

Chrono - Bug #42142

requêtes SQL api datetimes

27 avril 2020 16:13 - Frédéric Péters

Statut:	Fermé	Début:	27 avril 2020
Priorité:	Normal	Echéance:	
Assigné à:	Frédéric Péters	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		

Description

(noté sur un agenda virtuel avec beaucoup d'agendas, mais ça ne fait que cumuler un truc existant avec un seul agenda)

Il y a une requête ici qui se trouve répétée pas mal :

```
+++ b/chrono/agendas/models.py
@@ -429,7 +429,8 @@ class TimePeriod(models.Model):
    def get_time_slots(self, min_datetime, max_datetime, meeting_type):
        meeting_duration = datetime.timedelta(minutes=meeting_type.duration)
        duration = datetime.timedelta(minutes=self.desk.agenda.get_base_meeting_duration()) # <
--
```

Et il y a d'autres trucs db, dans `get_all_slots()`, pas analysés, j'ai ça tapé à l'arrache juste pour éliminer les requêtes SQL de mon profil de performances, c'est à faire en réfléchissant :

```
- agendas = agenda.get_real_agendas()
+ agendas = agenda.get_real_agendas().select_related()
...
+ time_periods_by_agenda = {}
  for agenda in agendas:
-     open_slots[agenda] = defaultdict(lambda: Intervals())
-
+     open_slots[agenda.id] = defaultdict(lambda: Intervals())
+     time_periods_by_agenda[agenda.id] = []
+
+     for period in TimePeriod.objects.all().prefetch_related('desk__agenda'):
+         if not period.desk or period.desk.agenda_id not in time_periods_by_agenda:
+             continue
+         time_periods_by_agenda[period.desk.agenda_id].append(period)
+     base_agenda_excluded_timeperiods = base_agenda.excluded_timeperiods.all()
...
-     for raw_time_period in TimePeriod.objects.filter(desk__agenda=agenda):
+     for raw_time_period in time_periods_by_agenda[agenda.id]:
+         for time_period in raw_time_period.get_effective_timeperiods(
-             base_agenda.excluded_timeperiods.all()
-         ):
+             base_agenda_excluded_timeperiods):
+             duration = (
```

Révisions associées

Révision 9d0b76c8 - 01 mai 2020 09:29 - Frédéric Péters

api: reduce number of sql queries in datetimes API (#42142)

Historique

#1 - 27 avril 2020 18:12 - Frédéric Péters

- Assigné à mis à Frédéric Péters

#2 - 27 avril 2020 18:12 - Frédéric Péters

- Statut changé de Nouveau à En cours

#4 - 27 avril 2020 20:40 - Frédéric Péters

- Fichier 0003-api-reduce-number-of-sql-queries-in-datetimes-API-42.patch ajouté

- Statut changé de En cours à Solution proposée

- Patch proposed changé de Non à Oui

(à l'origine sur le test, 28 requêtes).

#5 - 27 avril 2020 21:42 - Frédéric Péters

En résultat chiffré, avec les trois tickets je passe de 56s à 20s (en exécution locale).

#6 - 28 avril 2020 15:26 - Emmanuel Cazenave

```
if not period.desk or period.desk.agenda_id not in time_periods_by_agenda:
    continue
time_periods_by_agenda[period.desk.agenda_id].append(period)
```

Je comprends pas d'où sort cet attribut 'agenda_id' sur un Desk ...

#7 - 28 avril 2020 15:31 - Frédéric Péters

```
class Desk(models.Model):
    agenda = models.ForeignKey(Agenda, on_delete=models.CASCADE)
```

→ agenda_id pour la pk.

#8 - 28 avril 2020 16:46 - Emmanuel Cazenave

Ok je connaissais pas ce pattern.

```
for period in TimePeriod.objects.all().prefetch_related('desk__agenda')
```

Il y aurait pas un filtrage à faire sur les TimePeriod qui concernent les agendas dont on s'occupe plutôt que choper toutes les TimePeriod de la DB ?

(désolé c'est au fil de l'eau mais j'ai toujours du mal avec cette zone de code)

#9 - 28 avril 2020 17:09 - Frédéric Péters

Oui je pense qu'on peut faire ça,

```
- for period in TimePeriod.objects.all().prefetch_related('desk__agenda'):
+ for period in TimePeriod.objects.filter(desk__agenda__in=agendas).prefetch_related('desk__agenda'):
```

Côté nombre de requêtes SQL ça fait pareil.

Côté code ça permet de réduire :

```
- for period in TimePeriod.objects.all().prefetch_related('desk__agenda'):
-     if not period.desk or period.desk.agenda_id not in time_periods_by_agenda:
-         continue
+ for period in TimePeriod.objects.filter(desk__agenda__in=agendas).prefetch_related('desk__agenda'):
+     time_periods_by_agenda[period.desk.agenda_id].append(period)
```

Côté perf, sur ma requête (simulation Metz), ça revient au même.

#10 - 28 avril 2020 17:09 - Frédéric Péters

- Fichier 0001-api-reduce-number-of-sql-queries-in-datetimes-API-42.patch ajouté

#11 - 28 avril 2020 18:16 - Emmanuel Cazenave

- Statut changé de Solution proposée à Solution validée

#12 - 28 avril 2020 18:24 - Benjamin Dauvergne

Il y a des cas aussi ou c'est plus simple de gérer soit même le prefetch :

```
agendas = {agenda.id: agenda...}
desks {desk.id: desk...}
for desk in desks.values():
```

```
desk.agenda = agendas[desk.agenda_id]
tp = {time_period.id: timep...}
for tp in tp.values():
    tp.desk = desks[tp.desk_id]
```

si on arrive à faire tous les prefetch en une seule requête tant mieux sinon faut passer par là.

#13 - 01 mai 2020 09:29 - Frédéric Péters

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

```
commit 9d0b76c82a3a0ac8b77613dba160070clf4426ad
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Mon Apr 27 18:52:43 2020 +0200
```

```
api: reduce number of sql queries in datetimes API (#42142)
```

#14 - 03 mai 2020 20:16 - Frédéric Péters

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

Fichiers

0003-api-reduce-number-of-sql-queries-in-datetimes-API-42.patch	9,12 ko	27 avril 2020	Frédéric Péters
0001-api-reduce-number-of-sql-queries-in-datetimes-API-42.patch	9,11 ko	28 avril 2020	Frédéric Péters