

Combo - Development #16393

Ajouter un cellule Calendar

18 mai 2017 10:35 - Josué Kouka

<b>Statut:</b>	Fermé	<b>Début:</b>	18 mai 2017
<b>Priorité:</b>	Normal	<b>Echéance:</b>	09 juin 2017
<b>Assigné à:</b>	Josué Kouka	<b>% réalisé:</b>	0%
<b>Catégorie:</b>		<b>Temps estimé:</b>	0:00 heure
<b>Version cible:</b>		<b>Planning:</b>	
<b>Patch proposé:</b>	Oui		
<b>Description</b> Cette cellule devra afficher les créneaux disponibles d'évènement d'une source quelconque. En choisissant un créneau, l'utilisateur devra etre redirigé vers un formulaire wcs			
<b>Demandes liées:</b>			
Lié à Intégrations graphiques Publik - Development #16936: style pour les cel...		Fermé	16 juin 2017
Lié à Combo - Development #16955: javascript de support pour la cellule calen...		Fermé	17 juin 2017
Lié à Combo - Bug #16956: Cellule Calendar: reservation invalide pour créneau...		Rejeté	17 juin 2017

Révisions associées

Révision 56b47f5a - 18 juin 2017 13:06 - Josué Kouka

general: add booking calendar cell (#16393)

Historique

#2 - 18 mai 2017 10:37 - Serghei Mihai (congé, retour 15/05)

- Sujet changé de Ajouter un cellule Calendar à Ajouter un cellule Calendar

#3 - 18 mai 2017 10:41 - Josué Kouka

- Fichier 0003-add-calendar-cell-static-files-16393.patch ajouté
- Fichier 0002-add-chrono-app-view-and-utils-16393.patch ajouté
- Fichier 0001-add-calendar-cell-model-16393.patch ajouté
- Patch proposé changé de Non à Oui

#5 - 18 mai 2017 11:16 - Thomas Noël

Le patch 3 c'est une blague ? ;) J'imagine que oui, avec #16394

Un peu partout au début : Copyright (C) 2015 → 2017

Dans le patch 1 :

```
@classmethod
def is_enabled(cls):
    return is_chrono_enabled
```

où il manque des parenthèses après is\_chrono\_enabled (sinon ça sera toujours "oui"), mais en plus is\_chrono\_enabled n'existe pas encore (patch 2).

```
context['events_source_url'] = reverse('chrono-events', kwargs={'pk': self.pk})
context['events_booking_url'] = reverse('chrono-booking', kwargs={'pk': self.pk})
```

ça me semble pas utile, on doit pouvoir trouver ces URL dans le template, via {% url ... %} non ? Et là encore, ça marche pas sans le patch 2...

chrono/base.html il sert à quoi ?

Patch 2 : url.py avec chrono/EventsView/, on fait pas du java :)

```
def is_chrono_enabled():
    return get_chrono_service()
```

pour faire clean et moderne et plaisir à Fred, ajoute un bool()

Et zéro tests, ça m'étonne de ta part (y'a de quoi s'inspirer dans tests, genre tests/test\_newsletters\_cell.py ou test\_search.py)

#### #6 - 18 mai 2017 17:35 - Josué Kouka

- Fichier 0001-add-calendar-cell-model-16393.patch ajouté

Avec les corrections et tests

#### #7 - 19 mai 2017 09:51 - Serghei Mihai (congrés, retour 15/05)

Il manque les migrations.

#### #8 - 19 mai 2017 10:12 - Thomas Noël

Dans le README, je ne comprends pas :

session\_vars are defined such as:

```
{
  "my_session_var1_name": "value",
  "my_session_var2_name": "ref_to_received_key",
}
```

(et "defined")

init.py : les double espaces sont en trop ici

comme pour les autres apps, déplacer : static/chrono/chrono.js → static/js/chrono.js

la CSS est vraiment nécessaire ou c'est juste pour faire un peu plus joli ? static/chrono/chrono.css → static/css/chrono.css

Mais surtout dans la CSS, quand je vois dans la css des choses comme

```
.input, select {
  padding: 2px 5px;
}
```

je me dis que ça peut pas être bon, ça va casser l'affichage d'une page si on utilise ça. Selon moi la CSS ne devrait agir que sur des éléments chrono/fullcalendar. Cf <https://cdnjs.cloudflare.com/ajax/libs/fullcalendar/3.4.0/fullcalendar.css>

<div id="calendar"> : ça va marcher comment si on a deux cellules : il faut utiliser des data-truc pour envoyer les valeurs de la cellule au js qui va jouer dessus

params = '&'.join(['%s=%s' % (key, value) for key, value in params.items()]) : utiliser toujours urlencode dans la construction de querystring

HttpResponseRedirect, HttpResponseNotAllowed : importés mais non utilisés

(bon j'arrête là pour ce matin ;)

#### #9 - 19 mai 2017 10:21 - Serghei Mihai (congrés, retour 15/05)

Thomas Noël a écrit :

la CSS est vraiment nécessaire ou c'est juste pour faire un peu plus joli ? static/chrono/chrono.css → static/css/chrono.css

Je dis qu'il faut laisser fullcalendar appliquer sa CSS et les customisations se feront dans publik-base-theme si besoin.

Aussi il manque lib/moment.min.js dans le packaging de fullcalendar et ça pete tout à l'affichage.

#### #10 - 19 mai 2017 14:24 - Josué Kouka

- Fichier 0001-add-calendar-cell-model-16393.patch ajouté

Avec les corrections

#### #11 - 19 mai 2017 15:34 - Josué Kouka

- Fichier 0001-add-calendar-cell-model-16393.patch ajouté

#### #12 - 19 mai 2017 15:56 - Serghei Mihai (congrés, retour 15/05)

Dans le README:

```
{
  "nuser_id": "southpark",
  "date_debut": "$start",
  "date_fin": "$end"
}
```

donne

```
session_var_nursery_id=southapark&date_debut=2017-05-19T13:30:00&date_fin=2017-05-19T15:30:00
```

Ça ne doit pas être ?

```
session_var_nursery_id=southapark&session_var_date_debut=2017-05-19T13:30:00&session_var_date_fin=2017-05-19T15:30:00
```

Et dans ce cas préciser que les variables seront préfixées par session\_var\_.  
Et que \$start et \$end seront remplacés par les créneaux sélectionnés par le widget.

Ensuite, dans le selector jquery:

```
$("#div [data-calendar]").each(function(i, cal){
  renderCal("#" + cal.id);
});
```

il y a un espace entre div et [data-calendar] alors que tu vises les div ayant l'attribut data-calendar.

#### #13 - 19 mai 2017 16:06 - Josué Kouka

- Fichier 0001-add-calendar-cell-model-16393.patch ajouté

En prenant en compte les remarques

#### #14 - 19 mai 2017 16:26 - Serghei Mihai (congé, retour 15/05)

datetime, json, reverse dans models.py non-utilisés: virer.

Charger et utiliser les locales pour le widget fullcalendar, sinon il restera en anglais sur un site en français. Exemple:  
<http://git.entrouvert.org/debian/xstatic-fullcalendar.git/tree/xstatic/pkg/fullcalendar/data/demos/locales.html>

#### #15 - 19 mai 2017 18:01 - Josué Kouka

- Fichier 0001-add-calendar-cell-model-16393.patch ajouté

Pour la langue, les codes qui sont traduits correctement sont ceux codé sur 2 caractères.

#### #16 - 19 mai 2017 18:23 - Josué Kouka

- Fichier 0001-add-calendar-cell-model-16393.patch ajouté

Sans le json qui traîne

#### #17 - 20 mai 2017 13:48 - Thomas Noël

Eviter le git+http://git.entrouvert.org/... dans les requirements : pousser le paquet dans pypi → [HowDoWeDoPythonPackaging](http://HowDoWeDoPythonPackaging)

#### #18 - 22 mai 2017 09:37 - Josué Kouka

- Fichier 0001-add-calendar-cell-model-16393.patch ajouté

Thomas Noël a écrit :

Eviter le git+http://git.entrouvert.org/... dans les requirements : pousser le paquet dans pypi → [HowDoWeDoPythonPackaging](http://HowDoWeDoPythonPackaging)

J'ai re-soumis le patch dans le requirement pointant vers le depot EO. Je l'avais mis parce qu'il y'avait déjà un version dans PyPi (XStatic-fullcalendar (2.2.2.1)) et cette version n'est malheureusement pas à jour.

## #19 - 22 mai 2017 09:45 - Frédéric Péters

J'ai re-soumis le patch dans le requirement pointant vers le depot EO. Je l'avais mis parce qu'il y'avait déjà un version dans PyPi (XStatic-fullcalendar (2.2.2.1)) et cette version n'est malheureusement pas à jour.

Il faut alors coordination avec l'auteur du module pour mise à jour sur pypi.

## #20 - 22 mai 2017 12:35 - Frédéric Péters

```
context['chrono'] = json.dumps({key: str(value) for key, value in vars(self).items()
                               if key in include})
```

Normalement il y a déjà une variable "cell" dans le contexte, qui donne accès à tout ça.

```
url(r'^chrono/events/(?P<pk>[\w,-]+)/', EventsView.as_view(), name='chrono-events'),
url(r'^chrono/book/(?P<pk>[\w,-]+)/', BookingView.as_view(), name='chrono-booking'),
```

Préfixer avec api/.

```
if hasattr(settings, 'KNOWN_SERVICES') and settings.KNOWN_SERVICES.get('chrono'):
    return settings.KNOWN_SERVICES['chrono'].values()[0]
```

Attention à ne pas prendre un service marqué "secondary" (cas du déploiement multipublik).

```
form_url = models.URLField(_('Application form URL'))
```

Non, proposer un choix dans les formulaires existants (cf combo.apps.wcs).

```
events_source = models.URLField(_('Events source URL'))
```

Non plus, proposer le choix dans les agendas existants.

```
minimal_event_duration = models.TimeField(
    _('Minimal event duration'), default=parse_time('02:00'))
```

Cet attribut ne me semble pas utilisé.

```
business_hours_start = models.TimeField(
    _('Business hour start'), default=parse_time('08:00'))
```

s ou pas à "hour" ?

```
event_default_title = models.CharField(
    _('Event default title'), max_length=32)
```

Dans les tests c'est utilisé comme étant event\_default\_title='Available' et ça ne me semble pas tellement correspondre à la description.

