Passerelle - Development #42627

sms: ajouter une classe SMSConnectorForm

07 mai 2020 14:45 - Nicolas Roche

Statut:	Rejeté	**Début:**	07 mai 2020
Priorité:	Normal	**Echéance:**	
Assigné à:	Nicolas Roche	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

côté UI ça m'irait d'avoir un <select> avec les presets

Pour pouvoir ensuite modifier les widgets dans sms/forms.py

Demandes liées:

Lié à Passerelle - Development #39650: connecteurs SMS, option pour refuser l...	**Fermé**	**07 février 2020**

Historique

#1 - 07 mai 2020 14:45 - Nicolas Roche

- Lié à Development #39650: connecteurs SMS, option pour refuser les numéros surchargés ajouté

#2 - 07 mai 2020 15:11 - Nicolas Roche

- Fichier 0001-sms-add-an-SMSConnectorForm-class-42627.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

#3 - 15 mai 2020 18:11 - Frédéric Péters

Avoir ces imports de bouts de SMS proprement déplacés dans une application à part qui redébarquent importés à la volée dans une méthode ici me fait dire qu'il y a un gros problème.

Il y a déjà :

```
        if hasattr(app, 'get_form_class'):
            self.form_class = app.get_form_class()
```

pour permettre à un connecteur de taper sa propre classe de formulaire d'édition.

Qu'est-ce que ce patch essaie de faire/contourner ?

#4 - 19 mai 2020 12:25 - Nicolas Roche

Je n'arrivais pas à m'en sortir avec les classes AppConfig déclarées dans les fichiers *init*.py.
Je réalise que si je défini ce fichier pour chaque apps SMS, le code que tu mentionnes fait que ça marche tout seul (et donc mis à par ces fichiers ce patch ne contiendrait plus rien).

Le test n'est pas représentatif de ce que je cherche à faire (ajouter des champs de formulaire génériques pour les connecteurs SMS). Du coup je préfère fusionner ce patch avec #39650 dont le premier patch ajoute de tels champs
(et rejeter ce ticket).

#5 - 19 mai 2020 12:41 - Frédéric Péters

Pas bien sûr d'avoir compris ta réponse mais bien sûr rejeter ce ticket s'il n'a plus de raison d'être c'est ok.

#6 - 19 mai 2020 16:23 - Nicolas Roche

Dans mon précédent message, je n'avais pas testé l'ajout d'un connecteur SMS : le formulaire n'hérite plus des champs de bases comme par exemple l'id.

Finalement, je ne m'en sort pas sans la fabrique de formulaires (modelform_factory appelé en dessous des lignes que tu mentionnes) : je peux définir un formulaire pour chacun des connecteur SMS, dérivant du formulaire générique sms/forms.py...

```
from passerelle.sms.forms import SMSConnectorForm
from .models import ChoositSMSGateway

class ChoositSMSConnectorForm(SMSConnectorForm):
    class Meta:
        model = ChoositSMSGateway
        fields = '__all__'
```

... mais alors il me faudra tout réinventer pour gérer l'affichage ou non du slug, suivant qu'il s'agit d'un ajout ou d'une mise à jour. Donc, je perd tout l'avantage d'utiliser un connecteur dérivant de GenericConnectorMixin.

Je pense finalement que ma première proposition n'était si mauvaise pour permettre à un connecteur SMS d'hériter de champs de formulaires customisés communs à tous les connecteurs SMS.

#7 - 19 mai 2020 17:26 - Frédéric Péters

Je propose alors de rejeter ce ticket, parce que "avoir ces imports de bouts de SMS proprement déplacés dans une application à part qui redébarquent importés à la volée dans une méthode ici me fait dire qu'il y a un gros problème.".

Et que quelque chose se passe ailleurs, parce que sur ce ticket dont je ne pige ni le sens ni les explications successive, on n'arrivera nulle part.

#8 - 19 mai 2020 19:14 - Nicolas Roche

- Statut changé de Solution proposée à Rejeté

Fichiers

0001-sms-add-an-SMSConnectorForm-class-42627.patch	7,43 ko	07 mai 2020	Nicolas Roche