

Chrono - Development #41663

définition d'événements récurrents

14 avril 2020 11:56 - Frédéric Péters

Statut:	Fermé	Début:	14 avril 2020
Priorité:	Normal	Echéance:	
Assigné à:	Valentin Deniaud	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
Genre s'il y a un cinéclub tous les lundis.			
Détails de ce qui est possible dans la spec RRULE des ics, et UI à chercher dans des outils existants (exemples de capture attachée).			
Demandes liées:			
Lié à Publik - Development #49203: Événements récurrents		Fermé	08 décembre 2020
Précède Chrono - Development #51218: Évènements récurrents : pouvoir spécifie...		Fermé	

Révisions associées

Révision 7234aa74 - 17 février 2021 15:49 - Valentin Deniaud

agendas: do not save seconds in event start_datetime (#41663)

Révision a392213d - 17 février 2021 15:49 - Valentin Deniaud

agendas: make returning a queryset in get_open_events useless (#41663)

Révision a699e144 - 17 février 2021 15:49 - Valentin Deniaud

add support for recurring events (#41663)

Révision e0928162 - 17 février 2021 15:49 - Valentin Deniaud

manager: backport SplitDateTimeField fix (#41663)

Révision b0d89df3 - 17 février 2021 15:49 - Valentin Deniaud

manager: handle edition/deletion of recurring event (#41663)

Historique

#1 - 14 avril 2020 12:11 - Pierre Cros

Mon cas d'usage : mon association réserve la salle de mini-foot tous les lundis soirs, je veux pouvoir le faire avec une seule demande en indiquant un créneau et une durée de répétition (ou avoir un an par défaut comme durée de répétition, genre)

#2 - 14 avril 2020 12:30 - Mikaël Ates

Je ne sais pas si mon cas se prête à ce ticket.

Cas d'usage : Cours de langue.

Je veux gérer les inscriptions à l'événement "Cours d'Anglais tous les lundi de 14h30 à 16h à partir du 13 avril jusqu'au 6 juillet".

Je crée aujourd'hui un unique événement avec pour date le 13 avril.

J'ai dans mon WF un statut inscriptions du jour qui permet de voir toutes les personnes en activité aujourd'hui dans la vue de listing tous événements confondus. Ceci ne fonctionne donc que pour la première séance.

Le pointage des présences à chaque séance n'est pas non plus possible.

J'ai envisagé de créer un événement par cours mais cela demande de créer tous ces événements. Ils vont ensuite tous apparaître dans la liste des activités alors que je ne souhaiterais afficher qu'une entrée. La gestion avec une demande d'inscription à un de ces événements qui générerait autant de demandes d'inscription paraît complexe et ce sera inapproprié pour pointer un unique paiement. Avoir une unique demande semble donc plutôt à privilégier mais l'affichage inscription du jour ou le pointage resterait difficile.

Je ne sais pas si la gestion de la récurrence pourrait apporter des solutions à ce cas d'usage.

#3 - 14 avril 2020 12:35 - Frédéric Péters

J'ai envisagé de créer un événement par cours mais cela demande de créer tous ces événements.

Ce ticket est pour éviter ça, qu'il y ait un événement créé, duquel découle automatiquement les autres.

Par contre, tu sembles dans la suite dire que les inscriptions sont "mutualisées", ce ticket concernait dans l'idée des événements indépendants (l'utilisateur ne s'inscrit pas à tous les cinéclubs), mais ce serait une suite possible.

#5 - 10 septembre 2020 14:07 - Emmanuel Cazenave

- Fichier *google-agenda-1.png* ajouté

- Fichier *google-agenda-2.png* ajouté

Je me perds pas mal dans le défrichage de ce ticket, en particulier dans le lien à faire avec la spec RRULE qui introduit la définition d'exceptions.

D'autant qu'en termes d'interface, je trouve celles qui n'exposent pas d'exceptions plus facile d'usage. Genre google calendar (capture ci-jointes) où on définit une récurrence lors de la création de l'événement, et on peut ensuite supprimer les événements en 'trop'.