## #21 - 22 mai 2017 23:41 - Frédéric Péters

J'ai hésité à faire ce commentaire, parce que ça remet beaucoup sur la table, mais je ne suis pas sûr du tout que fullcalendar soit ce qu'on veuille ici. Je me disais surtout que ça faisait sans doute déjà tout, notamment un fonctionnement intelligent sur mobile, mais ce n'est absolument le cas, le glisser/tirer pour créer un événement y est tout à fait atroce à utiliser.

Outre que ça fonctionne trop mal sur mobile, la gestion des contraintes est peut-être possible mais pas encore présente dans le patch et pour évoluer je sens déjà les galères, on n'aura pas de maîtrise efficace du code, on hackera autour, je me dis qu'un système bien plus simple serait plus opportun : on dessinerait simplement un calendrier rempli de cases à cocher, avec du style par dessus et même un peu de js pour gérer les contraintes, puis un bouton et un petit mot disant "cliquez sur les créneaux souhaités puis sur le bouton de validation" et ça ferait le job tout à fait correctement.

(et peut-être assurer du js pour le changement fluide de semaine en semaine mais sur ce point j'hésiterais avec le code existant déjà pour recharger une cellule, mécanisme qui pourrait être réutilisé et combiné avec une lecture de la query string).

~~~

À essayer le patch pour de vrai, je parlais de cette ligne dans mon commentaire au-dessus :

```
context['chrono'] = json.dumps({key: str(value) for key, value in vars(self).items()
                               if key in include})
```

Mais avant même d'être inutile, elle ne marche tout simplement pas, le module json n'étant pas importé. (curieusement retiré en toute conscience quelques commentaires plus haut)

**#22 - 22 mai 2017 23:59 - Frédéric Péters**

un calendrier rempli de cases à cocher

Et en mode qui ne serait pas "permettre la jonction de plusieurs créneaux", ça serait simplement des liens. (mais bien que ce cas soit plus naturel pour chrono, laissons-le pour plus tard).

**#23 - 23 mai 2017 10:14 - Josué Kouka**

Frédéric Péters a écrit :

J'ai hésité à faire ce commentaire, parce que ça remet beaucoup sur la table, mais je ne suis pas sûr du tout que fullcalendar soit ce qu'on veuille ici. Je me disais surtout que ça faisait sans doute déjà tout, notamment un fonctionnement intelligent sur mobile, mais ce n'est absolument le cas, le glisser/tirer pour créer un événement y est tout à fait atroce à utiliser.

Outre que ça fonctionne trop mal sur mobile, la gestion des contraintes est peut-être possible mais pas encore présente dans le patch et pour évoluer je sens déjà les galères, on n'aura pas de maîtrise efficace du code, on hackera autour, je me dis qu'un système bien plus simple serait plus opportun : on dessinerait simplement un calendrier rempli de cases à cocher, avec du style par dessus et même un peu de js pour gérer les contraintes, puis un bouton et un petit mot disant "cliquez sur les créneaux souhaités puis sur le bouton de validation" et ça ferait le job tout à fait correctement.

(et peut-être assurer du js pour le changement fluide de semaine en semaine mais sur ce point j'hésiterais avec le code existant déjà pour recharger une cellule, mécanisme qui pourrait être réutilisé et combiné avec une lecture de la query string).

Ok je comprends ta position Fred. Et j'avoue que sur la gestion des contraintes c'est un peu galère, mais est ce que l'on peut commencer par ceci puis changer après.

Je comprends que c'est assez bizarre mais vu les contraintes je ne pense pas pouvoir tout refaire avant la fin de la semaine prochaine.

~ ~ ~

À essayer le patch pour de vrai, je parlais de cette ligne dans mon commentaire au-dessus :

[...]

Mais avant même d'être inutile, elle ne marche tout simplement pas, le module json n'étant pas importé. (curieusement retiré en toute conscience quelques commentaires plus haut)

Oui, une erreur de ma part.

**#24 - 23 mai 2017 10:16 - Frédéric Péters**

Ok je comprends ta position Fred. Et j'avoue que sur la gestion des contraintes c'est un peu galère, mais est ce que l'on peut commencer par ceci puis changer après.

S'il y avait eu quelque chose d'abouti, je n'aurais pas réagi. Ici ce n'est pas le cas du tout, je ne veux pas que ça passe en production, et il me semble qu'un plan simple à base de cases à cocher a plus de chances d'être réalisé correctement dans les temps, même si ça impose de reprendre une partie du travail.

**#25 - 23 mai 2017 10:35 - Josué Kouka**

Frédéric Péters a écrit :

Ok je comprends ta position Fred. Et j'avoue que sur la gestion des contraintes c'est un peu galère, mais est ce que l'on peut commencer par ceci puis changer après.

S'il y avait eu quelque chose d'abouti, je n'aurais pas réagi. Ici ce n'est pas le cas du tout, je ne veux pas que ça passe en production, et il me semble qu'un plan simple à base de cases à cocher a plus de chances d'être réalisé correctement dans les temps, même si ça impose de reprendre une partie du travail.

Ok je comprends.

**#26 - 23 mai 2017 10:53 - Serghei Mihai (congé, retour 15/05)**

Je viens de tomber sur ça: <http://www.jqueryscript.net/time-clock/Basic-Schedule-Calendar-Plugin-jQuery-Weekly-Scheduler.html>

IMHO, ça peut donner des idées pour placer les cases à cocher dans les zones sélectionnées et un bouton de soumission du formulaire.

## #27 - 23 mai 2017 11:04 - Frédéric Péters

IMHO, ça peut donner des idées pour placer les cases à cocher dans les zones sélectionnées et un bouton de soumission du formulaire.

Oui, un tableau avec les jours de la semaine en colonnes, et les créneaux en lignes, et dans chaque cellule, une case à cocher.

## #28 - 23 mai 2017 11:10 - Serghei Mihai (congé, retour 15/05)

J'imaginai même les cases à cocher cachées, mais faisons déjà avec les cases affichées, on verra pour le reste.

## #29 - 23 mai 2017 11:12 - Frédéric Péters

Je parlais d'HTML, CSS et JS après.

## #30 - 31 mai 2017 18:11 - Josué Kouka

- Fichier 0001-add-calendar-cell-model-16393.patch ajouté

## #31 - 31 mai 2017 18:41 - Frédéric Péters

Je n'appellerais pas ça Chrono, simplement calendar.

~~~

```
Combo calendar cell
=====
```

Manque deux =

Example of session\_var definition

Ça débarque plutôt soudainement et sans contexte. Je taperais plutôt en help\_text du champ en question, qui devrait s'appeler formdef\_url\_params, genre, qu'il y a des substitutions possibles, \$start prendra date/heure de début et \$end date/heure de fin.

Cela étant, pas la meilleure idée de siècle d'inventer un nouveau mécanisme de substitution; on a déjà importé un minimaliste ezt pour faire [whatever] dans les URL, et il y a le mécanisme de templates de django. Etc.

and all keys in the session\_vars dict will be prefixed by a <session\_var\_[key]>.

Moby.

```
print(formdef_references)
```

Ligne de debug en trop.

```
from .utils import get_calendar, get_agendas, get_formdefs
```

get\_formdefs.

```
class CalendarCellForm(forms.ModelForm):
...
class CalendarForm(forms.Form):
```

Lavender.

```
verbose_name = _('Calendar Cell')
```

Sérieusement j'encourage à lire les relectures et les patches, même quand tu n'es pas concerné, ça ne fait pas dix jours que concernant les cartes je disais qu'il fallait 'Map' et pas 'Map Cell'.

combo/apps/chrono/static/chrono/css/calendar.css

Je me passerais de ce fichier.

combo/apps/chrono/static/chrono/js/calendar.js

Plutôt lors d'un changement, marquer disabled les cases qui ne peuvent pas être cochées.

```
<h2>{{title}}</h2>
```

Permettre le titre vide, et donc ici mettre uniquement le <h2>... si un titre est défini.

```
<div id="calendar_{{cell.pk}}" data-calendar="{{cell.pk}}">
```

```

<form method="POST" action="{% url 'chrono-booking' cell.pk %}">
    {% csrf_token %}
    {{ form.as_div }}
    <input type="submit" value="Submit"/>
</form>
</div>

```

Dans Publik les formulaires s'affichent avec `.as_p()` (cf <http://styleguide.entrouvert.com/motifs/formulaires/>).

Mais ici je suis avant tout très surpris par l'absence d'itération sur des jours de la semaine, sur des créneaux horaires, je m'attendrais vraiment à un `<table>`, qui est la bonne structure pour cette info.

```

url(r'^chrono/events/(?P<pk>[\w,-]+)/', EventsView.as_view(), name='chrono-events'),
url(r'^chrono/book/(?P<pk>[\w,-]+)/', BookingView.as_view(), name='chrono-booking'),

```

api/ ou ajax/ en préfixe, pas chrono.

```
def humanize_date(datetime):
```

humanize c'est déjà utilisé et c'est autre chose. Et puis cette fonction ne semble pas être appelée.

```
+ 'xstatic.pkg.fullcalendar',
```

Non,

```
+ python-xstatic-fullcalendar (>= 3.4.0)
```

non

```
+ 'XStatic-fullcalendar>=3.4.0'
```

et non.

### #32 - 02 juin 2017 17:59 - Josué Kouka

- Fichier 0001-add-calendar-cell-model-16393.patch ajouté

### #33 - 06 juin 2017 11:14 - Frédéric Péters

J'écrivais :

```
from .utils import get_calendar, get_agendas, get_formsdef
```

```
get_formdefs.
```

Pas pris en compte, je ne relis rien d'autre.

