

## Passerelle - Development #10299

### logger tous les input/output de python-requests

15 mars 2016 09:12 - Josué Kouka

<b>Statut:</b>	Fermé	<b>Début:</b>	15 mars 2016
<b>Priorité:</b>	Normal	<b>Echéance:</b>	
<b>Assigné à:</b>	Josué Kouka	<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		
<b>Description</b>			

#### Révisions associées

##### Révision fd498cb6 - 06 avril 2016 18:39 - Josué Kouka

add logger to every requests call (#10299)

##### Révision cf64834e - 06 avril 2016 19:29 - Serghei Mihai (congé, retour 15/05)

logging: add missing migrations for 'log\_level' attribute (#10299)

#### Historique

##### #1 - 15 mars 2016 09:15 - Josué Kouka

- Statut changé de Nouveau à En cours

##### #2 - 15 mars 2016 12:09 - Josué Kouka

- Fichier 0001-add-logger-for-every-requests-call-10299.patch ajouté

- Patch proposed changé de Non à Oui

##### #3 - 15 mars 2016 15:03 - Josué Kouka

Pouvoir choisir de logger ou pas un connecteur ( checkbox interface connecteur )

```
if connector.log == True:
    connector.requests = passerelle.utils.RequestSessionWrapper(connector.log)
else:
    connector.requests = requests
```

##### #4 - 15 mars 2016 15:08 - Frédéric Péters

Dans mon idée, dans une classe de base des connecteurs,

```
@property
def requests(self):
    if self.debug_enabled:
        return passerelle.utils.LoggedRequests()
    else:
        return requests # le module
```

##### #5 - 16 mars 2016 14:36 - Josué Kouka

- Fichier 0001-add-logger-for-every-requests-call-10299.patch ajouté

- Tracker changé de Bug à Development

##### #6 - 16 mars 2016 15:35 - Josué Kouka

- Fichier 0001-add-logger-for-every-requests-call-10299.patch ajouté

##### #7 - 25 mars 2016 15:41 - Benjamin Dauvergne

- le log\_level = getattr(... dans requests()) ne sert à rien

- j'aurai nommé le logger passerelle.resource.%s.%s' % (slugify(self.\_\_class\_\_.\_\_name\_\_), self.slug)
- dans LoggedRequest.request() il faut aussi tronquer response.content quand il est passé dans le dico extra

PEP8:

- un espace après % quand on formate une chaîne
- pas de parenthèse autour des arguments à % quand il n'y en a qu'un
- pas d'espace autour de = dans les keyword arguments d'une fonction (install pyflake et cherche le module emacs qui va bien :)

Il manque les changements au packaging et aux settings par défaut (sinon tout ça ne sert à rien), genre

```
'loggers': {
    'passerelle.resource': {
        'level': 'DEBUG',
        'propagate': False,
        'handlers': ...,
    },
    ...
}
```

On met par défaut au niveau DEBUG puisque le vrai niveau se définit maintenant au niveau de chaque ressource.

Par défaut on envoie sur la console et dans les settings debian il faut reprendre la liste des handlers du logger racine venant de debian\_config\_common.py.

#### #8 - 25 mars 2016 17:16 - Serghei Mihai (congé, retour 15/05)

Benjamin Dauvergne a écrit :

- pas d'espace autour de = dans les keyword arguments d'une fonction (install pyflake et cherche le module emacs qui va bien :)

Il suffit de taper C-x-m-e-c

#### #9 - 25 mars 2016 18:05 - Josué Kouka

- Fichier 0001-add-logger-for-every-requests-call-10299.patch ajouté

#### #10 - 25 mars 2016 18:07 - Josué Kouka

Voici ce que ça donne

```
Mar 25 18:01:33 debian DEBUG passerelle.resource.parispocgru.wso2.request: Content-Length:38 Accept-Encoding:g
zip, deflate Accept:*/.* User-Agent:python-requests/2.7.0 CPython/2.7.11+ Linux/4.3.0-1-amd64 Connection:keep-a
live Content-Type:application/x-www-form-urlencoded
Mar 25 18:01:33 debian DEBUG passerelle.resource.parispocgru.wso2.request: content-length:485 server:nginx con
nection:keep-alive access-control-allow-credentials:true date:Fri, 25 Mar 2016 17:01:33 GMT access-control-all
ow-origin:* content-type:application/json
Mar 25 18:01:33 debian DEBUG passerelle.resource.parispocgru.wso2.request: '{\n  "args": {}, \n  "data": "", \n
  "files": {}, \n  "form": {\n    "stuff": "that is a simple thing", \n    "user": "josh"\n  }, \n  "headers"
: {\n    "Accept": "*/.*", \n    "Accept-Encoding": "gzip, deflate", \n    "Content-Length": "38", \n    "Conte
nt-Type": "application/x-www-form-urlencoded", \n    "Host": "httpbin.org", \n    "User-Agent": "python-reques
ts/2.7.0 CPython/2.7.11+ Linux/4.3.0-1-amd64"\n  }, \n  "json": null, \n  "origin": "80.12.92.47", \n  "url":
"https://httpbin.org/post"\n}\n'
```

#### #11 - 25 mars 2016 18:10 - Josué Kouka

- Fichier 0001-add-logger-for-every-requests-call-10299.patch ajouté

#### #12 - 25 mars 2016 19:13 - Benjamin Dauvergne

Tu ne reprends pas la liste des handlers du logger racine, le logger racine c'est celui nommé ''.

