Chrono - Development #15729

Gestion multi-guichets

03 avril 2017 14:26 - Frédéric Péters

Statut:	Fermé	**Début:**	03 avril 2017
Priorité:	Normal	**Echéance:**	28 juillet 2017
Assigné à:	Josué Kouka	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	

Description

Dans une plage horaire (TimePeriod) permettre de définir le nombre de places disponibles.

Ou bien gérer un identifiant ("guichet") supplémentaire par période horaire, et permettre de définir plusieurs fois la même tranche horaire ? (peut-être ça qui permettra de construire la gestion la plus facile).

```
Périodes horaires
=================

Guichet 1                       Guichet 2                   Guichet 3
---------                       ---------                   ---------

lundi / 10:00 → 17:00           lundi / 09:00 → 12:00
mardi / 10:00 → 17:00

[ajouter]                       [ajouter]                   [ajouter]
```

Cette option demande d'étendre l'objet Event existant pour permettre de garder une association rendez-vous→guichet. Et modifier l'API /datetimes pour "merger" les rendez-vous possibles aux mêmes heures (on ne laisse pas en frontoffice la possibilité de choisir le guichet). Et modifier l'API /fillslot pour enregistrer l'événement dans un des guichets libres. Etc. sans doute.

Demandes liées:

Lié à Chrono - Development #18414: Ajustements ui/ux au multiguichet	**Fermé**	**03 septembre 2017**

Révisions associées

Révision f851de07 - 04 septembre 2017 11:00 - Josué Kouka

add multiple desk management (#15729)

Révision c2365420 - 04 septembre 2017 12:41 - Josué Kouka

misc: add intervaltree to requirements.txt to fix test (#15729)

Historique

#2 - 07 juin 2017 10:02 - Josué Kouka

- Assigné à mis à Josué Kouka

#3 - 07 juin 2017 10:03 - Josué Kouka

- Echéance mis à 28 juillet 2017

- Assigné à Josué Kouka supprimé

#4 - 09 juin 2017 11:33 - Brice Mallet

- Assigné à mis à Josué Kouka

#5 - 24 juillet 2017 10:06 - Josué Kouka

- Statut changé de Nouveau à En cours

#6 - 25 juillet 2017 17:23 - Josué Kouka

- Fichier 0001-add-multiple-windows-feature-for-meeting-api-15729.patch ajouté

- Patch proposed changé de Non à Oui

Je colle un patch par rapport à ce que j'ai compris du besoin.
La solution que j'ai choisi est d'ajouter des disponibilités aux différentes TimePeriod et de lier à une réservation lors de la création de l'évènement avec places disponnibles = nombre de guichets.
Je ne sais pas si c'est la meilleure solution, mais c'est un patch pour au moins entamer une discussion.

#7 - 25 juillet 2017 17:51 - Frédéric Péters

Il n'y avait pas besoin de patch pour annoncer que tu prenais l'option "Dans une plage horaire (TimePeriod) permettre de définir le nombre de places disponibles.".

Au moment de l'écrire déjà, et par la suite et maintenant toujours, je continue à penser que la seconde approche est meilleure; qu'on gère du multi-guichet en ajoutant un réel objet guichet, pas en ajoutant un attribut "nombre de places" à une période horaire.

J'ai l'impression que ça se sent en allant un tout petit peu plus loin dans le besoin : ça pourra faciliter la gestion (cf ascii art dans la description), ça permettra par la suite de fournir à l'usager les indications pour se rendre au guichet en question (si jamais les guichets n'étaient pas côte à côte); également d'avoir pour les agents une vue des rendez-vous à leur guichet propre; etc.

#8 - 27 juillet 2017 18:59 - Josué Kouka

Frédéric Péters a écrit :

> Il n'y avait pas besoin de patch pour annoncer que tu prenais l'option "Dans une plage horaire (TimePeriod) permettre de définir le nombre de places disponibles.".
>
> Au moment de l'écrire déjà, et par la suite et maintenant toujours, je continue à penser que la seconde approche est meilleure; qu'on gère du multi-guichet en ajoutant un réel objet guichet, pas en ajoutant un attribut "nombre de places" à une période horaire.
>
> J'ai l'impression que ça se sent en allant un tout petit peu plus loin dans le besoin : ça pourra faciliter la gestion (cf ascii art dans la description), ça permettra par la suite de fournir à l'usager les indications pour se rendre au guichet en question (si jamais les guichets n'étaient pas côte à côte); également d'avoir pour les agents une vue des rendez-vous à leur guichet propre; etc.

Ok.

Je viens de pousser une branche ou l'on peut voir l'avancement http://git.entrouvert.org/chrono.git/log/?h=wip/multi-guichet-15729.
Il manque la gestion de l'import/export.

Merci pour vos retours

#9 - 27 juillet 2017 19:39 - Frédéric Péters

> chrono/manager/templates/chrono/manager_agenda_view.html

Je ne mettrais pas cette emphase sur les guichets; les écrans seraient, d'abord comme aujourd'hui :

```
Périodes horaires
=================

lundi / 10:00 → 17:00
mardi / 10:00 → 17:00
```

Puis, ajout d'un guichet, l'information s'ajoute à la liste des périodes horaires, c'est presque le mockup de la description :

```
Périodes horaires
=================

Guichet 1                    Guichet 2
---------                    ---------

lundi / 10:00 → 17:00
mardi / 10:00 → 17:00
```

Pour suivre vraiment la description, il y aurait déplacement de "Nouvelle période horaire" en bas mais je laisserais ça de côté pour le moment.

Intègre les tests directement dans les commits auxquels ils se rapportent. (mais l'objectif est peut-être d'unir tout en un unique patch, je ne sais pas)