#6 - 10 septembre 2020 14:14 - Emmanuel Cazenave

Mais pour contre balancer mon commentaire précédent sur le fait ne pas exposer la notion d'exception, il la demande qui me semble arriver gros comme une maison : j'ai supprimé le lundi 5 octobre parce que le gymnase devait être repeint mais finalement ça n'a pas lieu, comment je fais pour remettre le 5 octobre dans la série ?

#7 - 10 septembre 2020 14:53 - Valentin Deniaud

Emmanuel Cazenave a écrit :

Mais pour contre balancer mon commentaire précédent sur le fait ne pas exposer la notion d'exception, il la demande qui me semble arriver gros comme une maison : j'ai supprimé le lundi 5 octobre parce que le gymnase devait être repeint mais finalement ça n'a pas lieu, comment je fais pour remettre le 5 octobre dans la série ?

Un bouton « désactiver » plutôt que « supprimer » ?

#8 - 08 décembre 2020 07:55 - Stéphane Guiet

Publik Famille

Les temps d'accueils (= événements) d'une activité doivent être créés par lots avec une récurrence :

toutes les semaines, une fois tous les 15 jours, les 2ème et 4ème jeudis d'un mois = la période est variable

Cas d'usage : générer en une fois l'ensemble des temps repas de la restauration scolaire (les L-M-J-V) pour l'année scolaire.

Cas d'usage : générer les séances de Yoga du mardi soir une fois tous les 15 jours (S1 et S3 du mois) et du jeudi (S2 et S4).

#9 - 08 décembre 2020 09:03 - Frédéric Péters

Cas d'usage : générer les séances de Yoga du mardi soir une fois tous les 15 jours (S1 et S3 du mois) et du jeudi (S2 et S4).

Ok, pour toi tous les 15 jours c'est deux fois par mois; pour moi c'était "une semaine sur deux". (la différence se fait sur les mois qui voient parfois 5x le même jour de semaine, le "une semaine sur deux" demande de définir une date de départ). Est-ce que le "une semaine sur deux" est une récurrence à avoir ?

#10 - 22 décembre 2020 11:29 - Valentin Deniaud

- Assigné à mis à Valentin Deniaud

- Priorité changé de Bas à Normal

#11 - 24 décembre 2020 17:43 - Valentin Deniaud

J'ai codé quelques trucs pour me rendre compte, la direction que ça prend pour info :

- Un événement définit comme récurrent est l'évènement « parent » d'évènements « fils ». Les événements fils héritent du paramétrage du parent (label, places, durée...).
- L'évènement parent est visible dans le paramétrage, les événements fils sont exposés à la réservation et dans la vue mensuelle.
- Au moment de la création d'un événement récurrent, aucun événement fils n'est créé. Il y a une méthode qui permet d'obtenir à la volée la liste des événements fils entre deux bornes temporelles, pour les exposer dans /datetimes/ et la vue mensuelle.

- Un évènements fils est créé en base la première fois qu'on tente de le réserver ou d'y accéder dans le manager (pour genre l'annuler, j'imagine).

Ça m'a l'air de bien fonctionner et ça ne modifie pas de code existant (juste des ajouts).

À noter que ce que je décris là s'applique aux évènements « cinéclub tous les lundis ». La structure pour un évènement famille à inscription récurrente sera la même mais avec une logique différente, en cela qu'il faudra montrer l'évènement parent dans /datetimes/ et y rattacher les réservations, les évènements fils serviront juste au pointage etc.

#12 - 07 janvier 2021 18:10 - Valentin Deniaud

Pour info, version presque bien sur la branche avec une bonne partie des tests. Il reste à en écrire quelques uns pour tout couvrir, et à ajouter l'import/export.

Niveau spec RRULE je me contente d'utiliser dateutil.

Niveau interface, je propose un bête select avec des options courantes. Il y aurait ensuite à développer une vue « avancée » de paramétrage de la récurrence, j'imagine qu'il faudra pas mal de JS pour avoir quelque chose d'utilisable.

