

Chrono - Development #17685

"prébloccage" d'une réservation

18 juillet 2017 17:00 - Frédéric Péters

Statut:	Fermé	Début:	18 juillet 2017
Priorité:	Normal	Echéance:	
Assigné à:	Benjamin Dauvergne	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
On imagine un formulaire où on demande à choisir un événement mais derrière il y a encore des pages et pourtant on ne voudrait pas qu'au bout la personne soit déçue et n'ait pas sa place. Bref, imaginer une période de prébloccage.			
Demandes liées:			
Lié à Chrono - Development #23008: api: ajouter une action "lockslot" pour bl...		Rejeté	06 avril 2018
Lié à w.c.s. - Development #24635: Générer un identifiant pour un formdata dè...		Nouveau	19 juin 2018
Lié à Chrono - Development #54956: Revert de #17685 « "prébloccage" d'une rése...		Fermé	17 juin 2021
Lié à Chrono - Development #80489: "prébloccage" d'une réservation		Fermé	22 août 2023 25 octobre 2023
Dupliqué par Chrono - Development #20437: possibilité de verrouiller un crénea...		Rejeté	05 décembre 2017

Révisions associées

Révision 2f9bf16a - 11 juin 2021 07:43 - Benjamin Dauvergne

api: add lock_code parameter to fillslot and datetimes (#17685)

Historique

#2 - 18 juillet 2017 17:07 - Thomas Noël

Note dans #17681 : « À côté de ce côté interface on doit aussi réfléchir aux mécanismes pour éviter que ça ne soit abusé et sans doute d'autres points. »

#3 - 05 décembre 2017 12:19 - Frédéric Péters

- Dupliqué par Development #20437: possibilité de verrouiller un créneau horaire ajouté

#4 - 15 avril 2018 09:53 - Frédéric Péters

- Lié à Development #23008: api: ajouter une action "lockslot" pour bloquer des slots pendant x minutes ajouté

#6 - 15 avril 2018 17:00 - Thomas Noël

J'importe ici mes notes sur [#23008](#) :

On peut avoir des conflits de réservation si plusieurs personnes arrivent sur le front, voient les mêmes datetimes et sélectionnent le même. Lors de l'appel fillslot, seul le premier réussi, les suivants passent en échec.

On pourrait éviter que cela se produise dans le workflow de traitement de la demande, en appelant le webservice en condition de sortie pour vérifier qu'une place est encore disponible et pour la "locker" pendant un certain temps.

Quelque chose comme « /api/agenda/<id>/lockslot/<id>/ » qui renvoie la même chose que fillslot, mais ne lock en réalité que pendant 10 minutes. Ou alors en payload d'un appel à fillslot, avec un "lock:10" ou "lock:True".

Techniquement on imagine un Booking.auto_cancellation_datetime (None par défaut) qui s'il existe, effectue un cancel de la réservation à l'instant donné, via un cron lancé chaque 5 minute.

La difficulté : on ne pourra pas dans w.c.s. stocker l'id de la pré-réservation ainsi obtenue. Donc, dans le workflow, lors du vrai fillslot, il faudra accepter même si booking existe, et dans ce cas le confirmer.

#7 - 17 mai 2018 17:50 - Thomas Noël

- Priorité changé de Normal à Haut

#9 - 05 juin 2018 08:59 - Thomas Noël

- Statut changé de Nouveau à En cours

- Priorité changé de Haut à Normal

Voici une première proposition de fonctionnement.

Tout se passe à travers un "code de blocage" (lock_code) qui doit être unique dans chrono. Via wcs on utilisera la référence de la demande (nnn-mmm).

- On cherche d'abord la liste des dispo pour ce code, disons "MYLOCK". C'est la liste des dispo habituelle.
- Une fois le choix fait, on lance la réservation en mode "lock", en envoyant un fillslot avec le lock_code
 - cela se fera via une condition de sortie de page
 - cela créé un booking, qui a comme particularité d'avoir un lockcode et une date d'expiration de ce lock
- A partir de là, la liste des dispo ne contient plus cette résa
 - et donc les autres personnes ne voient plus la dispo
 - et même si elles l'avaient vu, elles ne peuvent plus la réserver (la condition de sortie va échouer)
 - Cependant, si on envoie le lockcode MYLOCK, la dispo est bien visible (cas d'un aller-retour sur la page de choix des resa)
 - Si on lock une autre dispo avec MYLOCK, la résa précédemment lockée ne l'est plus
- à la fin, pour valider vraiment la réservation, on la renvoie, avec le même lockcode et un confirm_after_lock=True. Cela créé un "vrai" booking, ie sans lockcode, ie non temporaire

Par défaut, un lock dure 10 minutes, ça se change via les settings ou lors de l'appel. Il manque encore dans ce code un cron de nettoyage des lock périmés.

