

Chrono - Bug #63358

Agenda virtuel : délais de réservation des agendas réels non honorés lors de la pose d'une réservation

30 mars 2022 12:38 - Emmanuel Cazenave

Statut:	Fermé	Début:	30 mars 2022
Priorité:	Normal	Echéance:	
Assigné à:	Emmanuel Cazenave	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		

Description

Un créneau qui sort via un agenda virtuel et qui vient d'un agenda réel X.

Sur l'appel fillslot la réservation est posée sur l'agenda réel Y, ce qui ne devrait pas être possible, violation du délai de réservation maximal de Y.

Révisions associées

Révision 89289ad6 - 31 mars 2022 14:41 - Emmanuel Cazenave

agendas: honor real agendas booking delay when booking through a virtual agenda (#63358)

Historique

#2 - 30 mars 2022 12:39 - Emmanuel Cazenave

- Sujet changé de Agenda virtuel : pose d'une réservation sur mauvais agenda à Agenda virtuel : pose d'une réservation sur mauvais agenda (violation du délai de réservation maximal)

#3 - 30 mars 2022 16:17 - Emmanuel Cazenave

- Fichier 0001-agendas-honor-real-agendas-booking-delay-when-bookin.patch ajouté

- Sujet changé de Agenda virtuel : pose d'une réservation sur mauvais agenda (violation du délai de réservation maximal) à Agenda virtuel : délais de réservation des agendas réels non honorés lors de la pose d'une réservation

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Bug grossier (je ne creuse pas pour savoir si là depuis le début ou introduit récemment).

Le contexte est un agenda virtuel sans délais de réservation min/max, dans cette situation les délais des agendas réels doivent être honorés.

Sur l'appel à datetimes, ils le sont, get_all_slots appelé sans start_datetime/end_datetime.

Sur l'appel à fillslot ils sont violés, get_all_slots appelé avec start_datetime = début du créneau à réserver et end_datetime = fin du créneau à réserver, ça provoque base_min_datetime=start_datetime et base_max_datetime=end_datetime, qui provoque je m'assoie sur les délais des agendas réels via :

```
used_min_datetime = base_min_datetime or get_min_datetime(agenda, start_datetime)
used_max_datetime = base_max_datetime or get_max_datetime(agenda, end_datetime)
```

#4 - 30 mars 2022 16:21 - Frédéric Péters

Sur l'appel à datetimes

Ok mais il était choisi comment le créneau dans #63297 ?

#5 - 30 mars 2022 16:44 - Emmanuel Cazenave

Frédéric Péters a écrit :

Ok mais il était choisi comment le créneau dans #63297 ?

(je ne suis pas sûr de comprendre la question)

C'est la situation de la description du ticket.

Un usager y fait une résa via un agenda virtuel, les agendas réels qui y sont inclus ont des délais de réservations différents, un créneau libre sort via datetimes parce que qu'il y a bien un créneau libre sur un agenda réel X et que c'est raccord avec les délais de X. Puis sur la pose de la réservation ça déraile, la résa devrait arriver sur X (ou n'importe quel autre agenda où c'est libre et raccord avec les délais) et elle arrive sur Y alors que ce créneau est pas raccord avec les délais de Y.

Ce qui explique #63297#note-11, en passant par l'agenda virtuel "au plus rapide" les gens arrivent à réserver pour le 26 avril dans "Mairie de quartier de Borny", tandis que s'ils choisissent le lieu (datetimes via agenda normal, délais de réservation honorés) ils ne voient pas de créneau libre.

#6 - 30 mars 2022 16:51 - Frédéric Péters

(je ne suis pas sûr de comprendre la question)

Si le paramétrage était correctement pris en compte dans l'API /datetimes (ce que je pensais avoir compris de ton commentaire), qu'elle ne renvoyait bien que les créneaux "licites", et comme c'est l'API /datetimes qui est utilisée pour afficher des créneaux (à moins qu'autre chose ici c'était ma question), comment est-ce qu'une sélection d'une valeur "illicite" a pu se faire ?

#7 - 30 mars 2022 16:53 - Frédéric Péters

Puis sur la pose de la réservation ça déraile, la résa devrait arriver sur X (ou n'importe quel autre agenda où c'est libre et raccord avec les délais) et elle arrive sur Y alors que ce créneau est pas raccord avec les délais de Y.

