donnée et le concours de Paul.
(je n'ai effectivement plus que la valeur _raw)
https://passerelle.dev.publik.love/csvdatasource/menu-pour-tous/query/choix/?qui={{ session_user_var_first_name }}

Scénario sans utiliser le code de suivi :

```
form_var_manger : tofu haché
form_var_manger_qui : Paul
form_var_manger_quoi : 2
form_var_manger_raw : 11
```

Scénario avec pause puis utilisation du code de suivi :

```
form_var_manger : None (<type 'NoneType'>)
form_var_manger_raw : 11
```

J'essaie d'écrire le test...

#4 - 23 octobre 2019 15:54 - Nicolas Roche

Question :
Est-ce que par chance test sur un champs commentaire qui utilise un élément du profil utilisateur suffirait ?

```
        fields.CommentField(id='1', type='comment',
            label='{{form_user_email}}'),
```

Parce qu'on retrouve également ce champ non renseigné suite à l'utilisation du code de suivi.

(et que le test est bien plus simple à écrire :)

```
def test_form_recall_logged_in_draft_using_tracking_code(pub):
    user = create_user(pub)
    formdef = create_formdef()
    formdef.fields = [
        fields.StringField(id='0', label='string',
            prefill={'type': 'string',
                     'value': 'here_1:{{form_user_email}}'}),
        fields.CommentField(id='1', type='comment',
            label='here_2:{{form_user_email}}'),
    ]
    formdef.enable_tracking_codes = True
    formdef.store()

    resp = login(get_app(pub), username='foo', password='foo').get('/test/')
    formdef.data_class().wipe()
    assert '<h3>Tracking code</h3>' in resp.body
    tracking_code = get_displayed_tracking_code(resp)
    assert tracking_code is not None
    assert 'here_1:foo@localhost' in resp.body
    assert 'here_2:foo@localhost' in resp.body
    resp = resp.forms[0].submit('submit')
    assert formdef.data_class().count() == 1
    formdata_id = formdef.data_class().select()[0].id

    # go back as anonymous
    pub.session_manager.session_class.wipe()
    resp = get_app(pub).get('/')
    resp.forms[0]['code'] = tracking_code
    resp = resp.forms[0].submit()
    assert resp.location == 'http://example.net/code/%s/load' % tracking_code
    resp = resp.follow()
    assert resp.location == 'http://example.net/test/%s' % formdata_id
    resp = resp.follow()
    assert resp.location.startswith('http://example.net/test/?mt=')
    resp = resp.follow()

    resp = resp.forms[1].submit('previous')
    assert 'here_1:foo@localhost' in resp.body
    assert 'here_2:foo@localhost' in resp.body  # <- ici on obtient seulement 'here_2:'
```

#5 - 23 octobre 2019 15:58 - Benjamin Dauvergne

Nicolas Roche a écrit :

> Question :
> Est-ce que par chance test sur un champs commentaire qui utilise une variable de session de l'utilisateur suffirait ?
> [...]
> Parce qu'on retrouve également ce champ non renseigné suite à l'utilisation du code de suivi.
> *(et que le test est bien plus simple à écrire :)*
> [...]

Oui le test est certainement plus simple comme ça et l'intitulé aurait pu être simplifié mais j'avais tellement la situation réelle en tête que j'ai parlé de source de donnée, le problème c'est juste que form_user_ n'est plus présent quand on reprend un formulaire via code de suivi. Ça ne devrait dépendre que de la valeur de formdata.user mais apparemment ce n'est pas le cas, formdata.user est perdu quelque part, je pense lors des nombreuses recréations d'un formdata depuis les données en session.

#6 - 28 octobre 2019 09:36 - Frédéric Péters

Ok donc la description c'est qu'en reprenant de manière anonyme la saisie d'une demande qui avait été initiée de manière authentifiée, il y aurait souhait de conserver l'utilisateur initialement à l'origine de la demande, c'est bien ça ?