Voilà, avant de chercher à peaufiner et notamment étendre les tests, je veux bien un avis sur :

- le principe
- l'intérêt de la chose sur les agenda events ? (je n'y ai pas trop réfléchi)
- la façon dont c'est codé, parce qu'il y a une partie un peu obscure qui est la suppression des lockcode qui suppriment des events sur un agenda de type meeting (rendez-vous), ces events étant en effet bidons.

#10 - 05 juin 2018 09:32 - Thomas Noël

- Fichier 0001-api-handle-lock-on-fillslots-and-datetimes-17685.patch ajouté

- Patch proposed changé de Non à Oui

Et le patch, donc.

#11 - 05 juin 2018 09:41 - Benjamin Dauvergne

Je ne comprends pas bien le besoin du lock_code, pourquoi ne pas simplement ajouter un paramètre booléen lock si celui-ci présent on pose lock_expiration. On renvoie le event.id ou un truc du genre. Sur un nouvel appel avec cet event_id on vire toutes les histoires de lock, c'est la confirmation de la réservation.

Ça évitera d'avoir à générer un lock_code coté w.c.s. (maintenant c'est peut-être volontaire de pouvoir envoyer un truc un peu intelligible comme l'identifiant du formulaire je ne sais pas trop).

Pour moi le code serait plus clair avec deux web-services plutôt que de tout surcharger sur un seul (un de pré-réservation l'autre de réservation).

#12 - 05 juin 2018 09:59 - Thomas Noël

Benjamin Dauvergne a écrit :

Je ne comprends pas bien le besoin du lock_code, pourquoi ne pas simplement ajouter un paramètre booléen lock si celui-ci présent on pose lock_expiration. On renvoie le event.id ou un truc du genre. Sur un nouvel appel avec cet event_id on vire toutes les histoires de lock, c'est la confirmation de la réservation.

Ouaip, c'est ce qui semble naturel... Mais la contrainte vient de w.c.s. : il peut rien stocker lors du formulaire. Il va juste lancer une condition de sortie, c'est tout (il ne peut rien faire du résultat). À ce stade le lock_code choisi par l'appelant est la seule solution que j'ai trouvée.

Un petit avantage, c'est que si le lock a finalement expiré entre temps mais que la confirmation réussit quand même (parce qu'il reste quand même une place), et bien tout se passe tranquillement. (parce qu'on ne cherche pas à valider une résa existante, mais bien à faire une résa, en considérant le lock comme libre).

#13 - 05 juin 2018 13:48 - Benjamin Dauvergne

Et du disposes de quoi pour construire ton lock code ? Un numéro de session ? Un id de draft ?

#14 - 05 juin 2018 14:51 - Thomas Noël

Benjamin Dauvergne a écrit :

Et du disposes de quoi pour construire ton lock code ? Un numéro de session ? Un id de draft ?

Oui, je pensais à form_number.

#15 - 13 juin 2018 19:27 - Benjamin Dauvergne

Ça m'a l'air ok.

#16 - 19 juin 2018 09:48 - Thomas Noël

- Patch proposed changé de Oui à Non

En fait ça ne marche pas, parce qu'on n'a pas de lock_code disponible à tout coup (form_number n'existe pas lors d'une saisie sur une demande qui n'a pas de mode brouillon)

#17 - 19 juin 2018 10:57 - Benjamin Dauvergne

- Lié à Development #24635: Générer un identifiant pour un formdata dès la saisie ajouté

#18 - 14 février 2020 12:17 - Thomas Noël

On a maintenant session_hash_id #39784 qui peut faire l'affaire en première approximation ; ce ticket peut reprendre vie.

