Passerelle - Bug #31114

Les logs des requêtes HTTP n'ont aucune logique

06 mars 2019 09:56 - Benjamin Dauvergne

Statut:	Fermé	**Début:**	06 mars 2019
Priorité:	Normal	**Echéance:**	
Assigné à:	Benjamin Dauvergne	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	

Description

Pour le connecteur CD06 ils utilisent le code 404 pour me renvoyer des informations en JSON or le code de notre surcharge de requests fait ça:

```
        if response is not None:
            message = message + ' (=> %s)' % response.status_code
            extra['response_status'] = response.status_code
            if self.logger.level == 10:  # DEBUG
                extra['response_headers'] = dict(response.headers.items())
                # log body only if content type is allowed
                if content_type_match(response.headers.get('Content-Type')):
                    content = response.content[:settings.LOGGED_RESPONSES_MAX_SIZE]
                    extra['response_content'] = repr(content)
            if response.status_code // 100 == 3:
                log_function = self.logger.warning
            elif response.status_code // 100 >= 4: <-- ***ICI***
                log_function = self.logger.error
```

et donc on reçoit ce genre de mail d'erreur :

```
[passerelle-departement13.test.entrouvert.org] ERROR (EXTERNAL IP): GET https://services-smh-test.
departement13.fr/api/situation/dossier/137908 (=> 404)
```

On ne devrait pas décider pour chaque réponse 3xx ou 4xx que faire un warning ou une erreur unilatéralement (et je ne sais pas pourquoi on ne gère pas 5xx parce que requests ne génère pas d'exception pour 500 non plus sans un raise_for_status()).

Il n'y a que le code du connecteur qui puisse décider de comment interpréter cela.

Je serai pour virer cela et tout logger en info. À la rigueur je veux bien qu'on ajoute un expected_status_codes pour lister ce qui parait normal du reste et dans ce cas uniquement monter le log au niveau erreur (le niveau warning est inutile, de toute façon on ne regarde pas les logs sauf si on constate une erreur).

Révisions associées

Révision ab7432fa - 07 mars 2019 15:21 - Benjamin Dauvergne

utils: add copyright (#31114)

Révision a8eb92ef - 07 mars 2019 15:21 - Benjamin Dauvergne

utils: PEP8ness (#31114)

Révision 941d0800 - 07 mars 2019 18:12 - Benjamin Dauvergne

allow ressources to disable error logging in requests wrapper (#31114)

- modify ProxyLogger to get request from the "extra" parameter like Django, it incurs that we need to clean extra from non JSON-serializable values before using it in the ResourceLog model.

- modify to_json() so that request is passed to every log calls as before commit 5286592cf, now we will get error mails containing Django request details.

- resource can set "log_requests_errors = False" so that the Request wrapper never log at the error/warning level.

- small annoyance: requests errors are logged two times, at the INFO level and the ERROR level.

- connector MDPH13 is adapted to use all this.

Historique

#1 - 06 mars 2019 09:56 - Benjamin Dauvergne

- Description mis à jour

#2 - 06 mars 2019 09:57 - Benjamin Dauvergne

- Description mis à jour

#3 - 06 mars 2019 09:58 - Benjamin Dauvergne

- Sujet changé de Les logs des requêtes HTTP sont en trop à Les logs des requêtes HTTP n'ont aucune logique

- Description mis à jour

#4 - 06 mars 2019 10:07 - Frédéric Péters

> Il n'y a que le code du connecteur qui puisse décider de comment interpréter cela.

Il est quand même plutôt commun que 404 soit réellement une erreur; du coup je serais plutôt pour garder ce comportement mais permettre à un connecteur de désactiver ça.

#5 - 06 mars 2019 10:21 - Benjamin Dauvergne

Frédéric Péters a écrit :

> > Il n'y a que le code du connecteur qui puisse décider de comment interpréter cela.
>
> Il est quand même plutôt commun que 404 soit réellement une erreur; du coup je serais plutôt pour garder ce comportement mais permettre à un connecteur de désactiver ça.