Erreurs à l'exécution des tests :

```
tests/test_api.py::test_datetimes_api_meetings_agenda[Europe/Brussels-mock_now0] FAILED
tests/test_api.py::test_datetimes_api_meetings_agenda[Europe/Brussels-mock_now1] FAILED
```

```
tests/test_api.py::test_datetimes_api_meetings_agenda[Europe/Brussels-mock_now2] FAILED
tests/test_api.py::test_datetimes_api_meetings_agenda[Asia/Kolkata-mock_now0] FAILED
tests/test_api.py::test_datetimes_api_meetings_agenda[Asia/Kolkata-mock_now1] FAILED
tests/test_api.py::test_datetimes_api_meetings_agenda[Asia/Kolkata-mock_now2] FAILED
tests/test_api.py::test_datetimes_api_meetings_agenda[Brazil/East-mock_now0] FAILED
tests/test_api.py::test_datetimes_api_meetings_agenda[Brazil/East-mock_now1] FAILED
tests/test_api.py::test_datetimes_api_meetings_agenda[Brazil/East-mock_now2] FAILED

tests/test_api.py::test_booking_cancellation_post_meeting_api[Europe/Brussels-mock_now0] FAILED
tests/test_api.py::test_booking_cancellation_post_meeting_api[Europe/Brussels-mock_now1] FAILED
tests/test_api.py::test_booking_cancellation_post_meeting_api[Europe/Brussels-mock_now2] FAILED
tests/test_api.py::test_booking_cancellation_post_meeting_api[Asia/Kolkata-mock_now0] FAILED
tests/test_api.py::test_booking_cancellation_post_meeting_api[Asia/Kolkata-mock_now1] FAILED
tests/test_api.py::test_booking_cancellation_post_meeting_api[Asia/Kolkata-mock_now2] FAILED
tests/test_api.py::test_booking_cancellation_post_meeting_api[Brazil/East-mock_now0] FAILED
tests/test_api.py::test_booking_cancellation_post_meeting_api[Brazil/East-mock_now1] FAILED
tests/test_api.py::test_booking_cancellation_post_meeting_api[Brazil/East-mock_now2] FAILED

tests/test_import_export.py::test_import_export[Europe/Brussels-mock_now0] FAILED
tests/test_import_export.py::test_import_export[Europe/Brussels-mock_now1] FAILED
tests/test_import_export.py::test_import_export[Europe/Brussels-mock_now2] FAILED
tests/test_import_export.py::test_import_export[Asia/Kolkata-mock_now0] FAILED
tests/test_import_export.py::test_import_export[Asia/Kolkata-mock_now1] FAILED
tests/test_import_export.py::test_import_export[Asia/Kolkata-mock_now2] FAILED
tests/test_import_export.py::test_import_export[Brazil/East-mock_now0] FAILED
tests/test_import_export.py::test_import_export[Brazil/East-mock_now1] FAILED
tests/test_import_export.py::test_import_export[Brazil/East-mock_now2] FAILED

tests/test_manager.py::test_add_event FAILED

tests/test_manager.py::test_add_event_as_manager FAILED
tests/test_manager.py::test_edit_event FAILED

tests/test_manager.py::test_edit_event_as_manager FAILED

tests/test_manager.py::test_meetings_agenda_add_time_period FAILED
```

#10 - 08 août 2017 18:59 - Frédéric Péters

(pense vraiment à ajouter une note dans le redmine quand tu pousses une nouvelle branche, idéalement en notant ce qui change et ce qui pourrait encore manquer).

```
self.event.desk = None
```

De loin il me semble que ça pourrait utile de conserver l'info sur le guichet assigné à une demande annulée.

```
+        # XXX: distinct on fields not supported by django sqlite backend
+        # so manually removing duplicates
```

Ne pas inclure de XXX si l'intention est juste de poser l'info en commentaire.

```
+            available_desk = Desk.get_available_desk(agenda, event)
+            event.desk = available_desk
+            event.save()
             event_pk = event.id
```

J'arrêterais avec une réponse "no more desk", plutôt qu'enregistrer l'inexistance d'un guichet disponible et dire "sold out" derrière; ça évitera aussi de mêler deux types de conditions différentes dans le bloc qui suit.

```
+
+.timeperiods {
+    position: relative;
+}
+
+.timeperiod {
+    float:left;
+    display:block;
+    margin:10px;
+}
```

Comment ne vois-tu pas dans le fichier que des tabulations sont utilisées, comment te retrouves-tu à taper un espace après les : et puis plus ?

```
+        <div class="timeperiod">
+            <h4>
+                <a href="{% if user_can_manage %}{% url 'chrono-manager-desk-edit' pk=desk.pk %}{% else %}#{%
```

```
 endif %}">
+                    {{ desk.label }}</a>
+            </h4>
```

Dans les mockups plus haut, quand il y a un seul guichet, on n'affiche pas son nom.

```
+                This desk doesn't have any time period yet.<a rel="popup" href="{{add_time_period_url}}">New
Time Period</a>
```

Et on n'a pas ce message, ni le "New Time Period" à cet endroit. J'écrivais « je laisserais ça de côté pour le moment » pour qu'on s'épargne un travail d'attention sur le rendu. Genre le lien collé au texte.

```
+    {% blocktrans %}
+    This agenda doesn't have any desk yet. Click on the "New Desk" button in
+    the top right of the page to add a first one.
+    {% endblocktrans %}
```

L'écran initial ici devrait être dans une situation où un guichet existe déjà, créé automatiquement.

Ça me fait donc noter que :

```
+class NewDeskForm(forms.ModelForm):
```

Il faudrait une valeur par défaut dans le champ "libellé" de la boite de dialogue, genre _('Desk #%s') % (len(current_desks)+1).

```
+def set_timeperiod_desk(apps, schema_editor):
+    TimePeriod = apps.get_model('agendas', 'TimePeriod')
+    Desk = apps.get_model('agendas', 'Desk')
+    for time_period in TimePeriod.objects.all():
+        desk, created = Desk.objects.get_or_create(
+            label='Guichet 1', slug='guichet-1', agenda=time_period.agenda)
+        time_period.desk = desk
+        time_period.save()
```

Manque un test unitaire pour couvrir cette migration.

Manquent aussi une converture pour le *unicode* de la classe Desk et pour la création de son slug si le premier slug trouvé n'est pas bon; n'hésite pas à exécuter les tests en activant le calcul de couverture de code chez toi. (--cov-report=html --cov=chrono/).

#11 - 22 août 2017 15:49 - Josué Kouka

J'ai mis à jour la branche

Frédéric Péters a écrit :

> (pense vraiment à ajouter une note dans le redmine quand tu pousses une nouvelle branche, idéalement en notant ce qui change et ce qui pourrait encore manquer).
>
> [...]
>
> De loin il me semble que ça pourrait utile de conserver l'info sur le guichet assigné à une demande annulée.
>
> [...]
>
> Ne pas inclure de XXX si l'intention est juste de poser l'info en commentaire.
>
> [...]
>
> J'arrêterais avec une réponse "no more desk", plutôt qu'enregistrer l'inexistance d'un guichet disponible et dire "sold out" derrière; ça évitera aussi de mêler deux types de conditions différentes dans le bloc qui suit.
>
> [...]
>
> Comment ne vois-tu pas dans le fichier que des tabulations sont utilisées, comment te retrouves-tu à taper un espace après les : et puis plus ?
>
> [...]

Corrigé

> Dans les mockups plus haut, quand il y a un seul guichet, on n'affiche pas son nom.
>
> [...]