Prochaines choses à faire, pour de futurs tickets sauf avis contraires :

- Paramétrage de la durée de répétition (date de début et de fin)
- Permettre les exceptions (avoir les jours fériés automatiques ce serait pas mal)
- Permettre une inscription récurrente pour la famille
- Regarder si il y a quelque chose à faire au niveau de l'import des ICS

#13 - 12 janvier 2021 16:33 - Valentin Deniaud

- Fichier 0001-agendas-do-not-save-seconds-in-event-start_datetime-.patch ajouté

- Fichier 0005-manager-handle-edition-deletion-of-recurring-event-4.patch ajouté

- Fichier 0003-trivial-factorize-event-forms-41663.patch ajouté

- Fichier 0004-add-support-for-recurring-events-41663.patch ajouté

- Fichier 0002-agendas-make-returning-a-queryset-in-get_open_events.patch ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Hop !

0001 : les évènements créés normalement via le manager n'ont pas de secondes mais ceux dans les tests créés vite fait avec start_datetime=now(), si. Ça m'embête à un endroit, d'où ce patch sans conséquence.

0002 : adaptation nécessaires pour get_open_events en prévision de la suite.

0003 : nécessaire à un moment pendant mon dev et puis plus vraiment, mais c'est pas méchant comme patch alors autant le garder.

0004 : le gros du boulot.

0005 : un bout du boulot que j'ai réussi à isoler, assez capital tout de même pour ne pas tout planter via le manager.

Bon à savoir pour la lecture du code, quand il est écrit « recurring event » ça désigne l'évènement parent sur lequel est définie une récurrence, et « event recurrence » c'est un des évènements fils qui en découlent.

#14 - 27 janvier 2021 18:01 - Emmanuel Cazenave

(premier round léger, suite plus tard).

0002

Dans chrono/agendas/models.py ça aurait pu être plus minimaliste en si le nom des nouveaux paramètres min_start et max_start ne reprenait pas des noms de variables interne à la fonction, là du coup ça fait un patch avec des lignes liées à ce renommage et ça obscurcit un peu l'essentiel je trouve.

L'utilisation de prefetched_queryset=True conjointement avec annotate_queryset=True provoquerait une erreur, je lèverai une erreur explicite dans ce cas avant que ça crash salement (genre un assert).

0004

Pas compris ce qui suit, quid d'un évènement récurrent dont la première occurrence a lieu dans un mois avec date de publication dans 15 jours ?

```
def in_bookable_period(self):
    if self.recurrence_rule is not None:
        return True
```

#15 - 27 janvier 2021 18:15 - Valentin Deniaud

- Statut changé de Solution proposée à En cours

Emmanuel Cazenave a écrit :

(premier round léger, suite plus tard).

Bonnes remarques, attends que je corrige ça pour continuer, je vais en plus rebaser sur [#50420](#) qui va faire sauter 0003 (et puis j'ai une idée pour enlever le bout moche de 0004 autour de existing_datetimes).

#16 - 28 janvier 2021 12:38 - Valentin Deniaud

- Fichier 0001-agendas-do-not-save-seconds-in-event-start_datetime-.patch ajouté
- Fichier 0005-manager-handle-edition-deletion-of-recurring-event-4.patch ajouté
- Fichier 0003-add-support-for-recurring-events-41663.patch ajouté
- Fichier 0002-agendas-make-returning-a-queryset-in-get_open_events.patch ajouté
- Fichier 0004-manager-backport-SplitDateTimeField-fix-41663.patch ajouté
- Statut changé de En cours à Solution proposée

Emmanuel Cazenave a écrit :

0002

Dans chrono/agendas/models.py ça aurait pu être plus minimaliste en si le nom des nouveaux paramètres min_start et max_start ne reprenait pas des noms de variables interne à la fonction, là du coup ça fait un patch avec des lignes liées à ce renommage et ça obscurcit un peu l'essentiel je trouve.

Pas réussi à ne pas renommer mais réussi à avoir un diff plus joli.

L'utilisation de prefetched_queryset=True conjointement avec annotate_queryset=True provoquerait une erreur, je lèverai une erreur explicite dans ce cas avant que ça crash salement (genre un assert).

