Chrono - Bug #44644

Crash sur /api/agenda/foo/meetings/bar/datetimes/ si des événements se chevauchent pour un même guichet

30 juin 2020 17:08 - Emmanuel Cazenave

Statut:	Fermé	**Début:**	30 juin 2020
Priorité:	Normal	**Echéance:**	
Assigné à:	Emmanuel Cazenave	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	Non

Description

Ce qui peut se produire (peut-être concurrence d'accès).

A l'arrivée ça fait :

```
Traceback (most recent call last):
  File "/usr/lib/chrono/manage.py", line 10, in <module>
    execute_from_command_line(sys.argv)
  File "/usr/lib/python3/dist-packages/django/core/management/__init__.py", line 364, in execute_f
rom_command_line
    utility.execute()
  File "/usr/lib/python3/dist-packages/django/core/management/__init__.py", line 356, in execute
    self.fetch_command(subcommand).run_from_argv(self.argv)
  File "/usr/lib/python3/dist-packages/hobo/multitenant/management/commands/tenant_command.py", li
ne 140, in run_from_argv
    klass.run_from_argv(args)
  File "/usr/lib/python3/dist-packages/django/core/management/base.py", line 283, in run_from_argv
    self.execute(*args, **cmd_options)
  File "/usr/lib/python3/dist-packages/django/core/management/base.py", line 330, in execute
    output = self.handle(*args, **options)
  File "/usr/lib/python3/dist-packages/hobo/multitenant/management/commands/runscript.py", line 34
, in handle
    runpy.run_module(module_name)
  File "/usr/lib/python3.5/runpy.py", line 208, in run_module
    return _run_code(code, {}, init_globals, run_name, mod_spec)
  File "/usr/lib/python3.5/runpy.py", line 85, in _run_code
    exec(code, run_globals)
  File "/tmp/t.py", line 24, in <module>
    for x in generator_of_unique_slots:
  File "/tmp/t.py", line 14, in unique_slots
    all_slots = list(get_all_slots(agenda, meeting_type, unique=True))
  File "/usr/lib/python3/dist-packages/chrono/api/views.py", line 163, in get_all_slots
    for desk_id, values in itertools.groupby(booked_events, lambda be: be[0])
  File "/usr/lib/python3/dist-packages/chrono/api/views.py", line 163, in <genexpr>
    for desk_id, values in itertools.groupby(booked_events, lambda be: be[0])
  File "/usr/lib/python3/dist-packages/chrono/interval.py", line 72, in from_ordered
    return cls(iterable, already_sorted=True)
  File "/usr/lib/python3/dist-packages/chrono/interval.py", line 57, in __init__
    raise ValueError('not well ordered: ! %s <= %s' % ((last_begin, last_end), (begin, end)))
```

Demandes liées:

Lié à Chrono - Bug #44676: rendez-vous : empecher au niveau base la création ...	**Fermé**	**01 juillet 2020**

Révisions associées

Révision 4dc54814 - 02 juillet 2020 06:22 - Emmanuel Cazenave

api: support overlapping events (#44644)

Révision 53a15007 - 02 juillet 2020 06:22 - Emmanuel Cazenave

api: support overlapping events (#44644)

Historique

#2 - 30 juin 2020 17:09 - Emmanuel Cazenave

- Statut changé de Nouveau à En cours

- Assigné à mis à Emmanuel Cazenave

#3 - 30 juin 2020 18:33 - Emmanuel Cazenave

- Fichier 0001-api-support-overlapping-events-44644.patch ajouté

- Statut changé de En cours à Solution proposée

- Patch proposed changé de Non à Oui

Le test reproduit bien la trace.

Le fix est un tir dans le noir, je comprends pas vraiment ce que fait IntervalSet.

#4 - 30 juin 2020 18:33 - Emmanuel Cazenave

Me suis trompé dans le nom de la branche : https://jenkins.entrouvert.org/job/chrono-wip/job/wip%252F44617-concurrent-events/.

#5 - 30 juin 2020 19:00 - Lauréline Guérin

Je pense que le fix serait plutôt:

```
diff --git a/chrono/api/views.py b/chrono/api/views.py
index 2f0bc6d..284ea1d 100644
--- a/chrono/api/views.py
+++ b/chrono/api/views.py
@@ -169,7 +169,7 @@ def get_all_slots(base_agenda, meeting_type, resources=None, unique=False):
             )
             .exclude(booking__cancellation_datetime__isnull=False)
             # ordering is important for the later groupby, it works like sort | uniq
-            .order_by('desk_id', 'start_datetime')
+            .order_by('desk_id', 'start_datetime', 'meeting_type__duration')
             .values_list('desk_id', 'start_datetime', 'meeting_type__duration')
         )
         # compute exclusion set by desk from all bookings, using
```

Pour être sûr que start + duration (= end) est bien dans le bon ordre

J'étais tombée sur ce cas pour les réservations de ressources, et j'ai pas pensé à vérifier l'existant

#6 - 30 juin 2020 19:01 - Lauréline Guérin

- Statut changé de Solution proposée à En cours

#7 - 30 juin 2020 20:14 - Thomas Noël

Sauf si j'ai rien compris, tout ça me semble contourner le vrai problème, à savoir que deux rendez-vous ont été pris sur le même guichet au même moment, parce qu'on n'a aucune contrainte SQL sur cela et que deux transactions vont être acceptées tranquillement.