Fait

> Et on n'a pas ce message, ni le "New Time Period" à cet endroit. J'écrivais « je laisserais ça de côté pour le moment » pour qu'on s'épargne un travail d'attention sur le rendu. Genre le lien collé au texte.
>
> [...]
>
> L'écran initial ici devrait être dans une situation où un guichet existe déjà, créé automatiquement.

Ajouté (j'imagine que modifier le label de ce guichet est inutile)

> Ça me fait donc noter que :
>
> [...]
>
> Il faudrait une valeur par défaut dans le champ "libellé" de la boite de dialogue, genre _('Desk #%s') % (len(current_desks)+1).
>
> [...]
>
> Manque un test unitaire pour couvrir cette migration.
>
> Manquent aussi une converture pour le *unicode* de la classe Desk et pour la création de son slug si le premier slug trouvé n'est pas bon; n'hésite pas à exécuter les tests en activant le calcul de couverture de code chez toi. (--cov-report=html --cov=chrono/).

Je rajoute d'ici peu et améliore la couverture.

#12 - 22 août 2017 16:42 - Josué Kouka

Josué Kouka a écrit :

> J'ai mis à jour la branche
>
> Frédéric Péters a écrit :
>
> > (pense vraiment à ajouter une note dans le redmine quand tu pousses une nouvelle branche, idéalement en notant ce qui change et ce qui pourrait encore manquer).
> >
> > [...]
> >
> > De loin il me semble que ça pourrait utile de conserver l'info sur le guichet assigné à une demande annulée.
> >
> > [...]
> >
> > Ne pas inclure de XXX si l'intention est juste de poser l'info en commentaire.
> >
> > [...]
> >
> > J'arrêterais avec une réponse "no more desk", plutôt qu'enregistrer l'inexistance d'un guichet disponible et dire "sold out" derrière; ça évitera aussi de mêler deux types de conditions différentes dans le bloc qui suit.
> >
> > [...]
> >
> > Comment ne vois-tu pas dans le fichier que des tabulations sont utilisées, comment te retrouves-tu à taper un espace après les : et puis plus ?
> >
> > [...]
>
> Corrigé
>
> > Dans les mockups plus haut, quand il y a un seul guichet, on n'affiche pas son nom.
> >
> > [...]
>
> Fait
>
> > Et on n'a pas ce message, ni le "New Time Period" à cet endroit. J'écrivais « je laisserais ça de côté pour le moment » pour qu'on s'épargne un travail d'attention sur le rendu. Genre le lien collé au texte.
> >
> > [...]
> >
> > L'écran initial ici devrait être dans une situation où un guichet existe déjà, créé automatiquement.

> Ajouté (j'imagine que modifier le label de ce guichet est inutile)
>
> > Ça me fait donc noter que :
> >
> > [...]
> >
> > Il faudrait une valeur par défaut dans le champ "libellé" de la boite de dialogue, genre _('Desk #%s') % (len(current_desks)+1).
> >
> > [...]
> >
> > Manque un test unitaire pour couvrir cette migration.
> >
> > Manquent aussi une converture pour le *unicode* de la classe Desk et pour la création de son slug si le premier slug trouvé n'est pas bon; n'hésite pas à exécuter les tests en activant le calcul de couverture de code chez toi. (--cov-report=html --cov=chrono/).
>
> Je rajoute d'ici peu et améliore la couverture.

J'ai rajouté le test sur la migration de données. D'apres le report du coverage je dois avoir couvert toutes mes modifications normalement.

#13 - 22 août 2017 17:36 - Frédéric Péters

```
+    @classmethod
+    def get_available_desk(cls, agenda, event):
+        event_datetime = localtime(event.start_datetime)
+        for desk in cls.objects.filter(agenda=agenda,
+                                       timeperiods__start_time__lte=event_datetime.time(),
+                                       timeperiods__end_time__gte=event_datetime.time()):
+            events = desk.event_set.filter(start_datetime=event_datetime)
+            if not events.exists():
+                return desk
+            if events.exclude(booking__cancellation_datetime__isnull=True).count() and events.count() == 1:
+                return desk
+        return None
```

Ça manque de commentaire pour que je sois sûr de comprendre et sûr que ça soit correct :

- condition 1 : not desk.event_set.filter(start_datetime=event_datetime).exists()
  - → il peut y avoir des rendez-vous de différents types, que se passe-t-il si un rendez-vous d'une heure a commencé dans ce guichet une demi-heure plus tôt ?
- condition 2 : events.exclude(booking__cancellation_datetime__isnull=True).count() and events.count() == 1
  - → simplement exclure les booking__cancellation_datetime__isnull = False de la recherche d'événement au-dessus, plutôt que ce drôle de truc ici, il me semble.

```
-            label='Guichet 1', slug='guichet-1', agenda=time_period.agenda)
+            label='Desk 1', slug='desk-1', agenda=time_period.agenda)
```

J'avais laissé passer le "Guichet 1" sur l'idée de ne pas demander à gérer les langues à ce niveau. Parce que imposer "Desk 1" à tout le monde, ça ne le fait pas du tout.

```
         busy_time_slots = Event.objects.filter(agenda=agenda,
                 start_datetime__gte=min_datetime,
-                start_datetime__lte=max_datetime + datetime.timedelta(meeting_type.duration))
+                start_datetime__lte=max_datetime + datetime.timedelta(meeting_type.duration)).exclude(desk__i
snull=True)
```

Mais il n'y a pourtant plus de event.desk = None ?

```
+            {% if object.desk_set.count > 1%}
```

Ça ne serait pas vraiment une relecture sans perdre du temps à pointer qu'il manque un espace là.

~~

Pour la page sous /manage/, je zappe les commentaires, je ferai des tickets (et vraisemblablement les patchs) plus tard.

#14 - 23 août 2017 19:10 - Josué Kouka

Branche mise à jour avec les modification

Frédéric Péters a écrit :

> [...]

> Ça manque de commentaire pour que je sois sûr de comprendre et sûr que ça soit correct :

J'en ai ajouté

- condition 1 : not desk.event_set.filter(start_datetime=event_datetime).exists()
  - → il peut y avoir des rendez-vous de différents types, que se passe-t-il si un rendez-vous d'une heure a commencé dans ce guichet une demi-heure plus tôt ?

J'ai ajouté un filtre sur le type de meetings.

- condition 2 : events.exclude(booking__cancellation_datetime__isnull=True).count() and events.count() == 1
  - → simplement exclure les booking__cancellation_datetime__isnull = False de la recherche d'événement au-dessus, plutôt que ce drôle de truc ici, il me semble.