#20 - 16 mars 2021 19:28 - Benjamin Dauvergne

- Assigné à mis à Benjamin Dauvergne

#21 - 16 mars 2021 19:28 - Benjamin Dauvergne

- Fichier 0001-api-add-lock_code-parameter-to-fillslot-and-datetime.patch ajouté

- Statut changé de En cours à Solution proposée

- Patch proposed changé de Non à Oui

J'ai commencé par reprendre le patch de Thomas et puis en bossant dessus je me suis rendu compte que c'était bien compliqué d'ajouter encore de nouveaux champs à Booking/Event&co. J'ai préféré introduire un nouveau modèle sur le coté pour stocker juste les locks. Il se nettoye tout seul (Delete à chaque fois que qu'on le requête) et ça permet de gérer les locks aussi sur les ressources au passage. Je n'ai pas particulièrement optimisé les requêtes dans get_all_slots() parce qu'il faut supposer qu'il n'y aura que très peu de locks vivant en même temps (à peu près autant que d'utilisateurs concurrent sur une période de longueur CHRONO_LOCK_DURATION).

Je n'ai pas particulièrement réfléchi à la réponse de fillslot quand on utilise lock_code, là ça retourne juste {'err': 0}.

J'ai ajouté le même genre d'index d'exclusion que sur les Event mais sans prévenir les bloquages en cas d'absence de l'extension, je me suis dit que ce stade était passé.

Le seul bémol par rapport à une implémentation plus compliqué c'est qu'il reste une très faible possibilité qu'un fillslot ait lieu au moment d'un fillslot?lock_code=XXX. La meilleur protection c'est de recommencer le fillslot?lock_code=XXX en fin de formulaire (et même à chaque page, ainsi ça prolonge le lock à chaque fois). Dans ce cas c'est impossible d'avoir d'erreur (si le formulaire a plus d'une page et que le choix du rdv n'est pas sur la dernière).

PS: j'ai aussi complètement bloqué la fonctionnalité pour le cas des évènements, je n'avais pas de test sous la main et je me suis dit que ce serait mieux de reprendre ça à froid si ça devient nécessaire un jour.

PS2: et bien sûr, mais ça n'a pas été dit avant explicitement, ça ne pourra jamais marcher avec un formulaire ayant des brouillons, ou plutôt si on prévoit des conditions de sortie de page un peu partout, ça devrait lever une erreur à la récupération du brouillon.

#22 - 17 mars 2021 00:18 - Benjamin Dauvergne

3 tests dont le comportement change vers minuit : <https://jenkins.entrouvert.org/job/chrono-wip/job/wip%252F17685--preblocage-d-une-reservation/2/>

#23 - 18 mai 2021 20:02 - Benjamin Dauvergne

- Fichier 0001-api-add-lock_code-parameter-to-fillslot-and-datetime.patch ajouté

Ça devient un peu fatigant de rebaser.

#24 - 02 juin 2021 17:27 - Valentin Deniaud

Qu'est-ce que fait le SQL de la migration ? EXCLUDE USING GIST(desk_id WITH =, tstzrange(start_datetime, end_datetime) WITH &&) franchement qui a inventé une syntaxe pareille. (l'explication peut s'accompagner d'un renommage de sql_forwards en quelque chose de plus parlant)

Mais avant de relire le code je voulais tester et justement, en essayant d'appliquer la migration,

```
Applying agendas.0088_lease...Traceback (most recent call last):
File "/home/vdeniaud/envs/publik-env-py3/lib/python3.7/site-packages/django/db/backends/utils.py", line 82,
in _execute
    return self.cursor.execute(sql)
psycopg2.errors.UndefinedObject: data type integer has no default operator class for access method "gist"
HINT: You must specify an operator class for the index or define a default operator class for the data type.
```

#25 - 02 juin 2021 17:34 - Frédéric Péters

je voulais tester

De bout en bout ? i.e. avec une démarche wcs ?

