Chrono - Bug #57305

API de modification d'un événement

27 septembre 2021 15:25 - Nicolas Roche

Statut:	Fermé	**Début:**	27 septembre 2021
Priorité:	Normal	**Echéance:**	
Assigné à:	Nicolas Roche	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

L'idée c'est de pouvoir modifier le nombre de places, changer le libellé ou les dates de début ou de fin par exemple.

Révisions associées

Révision 9a227239 - 06 octobre 2021 15:30 - Nicolas Roche

agendas: factorize recurring event update code (#57305)

Révision 030a355e - 06 octobre 2021 15:31 - Nicolas Roche

api: add an endpoint to patch an event (#57305)

Historique

#2 - 28 septembre 2021 18:52 - Nicolas Roche

- Assigné à mis à Nicolas Roche

#3 - 28 septembre 2021 19:58 - Nicolas Roche

- Fichier 0001-api-add-PATCH-action-on-event-endpoint-57305.patch ajouté

- Tracker changé de Support à Bug

- Statut changé de Nouveau à En cours

- Patch proposed changé de Non à Oui

Je réalise que j'ai ajouté un nouveau endpoint 'add-event' dans #47337
alors que j'aurais plutôt intérêt à regrouper les nouveaux verbes POST et PATCH sur le endpoint 'status' (qui ne fait qu'un GET actuellement).
Sauf que 'status' c'est pas trés explicite (j'aurais préféré 'event').
Est-ce que je dupplique le GET de 'status' dans un nouveau endpoint 'event' ?

(patch pour illustrer si ce n'est pas clair.)

#4 - 28 septembre 2021 20:33 - Frédéric Péters

Je ne sais pas ce que le patch est supposé illustrer. (je ne veux pas savoir)

Ça n'aurait aucun sens de faire l'ajout d'un événement sur un endpoint dont l'URL contient un identifiant d'événement.

Je comprends qu'il y a envie d'une URL "neutre" pour un événement, pour pouvoir y taper un PATCH pour le modifier.

Qu'en fait aujourd'hui on a :

- '^agenda/(?P<agenda_identifier>[\w-]+)/status/(?P<event_identifier>[\w:-]+)/$'
- '^agenda/(?P<agenda_identifier>[\w-]+)/bookings/(?P<event_identifier>[\w:-]+)/$'
- '^agenda/(?P<agenda_identifier>[\w-]+)/check/(?P<event_identifier>[\w:-]+)/$'

et que ça serait tellement plus net si on avait ^agenda/(?P<agenda_identifier>[\w-]+)/event/(?P<event_identifier>[\w:-]+)/(status ou bookings ou check ou autre).

Qu'il ne soit rien fait de cela ici.

Qu'il soit créé un url(r'^agenda/(?P<agenda_identifier>[\w-]+)/event/(?P<event_identifier>[\w:-]+)/$') et que ça prenne un PATCH pour modifier l'événement en question.

Qu'il ne soit rien fait rien discuté d'autre.

Pas de GET pas de POST pas de modification dans les URL existants.

#5 - 30 septembre 2021 18:41 - Nicolas Roche

- Fichier 0001-api-add-an-enpoint-to-patch-an-event-57305.patch ajouté

- Fichier 0002-agendas-factorize-recurrent-event-update-code-57305.patch ajouté

- Statut changé de En cours à Solution proposée

Merci Fred, remarque prise en compte.

Je propose 0001.

Et dans 0002 je propose également de factoriser le code du manager,
mais c'est pas très lisible donc peut-être de trop.

#6 - 05 octobre 2021 12:06 - Valentin Deniaud

Les parties # legacy access by sont à supprimer, on a pas à supporter l'ancien dans du nouveau (cf le get_object de EventBookings).

Je trouve que la validation est trop bien faite, il faudrait faire moins bien et plus simple : lever ValidationError comme c'est fait partout et ne pas chercher à collecter toutes les erreurs dans un dictionnaire pour toutes les rapporter en une fois.

Ensuite limiter la duplication de code de validation entre le formulaire et l'API me paraît effectivement nécessaire. Pour que 0002 soit plus propre tu pourrais essayer d'en faire un contextmanager qui serait une méthode de Event comme tes pre/post_update, et ça donnerait un truc genre

```
with self.instance.update_recurrences(self.changed_data, self.cleaned_data, self.protected_fields):
    super().save(*args, **kwargs)
```

(et niveau découpage de patches ça serait 0001 ajout de ce contextmanager et utilisation dans le formulaire et 0002 le reste)

Malgré ça il reste plein de code dupliqué mais je ne sais pas trop comment faire autrement :/ d'ailleurs tu as oublié de dupliquer la partie qui empêche de modifier certains champs d'une récurrence d'un évènement (le if self.instance.primary_event:).

#7 - 05 octobre 2021 17:47 - Nicolas Roche

- Fichier 0002-api-add-an-enpoint-to-patch-an-event-57305.patch ajouté

- Fichier 0001-agendas-factorize-recurrent-event-update-code-57305.patch ajouté

> Les parties # legacy access by sont à supprimer, on a pas à supporter l'ancien dans du nouveau (cf le get_object de EventBookings).

Fait, merci.

> Je trouve que la validation est trop bien faite, il faudrait faire moins bien et plus simple : lever ValidationError comme c'est fait partout et ne pas chercher à collecter toutes les erreurs dans un dictionnaire pour toutes les rapporter en une fois.

