Chrono - Bug #20791

Empecher le changement d'une periode horaire quand des reservations existent

19 décembre 2017 09:56 - Josué Kouka

Statut:	Rejeté	**Début:**	19 décembre 2017
Priorité:	Normal	**Echéance:**	
Assigné à:		**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Non	**Planning:**	Non

Description

Actuellement, il est possible de changer des periodes horaires quand bien meme des réservations existent. Du coup certaines reservations se retrouvent à cheval entre des périodes horaires ou non existantes dans les périodes horaires définies.

Historique

#2 - 19 décembre 2017 10:09 - Josué Kouka

- Statut changé de En cours à Rejeté

Je rejete le ticket, couvert dans #20732

#3 - 19 décembre 2017 10:12 - Josué Kouka

- Statut changé de Rejeté à En cours

Ma faute, #20732 ne le couvre pas en fait.

#4 - 19 décembre 2017 15:19 - Josué Kouka

- Fichier 0001-manager-prevent-time-period-changes-when-bookings-ex.patch ajouté

- Patch proposed changé de Non à Oui

Un patch auquel il manque l'autorisation de modification lors d'une extension de la période horaire (on ouvre plus tot ou on finit plus tard).
Par rapport à cette dernière une question, on autorise que les extensions dont le pas est égale au pgcd des des différents types de rendez-vous ou le permet quand il ne l'est pas ?

#5 - 19 décembre 2017 15:28 - Frédéric Péters

> on autorise que les extensions dont le pas est égale au pgcd des des différents types de rendez-vous ou le permet quand il ne l'est pas ?

Un effort de relecture, stp. Dans cette phrase et dans le patch.

Et si je comprends bien la question, on n'interdit pas et c'est déjà géré correctement (cf le test ajouté dans #20732 qui concerne cette situation).

#6 - 19 décembre 2017 15:39 - Josué Kouka

Frédéric Péters a écrit :

> Et si je comprends bien la question, on n'interdit pas et c'est déjà géré correctement (cf le test ajouté dans #20732 qui concerne cette situation).

Ok. Oui ça l'est au niveau de l'api mais je pense que empêcher un agent de changer la période horaire quand des rdvs existent est important. Dans le cas de Meaux, ils parlent de collisions alors qu'ils auraient du décaler les rdv déja pris avant de changer les périodes horaires.

#7 - 19 décembre 2017 15:56 - Frédéric Péters

Je te suggère de reformuler ton commentaire pas clair, je n'ai probablement pas compris la question.

- début 8h du matin, rendez-vous à 9h,
  - je change le début à 8h30 → pas de soucis, c'est ma réponse, on se contrefiche de coller aux pas de "une heure", ça serait 8h15 ou 8h55 ça serait bon aussi.
- début 8h du matin, rendez-vous à 8h,

- je change le début à 8h30 → alerte, il y a un rendez-vous qui entrerait en zone impossible. (comme les exceptions).

#8 - 19 décembre 2017 16:28 - Josué Kouka

Frédéric Péters a écrit :

> Je te suggère de reformuler ton commentaire pas clair, je n'ai probablement pas compris la question.
>
> - début 8h du matin, rendez-vous à 9h,
> - je change le début à 8h30 → pas de soucis, c'est ma réponse, on se contrefiche de coller aux pas de "une heure", ça serait 8h15 ou 8h55 ça serait bon aussi.
> - début 8h du matin, rendez-vous à 8h,
> - je change le début à 8h30 → alerte, il y a un rendez-vous qui entrerait en zone impossible. (comme les exceptions).

Oui, c'est exactement ça, en plus des cas suivants:

- fin 16h30, rendez-vous à 16h
  - je change la fin à 17h00 → pas de souci.
- fin 16h30, rendez-vous à 16h
  - je change la fin à 15h30 → alerte, il y a un rendez-vous qui entrerait en zone impossible. (comme les exceptions).

#9 - 19 décembre 2017 16:35 - Benjamin Dauvergne

SingleResponsibilityPrinciple: je bougerai cette ligne `if hasattr(self.instance, 'desk')`: dans changes_ok() ce n'est pas le boulot de clean() de savoir si changes_ok() peut être utilisé ou pas.

#10 - 19 décembre 2017 16:41 - Benjamin Dauvergne

Puisqu'on s'intéresse qu'aux event et pas aux booking en fait est-ce qu'on pourrait pas faire au lieu de

```
        bookings = Booking.objects.filter(
            event__desk=self.instance.desk, cancellation_datetime__isnull=True).select_related('event')
```

ça

```
        # use distinct() because we join with an implicit OneToMany relation
        events = Event.objects.filter(desk=self.instance.desk, booking__cancellation_datetime__isnull=True).di
stinct()
```

les localtime() c'est pour être sûr du calcul du weekday autour de minuit c'est cela ?

#11 - 19 décembre 2017 16:51 - Frédéric Péters

> pas aux booking

C'est bien aux réservations qu'on doit s'intéresser.

#12 - 20 décembre 2017 19:09 - Josué Kouka

- Fichier 0001-manager-prevent-time-period-changes-when-bookings-ex.patch ajouté

Un patch gerant les cas évoqué en amont.