Corrigé, mais l'exclusion des booking__cancellation_datetime__isnull = False ne suffit pas dans le cas ou :
1. On réserve un créneau (Booking.objects.count 1)
2. On l'annule (Booking.objects.count 1)
3. On re-réserve le meme créneau (Booking.objects.count == 2)
4. On éssaie de réserver le meme créneau encore une fois
Le seul moyen de savoir si ce créneau est annulé et n'a pas été "re-réserver" c'est de m'assurer que une seule résa existe et qu'elle est annulée.

> [...]
>
> J'avais laissé passer le "Guichet 1" sur l'idée de ne pas demander à gérer les langues à ce niveau. Parce que imposer "Desk 1" à tout le monde, ça ne le fait pas du tout.
>
> [...]

J'ai remis comme auparavant

> Mais il n'y a pourtant plus de event.desk = None ?

Vrai

> [...]
>
> Ça ne serait pas vraiment une relecture sans perdre du temps à pointer qu'il manque un espace là.
>
> ~~
>
> Pour la page sous /manage/, je zappe les commentaires, je ferai des tickets (et vraisemblablement les patchs) plus tard.

#15 - 23 août 2017 19:59 - Frédéric Péters

> Le seul moyen de savoir si ce créneau est annulé et n'a pas été "re-réserver" c'est de m'assurer que une seule résa existe et qu'elle est annulée.

L'explication m'a l'air tortueuse.

#16 - 24 août 2017 10:40 - Josué Kouka

Frédéric Péters a écrit :

> > Le seul moyen de savoir si ce créneau est annulé et n'a pas été "re-réserver" c'est de m'assurer que une seule résa existe et qu'elle est annulée.
>
> L'explication m'a l'air tortueuse.

T'avais raison. Un excluce suffisait. La branche est à jour.

#17 - 25 août 2017 11:34 - Josué Kouka

J'ai mis la branche à jour avec ajout de la gestion du chevauchement des évènements de différent types de rendez-vous.

#18 - 25 août 2017 11:54 - Frédéric Péters

Pointé dans celui sur les exceptions mais la même préoccupation est à avoir ici :

```
        busy_time_slots = [
            time_slot for time_slot in list(busy_time_slots) if not Desk.get_available_desk(agenda, time_slot)
]
```

get_available_desk fait nombre de guichets + 1 requêtes; c'est multiplié par le nombre de créneaux, qui peut facilement monter à des centaines.

On ne peut pas faire autant de requêtes.

#19 - 28 août 2017 16:05 - Josué Kouka

Frédéric Péters a écrit :

> Pointé dans celui sur les exceptions mais la même préoccupation est à avoir ici :
>
> [...]
>
> get_available_desk fait nombre de guichets + 1 requêtes; c'est multiplié par le nombre de créneaux, qui peut facilement monter à des centaines.

Ceci est vrai pour les exceptions, je vois comment l'optimiser. Mais dans le cas l'api datetime du multi guichet c'est multiplié par le nombre d'évènements existants dans la période ouvrées de cette agenda et non tous les créneaux. Ces évènements correspondent au différents créneaux occupés.
J'étais parti sur l'idée que je pouvais jouer sur la génération des créneaux mais j'avais tort car j'avais perdu de vue que c'était les évènements qui m'intéressaient (ça me fait penser que je dois ajouter une migration de données pour les évènements existants).
Je ne vois pas comment éviter de parcourir chaque guichet et vérifier pour chacun d'eux qu'il y a une résa en cours pour dans un créneau donné.

> On ne peut pas faire autant de requêtes.

#20 - 28 août 2017 18:28 - Josué Kouka

> J'étais parti sur l'idée que je pouvais jouer sur la génération des créneaux mais j'avais tort car j'avais perdu de vue que c'était les évènements qui m'intéressaient (ça me fait penser que je dois ajouter une migration de données pour les évènements existants).

J'ai mis à jour la branche avec la migration de données pour les évènements.

#21 - 28 août 2017 18:35 - Frédéric Péters

> Mais dans le cas l'api datetime du multi guichet c'est multiplié par le nombre d'évènements existants dans la période ouvrées de cette agenda et non tous les créneaux

Ok (je ne vérifie pas) mais sur une mairie chargée, ça revient au même.

#22 - 28 août 2017 22:23 - Frédéric Péters

J'étais parti pour étendre les tests pour avoir beaucoup de réservations, pour que le coût des requêtes SQL se ressente, mais.

Je vide toutes les réservations et comme il y a deux types de rendez-vous, 30 et 60 minutes, j'assure qu'on est sur un rendez-vous de 30 minutes :

```
+    Booking.objects.all().delete()
+    assert meeting_type.duration == 30
```

Vu que j'ai vidé Booking, dans cet extrait il y a juste le nombre de réservations qui diffère :

```
     resp = app.get('/api/agenda/meetings/%s/datetimes/' % meeting_type.id)
     event_id = resp.json['data'][1]['id']
     resp_booking = app.post('/api/agenda/%s/fillslot/%s/' % (agenda_id, event_id))
-    assert Booking.objects.count() == 2
+    assert Booking.objects.count() == 1
+    assert resp_booking.json['err'] == 0
     assert resp_booking.json['datetime'][:16] == localtime(Booking.objects.last().event.start_datetime
             ).isoformat()[:16]
```

Voilà le vrai ajout, je prend le deuxième créneau qui avait été envoyé par datetimes, je note bien qu'il est différent du précédent :

```
+    event_id = resp.json['data'][2]['id']
+    assert resp.json['data'][1]['id'] != resp.json['data'][2]['id']
```

J'essaie alors de le réserver :

```
+    resp_booking = app.post('/api/agenda/%s/fillslot/%s/' % (agenda_id, event_id))
+    assert resp_booking.json['err'] == 0
+    assert Booking.objects.count() == 2
```

Et ça échoue.

Pour info,

```
(Pdb) print resp.json['data'][1]['id']
1:2017-05-22-1030
(Pdb) print resp.json['data'][2]['id']
1:2017-05-22-1100
(Pdb) print resp_booking.json
{u'reason': u'no more desk available', u'err': 1}
```

Parce qu'on n'est jamais trop prudent, je suis passé sur la branche master pour vérifié, j'ai étendu test_booking_api_meeting avec une seconde réservation réussie, et ça passe. J'ai poussé ce commit ajoutant un test, j'ai modifié le nom d'une variable (resp → resp2) par facilité, ça fait une petite modif qui ne passe pas automatiquement lors du rebase, désolé. Après rebase, j'ai exécuté le test depuis la branche multi-guichet et comme attendu, il a échoué.