J'ai plutôt pris le parti d'ignorer annotate_queryset si prefetched_queryset=True.

0004

Pas compris ce qui suit, quid d'un évènement récurrent dont la première occurrence a lieu dans un mois avec date de publication dans 15 jours ?

Il faut avoir en tête que les évènements récurrents (self.recurrence_rule is not None) ne sont visibles que dans la page de paramétrage. Donc le problème que cette ligne résout c'est que si un évènement est récurrent à partir du premier janvier tous les jours, c'est moche de l'afficher comme hors de la période de réservation passé le premier janvier, puisqu'en vrai on peut réserver le 2, le 3, le 4 etc.

Par contre oui pour bien faire il faut faire attention à la date du début de récurrence, si on est le 1er octobre autant que cet évènement soit marqué comme hors de la période, je corrige en ce sens (et l'introduction de la date de fin de récurrence par la suite ira compléter cette partie). J'en profite pour bouger ce code vers le dernier patch, qui s'occupe explicitement du manager.

#17 - 28 janvier 2021 15:10 - Lauréline Guérin

Dans 0002, pourquoi tu changes le .exists() en bool(qs) ? C'est dommage de faire un select de plusieurs lignes quand un LIMIT 1 peut suffire, non ?

#18 - 28 janvier 2021 15:26 - Valentin Deniaud

Lauréline Guerin a écrit :

```
bool(qs)
```

Parce que précisément ce n'est plus un queryset à partir du patch d'après. Ça serait possible de construire un has_open_events qui ne travaille qu'avec un queryset même une fois que les évènements récurrents entrent dans la danse, mais c'est un peu de boulot et ça peut être fait dans un ticket annexe si nécessaire je pense.

#19 - 29 janvier 2021 09:03 - Lauréline Guérin

Question, si on crée un évènement avec une récurrence, et qu'une réservation se fait sur une date future de la récurrence. Si la récurrence est éditée (c'est éditable ?), et modifiée, en faisant en sorte que l'event fils réservé ne rentre plus dans la récurrence (genre on passe de tous les lundi à tous les mardis), ça a quel impact ?

#20 - 29 janvier 2021 10:31 - Valentin Deniaud

Lauréline Guerin a écrit :

Si la récurrence est éditée (c'est éditable ?)

Nop, pas quand une réservation est posée, c'est 0005. Et fonctionnellement je pense que c'est gérable ainsi : l'agent qui se retrouve dans ce cas ira annuler manuellement les événements qui ont des réservations, ce qui notifiera proprement les usagers, et ensuite la modif est permise. Si le besoin c'est de garder ces événements ça sera plus compliqué mais gérable, à coup d'exceptions bien choisies j'imagine.

Le truc améliorable dans l'histoire c'est qu'une modif qui simplement ajouterait des récurrences devrait être permise (mais pas capital non plus, souvent ça sera gérable en créant un deuxième événement).

#21 - 29 janvier 2021 17:52 - Emmanuel Cazenave

Je réfléchis tout haut façon pavé dans la marre : j'ai l'impression que le choix de ne pas persister en DB les événements futurs issus de la récurrence tant qu'aucune réservation n'est posée dessus amène son lot de problèmes.

- "Il faut avoir en tête que les événements récurrents (self.recurrence_rule is not None) ne sont visibles que dans la page de paramétrage" : j'ai peur qu'on perde un peu les gens (à commencer par nous mêmes). J'imagine que ce choix d'interface est lié à l'absence de persistance.

- des questions pour marquer des événements comme finalement non ouverts : ce qui amène [#50561](#) avec aussi son lot de question. Là aussi j'ai l'impression qu'on en arrive à imaginer gérer les choses avec des exceptions parce qu'on a pas de persistance, tandis que si on en avait on retomberait sur les cas classique "si un événement existe je peux le supprimer etc"

- si demain on veut pouvoir répondre à des demandes du types : "le ciné club du 11 janvier, exceptionnellement il commence à 15h00 au lieu de 18 heures", là aussi sans persistance ça s'annonce compliqué

Bref j'ai l'impression qu'il y a plus d'avenir à demander une date limite lors de la création de la récurrence et à créer les événements. Et qu'ensuite petit à petit on pourra ajouter des fonctionnalités dans l'interface sur modification de toute la série/modification d'une occurrence particulière.