#8 - 30 juin 2020 21:11 - Benjamin Dauvergne

Thomas Noël a écrit :

> Sauf si j'ai rien compris, tout ça me semble contourner le vrai problème, à savoir que deux rendez-vous ont été pris sur le même guichet au même moment, parce qu'on n'a aucune contrainte SQL sur cela et que deux transactions vont être acceptées tranquillement.

Oui, +1 pour Thomas qui a bien suivi en cours de SQL. Je dirai unicité sur (desk_id, start_datetime) mais faudra un index partiel avec une clause WHERE pour que ça ne s'applique que si cancellation_datetime__isnull=True; à l'endroit de la prise de rendez-vous il faudrait utiliser get-or-create sur (desk_id, start_datetime) on peut ajouter agenda_id mais il découle de desk_id normalement (le schéma n'est pas en forme normale je ne sais laquelle).

PS1: dans un autre ticket, la proposition de Laureline suffit ici (qui est la même que la mienne)
PS2: mais d'abord il faut vérifier si le cas ne s'est pas produit énormément ailleurs avant, sinon la migration ne passera jamais.

#9 - 01 juillet 2020 00:54 - Thomas Noël

Benjamin Dauvergne a écrit :

> Je dirai unicité sur (desk_id, start_datetime)

Ca suffira pas totalement (même si c'est déjà ça), on pourra toujours ajouter une résa 13h15-13h30 alors que 13h00-14h00 vient d'être reservé dans la transaction précédente. Je pense que c'est le genre de truc compliqué qu'on ne règle qu'à coup de procédures stockées (et on ne va pas aller vers ça, je pense). Et oui, tout ça est pour un autre ticket.

Mais je ne suis pas super chaud pour appliquer le patch proposé ici : il cache le problème sous le tapis (on ne devrait pas avoir besoin de ce patch si les écritures étaient bien contraintes).

#10 - 01 juillet 2020 08:53 - Benjamin Dauvergne

Thomas Noël a écrit :

> Benjamin Dauvergne a écrit :
>
> > Je dirai unicité sur (desk_id, start_datetime)
>
> Ca suffira pas totalement (même si c'est déjà ça), on pourra toujours ajouter une résa 13h15-13h30 alors que 13h00-14h00 vient d'être reservé dans la transaction précédente. Je pense que c'est le genre de truc compliqué qu'on ne règle qu'à coup de procédures stockées (et on ne va pas aller vers ça, je pense). Et oui, tout ça est pour un autre ticket.

Non il y a aussi des index d'exclusion en Postgresql, mais ça demande que end_datetime soit matérialisé dans la table on ne peut pas demander à postgres de le calculer à la volée depuis meeting_type__duration comme on le fait dans le code.

https://www.postgresql.org/docs/9.4/sql-createtable.html#SQL-CREATETABLE-EXCLUDE

Ça donnerait un truc comme ça :

```
EXCLUDE USING GIST (desk_id WITH =, tstzrange(start_datetime, end_datetime) WITH &&)
```

> Mais je ne suis pas super chaud pour appliquer le patch proposé ici : il cache le problème sous le tapis (on ne devrait pas avoir besoin de ce patch si les écritures étaient bien contraintes).

En attendant on a aucun moyen d'empêcher ce bug, il faut appliquer ce patch sinon quand le bug arrive ça empêche toute réservation, pour l'instant on peut vivre avec quelques rdv qui se chevauchent plutôt qu'avec des applications qui ne marchent pas.

#11 - 01 juillet 2020 10:38 - Thomas Noël

Benjamin Dauvergne a écrit :

> Mais je ne suis pas super chaud pour appliquer le patch proposé ici : il cache le problème sous le tapis (on ne devrait pas avoir besoin de ce patch si les écritures étaient bien contraintes).
>
> En attendant on a aucun moyen d'empêcher ce bug, il faut appliquer ce patch sinon quand le bug arrive ça empêche toute réservation, pour l'instant on peut vivre avec quelques rdv qui se chevauchent plutôt qu'avec des applications qui ne marchent pas.

Ok, allons-y alors. Un mix patch de Lauréline + test d'Emmanuel comme ça tout le monde est content ?

#12 - 01 juillet 2020 12:47 - Emmanuel Cazenave

- Fichier 0001-api-support-overlapping-events-44644.patch ajouté

- Statut changé de En cours à Solution proposée

Lauréline Guerin a écrit :

> Je pense que le fix serait plutôt:

Voilà.

#13 - 01 juillet 2020 14:25 - Thomas Noël

- Statut changé de Solution proposée à Solution validée

#14 - 01 juillet 2020 14:35 - Thomas Noël

- Lié à Bug #44676: rendez-vous : empecher au niveau base la création de conflit sur des rdv ajouté

#15 - 02 juillet 2020 06:23 - Frédéric Péters

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

Poussé, ainsi que vers une branche hotfix.

```
commit 53a150078b81d5dd5789a66eb7a1e4249bd3d100
Author: Emmanuel Cazenave <ecazenave@entrouvert.com>
Date:   Tue Jun 30 18:13:59 2020 +0200

    api: support overlapping events (#44644)
```

#16 - 02 juillet 2020 11:16 - Frédéric Péters

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

Fichiers

0001-api-support-overlapping-events-44644.patch	2,64 ko	30 juin 2020	Emmanuel Cazenave
0001-api-support-overlapping-events-44644.patch	2,75 ko	01 juillet 2020	Emmanuel Cazenave