### #34 - 06 juin 2017 11:16 - Frédéric Péters

Patient, je continue quand même, et :

```
def humanize_date(datetime):
```

humanize c'est déjà utilisé et c'est autre chose. Et puis cette fonction ne semble pas être appelée.

Pas pris en compte.

### #35 - 06 juin 2017 11:33 - Josué Kouka

- Fichier 0001-add-calendar-cell-model-16393.patch ajouté

Frédéric Péters a écrit :

Patient, je continue quand même, et :

```
def humanize_date(datetime):
```

humanize c'est déjà utilisé et c'est autre chose. Et puis cette fonction ne semble pas être appelée.

Pas pris en compte.

+1, avec la modification

#### #36 - 06 juin 2017 11:44 - Frédéric Péters

```
from .utils import get_calendar, get_agendas, get_formsdef
```

```
get_formdefs.
```

Pas pris en compte, je ne relis rien d'autre.

Toujours pas.

#### #37 - 06 juin 2017 12:01 - Frédéric Péters

Attention à bien prendre en compte l'ensemble des commentaires, ou à noter pourquoi ils ne le sont pas.

#### #38 - 06 juin 2017 14:24 - Serghei Mihai (congrès, retour 15/05)

- *Echéance mis à 09 juin 2017*

#### #39 - 06 juin 2017 17:43 - Josué Kouka

- *Fichier 0001-add-calendar-cell-model-16393.patch ajouté*

Je pense avoir fait le tour des demandes et tout devrait y être.

#### #40 - 07 juin 2017 10:39 - Frédéric Péters

```
combo/apps/chrono/templates/__init__.py
```

Fichier inutile.

Je n'appellerais pas ça Chrono, simplement calendar.

Toujours de cet avis, et la cellule même, 'Booking Calendar'.

```
title = models.CharField(_('Title'), max_length=128)
```

Dans la plupart des cellules (et ça devrait être le cas partout) on ne met pas le titre obligatoire.

```
agenda_reference = models.URLField(_('Events source URL'))
formdef_reference = models.URLField(_('Application form URL'))
```

Non, ça ne doit pas être des URL mais une référence indépendante, l'idée c'est qu'à l'export/import vers une autre plateforme, la référence vers le formulaire "réservation" de "wcs" continue à fonctionner. (cf combo.apps.wcs écrivais-je). Et pareil pour l'agenda.

```
        ('business_hour_start', models.TimeField(default=datetime.time(8, 0), verbose_name='Business h
our start')),
        ('business_hour_end', models.TimeField(default=datetime.time(18, 0), verbose_name='Business ho
ur end')),
```

Je ne vois plus nulle part ces attributs utilisés.

```
# assure that at least slot is selected
if len(self.data.keys()) == 1:
    raise ValueError(_('Please select slots'))
```

Ça rend l'affaire difficile à comprendre d'avoir la vérif sur 1 et non 0, je comprends que ça vient parce que dans data tu as aussi csrfmiddlewaretoken mais ici, je commencerais la méthode par alimenter une liste avec l'ensemble des slots, sur base d'un préfixe commun.

Dans cette méthode, aussi, il ne me semble pas y avoir une vérification sur la continuité de la sélection.

```
@csrf_exempt
```

Mais si l'objectif est de ne pas avoir de vérification CSRF (pourquoi ?) le plus simple c'est quand même aussi de ne pas mettre le {% csrf\_token %} dans le formulaire.



```
def save(self, *args, **kwargs):
    if not self.slug:
        self.slug = slugify(self.title)
    return super(CalendarCell, self).save(*args, **kwargs)
```

Non, on ne veut pas pour cette cellule que le slug soit rempli automatiquement, l'uniformité est importante.

```
{% trans "previews week" %}
```

Et un tas d'erreurs mineures de ce type ou de style, je garde pour plus tard, en me disant qu'à force de relecture tu pourras les voir par toi-même. (genre le </div> en trop, il a une incidence visuelle non négligeable).

Pour une fonction appelée une seule fois, genre `is_datetime_diff_valid`, plutôt que la placer dans un autre fichier, son code pourrait être intégré directement dans la vérification de validité.

```
data['start'] = values[0]
data['end'] = values[-1]
```

Ça va faire une comparaison entre le début du premier slot et le début du dernier slot, ça va donc échouer si on prend juste le minimum de nombre de slots.

```
for choice in calendar.object_list[0][1].values()[0]:
    calendar_table_headers.append(choice['label'])
```

Pas sûr de comprendre le raisonnement ici mais le résultat est mauvais :

```
(Pdb) calendar.object_list[0][1].values()
[[{'disabled': 'disabled', 'value': '2017-06-14T08:00:00', 'label': u'14 juin 2017'}, {'disabled': '', 'value': '2017-06-14T08:30:00', 'label': u'14 juin 2017'}, {'disabled': '', 'value': '2017-06-14T09:00:00', 'label': u'14 juin 2017'}, {'disabled': '', 'value': '2017-06-16T10:00:00', 'label': u'16 juin 2017'}]]
(Pdb) p context['calendar_table_headers']
[u'14 juin 2017']
```

Comme entête il faudrait aussi le 16 juin.

#### #41 - 08 juin 2017 06:39 - Josué Kouka

- Fichier `0001-add-calendar-cell-model-16393.patch` ajouté

J'ai pris en compte les remarques citées plus haut, par contre pour la dernière j'obtiens le resultat attendu

```
(Pdb) calendar.object_list[0][1].values()[0]
[{'disabled': False, 'value': '2017-06-12T08:00:00', 'label': u'12 juin 2017'}, {'disabled': False, 'value': '2017-06-13T08:00:00', 'label': u'13 juin 2017'}, {'disabled': False, 'value': '2017-06-14T08:00:00', 'label': u'14 juin 2017'}, {'disabled': False, 'value': '2017-06-15T08:00:00', 'label': u'15 juin 2017'}, {'disabled': False, 'value': '2017-06-16T08:00:00', 'label': u'16 juin 2017'}]
(Pdb) context['calendar_table_headers']
[u'12 juin 2017', u'13 juin 2017', u'14 juin 2017', u'15 juin 2017', u'16 juin 2017']
(Pdb)
```

#### #42 - 08 juin 2017 22:51 - Frédéric Péters

```
(Pdb) calendar
<Page 1 of 2>
(Pdb) calendar.object_list
((24, OrderedDict([('08:00', [{'disabled': 'disabled', 'value': '2017-06-14T08:00:00', 'label': u'14 juin 2017'}])), ('08:30', [{'disabled': '', 'value': '2017-06-14T08:30:00', 'label': u'14 juin 2017'}])), ('09:00', [{'disabled': '', 'value': '2017-06-14T09:00:00', 'label': u'14 juin 2017'}])), ('10:00', [{'disabled': '', 'value': '2017-06-16T10:00:00', 'label': u'16 juin 2017'}])))),)
(Pdb) calendar.object_list[0]
(24, OrderedDict([('08:00', [{'disabled': 'disabled', 'value': '2017-06-14T08:00:00', 'label': u'14 juin 2017'}])), ('08:30', [{'disabled': '', 'value': '2017-06-14T08:30:00', 'label': u'14 juin 2017'}])), ('09:00', [{'disabled': '', 'value': '2017-06-14T09:00:00', 'label': u'14 juin 2017'}])), ('10:00', [{'disabled': '', 'value': '2017-06-16T10:00:00', 'label': u'16 juin 2017'}]))))
(Pdb) calendar.object_list[0][1]
OrderedDict([('08:00', [{'disabled': 'disabled', 'value': '2017-06-14T08:00:00', 'label': u'14 juin 2017'}]), ('08:30', [{'disabled': '', 'value': '2017-06-14T08:30:00', 'label': u'14 juin 2017'}]), ('09:00', [{'disabled': '', 'value': '2017-06-14T09:00:00', 'label': u'14 juin 2017'}]), ('10:00', [{'disabled': '', 'value': '2017-06-16T10:00:00', 'label': u'16 juin 2017'}])])
(Pdb) calendar.object_list[0][1].values()
[[{'disabled': 'disabled', 'value': '2017-06-14T08:00:00', 'label': u'14 juin 2017'}, {'disabled': '', 'value': '2017-06-14T08:30:00', 'label': u'14 juin 2017'}, {'disabled': '', 'value': '2017-06-14T09:00:00', 'label': u'14 juin 2017'}, {'disabled': '', 'value': '2017-06-16T10:00:00', 'label': u'16 juin 2017'}]]
(Pdb) calendar.object_list[0][1].values()[0]
```

```
[{'disabled': 'disabled', 'value': '2017-06-14T08:00:00', 'label': u'14 juin 2017'}]
```

Et avant ça (pas pratique de réutiliser le même nom de variable :

```
(Pdb) p get_calendar(cell.get_calendar())
OrderedDict([(24, OrderedDict([('08:00', [{'disabled': 'disabled', 'value': '2017-06-14T08:00:00', 'label': u'14 juin 2017'}]), ('08:30', [{'disabled': '', 'value': '2017-06-14T08:30:00', 'label': u'14 juin 2017'}]), ('09:00', [{'disabled': '', 'value': '2017-06-14T09:00:00', 'label': u'14 juin 2017'}]), ('10:00', [{'disabled': '', 'value': '2017-06-16T10:00:00', 'label': u'16 juin 2017'}])]))])
```

Je trouve le code tout bonnement impossible à comprendre, et donc à déboguer.

#### #43 - 08 juin 2017 22:54 - Frédéric Péters

Je n'appellerais pas ça Chrono, simplement calendar.

Toujours de cet avis, et la cellule même, 'Booking Calendar'.

Pas explicite mais ce n'est pas juste le nom de la cellule, c'est également le nom de la classe.

#### #44 - 09 juin 2017 10:19 - Josué Kouka

- Fichier 0001-add-calendar-cell-model-16393.patch ajouté

J'ai pris en compte les remarques.

- Renommage de la cellule
- Le thead est calculé lors de la construction de calendrier

#### #45 - 09 juin 2017 10:21 - Frédéric Péters

calendar = get\_calendar(cell.get\_calendar()), je n'ai pas regardé le reste mais vraiment, une ligne pareille doit amener un gros questionnement sur le nommage.