J'espérais en fait que fonctionnellement c'était ok, qu'il était temps de se préoccuper des performances mais en fait non, ce n'est pas encore ok, il faut corriger ce cas-ci, et réfléchir à d'autres cas à ajouter dans les tests.

#23 - 28 août 2017 22:42 - Frédéric Péters

En l'espèce :

```
 def has_overlap(a, b):
     latest = max(a.start_datetime, b.start_datetime)
     earliest = min(a.end_datetime, b.end_datetime)
-    return latest <= earliest
+    return latest < earliest
```

Mais quelle manière peu lisible d'écrire ce code.

```
def has_overlap(a, b):
    # le premier se termine quand l'autre n'a pas encore commencé
    if a.end_datetime <= b.start_datetime:
        return False
    # ou il commence quand l'autre est déjà terminé
    if a.start_datetime >= b.end_datetime:
        return False
    return True
```

Vraiment, aucun bonus à faire tenir ça sur trois lignes.

#24 - 28 août 2017 23:13 - Frédéric Péters

- Fichier figure_1-1.png ajouté

Pour reprendre sur les performances, sur un agenda chargé :

```
@@ -715,14 +717,16 @@ def test_agenda_meeting_api_multiple_desk(app, meetings_agenda, user):
         desk=desk2)

     resp = app.get('/api/agenda/meetings/%s/datetimes/' % meeting_type.id)
-    event_id = resp.json['data'][1]['id']
-    resp_booking = app.post('/api/agenda/%s/fillslot/%s/' % (agenda_id, event_id))
-    assert Booking.objects.count() == 2
-    assert resp_booking.json['datetime'][:16] == localtime(Booking.objects.last().event.start_datetime
-            ).isoformat()[:16]
-
-    resp2 = app.get('/api/agenda/meetings/%s/datetimes/' % meeting_type.id)
-    assert len(resp.json['data']) == len([x for x in resp2.json['data'] if not x['disabled']]) + 2
+    for i, event_data in enumerate(resp.json['data'][1:]):
+        event_id = event_data['id']
+        resp_booking = app.post('/api/agenda/%s/fillslot/%s/' % (agenda_id, event_id))
+        assert Booking.objects.count() == 2+i
+        assert resp_booking.json['datetime'][:16] == localtime(Booking.objects.last().event.start_datetime
```

```
+                ).isoformat()[:16]
+
+        with CaptureQueriesContext(connection) as ctx:
+            resp2 = app.get('/api/agenda/meetings/%s/datetimes/' % meeting_type.id)
+            assert len(ctx.captured_queries) < XXX
```

On fixe XXX, le nombre de requêtes SQL pour /datetimes/ à combien ?

#25 - 31 août 2017 12:21 - Benjamin Dauvergne

On me demande mon avis (mais ça gagnerait à avoir une page wiki qui parle juste du modèle de donnée, parce que comme dirait un éminent inventeur de système de type Unix libre, l'important c'est les structures de donnée et leurs relations).

Déjà on gagnerait à utiliser des objets adaptés du type Interval (interval de temps fermé à gauche ouvert à droite) et IntervalSet, par exemple ça https://pypi.python.org/pypi/intervaltree qui est dans Debian.

Étant donné ces outils je ferai un truc comme cela sans appeler 36 APIs différentes qui limite les possibilités d'avoir un algo simple et efficace:

```
     # D'abord on range les périodes d'ouverture par jour
     tp_by_weekday = {}
     for tp in TimePeriod.objects.filter(agenda=agenda):
         tp_by_weekday.setdefault(tp.weekday, []).append(tp)
     # Maintenant on accumule les périodes d'ouverture pour la période globale concernée et rangé par guichet
     cursor = min_datetime
     open_ranges_by_desk = collections.defaultdict(lambda: IntervalTree())
     while cursor < max_datetime:
         weekday = cursor.date().weekday
         for tp in tp_by_weekday[weekday]:
             start = cursor.replace(hour=tp.start_time.hour, minute=tp.start_time.minute, second=tp.start_time
.second)
             if start < min_datetime:
                 start = min_datetime
             if start > max_datetime:
                 continue
             end = cursor.replace(hour=tp.end_time.hour, minute=tp.end_time.minute, second=tp.end_time.second)
             if end > max_datetime:
                 end = max_datetime
             if end < min_datetime:
                 continue
             open_slots_by_desk[tp.desk].addi(start, end)
         cursor = cursor + timedelta(day=1)
    # On enlève les périodes indisponibles
    for event in Event.objects.filter(agenda=agenda, start_datetime__gte=min_datetime,
                                      start_datetime__lte=max_datetime + datetime.timedelta(meeting_type.durat
ion)):
        open_slots_by_desk[event.desk].chop(event.start_datetime, envent.end_datetime)
# XXX: event.desk could be None...
    # Les périodes disponibles sont l'union des périodes disponible pour tous les guichets
    open_slots = reduce(open_slots_by_desk.values(), operator.__or__)


# Pour chaque time_slot on vérifie qu'il est contenu dans une période disponible (ça s'optimise en générant di
rectement les time_slot depuis l'IntervalTree je pense)
    for time_slot in all_time_slots:
        if open_slots[time_slot.start_datetime:time_slot.end_datetime]:
            entries.append(time_slot)
```

#26 - 31 août 2017 14:36 - Josué Kouka

Benjamin Dauvergne a écrit :

> On me demande mon avis (mais ça gagnerait à avoir une page wiki qui parle juste du modèle de donnée, parce que comme dirait un éminent inventeur de système de type Unix libre, l'important c'est les structures de donnée et leurs relations).
>
> Déjà on gagnerait à utiliser des objets adaptés du type Interval (interval de temps fermé à gauche ouvert à droite) et IntervalSet, par exemple ça https://pypi.python.org/pypi/intervaltree qui est dans Debian.
>
> Étant donné ces outils je ferai un truc comme cela sans appeler 36 APIs différentes qui limite les possibilités d'avoir un algo simple et efficace:
>
> [...]

Merci pour ton retour Benj. J'avais pu diminué le temps de réponse pour l'api datetime pour un agenda plein par 2, malheureusement ce temps était toujours supérieur à 1s.

Je regarde avec cet algo pour voir.

#27 - 01 septembre 2017 13:28 - Josué Kouka

J'ai mis à jour la branche avec les améliorations des performances **0.3s** en moyenne sur datetime.

#28 - 01 septembre 2017 13:36 - Frédéric Péters

> J'ai mis à jour la branche avec les améliorations des performances 0.3s en moyenne sur datetime.

