Gestion des évènements

Obtenir des informations sur un évènement

Les informations sur un évènement sont accessibles à l'adresse /api/agenda/SLUG-DE-LAGENDA/status/SLUG-DE-LEVENEMENT/ via la méthode HTTP GET.

Exemple

```
GET /api/agenda/foo-bar/status/event-slug/
```


```
{
    "err": 0,
    "id": "event-slug",
    "slug": "event-slug",
    "text": "Event",
    "label": "Event",
    "date": "2020-06-11",
    "datetime": "2020-06-11 10:00:00",
    "description": null,
    "pricing": null,
    "url": null,
    "disabled": true,
    "checked": false,
    "api": {
                "bookings_url": "http://chrono.dev.publik.love/api/agenda/foo-bar/bookings/event-slu
g/",
                "fillslot_url": "http://chrono.dev.publik.love/api/agenda/foo-bar/fillslot/event-slu
g/",
                "status_url": "http://chrono.dev.publik.love/api/agenda/foo-bar/status/event-slug/",
                "check_url": "http://chrono.dev.publik.love/api/agenda/foo-bar/check/event-slug/",
    },
    "places": {
        "available": 0,
        "full": true,
        "has_waiting_list": false,
        "reserved": 3,
        "total": 3
    }
}
```


Quand une liste d'attente existe, le dictionnaire places est complété ainsi :


```
    ...
    "places": {
        "available": 0,
        "full": true,
        "reserved": 5,
        "total": 30,
        "has_waiting_list": true,
        "waiting_list_total": 10,
        "waiting_list_reserved": 2,
        "waiting_list_available": 8,
        "waiting_list_activated": true
    }
```


Le booléen waiting_list_activated signifie que la prochaine réservation sera envoyée en liste d'attente.

Obtenir la liste des réservations d'un évènement

La liste des réservations d'un utilisateur pour un évènement donné est accessible à l'adresse /api/agenda/SLUG-DE-LAGENDA/bookings/SLUG-DE-LEVENEMENT/ via la méthode HTTP GET. Le paramètre user_external_id

est obligatoire pour désigner l'utilisateur.

Exemple

```
GET /api/agenda/foo-bar/bookings/event-slug/?user_external_id=xxx
```

```
{
    "err": 0,
    "data": [
        {"id": 1, "in_waiting_list": true},
        {"id": 2, "in_waiting_list": false}
    ]
}
```

Agenda évènement : Marquer un évènement comme pointé

Un évènement peut être marqué comme pointé via un appel à l'adresse /api/agenda/SLUG-DE-LAGENDA/check/SLUG-DE-LEVENEMENT/ via la méthode HTTP GET.

```
GET /api/agenda/foo-bar/check/event-slug/
```

```
{
    "err": 0,
}
```

Ajouter un évènement

L'ajout d'un évènement s'effectue par un appel à l'adresse /api/agenda/SLUG-DE-LAGENDA/event/ via la méthode HTTP POST.

Paramètres JSON, corps de la requête

Deux paramètres doivent obligatoirement être présents.

Nom	Description	Exemple
start_datetime	Date et heure	"2021-10-22 09:45", sous forme de gabarit {{today|date:"Y-m-d"}} {{today|time:"H:i"}}
places	Nombre de places	10

Paramètres optionnels.

Nom	Description	Exemple
label	Libellé optionnel pour identifier la date	"..."
duration	Durée (en minutes)	90
publication_datetime	Date·heure de publication	"2021-10-07 10:00", sous forme de gabarit, par exemple {{today|date:"Y-m-d 10:00"}}
waiting_list_places	Places dans la liste d'attente	3
description	Description	"..."
pricing	Tarif (text)	"2€", "gratuit", ...
url	URL vers un site tiers	"https://doc-publik.entrouvert.com"

Des paramètres supplémentaires permettent de définir un événement récurrent :

Nom	Description	Exemple
recurrence_days	Jours de la récurrence	"0,1,6" pour lundi, mardi et dimanche

recurrence_week_interval	Répéter toutes X semaines	1 (défaut), 2 ou 3
recurrence_end_date	Date de fin de la récurrence	"2021-10-20"

Les jours de la récurrence et la date de fin de récurrence doivent
être fournis pour créer un événement récurrent.

Exemple

```
POST /api/agenda/SLUG-DE-LAGENDA/event/
```

```
{
   "data" : {
      "date" : "2021-11-22",
      "datetime" : "2021-11-22 09:45:00",
      "description" : "Une description associée",
      "duration" : 90,
      "id" : "mon-evenement",
      "label" : "Mon événement",
      "pricing" : "2€",
      "recurrence_days" : [
         0,
         1,
         6
      ],
      "recurrence_end_date" : "2021-10-20",
      "recurrence_week_interval" : 2,
      "slug" : "mon-evenement",
      "text" : "Mon événement",
      "url" : "https://doc-publik.entrouvert.com"
   },
   "err" : 0
}
```

Mettre à jour un événement

Les événements sont modifiables à l'adresse /api/agenda/SLUG-DE-LAGENDA/event/SLUG-DE-LEVENEMENT/ via la méthode HTTP PATCH.

Paramètres JSON, corps de la requête

Les paramètres utilisables sont les mêmes que ceux de la requête d'ajout.
Il n'y a pas besoin de passer l'ensemble des paramètres, seul les paramètres fournis seront mis à jour.

Exemple

```
PATCH /api/agenda/foo-bar/event/mon-evenement/
```

```
(même retour que pour la méthode POST)
```

Il y a cependant certaines restrictions :

Restrictions sur les événements récurrents

On ne peut pas modifier les paramètres start_datetime, recurrence_days et recurrence_week_interval

```
{
    "err": 1,
    "err_class": "start_datetime cannot be modified because some recurrences have bookings attache
d to them."
}
```

On ne peut pas réduire la date de fin de récurrence
(recurrence_end_date) avant un la date de la dernière réservation faite sur l'événement.

```
{
    "err": 1,
    "err_class": "recurrence_end_date cannot be modified because bookings exist after this date."
}
```

Restrictions sur les occurrences des événements récurrents

Les champs publication_datetime et ceux définissant la récurrence (recurrence_days, recurrence_week_interval et recurrence_end_date) ne sont pas modifiables sur les récurrences d'événements récurrents.

```
{
    "err": 1,
    "err_class": "publication_datetime cannot be modified on an event recurrence"
}
```