#### #46 - 09 juin 2017 10:25 - Frédéric Péters

Non, on ne veut pas pour cette cellule que le slug soit rempli automatiquement, l'uniformité est importante.

Toujours vrai.

#### #47 - 09 juin 2017 10:43 - Josué Kouka

- Fichier 0001-add-calendar-cell-model-16393.patch ajouté

#### #48 - 09 juin 2017 10:52 - Frédéric Péters

Ok, j'ai capté où ça merde :

```
(Pdb) p calendar
OrderedDict([(24, OrderedDict([('08:00', [{'disabled': True, 'value': '2017-06-14T08:00:00', 'label': u'14 juin 2017'}]), ('08:30', [{'disabled': False, 'value': '2017-06-14T08:30:00', 'label': u'14 juin 2017'}]), ('09:00', [{'disabled': False, 'value': '2017-06-14T09:00:00', 'label': u'14 juin 2017'}]), ('10:00', [{'disabled': False, 'value': '2017-06-15T10:00:00', 'label': u'15 juin 2017'}, {'disabled': False, 'value': '2017-06-16T10:00:00', 'label': u'16 juin 2017'}]), ('days', [u'14 juin 2017', u'15 juin 2017', u'16 juin 2017'])]))])
(Pdb) p calendar[24]
OrderedDict([('08:00', [{'disabled': True, 'value': '2017-06-14T08:00:00', 'label': u'14 juin 2017'}]), ('08:30', [{'disabled': False, 'value': '2017-06-14T08:30:00', 'label': u'14 juin 2017'}]), ('09:00', [{'disabled': False, 'value': '2017-06-14T09:00:00', 'label': u'14 juin 2017'}]), ('10:00', [{'disabled': False, 'value': '2017-06-15T10:00:00', 'label': u'15 juin 2017'}, {'disabled': False, 'value': '2017-06-16T10:00:00', 'label': u'16 juin 2017'}]), ('days', [u'14 juin 2017', u'15 juin 2017', u'16 juin 2017'])])
(Pdb) p calendar[24].keys()
['08:00', '08:30', '09:00', '10:00', 'days']
```

Et donc, exemple, si un créneau n'existe pas le premier jour, il y a décalage dans le tableau pour la suite :

```
(Pdb) p calendar[24]['08:00']
[{'disabled': True, 'value': '2017-06-14T08:00:00', 'label': u'14 juin 2017'}]
```

```
(Pdb) p calendar[24]['10:00']
[{'disabled': False, 'value': '2017-06-15T10:00:00', 'label': u'15 juin 2017'}, {'disabled': False, 'value': '2017-06-16T10:00:00', 'label': u'16 juin 2017'}]
```

Pour la tranche "8h" la liste commence le 14 juin alors qu'elle commence le 15 juin pour la tranche "10h".

Je pense qu'il faut vraiment reprendre cette partie sur base de réelles structures, dans lesquelles tu pourras avoir des méthodes etc. plutôt que te baser sur des imbrications de dictionnaires et listes.

#### #49 - 09 juin 2017 11:02 - Frédéric Péters

Je pense qu'il faut vraiment reprendre cette partie sur base de réelles structures, dans lesquelles tu pourras avoir des méthodes etc. plutôt que te baser sur des imbrications de dictionnaires et listes.

Et ainsi, consolider une liste de jours et de créneaux, qui serviront pour la construction d'un tableau, qui doit ressembler à :

	12 juin	13 juin	14 juin	15 juin	16 juin
08:00		[ ]		[ ]	[ ]
08:30		[ ]		[ ]	[ ]
09:00	[ ]	[ ]		[ ]	[ ]
09:30	[ ]	[ ]		[ ]	[ ]
...					

#### #50 - 09 juin 2017 11:05 - Frédéric Péters

Ce qui correspond va-t-en comprendre comment à ce que j'écrivais il y a 17 jours :

un tableau avec les jours de la semaine en colonnes, et les créneaux en lignes, et dans chaque cellule, une case à cocher.

#### #51 - 13 juin 2017 20:02 - Josué Kouka

- Fichier 0001-add-calendar-cell-model-16393.patch ajouté

#### #52 - 14 juin 2017 08:36 - Frédéric Péters

Relecture uniquement, pas testé en vrai cette fois-ci.

```
+ def is_valid(self):
+     slots = getattr(self.data, 'getlist', lambda x: [])(self.slots)

+     # check that at least one slot is selected
+     if not slots:
+         raise ValueError(_('Please select slots'))

+     offset = self.cell.slot_duration.hour * 60 + self.cell.slot_duration.minute
+     start_dt = parse_datetime(slots[0])
+     end_dt = parse_datetime(slots[-1]) + datetime.timedelta(minutes=offset)
+     slots.append(end_dt.isoformat())

+     # check that all slots are part of the same day
+     for slot in slots:
+         if parse_datetime(slot).date() != start_dt.date():
+             raise ValueError(_('Please select slots of the same day'))

+     # check that slots datetime are contiguous
+     start = start_dt
+     while start <= end_dt:
+         if start.isoformat() not in slots:
+             raise ValueError(_('Please select contiguous slots'))
+         start = start + datetime.timedelta(minutes=offset)

+     # check that event booking duration >= minimal booking duration
+     min_duration = self.cell.minimal_event_duration.hour * 60 + self.cell.minimal_event_duration.minute
+     if not (end_dt - start_dt) >= datetime.timedelta(minutes=min_duration):
+         raise ValueError(_('Minimal booking duration is %s' % self.cell.minimal_event_duration.strftime('%H:%M')))

+     self.cleaned_data['start'] = start_dt.isoformat()
+     self.cleaned_data['end'] = end_dt.isoformat()
+     return True
```

Aérer là où ça ajoute de la lisibilité; ma suggestion ci-dessus.

```
+ formdef_url_params = JSONField(
+     _('Session vars'),
+     help_text=_('{{start}} will take booking start datetime and {{end}} will take booking end datetime'),
+     default={"session_var_start": "{{start}}", "session_var_end": "{{end}}"})
```

Je serais pour virer ce paramétrage obscur, qu'on commence simplement en filant à chaque fois booking\_start, booking\_end et booking\_agenda\_slug, il sera toujours temps plus tard de rendre ça paramétrable.

(booking\_agenda\_slug, c'est le slug qui se trouve dans agenda\_reference, pas le slug de la cellule.)

```
+ template_name = 'calendar/calendar_cell.html'
```

Suivons le changement de nom et appelons ça booking\_calendar\_cell.html.

```
+<div id="cal-{{cell.pk}}">
```

Pas tout à fait sûr de l'utilité d'un identifiant.

```
+ <div>
+     {% block calendar_table %}
+     {% calendar_table cell=cell %}
+     {% endblock %}
+ </div>
```

Je m'interroge quand même un peu sur le besoin de template tag ici.

```
+<div>
+     {% if calendar.has_other_pages %}
+     <p class="paginator">
+         {% if calendar.has_previous %}
+         <a href="?week_{{cell.slug}}={{calendar.previous_page_number}}">{% trans "previous week" %}</a>
```

Dans un commentaire précédent je dis que le slug n'est pas un élément obligatoire pour les cellules; du coup très bien il n'est plus généré automatiquement mais ça veut aussi dire qu'il ne peut pas être utilisé ici.

Comme plus haut, tu peux utiliser cell.id.

Sur le fond je continue à trouver curieux l'utilisation du mécanisme de pagination pour naviguer entre semaines, mais je passe.

```
+<div id="calendar_{{cell.pk}}" data-calendar="{{cell.pk}}">
+     <form id="form-cal-{{cell.pk}}" method="POST" action="{% url 'calendar-booking' cell.pk %}">
```

Ici non plus, pas bien sûr de l'utilité de taper des id sur les éléments.

```
+ <table id="cal-table-{{cell.pk}}">
+     {% for cal in calendar %}
+     <thead>
+         <tr>
+             {% for day in cal.days %}
+             <th>{{day.date|date:"SHORT_DATE_FORMAT"}}</th>
+             {% endfor %}
+         </tr>
+     </thead>
+     <tbody>
+         {% for slot in cal.get_slots %}
+         <tr>
```

Avoir une première colonne avec les libellés,

```
+         {% for day in cal.days %}
+         <td>
+             {% if day|available:slot %}
+             {% with value=day|get_slot:slot %}
+             <input type="checkbox" name="slots" value="{{value.label}}" id="{{value.label}}"/>
+             <label for="{{value.label}}">{{slot}}</label>
```

Qu'on ne répète pas 8:30 8:30 8:30 8:30 sur les lignes. (i.e. ici mettre uniquement la case à cocher).