(ici appel du pied à Benjamin pour qu'il la partage)

#26 - 03 juin 2021 17:04 - Benjamin Dauvergne

- Statut changé de Solution proposée à En cours

Ok, j

Valentin Deniaud a écrit :

Qu'est-ce que fait le SQL de la migration ? EXCLUDE USING GIST(desk_id WITH =, tstzrange(start_datetime, end_datetime) WITH &&) franchement qui a inventé une syntaxe pareille. (l'explication peut s'accompagner d'un renommage de sql_forwards en quelque chose de plus parlant)

Si c'est la première fois que la syntaxe SQL te parait bizarre c'est que tu es t

Mais avant de relire le code je voulais tester et justement, en essayant d'appliquer la migration, [...]

C'est bizarre parce qu'on utilise déjà une contrainte équivalente sur les event:

```
chrono/agendas/migrations/0053_event_date_range_constraint.py:EXCLUDE USING GIST(desk_id WITH =, tstzrange(start_datetime, _end_datetime) WITH &&)
```

et que ça passe crème sur jenkins et chez moi.

Est-ce que la migration 0053 n'échouerait pas chez toi ? (ton user n'est pas administrateur de postgresql).

Alors je peux cacher la non création de l'index de contrainte, encore comme dans 0053, si vous voulez.

Frédéric Péters a écrit :

De bout en bout ? i.e. avec une démarche wcs ?
(ici appel du pied à Benjamin pour qu'il la partage)

Ok je fais ça.

#27 - 03 juin 2021 17:26 - Benjamin Dauvergne

- Fichier 0003-to-fixup-rename-lease-migration.patch ajouté

- Fichier 0004-to-fixup-adapt-dependency-of-lease-migration.patch ajouté

- Fichier 0002-to-fixup-rename-sql-snippets.patch ajouté

- Fichier 0001-api-add-lock_code-parameter-to-fillslot-and-datetime.patch ajouté

Rebasé.

#28 - 03 juin 2021 17:42 - Valentin Deniaud

Benjamin Dauvergne a écrit :

Si c'est la première fois que la syntaxe SQL te paraît bizarre c'est que tu es t

Je n'ose imaginer de quel nom tu voulais me traiter !

Est-ce que la migration 0053 n'échouerait pas chez toi ? (ton user n'est pas administrateur de postgresql).

Pas testé, pas envie de shooter ma db ni de créer un autre tenant juste pour voir. Par contre j'ai googlé et il me manquait un CREATE EXTENSION btree_gist; et c'est désormais bon.

#29 - 03 juin 2021 17:50 - Benjamin Dauvergne

Valentin Deniaud a écrit :

Benjamin Dauvergne a écrit :

Si c'est la première fois que la syntaxe SQL te paraît bizarre c'est que tu es t

Je n'ose imaginer de quel nom tu voulais me traiter !

Je voulais écrire très tolérant :)

Est-ce que la migration 0053 n'échouerait pas chez toi ? (ton user n'est pas administrateur de postgresql).

Pas testé, pas envie de shooter ma db ni de créer un autre tenant juste pour voir. Par contre j'ai googlé et il me manquait un CREATE EXTENSION btree_gist; et c'est désormais bon.

Le CREATE EXTENSION btree_gist est justement dans la migration 0053.

#30 - 03 juin 2021 18:31 - Benjamin Dauvergne

- Fichier *datasource-agenda.wcs* ajouté
- Fichier *wscall-lock_rdv.wcs* ajouté
- Fichier *workflow-rdv.wcs* ajouté
- Fichier *form-test-agenda-lock.wcs* ajouté
- Statut changé de *En cours* à *Solution validée*

Exports de test, il faut:

- un appel webservice vers `api_fillslot_url` avec un seul paramètre `lock_code={{ session_hash_id }}` pour la condition de sortie de la page qui contient le sélecteur de RdV, pour vérifier la dispo et verrouiller (si un verrou pour le `lock_code` existe déjà il est supprimé et ça passe, donc on peut revenir en arrière et changer son RdV); l'appel peut resservir pour tous les formulaires, **si** le champ liste a toujours le même nom de variable (je vois que dans la doc on incite fortement à toujours le nommer "event", dans ce cas ça marchera, comme je ne connaissais pas l'usage ici la variable se nomme "rdv"),
- une source de donnée agenda spécifique parce qu'elle doit préciser `?lock_code={{ session_hash_id|urlencode }}` pour ne pas cacher le RdV préalablement verrouillé en cas de retour en arrière (et éviter le message d'erreur "la sélection est invalide")
- un formulaire/workflow de réservation de RdV bateau, juste l'appel à `api_fillslot_url` doit ajouter `lock_code={{ session_hash_id }}` et `confirm_after_lock=True` (en expression python).

