

Chrono - Development #58014

Rationaliser les APIError

20 octobre 2021 16:06 - Valentin Deniaud

Statut:	Fermé	Début:	20 octobre 2021
Priorité:	Normal	Echéance:	
Assigné à:	Valentin Deniaud	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		

Description

Quand je fais un patch dans chrono j'ai l'impression que mon code devient plus moche dès que je dois lever une exception APIError.

C'est dû au fait que l'instanciation de l'exception prend 2 à 4 arguments, dont l'argument `err_class` qui est une chaîne dupliquée, et l'argument `http_status` qui est bien verbeux en prenant des valeurs genre `status.HTTP_400_BAD_REQUEST`.

J'ai l'impression que ce serait facilement améliorable :

- Utiliser `gettext_noop()` pour ne plus dupliquer la chaîne décrivant l'erreur
- Avoir une exception `APIError400` qui revienne à passer `http_status=status.HTTP_400_BAD_REQUEST`

Et dans la plupart des cas on reviendrait à une exception qui ne prend en argument que le message d'erreur, ce qui me paraît plus sympatoche. Concrètement

```
raise APIError(
    _('invalid datetime for event %s') % event_identifieur,
    err_class='invalid datetime for event %s' % event_identifieur,
    http_status=status.HTTP_400_BAD_REQUEST,
)
```

devient

```
raise APIError400(N_('invalid datetime for event %s') % event_identifieur)
```

Idée plutôt à prendre ou à laisser ?

PS (de taille) : il faudrait en profiter pour définir quelles erreurs doivent être HTTP 200 et quelles autres HTTP 400, car j'ai l'impression que c'est fait au petit bonheur la chance. Dans le même ordre d'idée, statuer à quoi sert le paramètre `err_class` : si c'est à être utilisé dans les conditions des workflows ou autre, alors il ne faudrait pas que la chaîne puisse varier ? (comme dans l'exemple ci-dessus avec l'interpolation)

Révisions associées

Révision [1c8c5f44](#) - 18 novembre 2021 10:16 - Valentin Deniaud

api: make APIError less verbose (#58014)

Historique

#1 - 20 octobre 2021 18:00 - Nicolas Roche

Idée plutôt à prendre ou à laisser ?

(oui, moi ça me parle tout ça)

#2 - 04 novembre 2021 10:40 - Valentin Deniaud

- Assigné à mis à Valentin Deniaud

#3 - 04 novembre 2021 12:03 - Valentin Deniaud

- Fichier 0001-api-make-APIError-less-verbose-58014.patch ajouté
- Statut changé de Nouveau à Solution proposée
- Patch proposed changé de Non à Oui

Hop,

3 files changed, 101 insertions(+), 330 deletions(-)

Valentin Deniaud a écrit :

il faudrait en profiter pour définir quelles erreurs doivent être HTTP 200 et quelles autres HTTP 400

Vu avec Thomas :

- 400 veut dire que la requête est mauvaise (paramètre manquant...)
- 200 pour le reste (il y a eu une erreur mais c'est pas de ta faute, plus de place etc)

Et en fait j'étais pessimiste, cette règle semble déjà bien appliquée.

Normalement il ne faudrait jamais autre chose, or il y a plusieurs endroits où on lève des 404, mais je ne vais pas toucher à ça ici.

#4 - 04 novembre 2021 12:25 - Frédéric Péters

La description ne parle pas des traductions, je n'ai pas l'intention expliquant le () -> N() mais ça ne va pas marcher, gettext_noop (N_) ça sert juste à marquer la chaîne dispo à la traduction (le code est vraiment juste def gettext_noop(msg): return msg), ça fait que

```
raise APIErrorBadRequest(N_('bad datetime format: %s') % datetime_str)
```

va passer 'bad datetime format: quelque chose' à APIErrorBadRequest, et là-dedans le to_response() va faire _('bad datetime format: quelque chose') et il n'y aura pas cette chaîne dans le catalogue gettext.

#5 - 04 novembre 2021 14:47 - Valentin Deniaud

- Fichier 0001-api-make-APIError-less-verbose-58014.patch ajouté

Frédéric Péters a écrit :

je n'ai pas l'intention expliquant le () -> N()

Il s'agit de ne pas avoir de chaîne systématiquement dupliquée et donc de traduire à posteriori, la chaîne traduite devant se retrouver dans err_desc et la non traduite dans err_class.

Si il existe une méthode pour appliquer d'abord _() et ensuite pouvoir quand même accéder à la chaîne non traduite, ça irait tout à fait mais je n'en connais pas.

mais ça ne va pas marcher, le to_response() va faire _('bad datetime format: quelque chose') et il n'y aura pas cette chaîne dans le catalogue gettext.

En effet, gros trou dans la raquette. Du coup proposition d'une forme à la logging, raise APIError(N_('bad datetime format: %s'), 'quelque chose') et charge à APIError d'appliquer l'interpolation.

#6 - 15 novembre 2021 11:31 - Valentin Deniaud

- Priorité changé de Bas à Normal

Valentin Deniaud a écrit :

proposition d'une forme à la logging, raise APIError(N_('bad datetime format: %s'), 'quelque chose') et charge à APIError d'appliquer l'interpolation.

Je peux aussi écrire un checker pylint pour s'assurer que la forme soit respectée (pas sûr que ce soit nécessaire mais ça serait bien que ce patch avance, il va occasionner moult rebase chiants).

#7 - 17 novembre 2021 20:09 - Frédéric Péters

- Statut changé de Solution proposée à Solution validée

C'est très bien ainsi.

#8 - 18 novembre 2021 10:51 - Valentin Deniaud

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

```
commit 1c8c5f447b264d32a7a4b1a24907eb56ed0fd336
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date: Thu Nov 4 11:05:01 2021 +0100
```

```
api: make APIError less verbose (#58014)
```

#9 - 19 novembre 2021 00:17 - Frédéric Péters

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

Fichiers

0001-api-make-APIError-less-verbose-58014.patch	36,8 ko	04 novembre 2021	Valentin Deniaud
0001-api-make-APIError-less-verbose-58014.patch	38 ko	04 novembre 2021	Valentin Deniaud