

Combo - Development #23535

Gestion "sous-page automatique"

02 mai 2018 15:21 - Frédéric Péters

Statut: Fermé	Début: 02 mai 2018
Priorité: Normal	Echéance:
Assigné à:	% réalisé: 0%
Catégorie:	Temps estimé: 0:00 heure
Version cible:	Planning:
Patch proposed: Oui	
Description L'idée c'est de pouvoir créer une page, mettons /fiche-usager/, et de définir dans le paramétrage de celle-ci que sous /fiche-usager/ il faut "quelque chose" (a priori dans la même boîte de dialogue que celle où on édite aujourd'hui le slug); ce "quelque chose" se trouverait ensuite posé dans le contexte, à disposition pour tout le monde. (et il pourrait y avoir un traitement particulier sur le <user_id> pour poser concerned_user).	
Demandes liées:	
Lié à Publik - Development #19756: Personnalisation accrue du portail agent p...	Fermé 29 octobre 2018 30 juin 2018
Lié à Combo - Development #8521: Autoriser plusieurs pages à avoir le même slug	Fermé 06 octobre 2015

Révisions associées

Révision cae8d0cd - 18 août 2018 14:07 - Frédéric Péters

general: add "sub slug" to create variable pages (#23535)

Historique

#1 - 02 mai 2018 15:21 - Frédéric Péters

- Lié à Development #19756: Personnalisation accrue du portail agent pour en faire aussi la page d'entrée des agents d'accueil ajouté

#2 - 08 août 2018 22:28 - Frédéric Péters

- Fichier 0001-general-add-sub-slug-to-create-variable-pages-23535.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Ça ajoute un attribut "sub slug" aux pages. (traduction à discuter)

Il prend alors une regex avec un groupe nommé, genre (?P<blah>[a-z]+). Et l'info se retrouve ensuite dans le contexte de rendu.

Exemple, une page /plop/ avec sub slug à (?P<blah>[a-z]+), aller sur /plop/whatever/ mettra dans le contexte "blah" avec comme valeur "whatever".

Au-dessus de ça il y a quatre lignes pour reconnaître user_id et name_id, et dans ces situations taper l'utilisateur ainsi trouvé dans le contexte dans "selected_user" (pour compatibilité avec ce qu'on avait déjà).

La page peut elle-même contenir des sous-pages, genre "toto" comme sous-page de "plop", aller sur /plop/whatever/toto/ fera le rendu de la page "toto", avec également whatever dans le contexte. (l'idée étant ainsi de pouvoir faire des sous-pages de la fiche usager.

Le comportement actuel sur un accès à une page demandant un sub slug quand l'info manque de l'url, exemple /plop/, c'est simplement une 404. (j'imaginai à un moment altérer le comportement de redirect_url, mais ça fait un truc bizarre, je me dis que mon option préférée s'il fallait autre chose qu'une 404, ce serait une redirection vers la page parente.

#3 - 08 août 2018 22:30 - Frédéric Péters

- Lié à Development #8521: Autoriser plusieurs pages à avoir le même slug ajouté

#4 - 16 août 2018 10:49 - Frédéric Péters

Le comportement actuel sur un accès à une page demandant un sub slug quand l'info manque de l'url, exemple /plop/, c'est simplement une 404. (j'imaginai à un moment altérer le comportement de redirect_url, mais ça fait un truc bizarre, je me dis que mon option préférée s'il fallait autre chose qu'une 404, ce serait une redirection vers la page parente.

Pour avoir continué à hésiter, je viens de pousser une branche avec comme comportement la redirection vers la page parente; ça me semble le plus pratique.

#5 - 16 août 2018 14:35 - Thomas Noël

Il y a aussi l'éventuelle possibilité d'une info vide genre /plop/toto, mais le parts = [x for x in request.path_info.strip('/').split('/') if x] mange les // et on va se retrouver avec toto comme valeur pour le sub slug, pas bon. A priori ça serait bien de retirer le "if x" dans la gestion de parts, c'est à dire avoir juste parts = request.path_info.strip('/').split('/') mais je n'ai pas encore bien mesuré l'impact, ie à quoi sert le "if x".

#6 - 16 août 2018 14:37 - Thomas Noël

Le "if x" a toujours été là... pour moi il n'est pas très pertinent... ou bien je rate une idée ?

#7 - 16 août 2018 14:44 - Frédéric Péters

Le "if x" a toujours été là... pour moi il n'est pas très pertinent... ou bien je rate une idée ?

Peut-être bien qu'on a des liens avec par erreur // et ça permet d'assurer quand même que la page soit servie. De mon côté je ne toucherais pas à ça et continuerais à considérer que // comme / (je viens de vérifier et côté filesystem ça se passe comme ça).