J'écrivais :

12 juin	13 juin	14 juin	15 juin	16 juin
---------	---------	---------	---------	---------

```
08:00          [ ]          [ ]          [ ]
08:30          [ ]          [ ]          [ ]
09:00    [ ]    [ ]          [ ]          [ ]
09:30    [ ]    [ ]          [ ]          [ ]
...
```

Point.

```
+         <input type="submit" value="{% trans 'Validate' %}">
```

Plutôt faire cliquer les gens sur un bouton "Réserver".

```
+     url(r'^api/calendar/events/(?P<pk>[\w,-]+)/', EventsView.as_view(), name='calendar-events'),
```

J'ai l'impression que cet endpoint n'est plus utilisé; si ?

```
+     url(r'^api/calendar/book/(?P<pk>[\w,-]+)/', BookingView.as_view(), name='calendar-booking'),
```

Et cet endpoint n'est pas/plus une API, /calendar/book/... marcherait.

```
+def get_service(service_name, key=None):
+    if hasattr(settings, 'KNOWN_SERVICES') and settings.KNOWN_SERVICES.get(service_name):
+        services = settings.KNOWN_SERVICES[service_name]
+        if key:
+            return services.get(key)
+        return services
```

C'est curieux cette méthode qui à la fois retourne un service et plusieurs services. Plutôt avoir `get_service` et `get_services`.

```
+def get_agendas():
+    chronos = get_service('chrono')
+    references = []
+    for chrono_key, chrono_site in chronos.items():
+        url = urlparse.urljoin(chrono_site['url'], 'api/agenda/')
+        response = requests.get(url, headers={'accept': 'application/json'})
```

Plutôt que jouer à assembler des URL puis demander à requests de trouver le service approprié, faire :

```
response = requests.get('/api/agenda', remote_service=chrono_site, ...)
```

```
+     for agenda in response.json()['data']:
+         references.append((
+             '%s:%s' % (chrono_key, agenda['slug']), agenda['text']))
+     return references
```

En l'absence de données ça va renvoyer None plutôt qu'une liste vide.

```
+def get_formdefs():
+    wcs = get_service('wcs')
+    references = []
+    for wcs_key, wcs_site in wcs.items():
+        url = urlparse.urljoin(wcs_site['url'], 'api/formdefs/')
+        response = requests.get(url, headers={'accept': 'application/json'})
+        data = response.json()
+        if isinstance(data, (dict,)):
+            if data.get('err') == 1:
+                continue
```

Pour être sûr, juste `data.get('err')` (ça pourrait être 2, ou 3, etc.)

```
+         data = data.get('data')
```

Et mon vote pour la PEP qui propose `data` comme méthode équivalente au `get`.

```
+     for form in data:
+         references.append((
+             '%s:%s' % (wcs_key, form['slug']), form['title']))
+     return references
```

Pareil (jeu sur URL + retour de liste).

<

```
+def get_chrono_events(agenda_reference, **kwargs):
+    chrono_key, chrono_slug = agenda_reference.split(':')
```

```
+ chrono = get_service('chrono', key=chrono_key)
+ url = urlparse.urljoin(chrono['url'], 'api/agenda/%s/datetimes/' % chrono_slug)
+ response = requests.get(url, headers={'accept': 'application/json'}, **kwargs)
+ return response.json().get('data', [])
```

Pareil jeu sur URL.

Attention aussi, Chrono va retourner des slots qui ne sont pas disponibles, avec disabled: True, ça pourrait être géré ici mais on perdrait alors les vraies bornes de journée, c'est donc mieux de gérer ailleurs, mais ça doit être géré.

```
+def get_slots_range(start, end, offset):
+    dstart = start
+    # offset in minute
+    offset = offset.hour * 60 + offset.minute
+    while dstart <= end:
+        yield dstart.strftime('%H:%M')
```

Utiliser TIME\_FORMAT (ici et ailleurs) pour avoir un format d'heure localisé.

```
+def get_calendar(agenda_reference, offset):
+    calendar = {}
```

Très bien pour à l'intérieur avoir WeekCalendar mais c'est dommage d'avoir laissé cette fonction retourner un dictionnaire. (dont la seule utilisation par la suite me semble être d'y faire un .values())

```
+ # add day to week calendar
+ if not week_cal.has_day(event_datetime.date()):
+     day = WeekDay(event_datetime.date())
+     week_cal.days.append(day)
```

Attention tu accèdes ici directement à l'intérieur de la structure.

```
+ def __repr__(self):
+     return '%s - %s' % (self.date_time.isoformat(), self.available)
```

Je trouve pratique de laisser aux repr() une tête de repr(), par exemple ici : '<DaySlot date\_time=%s available=%s>'

```
+ def get_days(self):
+     return self.days
```

Pour moi get\_days devrait comme get\_slots produire une liste intégrale, que chaque semaine contienne sept jours. (et éventuellement un jour un paramétrage sur la cellule pour ne pas inclure les weekends, ou autres options pour contrôler plus finement ça).

```
+class EventsView(SingleObjectMixin, View):
+
+    http_method_names = ['get']
+    model = BookingCalendar
+
+    def get(self, request, *args, **kwargs):
+        cell = self.get_object()
+        data = get_chrono_events(cell.agenda_reference)
+        return JsonResponse(data, safe=False)
```

Le safe=False fait un peu peur; je préférerais simplement que la réponse soit JsonResponse({'data': data}). Mais je notais plus haut que je ne voyais pas à quoi cette URL servait.

```
+ @csrf_exempt
```

J'écrivais : « Mais si l'objectif est de ne pas avoir de vérification CSRF (pourquoi ?) ».

Je repose la question, pourquoi ?

```
+ 'chrono': {
+     'default': {'title': 'test', 'url': 'http://example.org',
```

Je ne réutiliserais pas <http://example.org>; au plus simple <http://chrono.example.org> serait ok.

```
+CHRONO_EVENTS = {
+    "data": [
+        {
+            "disabled": True,
+            "text": "19 mai 2017 08:00",
```

Curieux que dans tes tests tu marques tous les slots comme n'étant pas disponibles. Ça rejoint la remarque plus haut comme quoi faut gérer ça, et pointe qu'il manque des tests vérifiant le rendu.

Et pour finir, un des avantages d'une structure comme ici WeekCalendar, c'est aussi que \_a permet d'ajouter des tests unitaires dédiés, pour facilement vérifier toute une série de comportements.

#### #53 - 14 juin 2017 22:24 - Josué Kouka

- Fichier 0001-add-calendar-cell-model-16393.patch ajouté

#### #54 - 15 juin 2017 05:53 - Josué Kouka

Je serais pour virer ce paramétrage obscur, qu'on commence simplement en filant à chaque fois booking\_start, booking\_end et booking\_agenda\_slug, il sera toujours temps plus tard de rendre ça paramétrable.

Je l'ai gardé comme tel parce que l'on peut avoir besoin de passer des infos supplémentaires i.e le nom d'une crèche par exemple.

#### #55 - 15 juin 2017 09:27 - Frédéric Péters

Je l'ai gardé comme tel parce que l'on peut avoir besoin de passer des infos supplémentaires i.e le nom d'une crèche par exemple.

Si vraiment le truc est utile (pas convaincu, le slug de l'agenda préciserait la crèche), qu'il serve pour des paramètres additionnels, mais que de base, avec le champ vide, il y ait booking\_start, booking\_end et booking\_agenda\_slug.

#### #56 - 15 juin 2017 10:31 - Josué Kouka

- Fichier 0001-add-calendar-cell-model-16393.patch ajouté

Frédéric Péters a écrit :

Je l'ai gardé comme tel parce que l'on peut avoir besoin de passer des infos supplémentaires i.e le nom d'une crèche par exemple.

Si vraiment le truc est utile (pas convaincu, le slug de l'agenda préciserait la crèche), qu'il serve pour des paramètres additionnels, mais que de base, avec le champ vide, il y ait booking\_start, booking\_end et booking\_agenda\_slug.

#### #57 - 15 juin 2017 15:46 - Frédéric Péters

J'écrivais bien plus tôt :

Attention à ne pas prendre un service marqué "secondary" (cas du déploiement multipublik).

Et en local j'ai encore trace d'un patch qui vérifiait "secondary".

Mais ce patch-ci, non, et je l'ajoute pour tester et bim ça crashe.

```
File "/home/fred/src/eo/combo/combo/manager/templatetags/cells.py" in cell_form
 28.         context['form'] = form_class(instance=cell, prefix='c%s' % cell.get_reference())
File "/home/fred/src/eo/combo/combo/apps/calendar/forms.py" in __init__
 37.         agenda_references = get_agendas()
File "/home/fred/src/eo/combo/combo/apps/calendar/utils.py" in get_agendas
 56.         for agenda in result['data']:
```

```
Exception Type: KeyError at /manage/pages/28/
Exception Value: 'data'
```

J'ajoute de quoi ignorer les services secondaires et même trace, parce que l'appel se fait en passant un Nameld vide dans la query string.

J'ajoute without\_user pour ne pas passer ça et enfin je vois le paramétrage de la cellule.

"Session vars" que je trouve toujours aussi peu utile, a minimal, le taper au fond, en dernier paramètre inutile ?

Je vais voir ce que donne la page, il n'y a rien dedans, sauf un bouton "Livrer" (il faut ajouter du contexte à la chaîne pour la traduction, cf <https://docs.djangoproject.com/en/dev/topics/i18n/translation/>, "{% trans %}" also supports contextual markers using the context keyword"). Il faudrait aussi sans doute ne rien afficher du tout quand il n'y a pas d'événements trouvés, pas juste un bouton inutile.

Suite, pourquoi vide, même histoire, Nameld passé vide. → nécessaire d'ajouter des tests pour vérifier que tout fonctionne aussi bien en étant loggué que non.

Je modifie donc ça pour avancer et deux cases à cocher apparaissent désormais, je clique dessus, ça me dit que je ne peux pas parce que c'est le même jour, bien.

J'ajoute des événements pour un autre jour, celui qui est plein n'apparaît pas, bien, mais par contre, celui que j'ai mis dans la continuité de la journée n'apparaît pas non plus. C'est parce que l'heure du slot final est celle du dernier jour, il faudrait que peu importe, qu'on puisse avoir un vendredi se terminant à midi sans obliger les autres jours à se terminer à midi aussi. (tests à ajouter sur cette situation).

À continuer à regarder, il y a un <div> vide quand il n'y a pas de pagination, qui prend de l'espace inutilement. Aussi, le style des cellules assure que le div qui suit l'h2 aura un padding correct, mais tu mets toi deux div, ça fait que le tableau colle aux bords.

#### #58 - 15 juin 2017 16:21 - Frédéric Péters

Et parce que je me trouve toujours bien obligé de relire mes commentaires,

ici non plus, pas bien sûr de l'utilité de taper des id sur les éléments.

Pas modifié, pas commenté.

#### #59 - 15 juin 2017 16:22 - Frédéric Péters

```
+def get_services(service_name, key=None):  
+    if hasattr(settings, 'KNOWN_SERVICES') and settings.KNOWN_SERVICES.get(service_name):  
+        services = settings.KNOWN_SERVICES[service_name]  
+        return services
```

key, inutile, retour de None par défaut pas opportun.

etc.

