

Passerelle - Development #55687

emails sur les erreurs, le paramétrage d'adresses alternatives ne semble plus ok

19 juillet 2021 22:00 - Frédéric Péters

Statut:	Rejeté	Début:	19 juillet 2021
Priorité:	Normal	Echéance:	
Assigné à:	Valentin Deniaud	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
Fatigué par certaines traces qui de toute façon ne donnent pas d'infos (#55516) j'ai voulu reparamétrer un connecteur pour qu'elles soient plutôt envoyée dans un /dev/null noreply@entrouvert.com et ça ne semble pas fonctionner, on continue à recevoir les emails.			
Sans doute aussi quelque chose cassé par le passage à django 2.2, vraisemblablement le <code>override_settings</code> qui de base était prévu pour les tests ne marche plus ici.			
<pre>logging_parameters = self.connector.logging_parameters if logging_parameters.trace_emails: admins = [('', x) for x in logging_parameters.trace_emails.splitlines()] with override_settings(ADMINS=admins): getattr(self._logger, levelname.lower())(message, *args, **kwargs)</pre>			
Demandes liées:			
Lié à Hobo - Development #55824: Permettre un AdminEmailHandler custom		Rejeté	26 juillet 2021

Historique

#1 - 20 juillet 2021 10:07 - Paul Marillonnet

- Description mis à jour

#2 - 20 juillet 2021 14:07 - Valentin Deniaud

- Assigné à mis à Valentin Deniaud

#3 - 20 juillet 2021 15:26 - Valentin Deniaud

C'est un cas qui est testé, `test_trace_emails`, et je ne reproduis pas en local. Quelle est l'URL du connecteur où ça n'a pas marché ?

#4 - 20 juillet 2021 15:39 - Frédéric Péters

Tout ce qui est listé dans #55588#note-3 j'ai mis noreply@entrouvert.com dans "Adresses où recevoir les erreurs et traces critiques" et ça continue à faire des mails toutes les 5 minutes adressés à admin+....

#5 - 20 juillet 2021 15:41 - Frédéric Péters

Aussi, <https://passerelle.toodego.com/grandlyon-iodas/iodas/>, qui a noreply@entrouvert.com depuis des lustres et qui s'est mis à les envoyer à admin@ au passage en django 2.2. (à peu près 4000 mails inutiles reçus pour ce cas).

#6 - 20 juillet 2021 16:17 - Valentin Deniaud

OK, je ne peux pas pousser plus loin l'investigation, les git diff dans les sources de Django ne donnent pas grand chose.

Du coup, jeter cette solution qui sentait de toute façon le bricolage, et :

- Passer `send_emails_to=xxx` dans extra au moment de logger
- Aller modifier `hobo.multitenant.log.AdminEmailHandler` et lui faire utiliser ce paramètre

C'est l'approche qui me paraît recommandée dans la doc (https://docs.djangoproject.com/en/3.2/topics/logging/#django.utils.log.AdminEmailHandler.send_mail).

#7 - 20 juillet 2021 16:40 - Frédéric Péters

Moi ça me va. (en soit ça fait un paramétrage de Passerelle qui dépend d'Hobo, mais acceptons ça).

#8 - 21 juillet 2021 12:51 - Valentin Deniaud

- Fichier 0001-misc-fix-override-of-trace-emails-ipients-for-Dja.patch ajouté
- Tracker changé de Bug à Development
- Statut changé de Nouveau à Solution proposée
- Patch proposed changé de Non à Oui

Voilà, j'ai fait en sorte qu'il n'y ait pas de dépendance à hobo (c'était obligatoire pour pouvoir tester de toute manière), ensuite il faudra dupliquer le code du handler dans hobo pour que ça marche sur nos installations.

#9 - 21 juillet 2021 13:00 - Valentin Deniaud

- Fichier 0001-misc-fix-override-of-trace-emails-ipients-for-Dja.patch supprimé

#10 - 21 juillet 2021 13:04 - Valentin Deniaud

- Fichier 0001-misc-fix-override-of-trace-emails-ipients-for-Dja.patch ajouté

Le bon patch.

#11 - 22 juillet 2021 09:21 - Frédéric Péters

En fait pour le cas #55588#note-3 ça n'ira pas c'est un `logger.error()` direct de `cmislib`. (je serais là à modifier la config pour virer ces infos, peut-être via `clamp_to_warning` comme on fait pour d'autres).

Ou alors on pose comme objectif que tous les appels à logging fait par des modules appelés par un connecteur doivent suivre le paramétrage du connecteur (ce qui s'entend mais me semble compliqué).