Il faudrait voir pour que ça marche avec les sources de donnée agenda auto-générées en ajoutant automatiquement `lock_code={{ session_hash_id|urlencode }}`, Ça ne devrait pas poser de problème car si le formulaire n'utilise pas le verrouillage ça n'aura pas de conséquence. De même on pourrait automatiquement créer un web-service "lock_event" qui marchera, si la variable s'appelle "event" par convention.

Je pense que j'ai tout dit, de mes premiers tests ça marche nickel (un formulaire ouvert en navigation privé et un en navigation classique, le passage de la première page exclut bien le RdV pour l'autre formulaire avec un message "Désolé, ce rendez-vous est déjà pris.", des retours en arrière puis changement de RdV débloquent la situation).

#31 - 03 juin 2021 18:31 - Benjamin Dauvergne

- Statut changé de *Solution validée* à *Solution proposée*

#32 - 07 juin 2021 17:31 - Valentin Deniaud

Ça marche bien chez moi aussi, pas vraiment de remarques sur le fond, sur la forme :

Renommage de migration, mettre 0091 et pas juste 91.

```
+     If it is booked but match the lock code, report the slot as open.  
+  
+     ""
```

Ligne blanche en trop.

```
+     # only lock related to on of the agenda or the resource
```

to one of (commentaire pas si utile ceci-dit)

La méthode `as_interval` de `Lease` n'est pas utilisée.

Dans `fillslot` tu as mis un `allow_blank=True` et ensuite tu vérifies que le champ n'est pas vide, il faudrait virer les deux et ça marcherait pareil.

#33 - 08 juin 2021 15:17 - Benjamin Dauvergne

- Fichier `0007-to-fixup-remove-useless-comment.patch` ajouté
- Fichier `0003-to-fixup-rename-lease-migration.patch` ajouté
- Fichier `0004-to-fixup-adapt-dependency-of-lease-migration.patch` ajouté
- Fichier `0002-to-fixup-rename-sql-snippets.patch` ajouté
- Fichier `0009-to-fixup-let-DRF-handle-empty-lock_code-errors.patch` ajouté
- Fichier `0008-to-fixup-removed-unused-Lease.as_interval.patch` ajouté
- Fichier `0001-api-add-lock_code-parameter-to-fillslot-and-datetime.patch` ajouté
- Fichier `0005-to-fixup-rename-migration-from-91-to-0091.patch` ajouté
- Fichier `0006-to-fixup-remove-blank-line.patch` ajouté

Remarques prises en compte.

#34 - 10 juin 2021 10:01 - Valentin Deniaud

- Statut changé de *Solution proposée* à *Solution validée*

#35 - 11 juin 2021 07:43 - Benjamin Dauvergne

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

```
commit 2f9bf16a57bab183b3f2650d882ecf37348ed053  
Author: Benjamin Dauvergne <bdauvergne@entrouvert.com>  
Date: Tue Mar 16 14:29:41 2021 +0100
```

```
api: add lock_code parameter to fillslot and datetimes (#17685)
```

#36 - 17 juin 2021 17:39 - Benjamin Dauvergne

- Lié à *Development #54956: Revert de #17685 « "préblochage" d'une réservation »* ajouté

#37 - 17 juin 2021 17:41 - Benjamin Dauvergne

- Statut changé de *Résolu (à déployer)* à *Nouveau*

#38 - 17 juin 2021 17:41 - Benjamin Dauvergne

Commit en cours de revert.

#39 - 18 juin 2021 07:53 - Frédéric Péters

De [#54780#note-26](#),

Le code de pré-blocage peut fonctionner, mais plus mal, sans l'index, les gens qui n'ont pas une base à jour auront juste des bugs qu'on aura pas.

Identifiés, ils peuvent être pris en compte dans une nouvelle version de ce patch ? Ou bien il s'agit de bug "quand même parfois sous forte charge il y aura une réservation qui passera" ?