#22 - 01 février 2021 08:21 - Lauréline Guérin

+1, j'ai aussi l'impression qu'imposer une date de fin (max 1 an ?) à un événement récurrent et tout créer permettra plus de souplesse, notamment sur l'édition d'un des événements: on pourra proposer de modifier seulement celui-ci, tous (mais pas les passés ?), ou les à venir, un peu comme dans google calendar.

Et plus besoin de gérer les exceptions.

#23 - 01 février 2021 10:20 - Valentin Deniaud

Emmanuel Cazenave a écrit :

Je réfléchis tout haut façon pavé dans la marre : j'ai l'impression que le choix de ne pas persister en DB les événements futurs issus de la récurrence tant qu'aucune réservation n'est posée dessus amène son lot de problèmes.

C'est certes plus compliqué, mais à mon avis aucune appli d'agenda n'impose une date de fin de récurrence. Et j'imagine que c'est pas le genre de limitation facile à lever une fois qu'elle existe, d'où l'idée de ne pas se l'autoriser.

- "Il faut avoir en tête que les événements récurrents (self.recurrence_rule is not None) ne sont visibles que dans la page de paramétrage" : j'ai peur qu'on perde un peu les gens (à commencer par nous mêmes). J'imagine que ce choix d'interface est lié à l'absence de persistance.

Du tout, la persistance ne change rien à ce niveau là.

Un événement tous les jours pendant un an -> on ne va certainement pas afficher 365 événements sur cette page de paramétrage, le bon choix est bien entendu d'afficher seulement un événement avec le libellé « une fois tous les jours pendant 1 an », ce qui est fait ici.

- des questions pour marquer des événements comme finalement non ouverts : ce qui amène [#50561](#) avec aussi son lot de question. Là aussi j'ai l'impression qu'on en arrive à imaginer gérer les choses avec des exceptions parce qu'on a pas de persistance, tandis que si on en avait on retomberait sur les cas classique "si un événement existe je peux le supprimer etc"

Lecture trop rapide ici aussi, j'ai répondu à ta question [#50561#note-9](#).

- si demain on veut pouvoir répondre à des demandes du types : "le ciné club du 11 janvier, exceptionnellement il commence à 15h00 au lieu de 18 heures", là aussi sans persistance ça s'annonce compliqué

Eh non, pareil ça fonctionne déjà de manière transparente, même mécanisme que pour l'annulation.

Lauréline Guerin a écrit :

+1, j'ai aussi l'impression qu'imposer une date de fin (max 1 an ?) à un événement récurrent et tout créer permettra plus de souplesse, notamment sur l'édition d'un des événements: on pourra proposer de modifier seulement celui-ci, tous (mais pas les passés ?)

Marche déjà, comme dit plus haut, à expliciter par un travail de documentation (et tout à fait à peaufiner dans la suite).

Et plus besoin de gérer les exceptions.

On ne s'en sortira pas sans gérer les exceptions, ça ne va pas d'imaginer qu'on va faire plusieurs centaines de clics sur un agenda pour virer les vacances scolaires, ce à chaque fois qu'on ajoute un évènement récurrent.

Conclusion, je veux bien détricoter mais pour l'instant je ne vois pas d'argument valable en ce sens. Il me semble aussi qu'une bonne partie de ces discussions aurait pu être évitées en essayant la fonctionnalité pour de vrai avant de lire le code, peut-être qu'il faut que j'attache mes formulaires et workflow de test pour rendre ça plus accessible ?

#24 - 01 février 2021 10:56 - Valentin Deniaud

Valentin Deniaud a écrit :

Emmanuel Cazenave a écrit :

- si demain on veut pouvoir répondre à des demandes du types : "le ciné club du 11 janvier, exceptionnellement il commence à 15h00 au lieu de 18 heures", là aussi sans persistance ça s'annonce compliqué

Eh non, pareil ça fonctionne déjà de manière transparente, même mécanisme que pour l'annulation.