Ce qu'il faut ce n'est pas juste la durée, surtout pas une "moyenne" (moyenne sur des mesures prises avec une variété d'événements enregistrés ou moyenne sur exactement le même appel ?, genre).

Ce qui est important c'est avoir idée de la complexité ("grand O"), notamment par rapport au nombre de requêtes SQL exigées.

#29 - 01 septembre 2017 13:39 - Frédéric Péters

tests/test_time_periods.py::test_timeperiod_time_slots FAILED

#30 - 01 septembre 2017 13:53 - Frédéric Péters

> Ce qui est important c'est avoir idée de la complexité ("grand O"), notamment par rapport au nombre de requêtes SQL exigées.

Toujours une progression linéraire au nombre de réservations, cette fois-ci parfaite. (3 requêtes par réservation).

Ça vient de là :

```
    for event in agenda.event_set.filter(agenda=agenda, start_datetime__gte=min_datetime,
                                         start_datetime__lte=max_datetime + datetime.timedelta(meeting_type.du
ration)):
        if event.booking_set.filter(cancellation_datetime__isnull=False).exists():
            continue
```

#31 - 01 septembre 2017 14:16 - Frédéric Péters

```
--- a/chrono/api/views.py
+++ b/chrono/api/views.py
@@ -47,13 +47,21 @@ def get_open_slot_by_desk(agenda, meeting_type):
     open_slots_by_desk = defaultdict(lambda: IntervalTree())
     for time_period in agenda.timeperiod_set.all():
         for slot in time_period.get_time_slots(**time_period_filters):
-            open_slots_by_desk[time_period.desk].addi(slot.start_datetime, slot.end_datetime, slot.desk)
+            open_slots_by_desk[time_period.desk_id].addi(slot.start_datetime, slot.end_datetime, slot.desk)

     for event in agenda.event_set.filter(agenda=agenda, start_datetime__gte=min_datetime,
-                                         start_datetime__lte=max_datetime + datetime.timedelta(meeting_type.d
uration)):
-        if event.booking_set.filter(cancellation_datetime__isnull=False).exists():
+                                         start_datetime__lte=max_datetime + datetime.timedelta(meeting_type.d
uration)
+                               ).select_related('meeting_type').extra(
+                    select={
+                        'booking_count': '''SELECT
+                                            COUNT(*) FROM agendas_booking
+                                            WHERE agendas_booking.event_id = agendas_event.id
+                                              AND agendas_booking.cancellation_datetime IS NOT NULL'''}):
+        if event.booking_count:
             continue
-        open_slots_by_desk[event.desk].remove_overlap(event.start_datetime, event.end_datetime)
+        open_slots_by_desk[event.desk_id].remove_overlap(event.start_datetime, event.end_datetime)

     open_slots = reduce(operator.__or__, open_slots_by_desk.values())
     return open_slots
```

Et peu importe le nombre de réservations ça fait 12 requêtes.

#32 - 01 septembre 2017 14:17 - Frédéric Péters

À noter que c'est exactement l'exemple donné dans la documentation : https://docs.djangoproject.com/en/1.11/ref/models/querysets/#extra

#33 - 01 septembre 2017 15:40 - Josué Kouka

Frédéric Péters a écrit :

> tests/test_time_periods.py::test_timeperiod_time_slots FAILED

Corrigé.

> > J'ai mis à jour la branche avec les améliorations des performances 0.3s en moyenne sur datetime.
>
> Ce qu'il faut ce n'est pas juste la durée, surtout pas une "moyenne" (moyenne sur des mesures prises avec une variété d'événements enregistrés ou moyenne sur exactement le même appel ?, genre).
>
> Ce qui est important c'est avoir idée de la complexité ("grand O"), notamment par rapport au nombre de requêtes SQL exigées.

Comme tu l'as constaté la complexité était de O(n)

> Ça vient de là :

```
for event in agenda.event_set.filter(agenda=agenda, start_datetime__gte=min_datetime,
                                     start_datetime__lte=max_datetime + datetime.timedelta(meeting_type.du
ration)):
        if event.booking_set.filter(cancellation_datetime__isnull=False).exists():
            continue
```

Yep j'avais remarqué mais j'avais besoin de filtrer les résa annulées.

> Et peu importe le nombre de réservations ça fait 12 requêtes.

Je confirme O(1).

ps: branche à jour.

#34 - 01 septembre 2017 15:46 - Frédéric Péters

> > Et peu importe le nombre de réservations ça fait 12 requêtes.
>
> Je confirme O(1).

On écrit "Merci".

#35 - 01 septembre 2017 15:47 - Josué Kouka

Frédéric Péters a écrit :

> > > Et peu importe le nombre de réservations ça fait 12 requêtes.
> >
> > Je confirme O(1).
>
> On écrit "Merci".

Merci Frédéric.

#36 - 01 septembre 2017 15:48 - Frédéric Péters

:)

Maintenant, tu peux tout réunir dans un seul commit; la séparation actuelle n'est pas utile.

#37 - 01 septembre 2017 15:48 - Frédéric Péters

Et aussi merci à Benjamin qui a donné la piste à suivre.

#38 - 01 septembre 2017 15:51 - Frédéric Péters

En regardant la couverture des tests, tu verras que tu as laissé du code inutile. (et qu'il faut ajouter un test sur 0018_event_desk.py).

#39 - 01 septembre 2017 16:27 - Josué Kouka

Frédéric Péters a écrit :

> En regardant la couverture des tests, tu verras que tu as laissé du code inutile. (et qu'il faut ajouter un test sur 0018_event_desk.py).