#7 - 28 octobre 2019 09:53 - Nicolas Roche

Oui, c'est comme ça que je le comprend, maintenant je ne sais pas si c'est vraiment souhaitable.
Pour le CD06 à l'origine de ce ticket, le bug devrait être évité puisque le code de suivi a été désactivé :
https://dev.entrouvert.org/issues/36086#note-65

#8 - 28 octobre 2019 10:42 - Benjamin Dauvergne

Nicolas Roche a écrit :

> Oui, c'est comme ça que je le comprend, maintenant je ne sais pas si c'est vraiment souhaitable.
> Pour le CD06 à l'origine de ce ticket, le bug devrait être évité puisque le code de suivi a été désactivé :
> https://dev.entrouvert.org/issues/36086#note-65

C'est vrai que cela a un impact au niveau sécurité : selon les appels de WS fait pendant le remplissage ça donne accès à des parties du profil de l'usager (par exemple des données d'un logiciel métier, genre listes des enfants, des associations) sans que la personne ne se soit identifié.

Pour cela je peux comprendre qu'on conserve le comportement actuel mais dans ce cas il faudrait reprendre mon commentaire du #37107, sur un brouillon auquel un utilisateur est attaché il faudrait forcer l'authentification l'authentification soit obligatoire ou pas sur ce formulaire (actuellement ça n'arrive que si le formulaire est réservé aux utilisateurs authentifiés). Ça évite de trop modifier le comportement du mode brouillon/code de suivi classique et ça corrige l'éventuel souci de sécurité de mon premier paragraphe.

#9 - 15 novembre 2019 11:45 - Nicolas Roche

- Statut changé de Nouveau à Information nécessaire

- Planning changé de Non à Oui

> forcer l'authentification sur un brouillon auquel un utilisateur est attaché

en front et en backoffice également ?

> un brouillon auquel un utilisateur est attaché

on a déjà quelque-chose pour détecter ça ?

#10 - 18 novembre 2019 10:58 - Nicolas Roche

- Assigné à changé de Nicolas Roche à Benjamin Dauvergne

stp, 2 petites questions pour me mettre un peu sur la voie.

#11 - 18 novembre 2019 11:35 - Benjamin Dauvergne

- Description mis à jour

#12 - 18 novembre 2019 11:39 - Benjamin Dauvergne

Nicolas Roche a écrit :

> > forcer l'authentification sur un brouillon auquel un utilisateur est attaché
>
> en front et en backoffice également ?

En soumission backoffice ça n'a pas de sens (l'utilisateur est déjà authentifié, ce n'est juste pas le même) mais c'est un bon moyen de voir ce qui diffère (pourquoi form_var_user_ n'est pas accessible); je pense que c'est simplement qu'à la soumission ou récupération du brouillon en front le user est perdu (écrasé par session.user qui est vide) en soumission backoffice c'est forcément fait autrement (sinon ça ne marcherait tout simplement pas).

> > un brouillon auquel un utilisateur est attaché
>
> on a déjà quelque-chose pour détecter ça ?

Je ne comprends pas la question, il faut mettre en place : si form.user et pas session.user alors authentifier (possible qu'ensuite le formulaire ne soit pas accessible, si l'utilisateur authentifié n'est pas celui associé à la demande).

#15 - 19 novembre 2019 18:57 - Nicolas Roche

- Fichier 0001-forms-force-authentication-on-user-drafts-37095.patch ajouté

- Statut changé de Information nécessaire à Solution proposée

- Patch proposed changé de Non à Oui

Je propose ce patch pour forcer l'authentification sur un brouillon auquel un utilisateur est attaché.

#16 - 19 novembre 2019 19:00 - Frédéric Péters

- Assigné à changé de Benjamin Dauvergne à Nicolas Roche

Il va falloir attendre #36515 puis le rebaser et au moins apporter des modifications aux tests, pour ne pas utiliser resp.body.

#17 - 19 novembre 2019 19:01 - Nicolas Roche

- Lié à Bug #29218: Un numero de suivi généré depuis un brouillon de demande en mode non authentifié, ne peut être utilisé une fois connecté ajouté

#18 - 20 novembre 2019 15:19 - Nicolas Roche

- Fichier 0001-forms-force-authentication-on-user-drafts-37095.patch ajouté

Patch mis à jour avec resp.text et rebasage suite à l'intégration de #36515.

#19 - 20 novembre 2019 15:47 - Frédéric Péters

La description de ce ticket et le code du test dans le patch font référence à un problème que je ne comprends pas très bien, c'est marqué dans "# authenticated user retrieve form_user variables valuated".

Pour le premier champ, vu comme il y a préremplissage, que le brouillon va être sauvegardé, restauré, qu'on soit anonyme ou pas la valeur va être celle qui était dedans. J'ai l'impression qu'il faudrait tout à fait se dégager de ce ticket l'idée de form_user_whatever, qui n'a au final pas de rapport.

