

Chrono - Development #33489

API : lors de la création d'une réservation, permettre d'en annuler une autre

28 mai 2019 14:01 - Emmanuel Cazenave

Statut:	Fermé	Début:	28 mai 2019
Priorité:	Normal	Echéance:	05 juin 2019
Assigné à:	Emmanuel Cazenave	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description D'un mail : "ce que j'imaginai, côté API, c'est que l'appel à 'fillslot' puisse prendre en paramètre un identifiant de réservation existant, et que dans ce cas, le créneau qui était attaché à la réservation se libère en même temps que le nouveau est réservé."			

Révisions associées

Révision ab9ff7ad - 29 mai 2019 16:09 - Emmanuel Cazenave

api: cancel a booking when making a new one (#33489)

Historique

#1 - 28 mai 2019 14:02 - Emmanuel Cazenave

- Echéance mis à 12 juin 2019

#2 - 28 mai 2019 17:03 - Emmanuel Cazenave

- Fichier 0001-api-cancel-a-booking-when-making-a-new-one-33489.patch ajouté
- Statut changé de Nouveau à Solution proposée
- Patch proposed changé de Non à Oui

Partis pris peut-être peu éclairé d'ignorer le problème si la réservation qu'on cherche à annuler n'existe pas ou est déjà annulée.

#3 - 28 mai 2019 17:04 - Frédéric Péters

Partis pris peut-être peu éclairé d'ignorer le problème si la réservation qu'on cherche à annuler n'existe pas ou est déjà annulée.

Retourner une erreur dans ces situations.

#4 - 28 mai 2019 17:06 - Emmanuel Cazenave

- Echéance changé de 12 juin 2019 à 05 juin 2019

#5 - 28 mai 2019 17:40 - Emmanuel Cazenave

- Fichier 0001-api-cancel-a-booking-when-making-a-new-one-33489.patch ajouté

Voilà.

#6 - 28 mai 2019 17:48 - Frédéric Péters

Je me dis que ça pourrait être monté plus haut, histoire de ne pas avoir à calculer les créneaux dispos si la réservation ne peut de toute façon pas se faire.

Aussi, vérifier count, et là, au choix, soit obliger que ça soit nécessairement 1, soit prêter attention au primary_booking, et assurer que le count corresponde à celui qui avait servi à la réservation.

#7 - 28 mai 2019 18:34 - Emmanuel Cazenave

Flash : manque aussi d'encapsuler tout ça dans une transaction.

#8 - 28 mai 2019 19:13 - Emmanuel Cazenave

La totale.

#9 - 28 mai 2019 19:14 - Emmanuel Cazenave

- Fichier 0001-api-cancel-a-booking-when-making-a-new-one-33489.patch ajouté

Avec le patch.

#10 - 28 mai 2019 19:38 - Emmanuel Cazenave

Je me dis soudainement qu'on a besoin d'être plus subtil sur le calcul du nombre de places restantes : un événement pourrait être vu comme plein et la réservation refusée parce qu'on ne prends pas en compte les places libérées par l'annulation dans ce calcul.

#11 - 29 mai 2019 09:42 - Emmanuel Cazenave

Emmanuel Cazenave a écrit :

~~Je me dis soudainement qu'on a besoin d'être plus subtil sur le calcul du nombre de places restantes : un événement pourrait être vu comme plein et la réservation refusée parce qu'on ne prends pas en compte les places libérées par l'annulation dans ce calcul.~~

Ça n'a aucun sens.

#12 - 29 mai 2019 10:17 - Thomas Noël

Dans la réponse, s'il y a eu un cancel, je serais pour le dire, genre « response[Cancelled] = payload.get('cancel_booking_id') » dans l'idée que wcs stockera ça dans son retour, et on ainsi savoir que la réservation en a aussi annulé une autre (et le dire à l'utilisateur, avec juste un "if response_data_cancelled" dans le mail, ce genre de truc). Je suis d'accord qu'on le sait déjà car on a fait l'appel webservice pour, mais il peut avoir un workflow avec différentes branches, etc...

Aussi, en fait, au lieu de juste faire « if 'cancel_booking_id' in payload » on devrait plutôt faire un « if payload.get('cancel_booking_id'): » dans l'idée qu'un appel webservice avec un cancel_booking_id vide, c'est acceptable, ça veut juste dire "y'a rien à annuler". Ça simplifiera l'appel via les workflows -- bon, j'imagine que le « cancel_booking_id = serializers.IntegerField() » va nous faire chier.