Merci, j'ai rajouté le test dans 0018_event_desk.py pour couvrir le code inutile et mis à jour la branche.

```
Name                                                                    Stmts   Miss  Cover   Missing
-----------------------------------------------------------------------------------------------------
chrono/__init__.py                                                          0      0   100%
chrono/agendas/__init__.py                                                  0      0   100%
chrono/agendas/migrations/0001_initial.py                                   5      0   100%
chrono/agendas/migrations/0002_event.py                                     5      0   100%
chrono/agendas/migrations/0003_booking.py                                   6      0   100%
chrono/agendas/migrations/0004_booking_cancellation_datetime.py             5      0   100%
chrono/agendas/migrations/0005_event_label.py                               5      0   100%
chrono/agendas/migrations/0006_auto_20160707_1357.py                        7      0   100%
chrono/agendas/migrations/0007_auto_20160722_1135.py                        5      0   100%
chrono/agendas/migrations/0008_auto_20160910_1319.py                        5      0   100%
chrono/agendas/migrations/0009_auto_20160911_1640.py                        5      0   100%
chrono/agendas/migrations/0010_auto_20160918_1250.py                        5      0   100%
chrono/agendas/migrations/0011_meetingtype_slug.py                          5      0   100%
chrono/agendas/migrations/0012_manual_set_slugs_on_meeting_types.py        11      0   100%
chrono/agendas/migrations/0013_auto_20161028_1603.py                        5      0   100%
chrono/agendas/migrations/0014_booking_primary_booking.py                   5      0   100%
chrono/agendas/migrations/0015_auto_20170628_1137.py                        5      0   100%
chrono/agendas/migrations/0016_desk.py                                      5      0   100%
chrono/agendas/migrations/0017_timeperiod_desk.py                          14      0   100%
chrono/agendas/migrations/0018_event_desk.py                               16      0   100%
chrono/agendas/migrations/__init__.py                                       0      0   100%
chrono/agendas/models.py                                                  255      8    97%   38-40, 146-150
chrono/api/__init__.py                                                      0      0   100%
chrono/api/urls.py                                                          3      0   100%
chrono/api/views.py                                                       218      0   100%
chrono/formats/__init__.py                                                  0      0   100%
chrono/formats/fr/__init__.py                                               0      0   100%
chrono/formats/fr/formats.py                                                1      0   100%
chrono/manager/__init__.py                                                  0      0   100%
chrono/manager/forms.py                                                    82      0   100%
chrono/manager/management/__init__.py                                       0      0   100%
chrono/manager/management/commands/__init__.py                              0      0   100%
chrono/manager/management/commands/export_site.py                          13      0   100%
chrono/manager/management/commands/import_site.py                          16      0   100%
chrono/manager/urls.py                                                      3      0   100%
chrono/manager/utils.py                                                    14      0   100%
chrono/manager/views.py                                                   205      0   100%
chrono/manager/widgets.py                                                  62      0   100%
chrono/settings.py                                                         41      2    95%   121-122
chrono/urls.py                                                             13      0   100%
chrono/urls_utils.py                                                       41      8    80%   36-39, 49-51, 70
chrono/views.py                                                            25      2    92%   28, 44
chrono/wsgi.py                                                              4      0   100%
-----------------------------------------------------------------------------------------------------
TOTAL                                                                    1115     20    98%
```

#40 - 01 septembre 2017 16:29 - Frédéric Péters

> Merci, j'ai rajouté le test dans 0018_event_desk.py pour couvrir le code inutile et mis à jour la branche.

Le code inutile et pas couvert, c'est celui-ci :

```
chrono/agendas/models.py                                                  255      8    97%   38-40, 146-150
```

#41 - 01 septembre 2017 16:44 - Frédéric Péters

Attention dans ton message de commit tu parles de "counter" mais dans le code tu utilises "desk", c'est mieux de rester cohérent.

#42 - 01 septembre 2017 17:02 - Josué Kouka

Frédéric Péters a écrit :

> Merci, j'ai rajouté le test dans 0018_event_desk.py pour couvrir le code inutile et mis à jour la branche.

Le code inutile et pas couvert, c'est celui-ci :

[...]

OK c'est bon j'ai supprimé has_overlap et intersect.

Attention dans ton message de commit tu parles de "counter" mais dans le code tu utilises "desk", c'est mieux de rester cohérent.

Message de commit changé.

#43 - 01 septembre 2017 17:43 - Frédéric Péters

```
+    desk = models.ForeignKey('Desk', on_delete=models.CASCADE,
+                             related_name='timeperiods')
```

Quand c'est possible je préfère qu'on garde les noms par défaut, ça permet une compréhension immédiate.

```
+from collections import defaultdict
 import datetime
+from intervaltree import IntervalTree
+import operator
```

Avoir les imports de la bibliothèque standard groupés alphabétiquement au début. (→ déplacer l'intervaltree dessous, après une ligne blanche).

```
@@ -66,12 +66,32 @@ class MeetingTypeForm(forms.ModelForm):

 class TimePeriodForm(forms.ModelForm):
+
     class Meta:
```

Ça me fatigue toujours, ces ajoutes de lignes blanches sans même l'argument de cohérence, ce bout continuant avec :

```
+class NewDeskForm(forms.ModelForm):
+    class Meta:
+        model = Desk

<a rel="popup" href="{{add_time_period_url}}"> {% trans "New Time Period" %}</a>
```

Espace inutile au début du lien.

```
+        url(r'^agendas/(?P<pk>\w+)/add-time-period/(?P<desk>\w+)$', views.agenda_add_time_period,
```

<desk_pk> ou <desk_id> et \d+ si c'est un id attendu, <desk_slug> si c'est un slug.

```
+++ b/tests/test_data_migrations.py
```

Ok pour taper ça dans un nouveau fichier mais il faut alors y mettre tous les tests de ce type (y déplacer ceux qui sont actuellement dans test_agendas.py).

(Je n'ai pas encore testé pour de vrai ces derniers changements, c'est juste une relecture)

#44 - 01 septembre 2017 18:48 - Josué Kouka

- Fichier 0001-add-multiple-desk-management-15729.patch ajouté

Frédéric Péters a écrit :

> [...]
>
> Quand c'est possible je préfère qu'on garde les noms par défaut, ça permet une compréhension immédiate.

Ok, j'ai supprimé le related_name

> [...]

> Avoir les imports de la bibliothèque standard groupés alphabétiquement au début. (→ déplacer l'intervaltree dessous, après une ligne blanche).

Corrigé

> [...]
>
> Ça me fatigue toujours, ces ajoutes de lignes blanches sans même l'argument de cohérence, ce bout continuant avec :

Corrigé

> [...]
>
> [...]
>
> Espace inutile au début du lien.

Corrigé

> [...]
>
> <desk_pk> ou <desk_id> et \d+ si c'est un id attendu, <desk_slug> si c'est un slug.

Corrigé (desk_pk)

> [...]
>
> Ok pour taper ça dans un nouveau fichier mais il faut alors y mettre tous les tests de ce type (y déplacer ceux qui sont actuellement dans test_agendas.py).

J'ai déplacé test_meeting_type_slug_migration vers tests/test_data_migrations.py

> (Je n'ai pas encore testé pour de vrai ces derniers changements, c'est juste une relecture)

#45 - 01 septembre 2017 19:01 - Frédéric Péters

> > Avoir les imports de la bibliothèque standard groupés alphabétiquement au début. (→ déplacer l'intervaltree dessous, après une ligne blanche).
>
> Corrigé

après une ligne blanche.