Ça renverse complètement le travail plutôt que de faire on va devoir se défendre de ce que le framework pourrait faire et qu'on ne veut pas, c'est vraiment fatiguant, pour moi on ne doit logger une erreur/warning que si une exception arrive jusqu'à jsonresponse, si certains codent mal en mettant des try/except généraux vraiment ça me fatigue.

Je vais rajouter un paramètre raise_for_status=True qui forcera un appel à reponse.raise_for_status() et faire le tour des appelants pour voir qui check manuellement le status_code pour y mettre raise_for_status=False.

#6 - 06 mars 2019 10:26 - Benjamin Dauvergne

Benjamin Dauvergne a écrit :

> Je vais rajouter un paramètre raise_for_status=True qui forcera un appel à reponse.raise_for_status() et faire le tour des appelants pour voir qui check manuellement le status_code pour y mettre raise_for_status=False.

Et dans jsonresponse si c'est une RequestException, je loggerai une erreur plutôt qu'un warning (là aussi pas logique, on log une erreur au niveau requests puis un warning au niveau jsonreponse :/)

```
# passerelle.utils.jsonresponse
163             elif isinstance(e, RequestException):
164                 logger.warning("RequestException occurred while processing request", extra=extras)
```

#7 - 06 mars 2019 10:51 - Frédéric Péters

> Ça renverse complètement le travail plutôt que de faire on va devoir se défendre de ce que le framework pourrait faire et qu'on ne veut pas

C'était ton option proposée "À la rigueur je veux bien qu'on ajoute un expected_status_codes".

De mon côté je suis farouchement pour que le comportement par défaut soit pertinent et dans la situation d'une erreur 404, je trouve pertinent que ça soit loggué.

(côté combo, pareil, on loggue par défaut mais on a un log_errors qui peut être mis à False quand l'appelant ne souhaite pas le comportement)

#8 - 07 mars 2019 00:30 - Benjamin Dauvergne

- Fichier 0004-utils-raise-on-HTTP-error-and-unwanted-redirects-311.patch ajouté

- Fichier 0001-utils-add-copyright-31114.patch ajouté

- Fichier 0002-utils-PEP8ness-31114.patch ajouté

- Fichier 0003-utils-only-log-requests-response-as-info-31114.patch ajouté

- Fichier 0005-let-requests-handle-HTTP-errors-31114.patch ajouté

- Patch proposed changé de Non à Oui

C'est un peu long mais les modifications aux connecteurs sont minimes, je referai une passe pour virer plus de mocked_get/post/etc..

#9 - 07 mars 2019 10:25 - Frédéric Péters

On est d'accord que ce n'est pas du tout "À la rigueur je veux bien qu'on ajoute un expected_status_codes", que ça demande de modifier tous les connecteurs, que tu gères ceux dans passerelle mais que ça réserve des surprises à ceux qui sont en-dehors, etc. ?

#10 - 07 mars 2019 12:13 - Frédéric Péters

J'ai regardé de mon côté et ça casse les connecteurs développés par GL et le connecteur liege-rn, ça casse aussi l'idée passée dans la documentation comme quoi self.requests est uniquement un wrapper léger au-dessus de requests; bref, pas fan du tout de ce raise_for_status() comme nouveau comportement par défaut.

#11 - 07 mars 2019 12:30 - Benjamin Dauvergne

Frédéric Péters a écrit :

> J'ai regardé de mon côté et ça casse les connecteurs développés par GL et le connecteur liege-rn, ça casse aussi l'idée passée dans la documentation comme quoi self.requests est uniquement un wrapper léger au-dessus de requests; bref, pas fan du tout de ce raise_for_status() comme nouveau comportement par défaut.

Je veux bien mettre raise_for_status/redirect=False par défaut mais concernant les logs au niveau warning/error pour tout redirect/http_error ça veut qu'à liege et GL ils utilisent des redirect/http_error ils se mangent plein de mail d'erreur de warning dans les logs actuellement ?