ValidationError est levée par les serializers puis retranscrite en APIError dans les vues, pour avoir par exemple 'err=1'.
cf le test test_datetimes_api_user_external_id :

```
    assert resp.json['err'] == 1
    assert resp.json['err_desc'] == 'invalid payload'
    assert resp.json['errors']['user_external_id'] == [
        'user_external_id and exclude_user_external_id have different values'
    ]
```

Ici on ne peut pas détecter les erreurs champ par champ au niveau du serializer.
Ensuite, comme de toute façon il faut un dictionnaire, je me dis que ça ne coûte rien de retourner toutes les erreurs d'un coup (je trouve ça plus sympa quand il faut utiliser un WS).
Mais bon, dis-moi si tu veux que je sorte direct en erreur.

> Ensuite limiter la duplication de code de validation entre le formulaire et l'API me paraît effectivement nécessaire. Pour que 0002 soit plus propre tu pourrais essayer d'en faire un contextmanager qui serait une méthode de Event comme tes pre/post_update, et ça donnerait un truc genre
>
> with self.instance.update_recurrences(self.changed_data, self.cleaned_data, self.protected_fields):
> super().save(*args, **kwargs)

Fait.

> Malgré ça il reste plein de code dupliqué mais je ne sais pas trop comment faire autrement :/ d'ailleurs tu as oublié de dupliquer la partie qui empêche de modifier certains champs d'une récurrence d'un évènement (le if self.instance.primary_event:).

Fait, et j'ai ajouté ces 2 tests, merci :
- update unprotected fields of one of the event recurrencies
- try to update protected fields of one of the event recurrencies

#8 - 06 octobre 2021 11:32 - Valentin Deniaud

> Mais bon, dis-moi si tu veux que je sorte direct en erreur.

Oui je trouve que ça fait quand même un code plus lisible et cohérent avec le reste, le gain d'avoir tout rapporté en un coup me paraît trop marginal.

0001, le nom more_protected_field n'est pas très explicite, ça pourrait s'appeler exclude_fields et contenir la liste des champs à ne pas mettre à jour, genre exclude_fields=protected_fields + ['recurrence_end_date', 'frequency'] au moment de l'appel.

0002,

```
if payload.get(field) and payload.get(field) != getattr(event, field):
    changed_data.append(field)
```

Si je veux retirer la date de publication d'un évènement, je vais envoyer publication_date: null, et alors publication_date ne se retrouvera pas dans changed_data alors que ça devrait. J'imagine que le premier test devrait être un « in ».

#9 - 06 octobre 2021 13:51 - Nicolas Roche

- Fichier 0002-api-add-an-enpoint-to-patch-an-event-57305.patch ajouté

- Fichier 0001-agendas-factorize-recurrent-event-update-code-57305.patch ajouté

Remarques prises en compte.

#10 - 06 octobre 2021 14:04 - Valentin Deniaud

Dernier round de pinaillage :

0001, ça ne sert à rien d'avoir les arguments optionnels dans la méthode, vu qu'on passe tout à chaque fois.

0002, maintenant qu'il n'y a plus qu'une erreur dans chaque APIError, autant avoir l'erreur dans err_desc directement (ie remplacer 'invalid payload' par l'erreur en question et ne plus utiliser le paramètre errors).

#11 - 06 octobre 2021 14:56 - Nicolas Roche

- Fichier 0002-api-add-an-enpoint-to-patch-an-event-57305.patch ajouté

- Fichier 0001-agendas-factorize-recurrent-event-update-code-57305.patch ajouté

Fait.

#12 - 06 octobre 2021 15:13 - Valentin Deniaud

- Statut changé de Solution proposée à Solution validée

Top, dernière chose les titres de commit
« factorize recurrent event » -> recurring
« add an enpoint » -> enDpoint

#13 - 06 octobre 2021 15:34 - Nicolas Roche

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

```
commit 030a355ebf443a8d8fb3b6840343d0af4b4fa854
Author: Nicolas ROCHE <nroche@entrouvert.com>
Date:   Thu Sep 30 00:23:31 2021 +0200

    api: add an endpoint to patch an event (#57305)
```

```
commit 9a2272396b62b057c62caa1fe1ebc1fc51b96f1f
Author: Nicolas ROCHE <nroche@entrouvert.com>
Date:   Tue Oct 5 14:36:29 2021 +0200

    agendas: factorize recurring event update code (#57305)
```

#14 - 07 octobre 2021 22:17 - Frédéric Péters

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

Fichiers

0001-api-add-PATCH-action-on-event-endpoint-57305.patch	7,67 ko	28 septembre 2021	Nicolas Roche
0001-api-add-an-enpoint-to-patch-an-event-57305.patch	11 ko	30 septembre 2021	Nicolas Roche
0002-agendas-factorize-recurrent-event-update-code-57305.patch	5,81 ko	30 septembre 2021	Nicolas Roche
0002-api-add-an-enpoint-to-patch-an-event-57305.patch	12,2 ko	05 octobre 2021	Nicolas Roche
0001-agendas-factorize-recurrent-event-update-code-57305.patch	5,35 ko	05 octobre 2021	Nicolas Roche
0002-api-add-an-enpoint-to-patch-an-event-57305.patch	12,1 ko	06 octobre 2021	Nicolas Roche
0001-agendas-factorize-recurrent-event-update-code-57305.patch	5,31 ko	06 octobre 2021	Nicolas Roche
0002-api-add-an-enpoint-to-patch-an-event-57305.patch	12,1 ko	06 octobre 2021	Nicolas Roche
0001-agendas-factorize-recurrent-event-update-code-57305.patch	5,09 ko	06 octobre 2021	Nicolas Roche