#46 - 01 septembre 2017 20:31 - Josué Kouka

- Fichier 0001-add-multiple-desk-management-15729.patch ajouté

Frédéric Péters a écrit :

> > Avoir les imports de la bibliothèque standard groupés alphabétiquement au début. (→ déplacer l'intervaltree dessous, après une ligne blanche).
>
> Corrigé

> **après une ligne blanche**.

Ligne blanche ajoutée.

#47 - 01 septembre 2017 20:46 - Frédéric Péters

Prendre en compte #18101 plutôt qu'introduire un nouvel exemplaire de ce bug.

#48 - 01 septembre 2017 20:53 - Frédéric Péters

On a Agenda 1-----n Desk 1-----n TimePeriod; Et en même temps la relation directe Agenda 1-----n TimePeriod. Il me semble que cette seconde relation n'a plus lieu d'être.

En lien, lors de l'import en mode overwrite :

```
        elif data['kind'] == 'meetings':
            if overwrite:
                MeetingType.objects.filter(agenda=agenda).delete()
                TimePeriod.objects.filter(agenda=agenda).delete()
```

Même sans mon commentaire, il manque la suppression des guichets. (et avec mon commentaire, la ligne TimePeriod disparait).

#49 - 03 septembre 2017 12:55 - Josué Kouka

- Fichier 0001-add-multiple-desk-management-15729.patch ajouté

Frédéric Péters a écrit :

> Prendre en compte #18101 plutôt qu'introduire un nouvel exemplaire de ce bug.

J'ai mis la taille a 150.

> On a Agenda 1-----n Desk 1-----n TimePeriod; Et en même temps la relation directe Agenda 1-----n TimePeriod. Il me semble que cette >
> seconde relation n'a plus lieu d'être.
>
> En lien, lors de l'import en mode overwrite :
> Même sans mon commentaire, il manque la suppression des guichets. (et avec mon commentaire, la ligne TimePeriod disparait).

J'ai tout corrigé

#50 - 03 septembre 2017 13:15 - Frédéric Péters

Le coverage de chrono/manager/views.py n'est plus total.

#51 - 03 septembre 2017 14:31 - Frédéric Péters

- Lié à Development #18414: Ajustements ui/ux au multiguichet ajouté

#52 - 03 septembre 2017 15:24 - Josué Kouka

- Fichier 0001-add-multiple-desk-management-15729.patch ajouté

Frédéric Péters a écrit :

> Le coverage de chrono/manager/views.py n'est plus total.

J'ai ajouté des tests. La couverture est maintenant totale.

#53 - 03 septembre 2017 15:38 - Frédéric Péters

> J'ai ajouté des tests. La couverture est maintenant totale.

Dans test_meetings_agenda_add_time_period_as_manager, il faudrait quand même tester autre chose que les cas d'erreur.

Dans test_agenda_meeting_api_multiple_desk il ne faudrait pas qu'à l'occasion d'un changement on perde les performances, il faudrait un test assurant que le nombre de requêtes SQL est identique qu'il y ait une ou deux réservations.

#54 - 04 septembre 2017 10:15 - Josué Kouka

- Fichier 0001-add-multiple-desk-management-15729.patch ajouté

Frédéric Péters a écrit :

> > J'ai ajouté des tests. La couverture est maintenant totale.

> Dans test_meetings_agenda_add_time_period_as_manager, il faudrait quand même tester autre chose que les cas d'erreur.

Ok, j'ai ajouté des tests d'existence de certains d'éléments dans le dom et des tests d'acces lorsque les droits de modification sont donnés au manager.

> Dans test_agenda_meeting_api_multiple_desk il ne faudrait pas qu'à l'occasion d'un changement on perde les performances, il faudrait un test assurant que le nombre de requêtes SQL est identique qu'il y ait une ou deux réservations.

J'ai ajouté un test qui compare les nombre de requetes faites (datetime et fillslot) pour un agenda avec très peu de résa avec celles faites pour un agenda complet.

#55 - 04 septembre 2017 10:39 - Frédéric Péters

> J'ai ajouté un test qui compare les nombre de requetes faites (datetime et fillslot) pour un agenda avec très peu de résa avec celles faites pour un agenda complet.

Bien mais ça devient trop coûteux à l'exécution des tests et ça décourage de les lancer :/ Je propose de simplement faire :

```
-    for idx, event_data in enumerate(resp2.json['data'][2:]):
+    for idx, event_data in enumerate(resp2.json['data'][2:10]):
```

Ce qui chez moi passe l'exécution intégrale des tests de 169 secondes à 63 secondes et fournit une garantie proche.

#56 - 04 septembre 2017 11:00 - Josué Kouka

- Fichier 0001-add-multiple-desk-management-15729.patch ajouté

Frédéric Péters a écrit :

> J'ai ajouté un test qui compare les nombre de requetes faites (datetime et fillslot) pour un agenda avec très peu de résa avec celles faites pour un agenda complet.
>
> Bien mais ça devient trop coûteux à l'exécution des tests et ça décourage de les lancer :/ Je propose de simplement faire :
>
> [...]
>
> Ce qui chez moi passe l'exécution intégrale des tests de 169 secondes à 63 secondes et fournit une garantie proche.

Ok fait.

#57 - 04 septembre 2017 12:21 - Frédéric Péters

Ok, ack pour ça.

#58 - 04 septembre 2017 14:23 - Josué Kouka

- Statut changé de En cours à Résolu (à déployer)

```
commit f851de07df1288c46fa89fa02b2e16eb6dbf1f7e
Author: Josue Kouka <jkouka@entrouvert.com>
Date:   Fri Sep 1 15:01:07 2017 +0200

    add multiple desk management (#15729)
```

#59 - 04 décembre 2018 20:07 - Frédéric Péters

- Statut changé de Résolu (à déployer) à Fermé

Fichiers

0001-add-multiple-windows-feature-for-meeting-api-15729.patch	5,95 ko	25 juillet 2017	Josué Kouka
figure_1-1.png	20,1 ko	28 août 2017	Frédéric Péters
0001-add-multiple-desk-management-15729.patch	47 ko	01 septembre 2017	Josué Kouka
0001-add-multiple-desk-management-15729.patch	47 ko	01 septembre 2017	Josué Kouka

0001-add-multiple-desk-management-15729.patch	51,3 ko 03 septembre 2017	Josué Kouka
0001-add-multiple-desk-management-15729.patch	52,1 ko 03 septembre 2017	Josué Kouka
0001-add-multiple-desk-management-15729.patch	53,7 ko 04 septembre 2017	Josué Kouka
0001-add-multiple-desk-management-15729.patch	53,7 ko 04 septembre 2017	Josué Kouka