Pour le cas [#55687#note-5](#) je n'ai par contre pas d'explication.

Voilà si on en reste au logging de passerelle, c'est plus clean de faire comme tu proposes que de garder `override_settings`. (il y aurait à en retirer `l'import`).

#12 - 22 juillet 2021 11:03 - Valentin Deniaud

- Fichier 0001-misc-fix-override-of-trace-emails-ipients-for-Dja.patch ajouté

Frédéric Péters a écrit :

Ou alors on pose comme objectif que tous les appels à logging fait par des modules appelés par un connecteur doivent suivre le paramétrage du connecteur (ce qui s'entend mais me semble compliqué).

Nouvelle version qui va dans ce sens, ça devrait marcher pour le connecteur CMIS en tout cas.

Et dans hobo, plutôt un `getattr(settings, 'ADMIN_EMAIL_HANDLER_CLASS')` et hériter de ça plutôt que dupliquer le code.

#13 - 26 juillet 2021 10:47 - Valentin Deniaud

- Lié à Development #55824: Permettre un `AdminEmailHandler` custom ajouté

#14 - 27 juillet 2021 18:02 - Valentin Deniaud

- Fichier 0001-misc-fix-override-of-trace-emails-ipients-for-Dja.patch ajouté

Rebasé, ajout du paramétrage attendu par hobo et correction d'un elif mal fichu.

#15 - 27 juillet 2021 18:17 - Benjamin Dauvergne

- Statut changé de Solution proposée à En cours

Je vois un `addFilter` mais pas de `removeFilter`, c'est nettoyé quand tout ça ? Pour ce genre de modifications un peu globale ce serait mieux avec un `contextmanager`, genre `ProxyLogger.monkeypatch_root_logger()`, on doit pouvoir placer un `with monkey...()` : à quelques points stratégiques genre `GenericEndpointView.dispatch()`, `Job.run()` et dans la commande de cron autour de `getattr(connector, 'frequency')()`.

#16 - 28 juillet 2021 10:51 - Valentin Deniaud

- Fichier 0001-misc-fix-override-of-trace-emails-ipients-for-Dja.patch ajouté
- Statut changé de En cours à Solution proposée

Benjamin Dauvergne a écrit :

Je vois un addFilter mais pas de removeFilter, c'est nettoyé quand tout ça ?

En effet, gros oubli... Revoici avec un contextmanager.

#17 - 13 août 2021 08:31 - Frédéric Péters

- Statut changé de Solution proposée à Solution validée

#18 - 13 août 2021 10:26 - Benjamin Dauvergne

À mon avis ça ne vas pas régler le souci de mail inutiles sans traceback car ils viennent de plus haut, [#56139](#).

#19 - 13 août 2021 10:47 - Frédéric Péters

Peut-être que ça a un rapport non explicité avec l'absence de trace mais l'objet ici est de respecter les destinataires paramétrés (qu'il y ait ou pas de trace, à partir du moment où ça va vers le bon endroit, était pour moi un autre problème).

#20 - 13 août 2021 11:01 - Benjamin Dauvergne

Ce que je veux dire c'est tu as cru qu'il y a un problème avec la customisation du mail mais il n'y en a pas, `override_settings()[1]` marche certainement correctement, c'est juste que c'est loggé ailleurs, le code présent ici n'y changera rien vu que le filtre ne pourra pas s'appliquer à `django.utils.log.log_response` fait au niveau de Django bien après la fin du traitement par le endpoint.

--

Par ailleurs si ce code finit par être poussé, ce qui m'embête c'est que le `mail_admins` configuré ici ne s'appliquera pas puisque la configuration du LOGGING vient de `hobo/debian/debian_config_common.py`; il faudrait donc en plus un patch spécifique pour passerelle dans hobo.

¹ et si vraiment ça ne marchait plus, un simple

```
@contextmanager
def with_admins(new_admins):
    old_admins = settings.ADMINS
    try:
        settings.ADMINS = new_admins
        yield
    finally:
        settings.ADMINS = old_admins
```

suffirait, mais je pense que `override_settings` fait déjà plus ou moins un truc comme ça au dessus des `TenantSettings` et que ça marche encore.

#21 - 13 août 2021 11:22 - Frédéric Péters

Et quand tu écris "c'est loggé ailleurs", pour me clarifier/vérifier c'est "en plus de ce qui était en place et qui fonctionne le soucis est désormais (2.2) qu'il y a log en doublon et que c'est ce second log doublon qui est distribué à l'adresse générale".