#### #60 - 15 juin 2017 20:02 - Josué Kouka

- Fichier 0001-add-calendar-cell-model-16393.patch ajouté

Les requêtes sont désormais faites sans envoi du NameID

Et en local j'ai encore trace d'un patch qui vérifiait "secondary".

Je modifie donc ça pour avancer et deux cases à cocher apparaissent désormais, je clique dessus, ça me dit que je ne peux pas parce que c'est le même jour, bien.

Je n'arrive pas à reproduire ce comportement en local malheureusement.

À continuer à regarder, il y a un <div> vide quand il n'y a pas de pagination, qui prend de l'espace inutilement. Aussi, le style des cellules assure que le div qui suit l'h2 aura un padding correct, mais tu mets toi deux div, ça fait que le tableau colle aux bords.

J'ai mis le tout dans une seule div

"Session vars" que je trouve toujours aussi peu utile, a minimal, le taper au fond, en dernier paramètre inutile ?

Supprimé.

J'ajoute des événements pour un autre jour, celui qui est plein n'apparaît pas, bien, mais par contre, celui que j'ai mis dans la continuité de la journée n'apparaît pas non plus. C'est parce que l'heure du slot final est celle du dernier jour, il faudrait que peu importe, qu'on puisse avoir un vendredi se terminant à midi sans obliger les autres jours à se terminer à midi aussi. (tests à ajouter sur cette situation).

Corrigé.

#### #61 - 16 juin 2017 00:02 - Frédéric Péters

Je parlais de déplacer session vars en dernier attribut et là il est tout à fait supprimé, sans commentaire; quid des arguments qui étaient donnés pour le conserver ?

J'en suis à templatetags.py et cal\_slot\_available veut pas dire grand chose et derrière c'est en fait juste utilisé pour donner une valeur qui est en fait majoritairement ignorée parce que seul compte au final le value.available dedans.

```
<p class="paginator">
```

Je ne vois pas de balise qui fermerait ça.



(par ailleurs je sais que niveau style ça va être moche, << ou >>, sans moyen d'adapter ça via CSS, mais je passe cet aspect).

```
def get_services(service_name, exclude_secondary=False):
```

Je ne comprends pas comment on en arrive à un truc si compliqué, et à nouveau sans doute c'est amplifié par la manière de nommer les variables :

```
    for service, items in services.items():
```

En fait ici services ce serait les diffères servicedu type (service\_name) passé; et donc service ce serait un service particulier et items une liste de trucs mais qui est une liste mais qui en fait est un seul truc ? Très sérieusement je connais la structure de KNOWN\_SERVICES et cette fonction me laisse pantois.

```
    if isinstance(result, (dict,)):
```

Truc de forme que je comptais laisser passer mais la remarque étant faite, if instance(result, dict): est bien plus lisible.

```
def get_days(self):
    if self.days:
        fday = self.days[0].date
    else:
        fday = datetime.datetime.today()
```

Ici aussi question de nommer les variables. fday ? kesako ? Ça représente quoi ?

```
def test_cell_rendering(mocked_get, app, cell):
    ...
def test_cell_rendering_when_authenticated(mocked_get, app, cell, admin):
```

Les résultats me semblant devoir être tout à fait identiques, ça serait mieux calibrer selon moi si une fixture assurait les tests en diverses configurations d'User (a minima je laisserais User + le Use rcréé dans nulle part ailleurs et cie.).

#### #62 - 16 juin 2017 02:43 - Josué Kouka

- Fichier 0001-add-events-importation-from-csv-16428.patch ajouté

Frédéric Péters a écrit :

Je parlais de déplacer session vars en dernier attribut et là il est tout à fait supprimé, sans commentaire; quid des arguments qui étaient donnés pour le conserver ?

A 2 reprises tu as mis dans tes commentaire que tu n'en voyais pas l'utilité.

(par ailleurs je sais que niveau style ça va être moche, << ou >>, sans moyen d'adapter ça via CSS, mais je passe cet aspect).

Une suggestion ?

Ici aussi question de nommer les variables. fday ? kesako ? Ça représente quoi ?

ça représente la date sur laquelle je me base pour construire tous les autres jours de la semaine. J'ai renommé en base\_day

#### #63 - 16 juin 2017 09:45 - Frédéric Péters

(mauvais patch attaché)

A 2 reprises tu as mis dans tes commentaire que tu n'en voyais pas l'utilité.

Et tu as dit que c'était nécessaire pour passer une info crèche. Et maintenant ça ne l'est plus. C'est ce changement sans explication que je trouve déconcertant; le dernier commentaire demandait juste de déplacer ce champ en dernier.

Mais bien sûr, ça me va très bien qu'il n'existe plus.

(par ailleurs je sais que niveau style ça va être moche, << ou >>, sans moyen d'adapter ça via CSS, mais je passe cet aspect).

Une suggestion ?

Pour adapter par CSS, taper des classes (previous/next) (comme il y a une classe "current" actuellement).

Aussi, sur cette partie, quand on peut aller en arrière il est écrit "previous week" mais quand on ne peut pas il est écrit "<<", et même chose pour "next week" quand c'est un lien et ">>" quand on est au bout. C'est curieux. (== il faut utiliser le même libellé sur le span et le a);

~~

Ça crashe quand ça essaie d'afficher la cellule alors que celle-ci n'est pas configurée.

```
File "/home/fred/src/eo/combo/combo/apps/calendar/models.py", line 58, in render
    calendar = self.get_calendar()
File "/home/fred/src/eo/combo/combo/apps/calendar/models.py", line 52, in get_calendar
    return get_calendar(self.agenda_reference, self.slot_duration)
File "/home/fred/src/eo/combo/combo/apps/calendar/utils.py", line 97, in get_calendar
    events = get_chrono_events(agenda_reference)
File "/home/fred/src/eo/combo/combo/apps/calendar/utils.py", line 85, in get_chrono_events
    chrono_key, chrono_slug = agenda_reference.split(':')
ValueError: need more than 1 value to unpack
```

~~

À formater les champs "slot duration" et "minimal event duration" pour que l'édition se fasse au format hh:mm et non hh:mm:ss. Et m'intéresser à ces champs, une première chose c'est que le deuxième pourrait plutôt être appelé "minimal booking duration" et plus important, que ça ne devrait pas être des TimeField mais des DurationField.

À venir, il y aurait aussi à taper des classes CSS permettant de différencier sur une case du tableau les trois états possibles (pas de slot existant, slot existant mais complet, slot avec de la place).

#### #64 - 16 juin 2017 11:01 - Frédéric Péters

Aussi, utiliser `{{value.label}}` comme id sur les cases à cocher n'est pas ok parce que ça pourrait faire des doublons en présence de plusieurs cellules sur la page.

Je suggères là-dessus, pour en plus gagner un `<label>` permettant une plus grande souplesse de style :

```
-                                     <input type="checkbox" name="slots" value="{{value.label}}" id="{{value.l
    label}}"/>
+                                     <input type="checkbox" name="slots" value="{{value.label}}" id="slot-{{ce
    ll.id}}-{{value.label}}"/>
+                                     <label for="slot-{{cell.id}}-{{value.label}}"></label>
```

#### #65 - 16 juin 2017 11:05 - Frédéric Péters

À l'usage aussi, je supprimerais les ancrs sur les liens précédent/suivant,

```
-      <a href="?week_{{cell.pk}}={{ calendar.previous_page_number }}#cal-{{cell.pk}}">{% trans "previous we
    ek" %}</a>
+      <a href="?week_{{cell.pk}}={{ calendar.previous_page_number }}">{% trans "previous week" %}</a>
```

et

```
-      <a href="?week_{{cell.pk}}={{ calendar.next_page_number }}#cal-{{cell.pk}}">{% trans "next week" %}</
    a>
+      <a href="?week_{{cell.pk}}={{ calendar.next_page_number }}">{% trans "next week" %}</a>
```

Parce que sur un calendrier visible sans scroller, en haut de page, ça fait un comportement désagréable.

#### #66 - 16 juin 2017 11:09 - Frédéric Péters

Aussi, pour des questions de style (parce que ça permet `::after` et `::before`), on doit favoriser l'utilisation de `<button>`, plutôt que `<input type="submit">`.

```
-      <input type="submit" value="{% trans "Book" context "booking" %}">
+      <button class="submit-button">{% trans "Book" context "booking" %}</button>
```

#### #67 - 16 juin 2017 11:14 - Frédéric Péters

- Lié à Development #16936: style pour les cellules "calendrier de réservation" ajouté

#### #68 - 16 juin 2017 12:19 - Josué Kouka

- Fichier 0001-add-calendar-cell-model-16393.patch ajouté

Je pense avoir fait toutes les modifications demandées

À formater les champs "slot duration" et "minimal event duration" pour que l'édition se fasse au format hh:mm et non hh:mm:ss. Et m'intéresser à ces champs, une première chose c'est que le deuxième pourrait plutôt être appelé "minimal booking duration" et plus important, que ça ne devrait pas être des TimeField mais des DurationField.

Je n'ai pas formaté en H:M les DurationField à cause de avec [https://docs.djangoproject.com/fr/1.8/modules/django/utils/dateparse/#parse\\_duration](https://docs.djangoproject.com/fr/1.8/modules/django/utils/dateparse/#parse_duration) qui conseille %d %H:%M:%S.%f.

En définissant un slot\_duration comme 00:30 (cas avec widget qui formate en H:M), parse\_duration considère que ce sont des secondes. J'ai ajouté un help\_text décrivant le format attendu.

#### #69 - 16 juin 2017 12:45 - Frédéric Péters

help\_text='H:M:S', c'est vraiment cheap et ça manque de trad.