#8 - 16 août 2018 16:56 - Thomas Noël

Ok, le /// à l'intérieur ne compte pas (dans les fs je savais, pas sur http, on découvre tous les jours, tant mieux). Laissons donc ainsi.

Je serais pour afficher le sub slug dans combo/manager/templates/combo/page_view.html s'il existe, pour ne pas avoir à cliquer pour le voir.

Pas trop d'idée non plus pour un help_text, mais ça serait bien, histoire qu'on ne nous demande pas "c'est quoi ce truc?".

Donc en dehors de ces petits trucs d'UI, ça me semble tout bien (mais je dois relire wip/guichet dans son ensemble)

#9 - 17 août 2018 09:51 - Frédéric Péters

Je serais pour afficher le sub slug dans combo/manager/templates/combo/page_view.html s'il existe, pour ne pas avoir à cliquer pour le voir.

Fait.

Pas trop d'idée non plus pour un help_text, mais ça serait bien, histoire qu'on ne nous demande pas "c'est quoi ce truc?".

J'ai tapé un truc un peu technique quand même, "Regular expression to create variadic subpages. Matching named groups are exposed as context variables." mais ça gagnerait surtout à une page de documentation détaillant ça.

#10 - 17 août 2018 16:15 - Thomas Noël

Ici on devrait faire des try/except pour ne pas crasher si l'utilisateur n'existe pas (User.DoesNotExist), ou si le ctx['user_id'] ne représente pas un nombre (ValueError) :

```
def modify_global_context(request, ctx):
    if 'user_id' in ctx:
        ctx['selected_user'] = User.objects.get(id=ctx['user_id'])
    if 'name_id' in ctx and UserSAMLIdentifier:
        ctx['selected_user'] = UserSAMLIdentifier.objects.get(name_id=ctx['name_id']).user
```

deviendrait quelque chose comme ça :

```
def modify_global_context(request, ctx):
    if 'user_id' in ctx:
        try:
            ctx['selected_user'] = User.objects.get(id=ctx['user_id'])
        except (User.DoesNotExist, ValueError):
            pass
    if 'name_id' in ctx and UserSAMLIdentifier:
        try:
            ctx['selected_user'] = UserSAMLIdentifier.objects.get(name_id=ctx['name_id']).user
        except UserSAMLIdentifier.DoesNotExist:
            pass
```

Bon, le ValueError pourrait être géré via une regex correct slug, mais au moins les DoesNotExist

#11 - 17 août 2018 16:48 - Frédéric Péters

Poussé dans wip/guichet avec ce changement.

#12 - 17 août 2018 17:42 - Thomas Noël

- *Statut changé de Solution proposée à Solution validée*

Ack, avec ou sans cet ajout en fin de tests/test_public.py::test_sub_slug si tu veux :

```
# bad user id => no selected_user
page2.sub_slug = '(?P<user_id>[0-9a-z]+)'
page2.save()
resp = app.get('/users/9999999/', status=200)
assert 'XXYY' in resp.text
resp = app.get('/users/abc/', status=200)
assert 'XXYY' in resp.text
```

```
# custom behaviour for <name_id>, it will add the SAML user to context
with mock.patch('combo.public.views.UserSAMLIdentifier') as user_saml:
    class DoesNotExist(Exception):
        pass
    user_saml.DoesNotExist = DoesNotExist
    def side_effect(*args, **kwargs):
        name_id = kwargs['name_id']
        if name_id == 'foo':
            raise user_saml.DoesNotExist
        return mock.Mock(user=john_doe)
    mocked_objects = mock.Mock()
    mocked_objects.get = mock.Mock(side_effect=side_effect)
    user_saml.objects = mocked_objects
```

```
page2.sub_slug = '(?P<name_id>[0-9a-z.]+)'
page2.save()
resp = app.get('/users/john.doe/', status=200)
assert 'XXjohn.doeYY' in resp.text
# unknown name id => no selected_user
resp = app.get('/users/foo/', status=200)
assert 'XXYY' in resp.text
```

#13 - 18 août 2018 14:07 - Frédéric Péters

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

J'ai ajouté ces tests et poussé,

```
commit cae8d0cda594b7fefae762b5ae56dd10612d89e3
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Wed Aug 8 22:18:36 2018 +0200
```

```
general: add "sub slug" to create variable pages (#23535)
```

#14 - 23 décembre 2018 15:10 - Frédéric Péters

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

Fichiers

0001-general-add-sub-slug-to-create-variable-pages-23535.patch	13,5 ko	08 août 2018	Frédéric Péters
--	---------	--------------	-----------------