#13 - 20 décembre 2017 20:23 - Frédéric Péters

- Lire et répondre au commentaire posé par Benjamin.
  - « les localtime() c'est pour être sûr du calcul du weekday autour de minuit c'est cela ? »
- Assurer une couverture complète. (si on avait l'info automatiquement je ne lirais même pas le patch)
- Je n'aime pas du tout cet hasattr() qui certes fonctionne mais me semble contre-intuitif quand la classe TimePeriod a dans sa définition un attribut "desk"
  - Je lui préférerais un plus explicite je trouve if self.instance.desk_id is None: # no desk assigned yet, this is a new time period
- Je n'aime pas du tout "changes_ok" comme nom, la nomenclature "verbe_..." n'est pas respectée et c'est fourre-tout comme nom.
- Ce n'est pas les seuls mais ce n'est pas une raison : les tests tapent dans Event.start_datetime des datetime naïfs, ne devraient pas faire ça.

- self.instance.start_time <= event_startdt.time() < self.instance.end_time est faux
  - il faut considérer la durée du rendez-vous. (si j'ai des rendez-vous calé à 16h00 et prenant une heure, je ne peux pas réduire la journée à 16h30).

#14 - 21 décembre 2017 10:45 - Frédéric Péters

Et bien sûr aussi, tellement énorme, il ne faut pas lever d'alerte sur l'existence de rendez-vous passés.

Et par nécessité de reformuler les choses pour anticiper d'autres erreurs : il faut lever une alerte quand des rendez-vous existent dans le futur et que le changement demandé à la période horaire sortirait certains des rendez-vous des heures d'ouverture.

#15 - 22 décembre 2017 12:17 - Josué Kouka

- Fichier 0001-manager-prevent-time-period-changes-when-bookings-ex.patch ajouté

> les localtime() c'est pour être sûr du calcul du weekday autour de minuit c'est cela ?

Non, c'est parce que mes datetime dans mes tests n'étaient pas localisés.

> Assurer une couverture complète. (si on avait l'info automatiquement je ne lirais même pas le patch)

La ligne non couverte l'est maintenant

> Je n'aime pas du tout cet hasattr() qui certes fonctionne mais me semble contre-intuitif quand la classe TimePeriod a dans sa définition un attribut "desk"

Ok

> Je n'aime pas du tout "changes_ok" comme nom, la nomenclature "verbe_..." n'est pas respectée et c'est fourre-tout comme nom.

are_changes_valid devrait convenir alors

> Ce n'est pas les seuls mais ce n'est pas une raison : les tests tapent dans Event.start_datetime des datetime naïfs, ne devraient pas faire ça.

Ok

> Et par nécessité de reformuler les choses pour anticiper d'autres erreurs : il faut lever une alerte quand des rendez-vous existent dans le futur et que le changement demandé à la période horaire sortirait certains des rendez-vous des heures d'ouverture.

Done

#16 - 24 décembre 2017 10:08 - Frédéric Péters

> > > que le changement demandé à la période horaire sortirait certains des rendez-vous des heures d'ouverture

Parce qu'apparemment jamais suffisamment explicite : si l'admin change le jour d'ouverture du mardi au mercredi, ça peut mettre des rendez-vous hors zone d'ouverture.

> are_changes_valid devrait convenir alors

« et c'est fourre-tout comme nom. »

#17 - 12 janvier 2018 15:08 - Josué Kouka

- Fichier 0001-manager-prevent-time-period-changes-when-bookings-ex.patch ajouté

Frédéric Péters a écrit :

> > > > que le changement demandé à la période horaire sortirait certains des rendez-vous des heures d'ouverture

> Parce qu'apparemment jamais suffisamment explicite : si l'admin change le jour d'ouverture du mardi au mercredi, ça peut mettre des rendez-vous hors zone d'ouverture.

Vrai. je le prends en compte dans ce patch

> > are_changes_valid devrait convenir alors
>
> « et c'est fourre-tout comme nom. »

J'ai finalement tout dans la methode has_bookings_out_bounds.

#18 - 17 janvier 2018 17:32 - Josué Kouka

- Fichier 0001-manager-prevent-time-period-changes-when-bookings-ex.patch ajouté

En rebasant sur master

#19 - 21 avril 2018 14:21 - Thomas Noël

(j'ai pas relu, mais je signale juste qu'il ne s'applique plus sur master, à rebaser sans doute)

#20 - 13 novembre 2018 21:17 - Frédéric Péters

- Assigné à Josué Kouka supprimé

#21 - 14 novembre 2018 08:45 - Frédéric Péters

- Patch proposed changé de Oui à Non

#22 - 14 novembre 2023 16:38 - Valentin Deniaud

- Statut changé de En cours à Rejeté

- Planning mis à Non

On ne veut pas empêcher ça, et c'est sans conséquence sur les réservations existantes.

Fichiers

0001-manager-prevent-time-period-changes-when-bookings-ex.patch	3,32 ko	19 décembre 2017	Josué Kouka
0001-manager-prevent-time-period-changes-when-bookings-ex.patch	5,76 ko	20 décembre 2017	Josué Kouka
0001-manager-prevent-time-period-changes-when-bookings-ex.patch	7,19 ko	22 décembre 2017	Josué Kouka
0001-manager-prevent-time-period-changes-when-bookings-ex.patch	7,36 ko	12 janvier 2018	Josué Kouka
0001-manager-prevent-time-period-changes-when-bookings-ex.patch	7,12 ko	17 janvier 2018	Josué Kouka