

Chrono - Bug #72475

Les périodes horaires uniques se masquent les unes les autres

15 décembre 2022 11:25 - Valentin Deniaud

Statut:	Fermé	Début:	15 décembre 2022
Priorité:	Normal	Echéance:	
Assigné à:	Valentin Deniaud	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
Si elles sont même jour de la semaine + même heure...			
Demandes liées:			
Lié à Chrono - Development #76571: Revoir la logique dans get_effective_timep...			Nouveau 13 avril 2023

Révisions associées

Révision 46b70f6d - 15 décembre 2022 15:04 - Valentin Deniaud

agendas: compare all WeekTime attributes by default (#72475)

Historique

#2 - 15 décembre 2022 11:27 - Valentin Deniaud

- Tracker changé de Development à Bug

#3 - 15 décembre 2022 15:32 - Valentin Deniaud

- Fichier 0001-agendas-compare-all-WeekTime-attributes-by-default-7.patch ajouté

- Tracker changé de Bug à Development

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Joli bug.

Ça se joue dans Agenda.get_effective_time_periods_meetings et l'utilisation de itertools.groupby :

```
for weektime_interval, time_periods in itertools.groupby(
    time_periods.prefetch_related('desk').order_by('weekday', 'start_time', 'end_time'),
    key=TimePeriod.as_weektime_interval,
)
```

Ici on ordonne le queryset « time_periods » en SQL.

Or dans la doc de Python, <https://docs.python.org/3/library/itertools.html#itertools.groupby>

Generally, the iterable needs to already be sorted on the same key function.

Donc on voit ici qu'on est à moitié dans les clous, on n'a pas ordonné avec la même fonction mais avec du SQL qu'on espère équivalent.

Sauf que depuis que time_periods a gagné un champ weekday_indexes (puis date), ce n'est plus équivalent.

Le péché originel qui fait que ce n'est plus équivalent c'est [#65849](#).

Ce patch c'est moi qui me suit aperçu que la comparaison entre période d'exclusion et période horaire avec numéro de semaine se passait mal, et qui me suit dit « ok on va pas comparer ».

Mais si on ne compare plus on tombe sur le bug présent, on regroupe des périodes qui en fait ne sont pas les mêmes (pas le même weekday_indexes, puis pas la même date). Mais ce regroupement fautif est rare, masqué par l'utilisation borderline de itertools.groupby.

(en vrai j'ai peur de toucher à cette ligne avec itertools donc je la laisse tranquille, elle n'est pas en soit fautive elle rend juste l'analyse plus compliquée)

Tout ça pour dire, le fix c'est de revenir sur [#65849](#) et d'inverser la logique : plutôt que de ne jamais comparer, toujours tout comparer et court-circuiter la comparaison au seul endroit où ça foire.

(et là dessus on est pas aidés par la généralité du code dans interval.py, le code pourrait être nettement moins générique et beaucoup plus simple à suivre/faire évoluer)

#4 - 15 décembre 2022 15:46 - Emmanuel Cazenave

- Statut changé de Solution proposée à Solution validée

#5 - 15 décembre 2022 15:47 - Valentin Deniaud

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

```
commit 46b70f6d0533044bd03919927c39abe1a0bbe10c
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date: Thu Dec 15 15:00:25 2022 +0100
```

```
agendas: compare all WeekTime attributes by default (#72475)
```

#6 - 15 décembre 2022 16:16 - Transition automatique

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

#7 - 19 février 2023 04:42 - Transition automatique

Automatic expiration

#8 - 13 avril 2023 06:54 - Benjamin Dauvergne

- Tracker changé de Development à Bug

#9 - 13 avril 2023 07:01 - Benjamin Dauvergne

- Lié à Development #76571: Revoir la logique dans `get_effective_timeperiods` avec l'introduction de `weekday_indexes` ajouté

Fichiers

0001-agendas-compare-all-WeekTime-attributes-by-default-7.patch 4,06 ko 15 décembre 2022

Valentin Deniaud