Aussi/surtout, à passer sur les chaînes à traduire j'ai vu "lock_code does not work with events" qui correspond à ce commentaire plus haut :

PS: j'ai aussi complètement bloqué la fonctionnalité pour le cas des événements, je n'avais pas de test sous la main et je me suis dit que ce serait mieux de reprendre ça à froid si ça devient nécessaire un jour.

Les grosses affluences ça a souvent été pour les événements, je pense par exemple aux piscines à Toulouse.

De la description du ticket d'ailleurs : « choisir un événement » (c'est plus loin qu'il commence à être question de rendez-vous et conflits de réservation).

#40 - 22 décembre 2021 11:14 - Benjamin Dauvergne

- Assigné à Benjamin Dauvergne supprimé

Je me débranche de ce ticket, je préfère laisser ça à la la team chrono.

#41 - 12 décembre 2022 20:31 - Anaïs Ecuillon → en congés, retour le 30/04

- Tags mis à simplification agenda

#42 - 17 mars 2023 09:32 - Olivier Renard

Je me permets de réveiller ticket, suite à #75543.

Je ne sais pas si le préblocage de rendez vous est couvert dans ce ticket.

#43 - 17 mars 2023 10:18 - Benjamin Dauvergne

Olivier Renard a écrit :

Je me permets de réveiller ticket, suite à #75543.

Je ne sais pas si le préblocage de rendez vous est couvert dans ce ticket.

Justement ça ne couvre que les rendez-vous, pour moi ce n'était pas un problème.

#44 - 17 mars 2023 10:40 - Olivier Renard

Oui, peut être qu'il faut dissocier le préblocage d'un rdv et le préblocage d'une place d'un événement (je ne connais pas la complexité technique à le faire).

Il n'y a pas la même criticité.

Une événement a souvent un nombre de places, a potentiellement une file d'attente, ce qui donne moins de criticité entre le choix de la place dans le formulaire et l'appel webservice de réservation.

Un rdv a plus de criticité (de mon point de vue).

#45 - 17 mars 2023 10:56 - Pierre Cros

La description du ticket c'est les événements et Fred citait l'exemple des piscines prises d'assaut.

Mais c'est vrai qu'un cas d'usage souvent évoqué pour ce lock c'est les RdV pour avoir une carte de séjour qui sont épuisés dès la mise à disposition des créneaux avec un trafic (dans tous les sens du terme) de dingue.

Et bien sûr c'est mieux d'avoir ça au moins pour les RdV que de ne rien avoir du tout. Après ce serait pas mal que ce qui est fait pour les RdV ne soit pas à détricoter le jour où on se souciera des événements, quand même. Je ne sais pas si cette dimension peut être prise en compte.

#46 - 17 mars 2023 11:45 - Benjamin Dauvergne

Pierre Cros a écrit :

Et bien sûr c'est mieux d'avoir ça au moins pour les RdV que de ne rien avoir du tout. Après ce serait pas mal que ce qui est fait pour les RdV ne soit pas à détricoter le jour où on se souciera des événements, quand même. Je ne sais pas si cette dimension peut être prise en compte.

C'est totalement indépendant, il faudra une solution complètement différente de toute façon, parce que déjà gérer les événements et les rdv dans la même application n'avait pas de sens même initialement, on aurait pu faire deux applications. Le partage des modèles est totalement artificiel, il n'y a rien de commun dans le code entre les deux, à part qu'un événement et un rdv ont des date de début et de fin.

#47 - 17 mars 2023 13:00 - Benjamin Dauvergne

Le principal souci pour les événements c'est de déterminer un comportement quand une liste d'attente est configurée. Des gens pourront se retrouver en liste d'attente alors que les réservations précédentes verrouillées se libèrent. Il faudrait une gestion automatique de la liste d'attente pour repasser ces personnes sur la liste principale avant de nouvelles réservations; et il faudrait pouvoir notifier w.c.s. de ce changement sans que celui-ci ait à poller l'état de la réservation. Comportement qui aurait son intérêt même pour le cas des annulations. Ou alors ne pas accepter de réservation en liste d'attente tant que la liste principale n'est pas rempli de réservations fermes.

