

## Passerelle - Bug #7866

### Erreur lors de la vérification de la signature d'un API user

16 juillet 2015 16:45 - Serghei Mihai

<b>Statut:</b>	Fermé	<b>Début:</b>	16 juillet 2015
<b>Priorité:</b>	Normal	<b>Echéance:</b>	
<b>Assigné à:</b>	Serghei Mihai	<b>% réalisé:</b>	0%
<b>Catégorie:</b>		<b>Temps estimé:</b>	0:00 heure
<b>Version cible:</b>		<b>Planning:</b>	
<b>Patch proposed:</b>	Oui		

#### Description

Arrivé sur la plateforme de dev de Fontenay:

```
Internal Server Error: /teamnet-axel/fsb/family/
Traceback (most recent call last):
  File "/usr/lib/python2.7/dist-packages/django/core/handlers/base.py", line 111, in get_response
    response = wrapped_callback(request, *callback_args, **callback_kwargs)
  File "/usr/lib/python2.7/dist-packages/passerelle/urls_utils.py", line 52, in f
    return func(request, *args, **kwargs)
  File "/usr/lib/python2.7/dist-packages/django/views/generic/base.py", line 69, in view
    return self.dispatch(request, *args, **kwargs)
  File "/usr/lib/python2.7/dist-packages/django/views/generic/base.py", line 87, in dispatch
    return handler(request, *args, **kwargs)
  File "/usr/lib/python2.7/dist-packages/passerelle/urls_utils.py", line 123, in _wrapped_view
    if not is_authorized(request, obj, perm):
  File "/usr/lib/python2.7/dist-packages/passerelle/urls_utils.py", line 113, in is_authorized
    return set(users).intersection(get_request_users(request))
  File "/usr/lib/python2.7/dist-packages/passerelle/urls_utils.py", line 83, in get_request_users
    if check_query(query, signature_user.key):
  File "/usr/lib/python2.7/dist-packages/passerelle/base/signature.py", line 62, in check_query
    return check_string(unsigned_query, signature, key, algo=algo)
  File "/usr/lib/python2.7/dist-packages/passerelle/base/signature.py", line 66, in check_string
    signature2 = sign_string(s, key, algo=algo)
  File "/usr/lib/python2.7/dist-packages/passerelle/base/signature.py", line 43, in sign_string
    hash = hmac.HMAC(key, digestmod=digestmod, msg=s)
  File "/usr/lib/python2.7/hmac.py", line 72, in __init__
    self.outer.update(key.translate(trans_5C))
TypeError: character mapping must return integer, None or unicode
```

Réproduit également dans un shell python avec la clé en unicode.

#### Révisions associées

Révision bfec6d12 - 04 août 2015 17:39 - Serghei Mihai

fix key type in request signing (#7866)

Tests for anonymous and signed api access added

#### Historique

#1 - 16 juillet 2015 16:51 - Frédéric Péters

Ça se prêterait très bien à l'ajout d'un test.

#2 - 17 juillet 2015 10:11 - Serghei Mihai

- Fichier 0001-fix-key-type-in-request-signing-7866.patch ajouté

- Assigné à mis à Serghei Mihai

Le voici

### #3 - 17 juillet 2015 10:16 - Frédéric Péters

Il y a déjà une infra de tests dans tests/, il faudrait l'utiliser. (par ailleurs j'imaginai plutôt un test montant au moins jusqu'au is\_authorized).

### #4 - 22 juillet 2015 14:20 - Serghei Mihai

Tests d'accès à l'api d'un connecteur en mock-ant la réponse.

### #5 - 22 juillet 2015 14:20 - Serghei Mihai

- Fichier 0001-fix-key-type-in-request-signing-7866.patch ajouté

avec le patch

### #6 - 22 juillet 2015 15:01 - Frédéric Péters

Perso j'ai plutôt tendance à organiser les choses différemment; ici l'objectif que je mettais était de monter jusqu'à is\_authorized(), et d'en tester des variations, j'aurais utilisé un accès direct pour créer et faire varier l'environnement des objets (càd que j'aurais instancié un objet ApiUser plutôt que le créer via l'interface web, j'aurais instancié un connecteur, de la même façon).

Et les tests auraient donc été des variations autour de is\_authorized, par exemple :

- Accès anonyme (présent dans ton test)
- ApiUser avec keytype à SIGN (comme ton test), accès avec bonne signature, accès avec mauvaise signature
- ApiUser avec keytype à SIGN, accès avec de l'authent HTTP
- ApiUser avec keytype à API, bonne clé, mauvaise clé
- ApiUser sans keytype (et donc succès)
- ApiUser avec restriction d'IP

Aussi, aujourd'hui, sans passer par mock, il y a moyen de distinguer l'erreur d'authent en utilisant un connecteur SMS. (en passant un mauvais payload).

Pour ce ticket, je dirais que j'aimerais bien voir le test\_api\_access.py avec deux tests, qui instancient un environnement, et qui aurait deux tests, pour reprendre ma liste "Accès anonyme", "ApiUser avec keytype à SIGN, accès avec bonne signature".

### #7 - 29 juillet 2015 10:11 - Serghei Mihai

- Fichier 0001-fix-key-type-in-request-signing-7866.patch ajouté

Ok.

Je fais un ticket séparé pour rajouter les autres tests

### #8 - 29 juillet 2015 16:30 - Serghei Mihai

- Fichier 0001-fix-key-type-in-request-signing-7866.patch supprimé

### #9 - 29 juillet 2015 16:33 - Serghei Mihai

- Fichier 0001-fix-key-type-in-request-signing-7866.patch ajouté

### #10 - 04 août 2015 18:14 - Serghei Mihai

- Statut changé de Nouveau à Résolu (à déployer)

```
commit bfec6d12555aea171e71a921de9f50155e7f420d
Author: Serghei Mihai <smihai@entrouvert.com>
Date: Wed Jul 29 10:06:51 2015 +0200
```

```
fix key type in request signing (#7866)
```

```
Tests for anonymous and signed api access added
```

### #11 - 06 septembre 2015 20:35 - Serghei Mihai

- Statut changé de Résolu (à déployer) à Fermé

## Fichiers

0001-fix-key-type-in-request-signing.patch	800 octets	16 juillet 2015	Serghei Mihai
0001-fix-key-type-in-request-signing-7866.patch	1,96 ko	17 juillet 2015	Serghei Mihai
0001-fix-key-type-in-request-signing-7866.patch	5,08 ko	22 juillet 2015	Serghei Mihai
0001-fix-key-type-in-request-signing-7866.patch	2,92 ko	29 juillet 2015	Serghei Mihai