#22 - 13 août 2021 12:21 - Benjamin Dauvergne

Frédéric Péters a écrit :

Et quand tu écris "c'est loggé ailleurs", pour me clarifier/vérifier c'est "en plus de ce qui était en place et qui fonctionne le soucis est désormais (2.2) qu'il y a log en doublon et que c'est ce second log doublon qui est distribué à l'adresse générale".

Oui, dès qu'il y a une 500 due à une exception non "gérée"[1] il y a normalement un doublon, un mail envoyé à la bonne adresse (via un log explicite dans `to_json`) et un mail à la mauvaise sans trace et venant du `logger django.request` dans Django; s'il y a deux mails à la mauvaise adresse alors là tu auras raison `override_settings` ne marche pas, mais uniquement si ça concerne une connecteur avec des endpoints modernes (voir prochain paragraphe).

Par ailleurs ça ne marche pas non plus avec les connecteurs qui utilisent `to_json` directement parce qu'il ne passe pas `connector.logger` à `to_json` et le `override_settings` est fait dans `ProxyLogger._log`; donc pour ces code legacy il faudrait appliquer un décorateur automatiquement à toutes les vues qui cherche les `kwargs` `resource/slug` et tenter de s'en sortir pour trouver le bon connecteur (ça devrait couvrir 90% du code qui utilise directement `utils.to_json()` de ce que je vois) et modifier `settings.ADMINS`.

Pour le code actuel je trouverai aussi plus claire de faire ce `override_settings` au niveau de `GenericEndpointView` plutôt qu'au fin fond de `ProxyLogger`, ça permettra que ça marche aussi pour un `logging.error/exception` involontaire au milieu du code.

On a aussi le cas d'exceptions `APIError` avec `log_error = False` / `http_status = 500` mais je n'en trouve qu'un exemplaire dans `caluire_axel` (et à mon avis c'est une erreur, on devrait renvoyer une 200 comme d'habitude pour une erreur du logiciel métier) :

```
passerelle/contrib/caluire_axel/models.py-1238-         except (KeyError, ValueError):
passerelle/contrib/caluire_axel/models.py-1239-             raise APIError(
passerelle/contrib/caluire_axel/models.py-1240-                 'Wrong pay-invoice response',
```

```

passerelle/contrib/caluire_axel/models.py-1241-      err_code='pay-invoice-code-response-error',
passerelle/contrib/caluire_axel/models.py-1242-      data={
passerelle/contrib/caluire_axel/models.py-1243-          'regie_id': regie_id,
passerelle/contrib/caluire_axel/models.py-1244-          'family_id': family_id,
passerelle/contrib/caluire_axel/models.py-1245-          'invoice': invoice,
passerelle/contrib/caluire_axel/models.py-1246-          'post_data': post_data,
passerelle/contrib/caluire_axel/models.py-1247-          'kwargs': kwargs,
passerelle/contrib/caluire_axel/models.py-1248-          'result': result.json_response,
passerelle/contrib/caluire_axel/models.py-1249-      },
passerelle/contrib/caluire_axel/models.py:1250-      http_status=500,
passerelle/contrib/caluire_axel/models.py-1251-  )

```

¹ Non géré à lire le code c'est tout sauf `APIError` (sauf si on override `http_status` avec une valeur 5xx), `PermissionDenied`, `HTTP404` et `ObjectDoesNotExist`; `RequestException` génère une 500 en plus du log en erreur sauf si c'est géré plus bas et converti en `APIError` ou bien si le connecteur a `log_requests_error` à `False` ou là on aura la 500, un mail d'erreur de Django sans trace mais juste un log info au niveau du connecteur (et là pas de mail pas de trace). Je reconnais que c'est devenu bien imbitable avec toutes les couches.

#23 - 13 août 2021 12:38 - Frédéric Péters

En pratique je lis mais ça ne met pas en mesure de trouver ce qui quand quoi amène à faire un mail qui dit :

```

Report at /grandlyon-iodas/iodas/getProcedures
Internal Server Error: /grandlyon-iodas/iodas/getProcedures

```

ni donc ce qui pourrait être fait pour qu'il n'y ait pas ce mail.

(et je ne sais pas ce qui dans django 2.2 a changé qui fait que logging est encore plus nul aujourd'hui qu'avant).

(et cf [#56009#note-7](#) ces messages doublons apparaissent même dans w.c.s. qui fait `LOGGING = {}` qui devrait dire **fucking arrête**).

#24 - 13 août 2021 13:51 - Benjamin Dauvergne

Frédéric Péters a écrit :

En pratique je lis mais ça ne met pas en mesure de trouver ce qui quand quoi amène à faire un mail qui dit :

[...]

