

Passerelle - Development #42427

Déplacer le code clean_number utilisés par les connecteurs SMS dans SMSResource

04 mai 2020 09:51 - Nicolas Roche

| | | | |
|--|---------------|----------------------|------------------------|
| Statut: | Fermé | Début: | 04 mai 2020 |
| Priorité: | Normal | Echéance: | |
| Assigné à: | Nicolas Roche | % réalisé: | 0% |
| Catégorie: | | Temps estimé: | 0:00 heure |
| Version cible: | | Planning: | Non |
| Patch proposed: | Oui | | |
| Description | | | |
| Dans #39650 on va se baser sur la sortie de clean_number() pour valider les numéros autorisés, et donc on peut supposer que cette fonction sera également appelée par tous les nouveaux connecteurs SMS. | | | |
| Et ce sont les numéros passés par clean_numbers qui arrivent ici, donc on connaît la structure. | | | |
| Demandes liées: | | | |
| Lié à Passerelle - Development #39650: connecteurs SMS, option pour refuser l... | | Fermé | 07 février 2020 |

Révisions associées

Révision ae903d89 - 29 mai 2020 10:45 - Nicolas Roche

sms: move default country_code/trunk_prefix to base class (#42427)

Historique

#1 - 04 mai 2020 10:05 - Nicolas Roche

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

#2 - 04 mai 2020 10:06 - Frédéric Péters

Mais c'est déjà le cas :

```
passerelle/apps/choosit/models.py:         destinations = self.clean_numbers(destinations,  
passerelle/apps/mobytt/models.py:         destinations = self.clean_numbers(destinations,  
passerelle/apps/orange/models.py:         destinations = self.clean_numbers(  
passerelle/apps/ovh/models.py:         destinations = self.clean_numbers(destinations,  
passerelle/apps/oxyd/models.py:         destinations = self.clean_numbers(destinations,  
passerelle/base/models.py:     def clean_numbers(cls, destinations, default_country_code='33',
```

(?)

#3 - 04 mai 2020 13:58 - Nicolas Roche

Oui, puisque c'est généralisé, placer cet appel dans le code générique SMS, en dehors des connecteurs.

(Afin d'éviter que le code d'une potentielle nouvelle fonction de vérification des numéros autorisés soit appelé dans chacun des connecteurs).

#4 - 04 mai 2020 14:00 - Frédéric Péters

Je ne comprends toujours pas ce que tu veux faire ici, le code est déjà factorisé, unique, appelé par les différents connecteurs. (mais bon, je peux me taire et attendre le patch pour comprendre).

#5 - 06 mai 2020 16:47 - Nicolas Roche

- Fichier 0001-sms-factorize-clean_number-call-42427.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Je souhaiterais ne pas appeler le code depuis les différents connecteurs, du fait que :

- ils l'utilisent tous sans exception
- que la validation des numéros autorisés se basera dessus

Et donc je voudrais factoriser l'appel à cette fonction (cf ma [branche](#) sur [#39650](#)) :

```
@endpoint (perm='can_send_messages', methods=['post'])
def send(self, request, *args, **kwargs):
...
    data['to'] = self.clean_numbers(data['to'])
    self.authorize_numbers(data['to'])
...
    result = {'data': self.send_msg(data['message'], data['from'], data['to'], stop=stop)}
```

#6 - 06 mai 2020 17:09 - Frédéric Péters

Ok, je pense que ça aide de prendre les choses dans l'ordre,

1/ tous les connecteurs SMS définissent les attributs `default_country_code` & `default_trunk_prefix` → ils pourraient donc être dans la classe de base.

2/ poser qu'alors l'appel à `clean_numbers()`, qui a besoin de ces infps, peut également être déplacé au niveau de la classe de base.

3/ ne pas recopier les commentaires inopportuns en faisant ça,

FIXME: add regexp field, to check destination and from format

→ ça n'aura pas lieu ainsi, il y a un autre ticket en cours pour la validation

unfortunately it lacks a batch API...

→ lamentation inutile maintenant que le code ne va plus être prévu pour permettre à un fournisseur qui offrirait ça de l'exploiter.

Yeah France first !

(passons)

Et pour revenir à mes interrogations initiales, il ne s'agit donc pas du tout de factoriser le code de `clean_numbers`, mais bien les appels à cette méthode.

#7 - 07 mai 2020 14:43 - Nicolas Roche

- Fichier `0001-sms-factorize-clean_number-call-42427.patch` ajouté

#8 - 15 mai 2020 18:07 - Frédéric Péters

Le sujet du commit continue à m'interloquer autant que le sujet du ticket; plutôt faire genre "move default country_code/trunk_prefix to base class" ?

#9 - 18 mai 2020 16:37 - Nicolas Roche

- Fichier `0001-sms-move-default-country_code-trunk_prefix-to-base-c.patch` ajouté

- Sujet changé de *Factoriser le code clean_number utilisés par les connecteurs SMS à Déplacer le code clean_number utilisés par les connecteurs SMS dans SMSResource*

#10 - 25 mai 2020 16:47 - Frédéric Péters

- Statut changé de *Solution proposée* à *Solution validée*

#11 - 29 mai 2020 10:49 - Nicolas Roche

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

```
commit ae903d89c357b34eda22f616f0588d8b18c50677
Author: Nicolas ROCHE <nroche@entrouvert.com>
Date: Tue May 5 11:26:48 2020 +0200
```

```
    sms: move default country_code/trunk_prefix to base class (#42427)
```

#12 - 29 mai 2020 14:16 - Frédéric Péters

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

Fichiers

| | | | |
|--|---------|-------------|---------------|
| 0001-sms-factorize-clean_number-call-42427.patch | 15,8 ko | 06 mai 2020 | Nicolas Roche |
|--|---------|-------------|---------------|

| | | | |
|---|---------|-------------|---------------|
| 0001-sms-factorize-clean_number-call-42427.patch | 15,6 ko | 07 mai 2020 | Nicolas Roche |
| 0001-sms-move-default-country_code-trunk_prefix-to-base-c.patch | 15,7 ko | 18 mai 2020 | Nicolas Roche |