Je viens d'essayer et non ça ne marche pas, en fait actuellement le bouton « Options » est masqué sur les récurrences, la seule action supportée est l'annulation. Mon plan c'était d'avoir ensuite un ticket pour démasquer ce bouton et en même temps vérifier que les modifications à cet endroit marchent as expected, genre celle de l'heure (et je ne vois rien de bloquant à ce que ça fonctionne).

#25 - 05 février 2021 11:03 - Lauréline Guérin

- Fichier Capture d'écran de 2021-02-05 10-35-02.png ajouté

- Fichier Capture d'écran de 2021-02-05 10-34-56.png ajouté

- Fichier Capture d'écran de 2021-02-05 10-34-49.png ajouté

quelques remarques après avoir testé la branche:

- quand on crée un event on peut définir une date de publication; quel effet a-t-elle sur un event récurrent ? quel effet veut-on que cette date de publication ait ? (cf screenshots, le display est étrange). note: une occurrence avant la date de publication remonte dans l'api datetimes, sans qu'elle soit notée disabled, mais elle n'est pas réservable (échec au post sur fillslot, "event not bookable")

- j'ai supprimé une occurrence d'un event récurrent défini tous les lundi (ici, celui du 1er mars); il apparaît toujours dans la liste des events du mois de mars, et ouvert à la résa. j'ai aussi réussi à le réserver.

- note: je vais merger [#48924](#), pense à renommer ta migration :)

#26 - 05 février 2021 11:12 - Lauréline Guérin

c'est normal que les urls d'api changent après qu'on ait accédé à une occurrence ? j'imagine que c'est dû au stockage en DB ?

exemple, l'api datetimes me renvoie:

```
{
  "api": {
    "bookings_url": "https://chrono.dev.publik.love/api/agenda/41663-events-recurrents/bookings/41663-events-recurrents-event:2021-03-29-1000/",
    "fillslot_url": "https://chrono.dev.publik.love/api/agenda/41663-events-recurrents/fillslot/41663-events-recurrents-event:2021-03-29-1000/",
    "status_url": "https://chrono.dev.publik.love/api/agenda/41663-events-recurrents/status/41663-events-recurrents-event:2021-03-29-1000/"
  },
  "datetime": "2021-03-29 10:00:00",
  "description": "",
  "disabled": false,
  "id": "41663-events-recurrents-event:2021-03-29-1000",
  "places": {
    "available": 10,
    "full": false,
    "has_waiting_list": false,
    "reserved": 0,
    "total": 10
  },
  "pricing": null,
  "slug": "41663-events-recurrents-event:2021-03-29-1000",
  "text": "event 1 lundi (29 mars 2021 10:00)",
  "url": null
}
```

```
},
```

je fais un call:

```
% http GET https://chrono.dev.publik.love/api/agenda/41663-events-recurrents/status/41663-events-recurrents-event:2021-03-29-1000/ [0]
HTTP/1.1 200 OK
Allow: GET, HEAD, OPTIONS
Connection: keep-alive
Content-Length: 763
Content-Type: application/json
Date: Fri, 05 Feb 2021 10:08:16 GMT
Server: nginx/1.18.0
X-Frame-Options: SAMEORIGIN
```

```
{
  "api": {
    "bookings_url": "https://chrono.dev.publik.love/api/agenda/41663-events-recurrents/bookings/41663-events-recurrents-event-2021-03-29-1000/",
    "fillslot_url": "https://chrono.dev.publik.love/api/agenda/41663-events-recurrents/fillslot/41663-events-recurrents-event-2021-03-29-1000/",
    "status_url": "https://chrono.dev.publik.love/api/agenda/41663-events-recurrents/status/41663-events-recurrents-event-2021-03-29-1000/"
  },
  "datetime": "2021-03-29 10:00:00",
  "description": "",
  "disabled": false,
  "err": 0,
  "id": "41663-events-recurrents-event-2021-03-29-1000",
  "places": {
    "available": 10,
    "full": false,
    "has_waiting_list": false,
    "reserved": 0,
    "total": 10
  },
  "pricing": null,
  "slug": "41663-events-recurrents-event-2021-03-29-1000",
  "text": "event 1 lundi (29 mars 2021 10:00)",
  "url": null
}
```

un call à <https://chrono.dev.publik.love/api/agenda/41663-events-recurrents/status/41663-events-recurrents-event-2021-03-29-1000/> donne le même résultat