#48 - 19 mars 2023 13:16 - Benjamin Dauvergne

- Assigné à mis à Benjamin Dauvergne

#49 - 19 mars 2023 19:42 - Robot Gitea

- Statut changé de Nouveau à En cours

Benjamin Dauvergne (bdauvergne) a ouvert une pull request sur Gitea concernant cette demande :

- URL : <https://git.entrouvert.org/entrouvert/chrono/pulls/58>
- Titre : WIP: "préblocage" d'une réservation (#17685)
- Modifications : <https://git.entrouvert.org/entrouvert/chrono/pulls/58/files>

#50 - 19 mars 2023 19:43 - Benjamin Dauvergne

J'ai rebasé mon code et je suis reparti en partie du concept de Thomas, Lease est désormais un modèle fils de Booking ça donne le support sur les rendez-vous et les événements. Je dois rajouter encore des tests sur les événements et faire le tour des nouvelles APIs pour voir si ça doit impacter ou pas leur fonctionnement.

#51 - 24 mars 2023 23:00 - Benjamin Dauvergne

- Statut changé de En cours à Solution proposée

#52 - 24 avril 2023 16:58 - Olivier Renard

Je me permets une relance sur ce besoin.

#53 - 02 mai 2023 14:24 - Benjamin Dauvergne

Voilà c'est re-vert.

#57 - 22 août 2023 10:21 - Benjamin Dauvergne

- Statut changé de Solution proposée à En cours

#59 - 22 août 2023 12:43 - Benjamin Dauvergne

- Lié à Development #80489: "préblocage" d'une réservation ajouté

#60 - 22 août 2023 12:45 - Robot Gitea

Benjamin Dauvergne (bdauvergne) a fermé une pull request sur Gitea concernant cette demande.

#61 - 22 août 2023 13:52 - Benjamin Dauvergne

- Statut changé de En cours à Fermé

Fichiers

0001-api-handle-lock-on-fillslots-and-datetime-17685.patch	11,1 ko	05 juin 2018	Thomas Noël
0001-api-add-lock_code-parameter-to-fillslot-and-datetime.patch	31,5 ko	16 mars 2021	Benjamin Dauvergne
0001-api-add-lock_code-parameter-to-fillslot-and-datetime.patch	30,9 ko	18 mai 2021	Benjamin Dauvergne
0003-to-fixup-rename-lease-migration.patch	635 octets	03 juin 2021	Benjamin Dauvergne
0004-to-fixup-adapt-dependency-of-lease-migration.patch	823 octets	03 juin 2021	Benjamin Dauvergne
0002-to-fixup-rename-sql-snippets.patch	1,52 ko	03 juin 2021	Benjamin Dauvergne
0001-api-add-lock_code-parameter-to-fillslot-and-datetime.patch	31 ko	03 juin 2021	Benjamin Dauvergne
datasource-agenda.wcs	341 octets	03 juin 2021	Benjamin Dauvergne
wscall-lock_rdv.wcs	499 octets	03 juin 2021	Benjamin Dauvergne
workflow-rdv.wcs	9,05 ko	03 juin 2021	Benjamin Dauvergne
form-test-agenda-lock.wcs	2,3 ko	03 juin 2021	Benjamin Dauvergne
0007-to-fixup-remove-useless-comment.patch	1001 octets	08 juin 2021	Benjamin Dauvergne
0003-to-fixup-rename-lease-migration.patch	635 octets	08 juin 2021	Benjamin Dauvergne
0004-to-fixup-adapt-dependency-of-lease-migration.patch	823 octets	08 juin 2021	Benjamin Dauvergne
0002-to-fixup-rename-sql-snippets.patch	1,52 ko	08 juin 2021	Benjamin Dauvergne
0009-to-fixup-let-DRF-handle-empty-lock_code-errors.patch	1,48 ko	08 juin 2021	Benjamin Dauvergne
0008-to-fixup-removed-unused-Lease.as_interval.patch	697 octets	08 juin 2021	Benjamin Dauvergne
0001-api-add-lock_code-parameter-to-fillslot-and-datetime.patch	31 ko	08 juin 2021	Benjamin Dauvergne
0005-to-fixup-rename-migration-from-91-to-0091.patch	645 octets	08 juin 2021	Benjamin Dauvergne
0006-to-fixup-remove-blank-line.patch	725 octets	08 juin 2021	Benjamin Dauvergne