Ton exemple de sorti de log c'est avec quel configuration des logs ? Ça ne ressemble pas à ce qu'il y a dans hobo et dans passerelle il n'y a rien non plus on dirait. Surtout je ne comprend pas d'où vient le .request après le passerelle.resource.parpocgru.wso2.

Pour les logs de debug il faudrait peut-être ajouter un texte introductif, genre 'Request headers, ', 'Response headers, ', 'Response payload', et remplacer 'Payload' par 'Request payload, ' sinon on ne sait pas trop ce qu'on regarde.

En l'état les logs sur les headers sont très difficiles à lire, j'ajouterai un espace après le ':' et peut-être que je séparerai par des pipes juste pour voir ('|').

#### #13 - 25 mars 2016 19:15 - Thomas Noël

Josué Kouka a écrit :

Voici ce que ça donne

Y'a pas l'URL ?

Il faudrait donc l'URL, avec devant le verbe (GET/POST/...) ; mais aussi des indications "Request Headers:", "Request Payload:", "Response Headers:", "Response Payload:" sur les lignes qui suivent...

#### #14 - 29 mars 2016 08:57 - Josué Kouka

Benjamin Dauvergne a écrit :

Tu ne reprends pas la liste des handlers du logger racine, le logger racine c'est celui nommé "".

~~Normalement le 'handlers': LOGGING['handlers'] le fait.~~ Ah oui t'as raison, faut juste les lister pas, les redefinir

Ton exemple de sorti de log c'est avec quel configuration des logs ? Ça ne ressemble pas à ce qu'il y a dans hobo et dans passerelle il n'y a rien non plus on dirait. Surtout je ne comprend pas d'où vient le .request après le passerelle.resource.parispocgru.wso2.

C'est le resultat de log d'un passerelle en standalone. Le .request vient du fait que le logger est appelé dans la méthode request de @LoggedRequest

Pour les logs de debug il faudrait peut-être ajouter un texte introductif, genre 'Request headers, ', 'Response headers, ', 'Response payload', et remplacer 'Payload' par 'Request payload, ' sinon on ne sait pas trop ce qu'on regarde.

En l'état les logs sur les headers sont très difficiles à lire, j'ajouterai un espace après le ':' et peut-être que je séparerai par des pipes juste pour voir (' | ').

#### #15 - 29 mars 2016 11:17 - Josué Kouka

Thomas Noël a écrit :

Josué Kouka a écrit :

Voici ce que ça donne

Y'a pas l'URL ?

Il faudrait donc l'URL, avec devant le verbe (GET/POST/...) ; mais aussi des indications "Request Headers:", "Request Payload:", "Response Headers:", "Response Payload:" sur les lignes qui suivent...

J'avais juste pas coller les méthodes

#### #16 - 29 mars 2016 11:26 - Josué Kouka

- Fichier 0001-add-logger-for-every-requests-call-10299.patch ajouté

#### #17 - 30 mars 2016 01:39 - Benjamin Dauvergne

request\_reponse ?

La méthode n'est pas passé dans le dico extra.

Concernant les logs précédents je ne vois pas quelle chaîne de formatage fait que la fonction est ajoutée au domaine des log, tu pourrais regarder ?

#### #18 - 30 mars 2016 09:45 - Josué Kouka

- Fichier 0001-add-logger-for-every-requests-call-10299.patch ajouté

Benjamin Dauvergne a écrit :

Concernant les logs précédents je ne vois pas quelle chaîne de formatage fait que la fonction est ajoutée au domaine des log, tu pourrais regarder ?

Oui j'ai trouvé pourquoi, j'avais le bloc suivant dans mes settings

```
'formatters': {  
    'syslog': {
```

```
        'format': '%(levelname)s %(name)s.%(funcName)s: %(message)s',
    },
},
```

#### #19 - 30 mars 2016 10:41 - Benjamin Dauvergne

Ack.

#### #20 - 30 mars 2016 10:41 - Benjamin Dauvergne

Euh non pas ack, je veux bien des tests utilisant pytest-capturelog, merci.

