

## Chrono - Development #80489

### "préblocage" d'une réservation

22 août 2023 12:42 - Benjamin Dauvergne

<b>Statut:</b>	Fermé	<b>Début:</b>	22 août 2023
<b>Priorité:</b>	Normal	<b>Echéance:</b>	25 octobre 2023
<b>Assigné à:</b>	Benjamin Dauvergne	<b>% réalisé:</b>	0%
<b>Catégorie:</b>		<b>Temps estimé:</b>	0:00 heure
<b>Version cible:</b>		<b>Planning:</b>	Non
<b>Patch proposed:</b>	Non		

**Description**

Le préblocage a lieu via le passage d'un paramètre "lock\_code" contenant un chaîne quelconque non vide, les espaces en fin et début sont retirés pour éviter les bugs idiots de template.

Ce paramètre est disponible sur les vues datetimes et fillslot pour les rendez-vous et les évènements.

Lors d'une réservation (fillslot) avec lock\_code est créé un modèle Lease plus des modèles Event/Booking habituels. Le Lease est en relation 1-1 avec le Booking créé. Le Lease dispose d'une date d'expiration (contrôlé via le setting CHRONO\_LOCK\_DEFAULT\_DURATION mais par ailleurs contrôlable via un autre paramètre lock\_duration durant le fill\_slot) et d'un champ lock\_code qui reprendre la valeur passée.

lock\_duration est par ailleurs limité par les settings CHRONO\_LOCK\_MIN\_DURATION/CHRONO\_LOCK\_MAX\_DURATION.

```
chrono/settings.py:CHRONO_LOCK_MIN_DURATION = 5 * 60
chrono/settings.py:CHRONO_LOCK_DEFAULT_DURATION = 10 * 60
chrono/settings.py:CHRONO_LOCK_MAX_DURATION = 20 * 60
```

Régulièrement et sur différentes vues (datetimes et fillslot) sont appelés les fonctions clean\_meeting\_events\_with\_expired\_lease()/clean\_bookings\_with\_expired\_lease qui suppriment les Booking (et Event dans le cas des rendez-vous) dont l'objet Lease a expiré, ça évite d'avoir une tâche de fond pour faire cela, c'est fait au fil de l'eau et notamment lors des appels fillslot/datetimes, on a donc une vue toujours à jour des créneaux/évènements disponibles.

Dans w.c.s:  
L'appel avec lock\_code doit être en sortie de page où le choix du rendez-vous est fait en utilisant les condition de sortie de page. Cet appel doit être en définissant un appel de web-service. Ex.:

Définition du web-service :

```
Slug: rdv_fillslot_preblocage
Méthode: POST
URL: {{ form_var_rendez_vous_fillslot_url }}
Paramètre POST:
    lock_code={{ session_hash_id }}
```

Condition de sortie de page:

```
webservice.rdv_fill_slot_preblocage.err == 0
```

Si cet appel fillslot a réussi alors le rendez-vous/inscription est pré-bloqué, et on peut passer à la page suivante.cet appel peut-être sur toutes les sorties des pages suivantes il est idempotent (si le lease existe avec le même lock\_code il, le rendez-vous/l'inscription est supprimée et recréé à chaque fois, voir dans les blocs transaction.atomic() où se passent la réservation effective). À chaque appel fillslot, l'expiration du lease est prolongée, si l'appel est bien fait en sortie de la page finale ça évite que le fillslot définitif dans le workflow n'échoue.

Pour valider définitivement le rendez-vous/l'inscription, on ajoute à l'appel fillslot (celui qui est typiquement fait en début de workflow après réception de la demande) le lock\_code et un autre paramètre "confirm\_after\_lock" à la valeur true; la présence de ce paramètre permettra le fillslot comme avant mais ne recréera pas le Lease, on se retrouve avec un rendez-vous/une inscription classique.