J'écrivais :

Aussi, sur cette partie, quand on peut aller en arrière il est écrit "previous week" mais quand on ne peut pas il est écrit "<<", et même chose pour "next week" quand c'est un lien et ">>" quand on est au bout. C'est curieux. (== il faut utiliser le même libellé sur le span et le a);

Et tu prends plutôt l'option de ne plus inclure de <span> sur les bornes; ok ok mais ce n'était pas ce que je demandais et le patch ajoutant du style convenable à la cellule n'est du coup plus bon...

```
def get_chrono_events (agenda_reference, **kwargs):
```

kwargs n'est pas utilisé.

```
    return {}
    return result.get('data', [])
```

Dans un cas ça retourne un dictionnaire dans une autre une liste, ça ne va pas.

```
# offset = self.offset.hour * 60 + self.offset.minute
```

Du vieux code en commentaire ?

```
        raise ValueError(_(
            'Minimal booking duration is %s' % str_min_duration))
```

Il faut passer à gettext la chaîne avant la substitution. (gettext ne trouvera pas dans son catalogue la chaîne 'Minimal booking duration is 60')

À venir, il y aurait aussi à taper des classes CSS permettant de différencier sur une case du tableau les trois états possibles (pas de slot existant, slot existant mais complet, slot avec de la place).

Rien à ce sujet, si tu gardes ça pour plus tard, ok, mais tu dois créer un ticket immédiatement.

~~

```
def get_chrono_service():
    for chrono_key, chrono_site in get_services('chrono').iteritems():
        if chrono_site.get('secondary', True):
            return {chrono_key: chrono_site}
    return {}
```

Et pourquoi donc tu t'es dit que modifier cette partie-ci maintenant était une bonne idée, je ne sais pas, mais voilà donc que ça ne prend plus le chrono primaire, et c'est un commentaire déjà fait il y a 25 jours, et répété hier, et corrigé, puis cassé. Tu dois donc corriger et ajouter un test pour que ça n'arrive plus.

#### #70 - 16 juin 2017 13:04 - Frédéric Péters

J'écrivais :

Aussi, le style des cellules assure que le div qui suit l'h2 aura un padding correct, [...]

Et sans raison non plus, la dernière version du patch retire le <div>, perd donc le padding correct, etc.

#### #71 - 16 juin 2017 13:08 - Frédéric Péters

```
def get_formdefs():
```

Prend les démarches de tous les w.c.s., mais ne les différencie pas à l'affichage. (dans combo.apps.wcs on inclut le titre du site quand il y en a plusieurs; d'ailleurs on pourrait se demander pourquoi mal réimplémenter ce code ici plutôt qu'utiliser le code existant).

#### #72 - 16 juin 2017 14:39 - Josué Kouka

- Fichier 0001-add-calendar-cell-model-16393.patch ajouté

Et tu prends plutôt l'option de ne plus inclure de `<span>` sur les bornes; ok ok mais ce n'était pas ce que je demandais et le patch ajoutant du style convenable à la cellule n'est du coup plus bon...

J'ai remis les span avec `<span>{% trans "next week" %}`

À venir, il y aurait aussi à taper des classes CSS permettant de différencier sur une case du tableau les trois états possibles (pas de slot existant, slot existant mais complet, slot avec de la place).

J'ai rajouté 2 classes, available et unavailable parce que dans le cas où le slot n'existe pas ou est complet, je renvoie un objet avec available à False.

#### #73 - 16 juin 2017 15:41 - Frédéric Péters

`help_text='H:M:S'`, c'est vraiment cheap et ça manque de trad.

C'est toujours aussi cheap. `_( 'Format is hour:minutes:seconds' )` le serait moins.

```
+         <span>{% trans "previous week" %}</span>
```

Manque une classe. (pareil sur le next)

~~

```
+def test_get_chrono_service(settings):
+    service = get_chrono_service()
+    for key, site in service.iteritems():
+        assert key == 'default'
```

C'est très curieux et compliqué pour la lecture de voir ce `get_chrono_service()` singulier suivi par une boucle.

C'est pareil dans la méthode `get_agendas()`.

`get_chrono_service()`, il devrait retourner

```
{
    'title': 'test',
    'url': 'http://chrono.example.org',
    ...
},
```

et non pas :

```
{
    'default': {
        'title': 'test',
        'url': 'http://chrono.example.org',
        ...
    },
}
```

#### #74 - 16 juin 2017 15:57 - Frédéric Péters

À venir, il y aurait aussi à taper des classes CSS permettant de différencier sur une case du tableau les trois états possibles (pas de slot existant, slot existant mais complet, slot avec de la place).

J'ai rajouté 2 classes, available et unavailable parce que dans le cas où le slot n'existe pas ou est complet, je renvoie un objet avec available à False.

Et si je parle de trois états, ce n'est pas pour avoir deux classes, c'est pour en avoir trois. Il est vraiment utile de pouvoir différencier un créneau plein (genre rouge) d'un créneau qui n'existe pas (genre gris).

#### #75 - 16 juin 2017 16:15 - Josué Kouka

- Fichier `0001-add-calendar-cell-model-16393.patch` ajouté

Frédéric Péters a écrit :

À venir, il y aurait aussi à taper des classes CSS permettant de différencier sur une case du tableau les trois états possibles (pas de slot existant, slot existant mais complet, slot avec de la place).

J'ai rajouté 2 classes, available et unavailable parce que dans le cas où le slot n'existe pas ou est complet, je renvoie un objet avec available à False.

Et si je parle de trois états, ce n'est pas pour avoir deux classes, c'est pour en avoir trois. Il est vraiment utile de pouvoir différencier un créneau plein (genre rouge) d'un créneau qui n'existe pas (genre gris).

J'ai ajouté la classe absent pour les créneaux non existants en plus des modifs liées aux précédents commentaires.

#### #76 - 16 juin 2017 17:03 - Josué Kouka

- Fichier 0001-add-calendar-cell-model-16393.patch ajouté

Rajout d'un help\_text oublié

#### #77 - 16 juin 2017 17:57 - Frédéric Péters

```
{% else %}
    {% if value.available %}
```

Faut utiliser elif plutôt qu'imbriquer.

```
+ for agenda in result.get('data'):
+     references.append((
+         '%s:%s' % (chrono['title'], agenda['slug']), agenda['text']))
+     return references
```

Non, on ne peut pas perdre la référence à un agenda au moindre changement de titre du site chrono.

#### #78 - 16 juin 2017 18:18 - Josué Kouka

- Fichier 0001-add-calendar-cell-model-16393.patch ajouté

Frédéric Péters a écrit :

[...]

Faut utiliser elif plutôt qu'imbriquer.

[...]

Non, on ne peut pas perdre la référence à un agenda au moindre changement de titre du site chrono.

Ok, j'inclus la clé du service au moment où je récupère le chrono\_site.

#### #79 - 16 juin 2017 18:36 - Frédéric Péters

Pour poser une clé, on fait plutôt :

```
- chrono_site.update({'slug': chrono_key})
+ chrono_site['slug'] = chrono_key

+ wcs_key, wcs_slug = cell.formdef_reference.split(':')
+ wcs = get_wcs_services().get(wcs_key)
+ response = requests.get('api/formdefs/', remote_service=wcs, without_user=True)
+ for formdef in response.json():
+     if formdef['slug'] == wcs_slug:
+         break
+ url = '%s/?%s' % (formdef['url'], urllib.urlencode(session_vars))
+ return url
```

Il me semble inutile d'appeler w.c.s. ici, on a l'url de base, on y ajoute le slug de la démarche, et ça nous fait l'url; genre :

```
url = '%s/%s/?%s' % (wcs['url'], wcs_slug, urllib....
```

~~

Je dois encore tester.

## #80 - 16 juin 2017 18:46 - Josué Kouka

- Fichier 0001-add-calendar-cell-model-16393.patch ajouté

```
url = '%s/%s/%s' % (wcs['url'], wcs_slug, urllib....
```

+1

## #81 - 16 juin 2017 19:15 - Frédéric Péters

```
url = '%s/%s/%s' % (wcs['url'], wcs_slug, urllib....
```

+1

Nope, en vrai ça fait deux / à la racine.

~~

À l'utilisation :

Sur une erreur au submit la page retourne à la première semaine.

Des cellules du tableau sont marquées avec "unavailable" alors que le créneau n'existe même pas (et devrait donc avoir la classe "absent"). Parce que ça tape "absent" uniquement quand le jour est totalement vide (si je lis get\_availability() correctement).

## #82 - 16 juin 2017 19:31 - Josué Kouka

- Fichier 0001-add-calendar-cell-model-16393.patch ajouté

Frédéric Péters a écrit :

```
url = '%s/%s/%s' % (wcs['url'], wcs_slug, urllib....
```

+1

Nope, en vrai ça fait deux / à la racine.

~~

Merci

À l'utilisation :

Sur une erreur au submit la page retourne à la première semaine.

Des cellules du tableau sont marquées avec "unavailable" alors que le créneau n'existe même pas (et devrait donc avoir la classe "absent"). Parce que ça tape "absent" uniquement quand le jour est totalement vide (si je lis get\_availability() correctement).

Corrigé

```
@@ -143,7 +143,7 @@ class WeekDay(object):
     if slot.date_time.time() == slot_time:
         return slot
     slot_datetime = datetime.datetime.combine(self.date, slot_time)
-     return DaySlot(slot_datetime, False)
+     return DaySlot(slot_datetime, False, exist=False)
```

## #83 - 16 juin 2017 19:38 - Josué Kouka

- Fichier 0001-add-calendar-cell-model-16393.patch ajouté

Suppression du debug

#### #84 - 16 juin 2017 19:42 - Frédéric Péters

Sur une erreur au submit la page retourne à la première semaine.

Pas de changement à ce niveau.

De l'interdiff :

```
- <td class="absent"></td>
+ <td class="absent">Absent</td>
```

Je ne comprends vraiment pas pourquoi tu continues à faire des modifications sans rapport, ça fait que ça n'en finit pas, par exemple ici je dois te dire que ça manque de traduction. Mais surtout que ça va être moche comme tout un tableau plein de "Absent" et qu'il vaudrait mieux défaire ça.

```
- <td class="unavailable"></td>
+ <td class="unavailable">Unavailable</td>
```

Pareil.

```
- url = '%s/%s/?%s' % (wcs['url'], wcs_slug, urllib.urlencode(session_vars))
+ url = '/%s/%s/?%s' % (wcs['url'], wcs_slug, urllib.urlencode(session_vars))
```

Il est temps d'aller dormir.

