

Passerelle - Development #29322

ne pas utiliser override_settings

22 décembre 2018 13:38 - Frédéric Péters

Statut:	Nouveau	Début:	22 décembre 2018
Priorité:	Bas	Echéance:	
Assigné à:		% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposed:	Non		

Description

Parce qu'incompatible avec l'utilisation potentielle de threads par des connecteurs (cf [#29320](#)).

Mon idée serait,

```
--- a/passerelle/base/models.py
+++ b/passerelle/base/models.py
@@ -579,12 +579,11 @@ class ProxyLogger(object):

        ResourceLog.objects.create(**attr)

-        admins = settings.ADMINS
        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)
+                kwargs.setdefault('extra', {})
+                kwargs['extra']['admins'] = [(' ', x) for x in logging_parameters.trace_emails.splitlines()]
+            getattr(self._logger, levelname.lower())(message, *args, **kwargs)

        def exception(self, message, *args, **kwargs):
            kwargs['exc_info'] = 1
```

Puis côté handler, quelque chos du genre :

```
class TraceHandler(django.utils.log.AdminEmailHandler):
    def emit(self, record):
        self.admins = getattr(record, 'admins', settings.ADMINS)
        return super(TraceHandler, self).emit(record)

    def send_mail(self, subject, message, *args, **kwargs):
        if not self.admins:
            return
        mail = EmailMultiAlternatives(
            '%s%s' % (settings.EMAIL_SUBJECT_PREFIX, subject), message,
            settings.SERVER_EMAIL, [a[1] for a in self.admins],
            connection=self.connection(),
        )
        if kwargs.get('html_message'):
            mail.attach_alternative(kwargs['html_message'], 'text/html')
        mail.send(fail_silently=kwargs.get('fail_silently', False))
```

qui serait à déclarer en remplacement de l'handler "mail_admins" sauf que là ça commence à toucher de trop près à logging et du coup ça devient pour moi incompréhensible (soit je me retrouver à devoir répéter toute la configuration de logging par défaut de django et ça foire quand même, soit déclarer le handler sous un nouveau nom mais du coup l'admin reçoit quand même toujours la trace).