#27 - 08 février 2021 10:22 - Valentin Deniaud

Lauréline Guerin a écrit :

- quand on crée un event on peut définir une date de publication; quel effet a-t-elle sur un event récurrent ? quel effet veut-on que cette date de publication ait ?

Je ne sais pas, je trouve que ça fait un peu doublon avec la date de début de récurrence mais il y a sûrement des cas d'usages que je n'ai pas en tête.

une occurrence avant la date de publication remonte dans l'api datetimes, sans qu'elle soit notée disabled, mais elle n'est pas réservable (échec au post sur fillslot, "event not bookable")

OK j'essaye de corriger ça, il doit manquer un bout de filter quelque part pour ne pas qu'elle remonte dans datetimes.

- j'ai supprimé une occurrence d'un event récurrent défini tous les lundi (ici, celui du 1er mars); il apparaît toujours dans la liste des events du mois de mars, et ouvert à la résa. j'ai aussi réussi à le réserver.

Effectivement, à régler au niveau de l'interface, il ne faut pas autoriser la suppression de ces évènements, tout doit passer par l'annulation.

c'est normal que les urls d'api changent après qu'on ait accédé à une occurrence ? j'imagine que c'est dû au stockage en DB ?

Ouep et c'est bien couvert par les tests, mais rétrospectivement je ne suis pas super convaincu de cette manière de faire : c'était par mimétisme avec le retour de /datetimes/ pour les rendez-vous, et comme « : » n'est pas un caractère valide pour un slug (même si en vrai ça passe), je le remplace par « - » à la sauvegarde. Peut-être plutôt utiliser genre « -- » comme séparateur, caractère valide pour un slug et slug.split('--') donnerait les deux

composantes de manière sûre.

#28 - 08 février 2021 16:09 - Valentin Deniaud

Valentin Deniaud a écrit :

Lauréline Guerin a écrit :

c'est normal que les urls d'api changent après qu'on ait accédé à une occurrence ? j'imagine que c'est dû au stockage en DB ?

Ouep et c'est bien couvert par les tests, mais rétrospectivement je ne suis pas super convaincu de cette manière de faire : c'était par mimétisme avec le retour de /datetimes/ pour les rendez-vous, et comme « : » n'est pas un caractère valide pour un slug (même si en vrai ça passe), je le remplace par « - » à la sauvegarde. Peut-être plutôt utiliser genre « -- » comme séparateur, caractère valide pour un slug et slug.rsplit('--') donnerait les deux composantes de manière sûre.

Bon à relire le code non, c'est quand même bien utile de pouvoir différencier les évènements à créer (caractère ':' dans le slug) des autres.

Autres remarques appliquées, il faudra affiner plus tard ce qu'on veut faire de la date de publication, affichage et tout.

#29 - 08 février 2021 16:20 - Valentin Deniaud

Valentin Deniaud a écrit :

Ouep et c'est bien couvert par les tests, mais rétrospectivement je ne suis pas super convaincu de cette manière de faire [...]

(en fait en tapant ça je me suis juste emmêlé les pinceaux avec une idée qui m'est venue en parallèle : ce qui est pas top c'est d'au final faire slug.replace(':', '-') au moment de save, parce qu'on se retrouve avec un slug du type my-event-2021-01-26-1305 et que si un jour on veut séparer texte et datetime c'est pas rigolo, et donc ce serait mieux d'avoir slug.replace(':', '--') qui permet de séparer avec un rsplit bien senti. Il y aurait plein de tests à modifier mais je vais sûrement le faire, sans incidence sur le reste de toute façon.)