Ensuite, if session.is_anonymous_submitter(filled): si dans la session on a noté que l'usager devant son clavier est bien à l'origine de la demande, pourquoi alors lui refuser dans certains cas ?

Si je comprends bien ici, on est sur une décision de ne pas autoriser un utilisateur non connecté à charger un code de suivi associé à un utilisateur. Plutôt que faire ça à cet endroit, ça devrait plutôt, je trouve, être fait au moment de la demande de chargement, dans le load, ajouter un truc genre

```
if formdata.user_id and not get_request().user:
    # anonymous user asked to load a tracking code associated with an user, don't load, ask for authentication
 instead
    return redirect(login ?next=url du formdata)
```

#20 - 21 novembre 2019 16:21 - Nicolas Roche

- Fichier 0001-forms-force-authentication-on-user-drafts-37095.patch ajouté

> session.is_anonymous_submitter(filled) :
> dans la session on a noté que l'usager devant son clavier est bien à l'origine de la demande

D'après pdb, j'ai l'impression qu'il s'agit de l'usager qui fait le chargement de la demande

- cf https://dev.entrouvert.org/issues/3031 ajoute une référence au formdata dans la session de l'utilisateur anonyme
- cf https://dev.entrouvert.org/issues/10586 allow logged in users to benefit from tracking codes

/wcs/forms/root.py :

```
class TrackingCodeDirectory(Directory):
...
    def load(self):
    ...
        get_session().mark_anonymous_formdata(formdata)
```

exemple de la pile d'appel via les tests :

```
pub.session_manager.session_class.wipe()  #  demande anonyme
resp.forms[0]['code'] = tracking_code
resp = resp.forms[0].submit()
assert resp.location == 'http://example.net/code/%s/load' % tracking_code
resp = resp.follow()

## pile
  /home/nroche/src/wcs/wcs/forms/root.py(177)load()
  /home/nroche/src/wcs/wcs/sessions.py(57)mark_anonymous_formdata()
```

> ça devrait plutôt être fait au moment de la demande de chargement, dans le load,

oui, du coup ça répond à ma remarque ci-dessus.

A noter qu'avec ce patch, un agent ne semble pas (tests à l'usage) pouvoir utiliser son compte pour s'identifier lors de la récupération d'un brouillon (ok pour les demandes, qui elles, sont redirigées en backoffice) via un code de suivi posé sur le portail agent (mais vu la remarque ci-dessous ce n'est pas plus mal).

Je pose cette question pour approfondir à l'occasion :
Est-ce que les administrateur sont sensés pouvoir récupérer les brouillons des usagers via le code de suivi (en backoffice) ?
Cela me pose question parce qu'alors, les champs pré-remplis prennent les données de l'administrateur (nom, prénom, téléphone...)

#21 - 22 novembre 2019 09:58 - Frédéric Péters

> D'après pdb, j'ai l'impression qu'il s'agit de l'usager qui fait le chargement de la demande

Taper le code de suivi = devenir/être la personne à l'origine de la demande.

> Est-ce que les administrateur sont sensés pouvoir récupérer les brouillons des usagers via le code de suivi (en backoffice) ?

Je dirais que non, mais

> Cela me pose question parce qu'alors, les champs pré-remplis prennent les données de l'administrateur (nom, prénom, téléphone...)

Pour ce qui est saisie depuis le backoffice les infos de l'agent ne sont pas dans form_user_*.

#22 - 22 novembre 2019 11:00 - Nicolas Roche

> Pour ce qui est saisie depuis le backoffice les infos de l'agent ne sont pas dans form_user_*.

Oui (ok pour les demandes, qui elles, sont redirigées en backoffice), mais pour les brouillons :
en testant ce que voit un administrateur qui récupère un brouillon en backoffice,
concernant un formulaire qui utilise un champ avec l'option "Champ utilisateur" -> "Nom" comme paramètre "Préremplir",
cela me conduit dans wcs/fields.py::Field::get_prefill_value() :

```
        elif t == 'user' and user:
            x = self.prefill.get('value')
 ...
                userform = user.get_formdef()
                for userfield in userform.fields:
                    if userfield.id == x:
                        return (user.form_data.get(x)...
```

```
## et via pdb
> /home/nroche/src/wcs/wcs/fields.py(354)get_prefill_value()
(Pdb) print __return__
('admin', False)
```

Mais en tout cas ça colle avec la définition donnée juste au dessus :

> Taper le code de suivi = devenir/être la personne à l'origine de la demande.