Appel webservice de réservation définitive:

```
Méthode: POST
URL: {{ form_var_rendez_vous_fillslot_url }}
Paramètre POST:
```

```
lock_code={{ session_hash_id }}
confirm_after_lock=true
```

Autre commentaire:

Des bidouilles sont faites pour les endpoints datetimes:

- pour les évènements au niveau d'accessor de Event et de api/views:get\_events\_from\_slots pour avoir une vue des places disponibles des évènements excluant ou pas les inscriptions pré-bloquées
- pour les rendez-vous dans Agenda.get\_all\_slots() (exclusion de certains évènements en fonction de la présence du paramètre lock\_code, sur les agendas concernés, sur les ressources et sur les évènements agrégés en fonction du user\_external\_id)

Coté w.c.s on peut utiliser les sources de données datetimes automatiques mais elle ne passent pas de paramètre lock\_code et donc l'expérience n'est pas optimale, par exemple si on revient sur la page de sélection du créneau le slot en cours de préblocage ne sera pas disponible. Pour pallier à cela il faut créer une source de donnée manuelle qui passera le paramètre lock\_code={{ session\_hash\_id }}.

#### Demandes liées:

Lié à Chrono - Development #17685: "pré blocage" d'une réservation	Fermé	18 juillet 2017
Lié à Chrono - Development #82774: affichage en backoffice d'un pré blocage	Fermé	24 octobre 2023
Lié à Chrono - Support #83635: Pré blocage - rdv pré-réservé hier non libéré a...	Fermé	17 novembre 2023
Lié à Chrono - Support #83636: Pré blocage - condition sortie de page échoue e...	Fermé	17 novembre 2023
Lié à Chrono - Support #83643: Pré blocage - créneau pré-réservé est libéré qu...	Fermé	17 novembre 2023

## Historique

### #1 - 22 août 2023 12:43 - Benjamin Dauvergne

- Lié à Development #17685: "pré blocage" d'une réservation ajouté

### #2 - 22 août 2023 12:44 - Benjamin Dauvergne

- Tags mis à simplification agenda

### #3 - 22 août 2023 12:47 - Robot Gitea

- Statut changé de Nouveau à Solution proposée

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

- URL : <https://git.entrouvert.org/entrouvert/chrono/pulls/135>
- Titre : Pré blocage d'une réservation (#80489)
- Modifications : <https://git.entrouvert.org/entrouvert/chrono/pulls/135/files>

### #4 - 22 août 2023 13:45 - Benjamin Dauvergne

- Description mis à jour

### #5 - 24 août 2023 15:19 - Valentin Deniaud

Benjamin Dauvergne a écrit :

Ce paramètre est disponible sur les vues datetimes et fillslot pour les rendez-vous et les évènements.

Ça aurait été chouette de séparer les deux genre 1/ ajout pour les agenda rdv 2/ ajout pour les évènements, avec éventuellement le 2/ dans un second temps et un second ticket. Avec un patch/sujet aussi gros la relecture risque d'être soit tortueuse, soit bâclée.

Je vais tenter de mettre ici les remarques de fond et de garder gitea pour les remarques de forme, où le commentaire ligne à ligne a du sens.

un autre paramètre lock\_duration durant le fill\_slot

lock\_duration est par ailleurs limité par les settings CHRONO\_LOCK\_MIN\_DURATION/CHRONO\_LOCK\_MAX\_DURATION.

Pour moi c'est à retirer, ça sera remis si quelqu'un le demande (jamais).

Régulièrement et sur différentes vues (datetimes et fillslot) sont appelés les fonctions

clean\_meeting\_events\_with\_expired\_lease()/clean\_bookings\_with\_expired\_lease qui suppriment les Booking (et Event dans le cas des rendez-vous) dont l'objet Lease a expiré, ça évite d'avoir une tâche de fond pour faire cela

Je ne vois que des avantages à la tâche de fond, le jour où il y a un bug avec ton approche on doit grepper les lignes qui nettoient les locks, en

trouver 10, vérifier tous ces chemins de code pour un éventuel oubli.

En plus ça ajoute une requête inutile dans 99% des appels à chrono sur nos instances.

Donc pour moi un coup de cron toutes les [2, 5] minutes c'est très bien, en plus la durée du lock d'un créneau est hyper variable comme tu le notes puisqu'elle dépend du nombre de pages du formulaires. Personne ne remarquera si les créneaux sont lockés 12 minutes et pas 10.