#30 - 11 février 2021 11:05 - Valentin Deniaud

Valentin Deniaud a écrit :

je vais sûrement le faire

Fait et c'est tout bon, relecture can go on.

#31 - 17 février 2021 14:33 - Valentin Deniaud

- Précède Development #51218: Évènements récurrents : pouvoir spécifier une date de fin de récurrence ajouté

#32 - 17 février 2021 15:24 - Emmanuel Cazenave

- Statut changé de Solution proposée à Solution validée

J'ai joué avec l'interface et refait un tour sur le code.

Sur l'idée qu'on a plus gagner à lancer et corriger le tir que de rester des plombs dans une review, let's go.

#33 - 17 février 2021 15:50 - Valentin Deniaud

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

```
commit b0d89df301304b17654caba5078aa5a066faa432
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date: Thu Jan 28 12:33:43 2021 +0100
```

```
manager: handle edition/deletion of recurring event (#41663)
```

```
commit e09281624b8027549489b535c17cf71496799a0d
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date: Wed Jan 13 14:59:36 2021 +0100
```

```
manager: backport SplitDateTimeField fix (#41663)
```

```
commit a699e144b4ef8d9285d352842acbfd4e3af208a0
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date: Tue Dec 22 17:26:29 2020 +0100
```

add support for recurring events (#41663)

commit a392213dce7007a4f6399bbb8872487af5775d89
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date: Wed Jan 6 12:26:45 2021 +0100

agendas: make returning a queryset in get_open_events useless (#41663)

commit 7234aa74d30296fa2662f36bac7f55ec100cdac7
Author: Valentin Deniaud <vdeniaud@entrouvert.com>
Date: Tue Jan 5 12:16:21 2021 +0100

agendas: do not save seconds in event start_datetime (#41663)

#34 - 18 février 2021 09:17 - Frédéric Péters

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

#35 - 21 février 2021 16:43 - Frédéric Péters

- Lié à Development #49203: Événements récurrents ajouté

Fichiers

stamina-20070203-diffusion-recurrence.png	31,5 ko	14 avril 2020	Frédéric Péters
Screenshot_2020-04-14 calendar-edit.png	14,9 ko	14 avril 2020	Frédéric Péters
google-agenda-2.png	120 ko	10 septembre 2020	Emmanuel Cazenave
google-agenda-1.png	120 ko	10 septembre 2020	Emmanuel Cazenave
0001-agendas-do-not-save-seconds-in-event-start_datetime-.patch	982 octets	12 janvier 2021	Valentin Deniaud
0005-manager-handle-edition-deletion-of-recurring-event-4.patch	10,2 ko	12 janvier 2021	Valentin Deniaud
0003-trivial-factorize-event-forms-41663.patch	1,51 ko	12 janvier 2021	Valentin Deniaud
0004-add-support-for-recurring-events-41663.patch	35,3 ko	12 janvier 2021	Valentin Deniaud
0002-agendas-make-returning-a-queryset-in-get_open_events.patch	5,63 ko	12 janvier 2021	Valentin Deniaud
0001-agendas-do-not-save-seconds-in-event-start_datetime-.patch	982 octets	28 janvier 2021	Valentin Deniaud
0005-manager-handle-edition-deletion-of-recurring-event-4.patch	10,6 ko	28 janvier 2021	Valentin Deniaud
0003-add-support-for-recurring-events-41663.patch	37,9 ko	28 janvier 2021	Valentin Deniaud
0002-agendas-make-returning-a-queryset-in-get_open_events.patch	5,43 ko	28 janvier 2021	Valentin Deniaud
0004-manager-backport-SplitDateTimeField-fix-41663.patch	1,17 ko	28 janvier 2021	Valentin Deniaud
Capture d'écran de 2021-02-05 10-34-56.png	53,5 ko	05 février 2021	Lauréline Guérin
Capture d'écran de 2021-02-05 10-35-02.png	124 ko	05 février 2021	Lauréline Guérin
Capture d'écran de 2021-02-05 10-34-49.png	33,4 ko	05 février 2021	Lauréline Guérin