Donc, une autre question :
Est-ce que c'est voulu que les utilisateurs utilisent le code de suivi pour récupérer leur brouillon ?

#23 - 22 novembre 2019 11:09 - Frédéric Péters

> Est-ce que c'est voulu que les utilisateurs utilisent le code de suivi pour récupérer leur brouillon ?

"utilisateurs" au sens large ou "utilisateurs connectés" ? Au sens large le code de suivi est justement là pour des utilisateurs non connectés, qui n'ont pas d'autre moyen pour revenir sur un brouillon. Au sens "utilisateurs connectés", perso, je pense que le code de suivi ne sert à rien.

#24 - 22 novembre 2019 13:18 - Benjamin Dauvergne

Frédéric Péters a écrit :

> Est-ce que c'est voulu que les utilisateurs utilisent le code de suivi pour récupérer leur brouillon ?

"utilisateurs" au sens large ou "utilisateurs connectés" ? Au sens large le code de suivi est justement là pour des utilisateurs non connectés, qui n'ont pas d'autre moyen pour revenir sur un brouillon. Au sens "utilisateurs connectés", perso, je pense que le code de suivi ne sert à rien.

Il restera le cas d'un brouillon commencé non connecté et repris en mode connecté, le cas des brouillons est particulier.

#25 - 23 novembre 2019 16:10 - Nicolas Roche

Oui, le patch proposé ne convre que (3) et (4), là tu étends le ticket à (1) et (2) :
je pense que j'ai quelques problèmes de cache et donc que ces tableaux ne sont peut-être pas corrects, notamment pour placer la frontière entre (1) et (2) et l'aspect aléatoire de (5)

Accès aux brouillons via le code de suivi :

commencé/repris	Anonyme	User1	User2	Agent, front	Agent, back
Anonyme	oui	oui (1)	oui (1)	oui (1)	oui (2)
Agent (saisie)	oui	oui (1)	oui (1)	oui (1)	oui (2)
User1	(3)	oui	non	non	oui (2)

Accès aux demandes via le code de suivi :

soumise/vue	Anonyme	User1	User2	Agent, front	Agent, back
Anonyme	oui	oui	oui	oui, front	oui, front
Agent (saisie)	oui	oui	oui	oui, front	oui, front
User1	(4)	oui	non	oui, (5)	oui, (5)

1. on repasse en utilisateur anonyme.
   si on se connecte depuis la page du brouillon, alors on passe en (2)
2. les champs sont pré-remplis en utilisant les variables de l'utilisateur,
   ensuite le brouillon est ré-affecté à l'utilisateur.
3. redirection vers une page de login,
   seul User1 peut se connecter (Agent ne peut pas)
4. redirection vers une page de login,
   seul User1 et Agent peuvent se connecter,
   Agent est redirigé en backoffice.
5. Parfois l'agent est redirigé en backoffice

(2) => peut-être que les brouillons ne devrait pas être accessibles à l'agent.

(mince, 25ème message, j'ai perdu)

#26 - 23 novembre 2019 16:40 - Frédéric Péters

> (mince, 25ème message, j'ai perdu)

Oui, il y a trop de distance par rapport à un sujet de ticket qui n'a plus vraiment de rapport, il faudrait repartir de zéro et exprimer dans un nouveau ticket un problème et une proposition; je serais pour rejeter ce ticket.

#27 - 01 décembre 2019 11:11 - Nicolas Roche

- Statut changé de Solution proposée à Rejeté

Repris à zéro dans sa globalité dans #38077, et plus spécifiquement dans #38079 pour la solution proposée.

Fichiers

form-commande-resto.wcs	1,88 ko	23 octobre 2019	Nicolas Roche
menu-vegan.ods	10,2 ko	23 octobre 2019	Nicolas Roche
0001-forms-force-authentication-on-user-drafts-37095.patch	3,38 ko	19 novembre 2019	Nicolas Roche

0001-forms-force-authentication-on-user-drafts-37095.patch	3,38 ko	20 novembre 2019	Nicolas Roche
0001-forms-force-authentication-on-user-drafts-37095.patch	2,24 ko	21 novembre 2019	Nicolas Roche