#12 - 07 mars 2019 14:10 - Frédéric Péters

> Je veux bien mettre raise_for_status/redirect=False par défaut

(ok)

> mais concernant les logs au niveau warning/error pour tout redirect/http_error ça veut qu'à liege et GL ils utilisent des redirect/http_error ils se mangent plein de mail d'erreur de warning dans les logs actuellement ?

Quand il y a une erreur il y a un mail, l'idée étant que quand tout va bien il n'y a pas de mail.

Pour revenir à l'origine de ce ticket, on est sur un connecteur où un code d'erreur est une réponse légitime, c'est une exception et c'est à ce titre que je ne trouve pas opportun de modifier le comportement usuel.

#13 - 07 mars 2019 15:20 - Benjamin Dauvergne

Frédéric Péters a écrit :

> > Je veux bien mettre raise_for_status/redirect=False par défaut
>
> (ok)
>
> > mais concernant les logs au niveau warning/error pour tout redirect/http_error ça veut qu'à liege et GL ils utilisent des redirect/http_error ils se mangent plein de mail d'erreur de warning dans les logs actuellement ?
>
> Quand il y a une erreur il y a un mail, l'idée étant que quand tout va bien il n'y a pas de mail.

Je ne suis pas vraiment d'accord je récapitule ce qui se passe actuellement :

- réponse 3xx : log warning, pas de mail dans Request, après traitement par le connecteur lui même ou pas, en tout cas ça peut finir par planter dans le connecteur lui même si vraiment ce n'est pas attendu et ça renvoie n'importe où, donc mail mais sans forcément la raison d'origine, il faut croiser un peu
- réponse 4xx : log erreur dans Request, donc mail, éventuellement erreur plus loin encore dans le connecteur, re-mail
- autre RequestException : log erreur dans Request.log_http_request, puis log warning dans json_response, pas de mail au niveau du traitement d'erreur dans jsonresponse, sauf si c'est traité avant par du code dans le connecteur

Je vais ajouter un error_log=True pour pouvoir désactiver ça.

> Pour revenir à l'origine de ce ticket, on est sur un connecteur où un code d'erreur est une réponse légitime, c'est une exception et c'est à ce titre que je ne trouve pas opportun de modifier le comportement usuel.

J'ai lu beaucoup de connecteurs hier soir, ça m'embête que du code gérant les erreurs soit l'exception :/ Donc mon nouveau plan :

- lors de l'instanciation de Request récupérer sur le connecteur deux paramètres raise_for_status et error_log par défaut à False et True
- j'oublie raise_for_redirect qui en fait ne m'intéressais pas je l'avais juste mis parce que tu trouvais les redirections comme étant un problème
- voir au niveau de to_json() si je peux détecter error_log à False et dans ce cas logger les RequestException proprement en erreur (faut bien les logger quelque part si on oublie des les gérer dans le connecteur)
- sur mon connecteur MDPH13 poser les valeurs True et False et laisser tout les autres se débrouiller avec tout ça
- je vais ouvrir un autre ticket avec mon code fake_http que je trouve sympatoche

#14 - 07 mars 2019 18:27 - Benjamin Dauvergne

- Assigné à mis à Benjamin Dauvergne

#15 - 07 mars 2019 18:27 - Benjamin Dauvergne

- Fichier 0001-utils-add-copyright-31114.patch ajouté

- Fichier 0003-allow-ressources-to-disable-error-logging-in-request.patch ajouté

- Fichier 0002-utils-PEP8ness-31114.patch ajouté

- Statut changé de Nouveau à Solution proposée

Voilà, finalement j'ai modifié au minimum to_json(), je lui ai juste appris à logger des erreurs Requests comme les autres, aussi j'ai corrigé un problème où on ne passait plus la requête Django dans nos logs et donc on ne bénéficiait pas de toute la mécanique dans le rendu des traces par mail.

Au passage j'ai identifié un problème de l'impossibilité de logger dans un block atomic() si il y a une exception, il faudrait déférer la sauvegarde des ResourceLog via transaction.on_commit().