#### #6 - 27 août 2023 10:54 - Benjamin Dauvergne

- Statut changé de Solution proposée à En cours

#### #7 - 30 août 2023 18:07 - Benjamin Dauvergne

Valentin Deniaud a écrit :

Benjamin Dauvergne a écrit :

Ce paramètre est disponible sur les vues datetimes et fillslot pour les rendez-vous et les évènements.

Ça aurait été chouette de séparer les deux genre 1/ ajout pour les agenda rdv 2/ ajout pour les évènements, avec éventuellement le 2/ dans un second temps et un second ticket. Avec un patch/sujet aussi gros la relecture risque d'être soit tortueuse, soit bâclée.

Ok je vais faire ça.

un autre paramètre lock\_duration durant le fill\_slot  
lock\_duration est par ailleurs limité par les settings CHRONO\_LOCK\_MIN\_DURATION/CHRONO\_LOCK\_MAX\_DURATION.

Pour moi c'est à retirer, ça sera remis si quelqu'un le demande (jamais).

Ok, c'est retiré.

Régulièrement et sur différentes vues (datetimes et fillslot) sont appelés les fonctions clean\_meeting\_events\_with\_expired\_lease()/clean\_bookings\_with\_expired\_lease qui suppriment les Booking (et Event dans le cas des rendez-vous) dont l'objet Lease a expiré, ça évite d'avoir une tâche de fond pour faire cela

Je ne vois que des avantages à la tâche de fond, le jour où il y a un bug avec ton approche on doit grepper les lignes qui nettoient les locks, en trouver 10, vérifier tous ces chemins de code pour un éventuel oubli.

En plus ça ajoute une requête inutile dans 99% des appels à chrono sur nos instances.

Ok, transformé en commande lancé via cron/uwsgi.

#### #8 - 31 août 2023 18:25 - Benjamin Dauvergne

- Statut changé de En cours à Solution proposée

#### #12 - 12 septembre 2023 10:54 - Anaïs Ecuivillon

Financement proposé au club via : <https://publik.tracim.fr/ui/workspaces/1/contents/html-document/2749>

#### #13 - 20 septembre 2023 17:25 - Robot Gitea

- Statut changé de Solution proposée à Solution validée

Valentin Deniaud (vdeniaud) a approuvé une pull request sur Gitea concernant cette demande :

- URL : <https://git.entrouvert.org/entrouvert/chrono/pulls/135>

#### #16 - 28 septembre 2023 16:40 - Benjamin Dauvergne

- Echéance mis à 25 octobre 2023

#### #17 - 17 octobre 2023 12:42 - Anaïs Ecuivillon

Le financement est bouclé.

Ce développement est chiffré à 16 000 € HT dont 6 000 € HT par la Mairie de Villejuif, 5 000 € HT par le Conseil départemental de la Haute-Garonne, 3 000 € HT par la ville de Nanterre et 2 000 € HT par la Ville de Lille.

Est-ce que je peux annoncer une livraison ~~début~~ fin novembre ? => ok

**#18 - 24 octobre 2023 14:38 - Frédéric Péters**

- Lié à *Development #82774: affichage en backoffice d'un préblocage ajouté*

**#19 - 16 novembre 2023 10:53 - Robot Gitea**

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

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

- URL : <https://git.entrouvert.org/entrouvert/chrono/pulls/135>
- Titre : Préblocage d'une réservation (#80489)
- Modifications : <https://git.entrouvert.org/entrouvert/chrono/pulls/135/files>

**#20 - 16 novembre 2023 12:15 - Transition automatique**

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

**#21 - 17 novembre 2023 08:12 - Anaïs Ecuivillon**

- Lié à *Support #83635: Préblocage - rdv pré-réservé hier non libéré après 12h ajouté*

**#22 - 17 novembre 2023 08:17 - Anaïs Ecuivillon**

- Lié à *Support #83636: Préblocage - condition sortie de page échoue en page 2 ajouté*

**#23 - 17 novembre 2023 09:11 - Anaïs Ecuivillon**

- Lié à *Support #83643: Préblocage - créneau pré-réservé est libéré quand j'effectue une nouvelle demande ajouté*

**#24 - 21 janvier 2024 04:42 - Transition automatique**

Automatic expiration