Je vois clairement deux erreurs loggés dans syslog pour le même problème :

```

Aug 13 03:09:57 eddv-gdlyon-gnm-front-1 uwsgi[23783]: passerelle ERROR passerelle.toodego.com 10.128.2.162 - r
:7F38C5D58940 Error occurred while processing request#012Traceback (most recent call last):#012 File "/usr/li
b/python3/dist-packages/passerelle/utils/jsonresponse.py", line 128, in api#012 resp = f(*args, **kwargs)#0
12 File "/usr/lib/python3/dist-packages/passerelle/views.py", line 527, in perform#012 result = self.endpo
int(request, **params)#012 File "/usr/lib/python3/dist-packages/grandlyon_iodas/models.py", line 91, in getPr
ocedures#012 resp = self.get_client().service.ODA_getProceduresSIH(dpap, typepro, datenais, nom, pren)#012
File "/usr/lib/python3/dist-packages/suds/client.py", line 566, in __call__#012 return client.invoke(args,
kwargs)#012 File "/usr/lib/python3/dist-packages/suds/client.py", line 705, in invoke#012 result = self.s
end(soapenv)#012 File "/usr/lib/python3/dist-packages/suds/client.py", line 753, in send#012 return self.p
rocess_reply(reply.message, None, None)#012 File "/usr/lib/python3/dist-packages/suds/client.py", line 814, i
n process_reply#012 raise WebFault(fault, replyroot)#012suds.WebFault: Server raised fault: 'Error in sessi
on'
Aug 13 03:09:57 eddv-gdlyon-gnm-front-1 uwsgi[23783]: passerelle ERROR passerelle.toodego.com 10.128.2.162 - r
:7F38C5D58940 Internal Server Error: /grandlyon-iodas/iodas/getProcedures

```

La première c'est bien le log via `ProxyLogger` (t'as du recevoir un mail avec une trace), la deuxième c'est celle sur `django.logger`, t'as du en recevoir un aussi mais sans trace. Si tu ne reçois aucun mail pour la première c'est qu'il y a un souci entre `ProxyLogger` / `AdminEmailHandler` qu'il faut examiner dans le cadre de ce connecteur (parce que visiblement ça marche ailleurs).

ni donc ce qui pourrait être fait pour qu'il n'y ait pas ce mail.

Comme je ne sais pas si tu as reçu un ou deux mails, je ne saurai dire, mais pour celui venant de `django.logger`, le ticket [#56139](#) le supprimera.

(et je ne sais pas ce qui dans django 2.2 a changé qui fait que logging est encore plus nul aujourd'hui qu'avant).

C'est l'introduction de `log_response` à la place d'appels explicite à logging, `log_response` déduit le niveau du code HTTP, avant c'était le chemin du code (et notamment le fait d'intercepter ou pas une exception) qui déterminait le niveau, en 1.11 pas d'exception alors niveau info.

(et cf [#56009#note-7](#) ces messages doublons apparaissent même dans w.c.s. qui fait `LOGGING = {}` qui devrait dire **fucking arrête**).

Oui parce que ça n'a jamais été ça qu'il faut faire, c'est `LOGGING_CONFIG = None`

```
def configure_logging(logging_config, logging_settings):
    if logging_config:
        # First find the logging configuration function ...
        logging_config_func = import_string(logging_config)

        logging.config.dictConfig(DEFAULT_LOGGING)

    # ... then invoke it with the logging settings
    if logging_settings:
        logging_config_func(logging_settings)
```

je cite le code mais c'est écrit dans la doc¹ aussi.

¹<https://docs.djangoproject.com/fr/3.2/topics/logging/#configuring-logging>

#25 - 13 août 2021 14:16 - Frédéric Péters

je ne saurai dire, mais pour celui venant de django.logger, le ticket [#56139](#) le supprimera.

Ok très bien.

#26 - 16 août 2021 18:32 - Benjamin Dauvergne

- Statut changé de Solution validée à Rejeté

Je rejette, je ne pense pas qu'il y ait un bug ici (ou alors il faut reformuler le problème).

Fichiers

0001-misc-fix-override-of-trace-emails-recipients-for-Dja.patch	4,07 ko	21 juillet 2021	Valentin Deniaud
0001-misc-fix-override-of-trace-emails-recipients-for-Dja.patch	5,7 ko	22 juillet 2021	Valentin Deniaud
0001-misc-fix-override-of-trace-emails-recipients-for-Dja.patch	6,13 ko	27 juillet 2021	Valentin Deniaud
0001-misc-fix-override-of-trace-emails-recipients-for-Dja.patch	8,53 ko	28 juillet 2021	Valentin Deniaud