#13 - 29 mai 2019 12:03 - Emmanuel Cazenave

- Statut changé de Solution proposée à Information nécessaire

- Assigné à changé de Emmanuel Cazenave à Pierre Cros

Thomas Noël a écrit :

Dans la réponse, s'il y a eu un cancel, je serais pour le dire, genre « response[Cancelled] = payload.get('cancel_booking_id') » dans l'idée que wcs stockera ..

Pris en compte, j'ai mis cancelled_booking_id.

Aussi, en fait, au lieu de juste faire « if 'cancel_booking_id' in payload » on devrait plutôt faire un « if payload.get('cancel_booking_id'): » ...

Ça ouvre quand même en grand la porte à des annulations qui ne se font pas parce que paramétrage de l'action d'appel webservice foirée et pas de remontée d'erreur.

Input needed : Pierre, est-ce que dans l'utilisation que tu imagines tu as besoin que la même action d'appel webservice fasse tantôt une annulation tantôt n'en fasse pas ? (ça nous rassurerait que dans un appel donné, si le paramètre cancel_booking_id est passé, on n'autorise pas que la valeur en fasse soit vide)

#14 - 29 mai 2019 15:01 - Thomas Noël

Emmanuel Cazenave a écrit :

Aussi, en fait, au lieu de juste faire « if 'cancel_booking_id' in payload » on devrait plutôt faire un « if payload.get('cancel_booking_id'): » ...

Ça ouvre quand même en grand la porte à des annulations qui ne se font pas parce que paramétrage de l'action d'appel webservice foirée et pas de remontée d'erreur.

Effectivement, laissons tomber pour l'instant. On verra si c'est vraiment utile pour remplacer par quelque chose comme -1 ou 0... (oui, je sais, c'est moche)

Tu rajoutes un test qui faile avec params={'cancel_booking_id': ''} ? (pour le jour où ça changera) (oui, j'aime être pénible)

#15 - 29 mai 2019 16:10 - Emmanuel Cazenave

- Fichier 0001-api-cancel-a-booking-when-making-a-new-one-33489.patch ajouté
- Statut changé de Information nécessaire à Solution proposée

Thomas Noël a écrit :

Tu rajoutes un test qui faile avec params={'cancel_booking_id': ''} ? (pour le jour où ça changera) (oui, j'aime être pénible)

Voilà (tout à la fin du test).

#16 - 29 mai 2019 17:27 - Thomas Noël

- Statut changé de Solution proposée à Solution validée
- Assigné à changé de Pierre Cros à Emmanuel Cazenave

#17 - 03 juin 2019 09:37 - Pierre Cros

Si je comprends correctement, tu peux committer Manu ? Je testerai après déploiement sur la recette.

#18 - 03 juin 2019 10:36 - Pierre Cros

Un peu tardivement je vois la question sur les événements mais je comprends pas l'utilité de déplacer dans un événement, pour moi c'est un truc limité au RdV.

Mais c'est peut-être qu'on peut déplacer d'un événement à un autre ce que je n'avais pas compris.

#19 - 03 juin 2019 11:48 - Emmanuel Cazenave

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

Pierre Cros a écrit :

Mais c'est peut-être qu'on peut déplacer d'un événement à un autre ce que je n'avais pas compris.

Oui pas de restrictions sur rdv/événement.

Author: Emmanuel Cazenave <ecazenave@entrouvert.com>
Date: Tue May 28 16:32:30 2019 +0200

```
api: cancel a booking when making a new one (#33489)
```

#20 - 03 juin 2019 20:16 - Frédéric Péters

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

Fichiers

0001-api-cancel-a-booking-when-making-a-new-one-33489.patch	3,5 ko	28 mai 2019	Emmanuel Cazenave
0001-api-cancel-a-booking-when-making-a-new-one-33489.patch	3,94 ko	28 mai 2019	Emmanuel Cazenave
0001-api-cancel-a-booking-when-making-a-new-one-33489.patch	7,34 ko	28 mai 2019	Emmanuel Cazenave
0001-api-cancel-a-booking-when-making-a-new-one-33489.patch	8 ko	29 mai 2019	Emmanuel Cazenave