

Passerelle - Bug #53697

échec de test_esirius.test_pre_request[xxx]

03 May 2021 06:38 PM - Frédéric Péters

Status:	Solution déployée	Start date:	03 May 2021
Priority:	Normal	Due date:	
Assignee:	Nicolas Roche	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		Planning:	Yes
Patch proposed:	Yes		

Description

Suite à [#53594](#),

<https://jenkins.entrouvert.org/job/passerelle/2126/>

```
connector = <ESirius: >, secret_key = 'xxx'

@pytest.mark.parametrize('secret_key', ['xxx', ''])
def test_pre_request(connector, secret_key):
    @httmock.urlmatch(netloc='esirius.example.net', path='/an/uri/', method='GET')
    def esirius_mock(url, request):
        assert request.headers['Accept'] == 'application/json; charset=utf-8'
        assert bool(request.headers.get('token_info_caller')) == bool(secret_key)
        if secret_key:
            assert request.headers['token_info_caller'][:42] == b'f3G6sjRZETBam6vcdrAxmvJQTX5h
h60jZ8XlU06SMo'
        return httmock.response(200)

    connector.secret_key = secret_key
    connector.save()
    with httmock.HTTMock(esirius_mock):
>         connector.request('an/uri/', method='get', params="somes")

tests/test_esirius.py:122:
-----
passerelle/apps/esirius/models.py:152: in request
    response = self.requests.request(method=method, url=url, headers=headers, params=params, json=
json)
passerelle/utils/__init__.py:300: in request
    response = super(Request, self).request(method, url, **kwargs)
/tmp/tox-jenkins/passerelle/py3-djangol11-black-coverage/lib/python3.7/site-packages/requests/sess
ions.py:542: in request
    resp = self.send(prepare_request(method=method, url=url, headers=headers, params=params, json=
json))
passerelle/utils/__init__.py:317: in send
    response = super(Request, self).send(request, **kwargs)
/tmp/tox-jenkins/passerelle/py3-djangol11-black-coverage/lib/python3.7/site-packages/httmock.py:16
3: in _fake_send
    response = self.intercept(request, **kwargs)
/tmp/tox-jenkins/passerelle/py3-djangol11-black-coverage/lib/python3.7/site-packages/httmock.py:22
1: in intercept
    res = first_of(self.handlers, url, request)
/tmp/tox-jenkins/passerelle/py3-djangol11-black-coverage/lib/python3.7/site-packages/httmock.py:14
1: in first_of
    res = handler(*args, **kwargs)
/tmp/tox-jenkins/passerelle/py3-djangol11-black-coverage/lib/python3.7/site-packages/httmock.py:10
0: in inner
    return func(self_or_url, url_or_request, *args, **kwargs)
-----

url = SplitResult(scheme='https', netloc='esirius.example.net', path='/an/uri/', query='somes', fr
agment='')
```

```
request = <PreparedRequest [GET]>

    @httpmock.urlmatch(netloc='esirius.example.net', path='/an/uri/', method='GET')
    def esirius_mock(url, request):
        assert request.headers['Accept'] == 'application/json; charset=utf-8'
        assert bool(request.headers.get('token_info_caller')) == bool(secret_key)
        if secret_key:
>         assert request.headers['token_info_caller'][:42] == b'f3G6sjRZETBam6vcdRAxmvJQTX5hh6Oj
Z8X1UO6SMo'
E         AssertionError: assert b'f3G6sjRZETB...6OjILg+vGd6TG' == b'f3G6sjRZETB...6OjZ8X1UO6SMo
'
E             At index 32 diff: b'I' != b'Z'
E             Use -v to get the full diff
```

Related issues:

Related to Passerelle - Development #53594: TypeError: Object of type builtin...

Solution déployée 06 Apr 2021

Associated revisions

Revision 59186c7c - 04 May 2021 04:45 PM - Nicolas Roche

esirius: ignore DES timestamp within tests (#53697)

History

#1 - 03 May 2021 06:39 PM - Frédéric Péters

- Description updated

#2 - 03 May 2021 06:43 PM - Frédéric Péters

- Related to Development #53594: TypeError: Object of type builtin_function_or_method is not JSON serializable added

#3 - 04 May 2021 09:31 AM - Nicolas Roche

- Assignee changed from Benjamin Dauvergne to Nicolas Roche

Ça me semble lié à la version de cryptodome : ça passe avec

```
$ pip install pycryptodomex==3.9.9
```

alors que sinon ça ne passe plus, même si je reviens au commit qui introduit le connecteur.

Dans le test je coupe la partie de la chaîne crypté liée à l'heure du chiffrement (pour avoir un test reproductible sans avoir à utiliser freeze pour autant).

Il semblerait que cette partie prend 10 octets de plus, j'essaye de comprendre pourquoi...

#4 - 04 May 2021 10:08 AM - Nicolas Roche

- Patch proposed changed from No to Yes

- Status changed from Nouveau to Solution proposée

- File 0001-esirius-ignore-DES-timestamp-within-tests-53697.patch added

Je n'ai réussi pas trouver le commit qui explique la régression sur le test.

J'imagine juste quelque chose comme

<https://github.com/Legrandin/pycryptodome/commit/853054937e29e6cd3dc61228be1dc768b688460b#diff-fe8b2c4039525ca3fead7acd9867edecb9f87d92764718533fa68f2f2e81d1f5>

Dans les tests il y en a un autre qui valide le token complet (test_token : avec une date fixée par freeze) et donc cette erreur ne montre pas une régression.

Le test qui plante ici ne teste que la partie supposée fixe du token.

Il est redondant et on pourrait même le retirer.

#5 - 04 May 2021 01:04 PM - Benjamin Dauvergne

- *Planning changed from No to Yes*

Tu ne veux pas juste décoder le token et vérifier ce qui doit l'être ? Ça me semble plus sûr.

#6 - 04 May 2021 02:50 PM - Nicolas Roche

- *File 0001-esirius-ignore-DES-timestamp-within-tests-53697.patch added*

(fait)

#7 - 04 May 2021 04:29 PM - Benjamin Dauvergne

- *Status changed from Solution proposée to Solution validée*

#8 - 04 May 2021 05:17 PM - Nicolas Roche

- *Status changed from Solution validée to Résolu (à déployer)*

(quelqu'un l'a déjà poussé pour moi)

```
commit 59186c7cf9bb9c6adaeca379dea8947f75ee98ee
Author: Nicolas ROCHE <nroche@entrouvert.com>
Date: Tue May 4 09:56:32 2021 +0200
```

```
esirius: ignore DES timestamp within tests (#53697)
```

#9 - 04 May 2021 11:17 PM - Frédéric Péters

- *Status changed from Résolu (à déployer) to Solution déployée*

Files

0001-esirius-ignore-DES-timestamp-within-tests-53697.patch	1.31 KB	04 May 2021	Nicolas Roche
0001-esirius-ignore-DES-timestamp-within-tests-53697.patch	2.68 KB	04 May 2021	Nicolas Roche