#16 - 08 mars 2019 01:16 - Benjamin Dauvergne

- Fichier 0001-utils-add-copyright-31114.patch ajouté

- Fichier 0004-add-bool-to-json-serializable-types-to-be-rebased.patch ajouté

- Fichier 0003-allow-ressources-to-disable-error-logging-in-request.patch ajouté

- Fichier 0002-utils-PEP8ness-31114.patch ajouté

ajouté bool aux types JSON sérialisables.

#17 - 11 mars 2019 10:43 - Frédéric Péters

- Statut changé de Solution proposée à Solution validée

> - small annoyance: requests errors are logged two times, at the INFO level and the ERROR level.

Petite question sur les implications ici (deux mails ? deux lignes dans la db ?).

À part ça, ok, et à très vite passer avec tag et mise à jour des serveurs etc.

#18 - 11 mars 2019 10:47 - Benjamin Dauvergne

Frédéric Péters a écrit :

- small annoyance: requests errors are logged two times, at the INFO level and the ERROR level.

Petite question sur les implications ici (deux mails ? deux lignes dans la db ?).

Un seul mail mais deux lignes en DB et dans syslog,etc..

À part ça, ok, et à très vite passer avec tag et mise à jour des serveurs etc.

D'ac, merci.

#19 - 11 mars 2019 15:03 - Frédéric Péters

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

Et,

```
commit 9bcc8d7a067188363af3dab2b6fac6d5ca9ae3c8
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date:   Fri Mar 8 01:16:08 2019 +0100

    add bool to json serializable types (to be rebased)

commit 941d0800949ccfae08bd63344e6a74c5d751584a
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date:   Wed Mar 6 10:10:35 2019 +0100

    allow ressources to disable error logging in requests wrapper (#31114)

    - modify ProxyLogger to get request from the "extra" parameter like Django, it
      incurs that we need to clean extra from non JSON-serializable values
      before using it in the ResourceLog model.

    - modify to_json() so that request is passed to every log calls as
      before commit 5286592cf, now we will get error mails containing Django
      request details.

    - resource can set "log_requests_errors = False" so that the Request
      wrapper never log at the error/warning level.

    - small annoyance: requests errors are logged two times, at the INFO
      level and the ERROR level.

    - connector MDPH13 is adapted to use all this.

commit a8eb92ef6fb248c62c23552b7cefa462a97d6989
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>
Date:   Wed Mar 6 10:02:57 2019 +0100

    utils: PEP8ness (#31114)
```

#20 - 11 mars 2019 15:16 - Frédéric Péters

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

#21 - 12 avril 2019 10:27 - Benjamin Dauvergne

- Statut changé de Solution déployée à Fermé

Fichiers

0004-utils-raise-on-HTTP-error-and-unwanted-redirects-311.patch	1,9 ko	06 mars 2019	Benjamin Dauvergne
0001-utils-add-copyright-31114.patch	1,26 ko	06 mars 2019	Benjamin Dauvergne
0002-utils-PEP8ness-31114.patch	3,99 ko	06 mars 2019	Benjamin Dauvergne
0003-utils-only-log-requests-response-as-info-31114.patch	1,83 ko	06 mars 2019	Benjamin Dauvergne
0005-let-requests-handle-HTTP-errors-31114.patch	158 ko	06 mars 2019	Benjamin Dauvergne
0001-utils-add-copyright-31114.patch	1,26 ko	07 mars 2019	Benjamin Dauvergne
0003-allow-ressources-to-disable-error-logging-in-request.patch	16,7 ko	07 mars 2019	Benjamin Dauvergne
0002-utils-PEP8ness-31114.patch	3,99 ko	07 mars 2019	Benjamin Dauvergne

0001-utils-add-copyright-31114.patch	1,26 ko	08 mars 2019	Benjamin Dauvergne
0004-add-bool-to-json-serializable-types-to-be-rebased.patch	984 octets	08 mars 2019	Benjamin Dauvergne
0003-allow-ressources-to-disable-error-logging-in-request.patch	16,7 ko	08 mars 2019	Benjamin Dauvergne
0002-utils-PEP8ness-31114.patch	3,99 ko	08 mars 2019	Benjamin Dauvergne