

## Chrono - Bug #52995

passer hh:mm à date\_start/date\_end sur /datetimes de rendez-vous n'est pas pris en compte

12 avril 2021 18:10 - Frédéric Péters

<b>Statut:</b>	Fermé	<b>Début:</b>	12 avril 2021
<b>Priorité:</b>	Normal	<b>Echéance:</b>	
<b>Assigné à:</b>	Lauréline Guérin	<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>	Oui		

### Description

Malgré [#51986](#), seules les dates sont prises en compte pour les rendez-vous. (via [#52992](#)).

Il me semble que ça pourrait venir de `get_time_slots` qui perd l'heure ici :

```
# make sure datetime in local timezone, it's ABSOLUTELY necessary
# to have stable event ids in the API.
real_min_datetime = real_min_datetime.replace(
    hour=12
) # so aware datetime will be int the dst of the day
```

(sauf que ça n'expliquerait pas pourquoi ça foire aussi sur `date_end`).

De manière très naïve, je

```
--- a/chrono/api/views.py
+++ b/chrono/api/views.py
@@ -310,6 +310,8 @@ def get_all_slots(
    )
        if unique and unique_booked.get(timestamp) is booked:
            continue
+
+
        if start_datetime < base_min_datetime or start_datetime >= base_max_datetime:
            continue
        unique_booked[timestamp] = booked
        yield TimeSlot(
            start_datetime=start_datetime, end_datetime=end_datetime, desk=desk, full
=booked
```

mais ça me semble plutôt rattraper les choses que gérer convenablement l'affaire.

### Révisions associées

Révision [0af9b0b6](#) - 15 avril 2021 15:36 - Lauréline Guérin

api: fix date\_start & meetings/datetimes ([#52995](#))

### Historique

**#2 - 13 avril 2021 15:40 - Lauréline Guérin**

- Assigné à mis à Lauréline Guérin

**#3 - 15 avril 2021 09:58 - Lauréline Guérin**

- Fichier `0001-api-fix-date_start-meetings-datetimes-52995.patch` ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

dans `get_time_slots`, on prend `min_datetime`, on y ajoute éventuellement 7 jours si on déborde sur le jour suivant, puis on le déplace à midi, on y applique l'heure de la `timeperiod`, et on déroule les créneaux possibles jusqu'à ne pas dépasser `max_datetime`.

=> si `min_datetime` est à midi, et la `timeperiod` démarre à 10H, on se retrouve avec des slots qui démarrent avant `min_datetime`

il manque juste un truc: démarrer à `max(event_datetime, min_datetime)`

mes tests étaient mal écrits du coup, c'est corrigé.

#### #4 - 15 avril 2021 15:22 - Thomas Noël

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

Je valide car l'explication du max() et les tests me vont bien, même si j'avoue être un peu loin de comprendre l'algo dans ses détails exacts.

#### #5 - 15 avril 2021 16:07 - Lauréline Guérin

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

```
commit 0af9b0b63dcf9df4dba0916b8296721030ea426a
Author: Lauréline Guérin <zebuline@entrouvert.com>
Date: Thu Apr 15 09:55:11 2021 +0200
```

```
api: fix date_start & meetings/datetimes (#52995)
```

#### #6 - 15 avril 2021 20:17 - Frédéric Péters

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

### Fichiers

---

0001-api-fix-date_start-meetings-datetimes-52995.patch	3,46 ko	15 avril 2021	Lauréline Guérin
--	---------	---------------	------------------