#### #85 - 16 juin 2017 19:43 - Frédéric Péters

Je ne comprends vraiment pas pourquoi tu continues à faire des modifications sans rapport, ça fait que ça n'en finit pas, par exemple ici je dois te dire que ça manque de traduction. Mais surtout que ça va être moche comme tout un tableau plein de "Absent" et qu'il vaudrait mieux défaire ça.

C'était donc du debug...

Le reste de mon commentaire reste d'application.

#### #86 - 16 juin 2017 20:04 - Josué Kouka

- Fichier 0001-add-calendar-cell-model-16393.patch ajouté

- retour sur la bonne semaine après une erreur
- url de redirection ok

#### #87 - 17 juin 2017 13:05 - Frédéric Péters

url de redirection ok

Non, tu as remis le code qui n'allait pas, qui provoque deux / à la racine.

```
https://wcs.example.net//foobar/?session_var_booking_agenda_slug=reunions-dinformation&session_var_booking_start=2017-06-21T08:30:00&session_var_booking_end=2017-06-21T09:30:00
                        ^^
```

→

```
--- a/tests/test_calendar.py
+++ b/tests/test_calendar.py
@@ -213,6 +213,7 @@ def test_cell_rendering(mocked_get, client, cell):
     resp.form.set('slots', True, 2)
     resp = resp.form.submit()
     parsed = urlparse.urlparse(resp.url)
+   assert parsed.path == '/test/'
     qs = urlparse.parse_qs(parsed.query)
     assert qs['session_var_booking_agenda_slug'] == ['test']
     assert qs['session_var_booking_start'] == ['2017-06-13T08:00:00']
```

(et dans les tests, en passant, il est astucieux de ne pas tout nommer de la même manière, avoir "test" à la fois comme slug de la page, de l'agenda

et du formdef, ça permettrait de passer à côté de bug où l'un est pris pour un autre.)

#### #88 - 17 juin 2017 15:32 - Josué Kouka

Frédéric Péters a écrit :

url de redirection ok

Non, tu as remis le code qui n'allait pas, qui provoque deux / à la racine.

J'ai étrangement en local la bonne url

```
http://wcs.debian.local/demande-de-place-en-creche/?session_var_booking_agenda_slug=southpark&session_var_booking_start=2017-06-29T09%3A00%3A00&session_var_booking_end=2017-06-29T10%3A00%3A00
```

En ajoutant le / devant voici le resultat en local

```
http://combo.debian.local/http://wcs.debian.local/demande-de-place-en-creche/?session_var_booking_agenda_slug=southpark&session_var_booking_start=2017-06-29T09%3A00%3A00&session_var_booking_end=2017-06-29T10%3A00%3A00
```

Surement un pepin chez moi, je cherche son origine.

[...]

→

[...]

(et dans les tests, en passant, il est astucieux de ne pas tout nommer de la même manière, avoir "test" à la fois comme slug de la page, de l'agenda et du formdef, ça permettrait de passer à côté de bug où l'un est pris pour un autre.)

ok

#### #89 - 17 juin 2017 15:36 - Frédéric Péters

- Lié à Development #16955: javascript de support pour la cellule calendrier de réservation ajouté

#### #90 - 17 juin 2017 15:43 - Frédéric Péters

Pourtant pas compliqué d'ajouter :

```
+ assert parsed.path == '/test/'
```

Ça prend l'url ici :

```
'default': {'title': 'test', 'url': 'http://127.0.0.1:8999/'},
```

Ça ajoute '/test/...'.

Ça fait donc <http://127.0.0.1:8999/test/>...

J'ai mis des mots, j'ai mis un exemple en pointant sur les deux /, j'ai mis une ligne à ajouter au test.

Je ne comprends pas ce qui t'a mis en tête qu'une solution serait d'« ajouter un / devant ».

#### #91 - 17 juin 2017 16:05 - Josué Kouka

- Fichier 0001-add-calendar-cell-model-16393.patch ajouté

Frédéric Péters a écrit :

Pourtant pas compliqué d'ajouter :

[...]

Ça prend l'url ici :

[...]

Ça ajoute '/test/...'.



Ça fait donc [http://127.0.0.1:8999/test/...](http://127.0.0.1:8999/test/)

J'ai mis des mots, j'ai mis un exemple en pointant sur les deux /, j'ai mis une ligne à ajouter au test.

Je ne comprends pas ce qui t'a mis en tête qu'une solution serait d'« ajouter un / devant ».

Excuses moi, je t'ai très mal lu.

#92 - 17 juin 2017 16:13 - Frédéric Péters

```
- url = '%s/%s/?%s' % (wcs['url'], wcs_slug, urllib.urlencode(session_vars))
+ url = urlparse.urljoin(wcs['url'], wcs_slug)
+ url += '/?%s' % urllib.urlencode(session_vars)
```

Soupir de complexité; il s'agit **simplement** de retirer le / entre le wcs['url'] (qui le contient déjà) et la suite :

```
- url = '%s/%s/?%s' % (wcs['url'], wcs_slug, urllib.urlencode(session_vars))
+ url = '%s%s/?%s' % (wcs['url'], wcs_slug, urllib.urlencode(session_vars))
```

~~

Autre chose, invitation à créer un ticker pour corriger ça plus tard, la vérification is\_valid() va foirer avec un élément à 23h30. (parce que 30 minutes sont ajoutées et cet élément 30 minutes plus tard est ajouté à la liste des slots et à la vérification "même jour", ça foire).

#93 - 17 juin 2017 16:21 - Josué Kouka

- Lié à Bug #16956: Cellule Calendar: reservation invalide pour créneaux réservés proche de minuit. ajouté

#94 - 17 juin 2017 16:25 - Josué Kouka

- Fichier 0001-add-calendar-cell-model-16393.patch ajouté

Je resoumets le patch en le faisant simplement

#95 - 18 juin 2017 13:07 - Frédéric Péters

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

Modifié pour :

- réécrire le message de commit pour mentionner booking calendar ;
- virer le README qui n'apporte rien ;
- retirer la méthode is\_available() qui n'est jamais appelée.

```
commit 56b47f5a50e7fc550f863ed77ce8d684fd84e685
Author: Josue Kouka <jkouka@entrouvert.com>
Date: Wed May 17 18:19:07 2017 +0200
```

```
general: add booking calendar cell (#16393)
```

#96 - 10 août 2017 18:48 - Josué Kouka

- Statut changé de Résolu (à déployer) à Fermé

Fichiers

0002-add-chrono-app-view-and-utils-16393.patch	6,14 ko	18 mai 2017	Josué Kouka
0001-add-calendar-cell-model-16393.patch	6,71 ko	18 mai 2017	Josué Kouka
0003-add-calendar-cell-static-files-16393.patch	1,07 Mo	18 mai 2017	Josué Kouka
0001-add-calendar-cell-model-16393.patch	19,9 ko	18 mai 2017	Josué Kouka
0001-add-calendar-cell-model-16393.patch	18,8 ko	19 mai 2017	Josué Kouka
0001-add-calendar-cell-model-16393.patch	23,2 ko	19 mai 2017	Josué Kouka
0001-add-calendar-cell-model-16393.patch	23,3 ko	19 mai 2017	Josué Kouka
0001-add-calendar-cell-model-16393.patch	23,4 ko	19 mai 2017	Josué Kouka
0001-add-calendar-cell-model-16393.patch	23,4 ko	19 mai 2017	Josué Kouka
0001-add-calendar-cell-model-16393.patch	23 ko	22 mai 2017	Josué Kouka
0001-add-calendar-cell-model-16393.patch	24,2 ko	31 mai 2017	Josué Kouka
0001-add-calendar-cell-model-16393.patch	27,9 ko	02 juin 2017	Josué Kouka

0001-add-calendar-cell-model-16393.patch	27,8 ko	06 juin 2017	Josué Kouka
0001-add-calendar-cell-model-16393.patch	27,9 ko	06 juin 2017	Josué Kouka
0001-add-calendar-cell-model-16393.patch	31,4 ko	08 juin 2017	Josué Kouka
0001-add-calendar-cell-model-16393.patch	31,5 ko	09 juin 2017	Josué Kouka
0001-add-calendar-cell-model-16393.patch	30,9 ko	09 juin 2017	Josué Kouka
0001-add-calendar-cell-model-16393.patch	33 ko	13 juin 2017	Josué Kouka
0001-add-calendar-cell-model-16393.patch	33,4 ko	14 juin 2017	Josué Kouka
0001-add-calendar-cell-model-16393.patch	33,7 ko	15 juin 2017	Josué Kouka
0001-add-calendar-cell-model-16393.patch	35,6 ko	15 juin 2017	Josué Kouka
0001-add-events-importation-from-csv-16428.patch	7,36 ko	16 juin 2017	Josué Kouka
0001-add-calendar-cell-model-16393.patch	34,1 ko	16 juin 2017	Josué Kouka
0001-add-calendar-cell-model-16393.patch	34,5 ko	16 juin 2017	Josué Kouka
0001-add-calendar-cell-model-16393.patch	34,7 ko	16 juin 2017	Josué Kouka
0001-add-calendar-cell-model-16393.patch	34,8 ko	16 juin 2017	Josué Kouka
0001-add-calendar-cell-model-16393.patch	34,7 ko	16 juin 2017	Josué Kouka
0001-add-calendar-cell-model-16393.patch	34,6 ko	16 juin 2017	Josué Kouka
0001-add-calendar-cell-model-16393.patch	34,6 ko	16 juin 2017	Josué Kouka
0001-add-calendar-cell-model-16393.patch	34,6 ko	16 juin 2017	Josué Kouka
0001-add-calendar-cell-model-16393.patch	34,7 ko	16 juin 2017	Josué Kouka
0001-add-calendar-cell-model-16393.patch	34,8 ko	17 juin 2017	Josué Kouka
0001-add-calendar-cell-model-16393.patch	34,7 ko	17 juin 2017	Josué Kouka