Mais avec ça j'ai donc compris qu'il n'y avait pas prise de créneau illicite en soit, c'est "juste" que le créneau licite était ensuite enregistré sur un agenda où il ne l'était pas.

#8 - 31 mars 2022 08:16 - Lauréline Guérin

Sur l'appel à fillslot ils sont violés, get_all_slots appelé avec start_datetime = début du créneau à réserver et end_datetime = fin du créneau à réserver

C'était probablement pour optimiser l'appel à get_all_slots, et réduire la fenêtre de recherche de créneaux dans le cas où on sait déjà où ça doit tomber.

Plutôt que faire

```
used_min_datetime = base_min_datetime
if base_agenda.minimal_booking_delay is None:
    used_min_datetime = get_min_datetime(agenda, start_datetime)
used_max_datetime = base_max_datetime
if base_agenda.maximal_booking_delay is None:
    used_max_datetime = get_max_datetime(agenda, end_datetime)
```

pourquoi pas faire du min/max ?

#9 - 31 mars 2022 10:45 - Emmanuel Cazenave

Lauréline Guerin a écrit :

pourquoi pas faire du min/max ?

Jusqu'à maintenant l'interface des agendas virtuels est :

- si pas de délais des réservation min/max sur l'agenda viruel, ceux des agendas réels sont utilisé
- sinon les délais de l'agenda virtuel prennent le pas

Si je ne m'abuse ta proposition ferait passer l'interface à :

- utilisation des délais les plus (ou les moins selon la gueule du patch) restrictifs

#10 - 31 mars 2022 10:46 - Emmanuel Cazenave

Emmanuel Cazenave a écrit :

Jusqu'à maintenant l'interface des agendas virtuels est :

Et ce ticket montre que ça marchait moyennement, mais cette interface était NOTRE PROJET.

#11 - 31 mars 2022 11:15 - Lauréline Guérin

je pensais à:

```
used_min_datetime = max(base_min_datetime, get_min_datetime(agenda, start_datetime))
used_max_datetime = min(base_max_datetime, get_max_datetime(agenda, end_datetime))
```

(ok, ça marche pas parce que base_xx_datetime peut être null)

(tu peux essayer de ne pas mélanger les tests datetimes/fillslots dans un module qui s'appelle datetimes ? :)

#12 - 31 mars 2022 13:35 - Emmanuel Cazenave

- Fichier 0001-agendas-honor-real-agendas-booking-delay-when-bookin.patch ajouté

Lauréline Guerin a écrit :

(tu peux essayer de ne pas mélanger les tests datetimes/fillslots dans un module qui s'appelle datetimes ? :)

Voilà.

#13 - 31 mars 2022 14:19 - Lauréline Guérin

- Statut changé de Solution proposée à Solution validée

c'est pas vraiment ce que j'avais en tête :)

#14 - 31 mars 2022 14:42 - Emmanuel Cazenave

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

```
commit 89289ad665ad2eccc0220c2b38ad51d3f399f6
Author: Emmanuel Cazenave <ecazenave@entrouvert.com>
Date: Wed Mar 30 15:34:28 2022 +0200
```

```
agendas: honor real agendas booking delay when booking through a virtual agenda (#63358)
```

#15 - 31 mars 2022 14:46 - Emmanuel Cazenave

Lauréline Guerin a écrit :

c'est pas vraiment ce que j'avais en tête :)

Si ce que tu avais en tête était un découpage en plusieurs tests différents : les assertions sur le résultats de datetimes sont juste là pour vérifier que l'environnement est bien mis en situation de reproduire le bug, et ce sont les assertions postérieures à l'appel à fillslot qui reproduisent le bug = ya pas de lieu d'écrire un test part dans test_meetings_datetimes.py.

#16 - 04 avril 2022 18:17 - Transition automatique

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

#17 - 05 juin 2022 04:42 - Transition automatique

Automatic expiration

Fichiers

0001-agendas-honor-real-agendas-booking-delay-when-bookin.patch	4,73 ko	30 mars 2022	Emmanuel Cazenave
0001-agendas-honor-real-agendas-booking-delay-when-bookin.patch	4,87 ko	31 mars 2022	Emmanuel Cazenave