#### #21 - 01 avril 2016 17:45 - Josué Kouka

- Fichier 0001-add-logger-for-every-requests-call-10299.patch ajouté

#### #22 - 01 avril 2016 18:08 - Benjamin Dauvergne

Ton test dépend d'un appel à httpbin, regarde du coté de <https://pypi.python.org/pypi/httpmock/> pour éviter ça.

```
from httpmock import urlmatch, HTTPMock

@urlmatch(netloc=r'httpbin\.org$')
def httpbin_mock(url, request):
    return 'Feeling lucky, punk?'

with HTTPMock(httpbin_mock):
    response = requests.post(url, json={'name': 'josh'})
```

Il faut ajouter httpmock aux dépendance tox aussi.

#### #23 - 01 avril 2016 18:53 - Josué Kouka

- Fichier 0001-add-logger-for-every-requests-call-10299.patch ajouté

#### #24 - 01 avril 2016 21:00 - Benjamin Dauvergne

Ack.

#### #25 - 05 avril 2016 10:40 - Josué Kouka

J'attends la fin du POC Paris pour pouvoir pousser (Normalement demain dans l'après midi)

#### #26 - 06 avril 2016 18:55 - Benjamin Dauvergne

<https://jenkins.entrouvert.org/job/passerelle/1291/changes>

```
self = <django.db.backends.sqlite3.base.SQLiteCursorWrapper object at 0x35308a0>
query = 'SELECT "fake_family_fakefamily"."id", "fake_family_fakefamily"."title", "fake_family_fakefamily"."slug", "fake_family...vel", "fake_family_fakefamily"."jsondatabase" FROM "fake_family_fakefamily" ORDER BY "fake_family_fakefamily"."id" ASC'
params = ()

def execute(self, query, params=None):
    if params is None:
        return Database.Cursor.execute(self, query)
    query = self.convert_query(query)
> return Database.Cursor.execute(self, query, params)
E OperationalError: no such column: fake_family_fakefamily.log_level

.tox/coverage-django18/local/lib/python2.7/site-packages/django/db/backends/sqlite3/base.py:485: OperationalError
```

Tu aurais du le voir en lançant les tests il me semble.

#### #27 - 06 avril 2016 19:09 - Serghei Mihai (congé, retour 15/05)

- Fichier 0001-logging-add-missing-migrations-for-log\_level-attribu.patch ajouté

#### #28 - 06 avril 2016 19:33 - Serghei Mihai (congé, retour 15/05)

Patch poussé: build back to normal

commit cf64834eab3690aa74206ae3e4f25750dee9ab68

Author: Serghei Mihai <smihai@entrouvert.com>  
Date: Wed Apr 6 19:08:10 2016 +0200

logging: add missing migrations for 'log\_level' attribute (#10299)

**#29 - 06 avril 2016 19:33 - Serghei Mihai (congés, retour 15/05)**

- Statut changé de *En cours à Résolu* (à déployer)

**#30 - 06 avril 2016 19:52 - Josué Kouka**

Benjamin Dauvergne a écrit :

<https://jenkins.entrouvert.org/job/passerelle/1291/changes>

[...]

Tu aurais du le voir en lançant les tests il me semble.

Désolé, j'avais des apps non activé malheureusement. Je serai plus prudent les prochaines fois

**#31 - 06 avril 2016 19:53 - Josué Kouka**

Serghei Mihai a écrit :

Patch poussé: build back to normal

[...]

Merci

**#32 - 04 août 2018 12:33 - Benjamin Dauvergne**

- Statut changé de *Résolu* (à déployer) à *Fermé*

**Fichiers**

0001-add-logger-for-every-requests-call-10299.patch	1,5 ko	15 mars 2016	Josué Kouka
0001-add-logger-for-every-requests-call-10299.patch	16,3 ko	16 mars 2016	Josué Kouka
0001-add-logger-for-every-requests-call-10299.patch	28,2 ko	16 mars 2016	Josué Kouka
0001-add-logger-for-every-requests-call-10299.patch	30,3 ko	25 mars 2016	Josué Kouka
0001-add-logger-for-every-requests-call-10299.patch	30,3 ko	25 mars 2016	Josué Kouka
0001-add-logger-for-every-requests-call-10299.patch	30,4 ko	29 mars 2016	Josué Kouka
0001-add-logger-for-every-requests-call-10299.patch	30,4 ko	30 mars 2016	Josué Kouka
0001-add-logger-for-every-requests-call-10299.patch	32,5 ko	01 avril 2016	Josué Kouka
0001-add-logger-for-every-requests-call-10299.patch	33 ko	01 avril 2016	Josué Kouka
0001-logging-add-missing-migrations-for-log_level-attribu.patch	10,5 ko	06 avril 2016	Serghei Mihai (congés, retour 15/05)