

راهنمای فارسی استفاده از وب سرویس الوکام

مطابق با نسخه 4.24.0 الوکام

ضمن تشکر از اینکه وقت خود را در اختیار تیم فنی الوکام قرار داده اید ذکر چند نکته هنگام مطالعه مستند ضروری است :

1. این مستند صرفاً توضیحات کاملتری برای نمونه api های پرکاربردی است که در آدرس <https://pnlapi.alocom.co/api/documentation> قابل دسترس هستند لذا ممکن است این مستند تمامی api ها را شامل نشود. لذا بهتر است برای مشاهده تمام api های وب سرویس الوکام و انجام تست آن ها به لینک یاد شده مراجعه شود.
2. در خواست هایی که در ادامه مستندات آمده است شامل سه بخش است : الف : نمونه درخواست ، ب: بدنه یا body درخواست ، ج: نمونه پاسخ یا response . مسیر یا route نمونه درخواست به صورت نسبی یا relative آورده شده است . بدیهی است که برای ارسال درخواست به وب سرور این api باید به صورت کامل و با آدرس request + <https://pnlapi.alocom.co/> ارسال شود . برای مثال درخواست لاگین به صورت کامل به این صورت است :
<https://pnlapi.alocom.co/api/v1/auth/login>
3. برای ارسال درخواست ها حتماً باید ابتدا در سامانه الوکام ثبت نام کرده باشید و با تهیه یک سرویس از امکانات وب سرویس به صورت کامل استفاده فرمایید . لازم است برای فراخوانی یا call درخواست ها (به جز لاگین) در هدر مربوطه توکن قرار داده شود توکن با لاگین در سامانه به دست می آید. بعد از به دست آوردن توکن برای تنظیم توکن باید در header با کلید واژه Authorization به شیوه زیر عمل شود.

Authorization : Bearer etJ0eXaiOIJ..

POST**/api/v1/auth/logout** Logout

توسط این api کاربر می تواند از سیستم خارج شود کافی است آن را اجرا کنید و توکن کاربر باطل می شود.

نمونه Response به صورت زیر است:

Responses		
Code	Description	Links
200	شما با موفقیت خارج شدید	No links
500	Server Error	No links

Services

POST

`/api/v1/agents/services/index` Agent Service List

توسط این api کاربر می تواند لیست تمام سرویس های مربوط به خود را مشاهده کند .

فیلد `take` و `skip` فیلد های اجباری می باشند.

Take و `skip` چه هستند؟

Take به معنی تعداد رکورد هایی است که شما از دیتابیس دریافت میکنید

Skip را با یک مثال توضیح خواهم داد، فرض کنید ما می خواهیم 5 رکورد اول را دریافت نکنیم و از رکورد 6 به بعد را دریافت کنیم یعنی باید 5 رکورد را رد (`skip`) کنیم در صورت صفر بودن این مقدار از اولین رکورد دریافت انجام خواهد شد.

و مابقی فیلد ها اختیاری هستند که شامل:

Title که عنوان سرویس است و می تواند داخل `body` به عنوان فیلتر ارسال شود.

Type که نوع سرویس است که یک فیلتر چند مقداره است و می تواند مقادیر `class` و یا `conference`

را داشته باشد و شما می توانید یکی از این عناوین یا هر دو را به صورت ارایه ای به عنوان فیلتر ارسال کنید

فیلد `status` که وضعیت سرویس را مشخص میکند بنابر این می تواند مقادیر

`active,inactive,expired`

را داشته باشد همچنین مقادیر این فیلد را می توانید به صورت ارایه ای و چند مقداره ارسال کنید

مثال:

```
{  
  "title": "mahyar",  
  "type": "1212Msdcd",  
  "status": "active",  
  "take": 15,  
  "skip": 0  
}
```

نوع response نمونه را می توانید در صفحه بعد مشاهده کنید:

Code	Description	Links
200	<p>Media type</p> <p><input type="text" value="application/json"/></p> <p>Controls Accept header.</p> <p>Example Value Schema</p> <pre> { "data": [{ "agentServiceId": 74, "serviceId": 23, "title": "فیزیک", "type": "class", "time": { "start_date": "1400/5/9", "start_time": "18:45", "end_date": "1401/9/9", "end_time": "18:45" }, "max_event_user": null, "duration": { "time_duration": 1, "used_time": "-", "remain_time": "-" }, "capacity": { "consumed_disk_capacity": "-", "disk_capacity": 1, "remain_disk_capacity": "-" }, "status": "active" }] } </pre>	No links

GET `/api/v1/agents/services/active-services` Agent get active services

توسط این Api کاربر می تواند سرویس های فعال مربوط به خود را مشاهده کند.

نمونه response به صورت زیر می باشد

```

"activeServices": [
  {
    "disk_capacity": 500,
    "time_duration": 12,
    "max_event_user": 100,
    "credit_duration": 155,
    "max_online_user": 100,

```

```
"service_id": 7,  
"title": "سرویس کنندگان ۲",  
"type": "class",  
"agent_service_id": 14,  
"expired_at": "1400/10/2"  
},  
{  
"disk_capacity": 500,  
"time_duration": 4,  
"max_event_user": 5,  
"credit_duration": 124,  
"max_online_user": 5,  
"service_id": 8,  
"title": "سرویس شدگان ۳",  
"type": "class",  
"agent_service_id": 15,  
"expired_at": "1400/9/2"  
},  
{  
"disk_capacity": 500,  
"time_duration": 6,  
"max_event_user": 50,  
"credit_duration": 496,
```

```
"max_online_user": 50,  
"concurrent_video": 5,  
"concurrent_voice": 5,  
"service_id": 10,  
"title": "سمبلیک",  
"type": "class",  
"agent_service_id": 16,  
"expired_at": "1401/9/3"  
},
```

```
GET /api/v1/agents/services/{service}/agent-services/{agentservice}
```

توسط این api کاربر می تواند اطلاعات یک سرویس خاص را مشاهده کند.

داخل route این api دو پارامتر وجود دارد یکی Service که id سرویسی است که می خواهد مشاهده کند

و دومی ID مربوط به agent service است.

لازم به ذکر است این ID مربوط به سرویس خریداری شده از سمت agent است و می توانید آن را از api

قبلی دریافت کنید و در اینجا جایگذاری کنید.

نمونه Response به صورت زیر است:



```
"serviceInfo": {
  "agent_service_id": 12,
  "service_id": 3,
  "title": "سرویس",
  "status": "expired",
  "type": "class",
  "expired_at": "1400/6/2",
  "created_at": "1400/5/2",
  "consumed_disk_capacity": "میزان مصرف شده",
  "remain_disk_capacity": "فضای باقیمانده",
  "used_time": "برگزار شده",
  "remain_time": "زمان باقیمانده",
  "disk_capacity": 100,
  "time_duration": 20,
  "max event user": 10.
```

GET

/api/v1/agents/services/suggested-services

توسط این api برگزارکننده می تواند سرویس های پیشنهادی را مشاهده کند.

نمونه Response به صورت زیر می باشد:

```
"data": {
  "suggestedServices": [
    {
      "id": 5,
      "title": "service-test-1",
      "price": 0,
      "discount": 0,
      "credit_duration": 1,
      "time_duration": 1,
```

```
"concurrent_voice": 1,  
"concurrent_video": 1,  
"max_online_user": 1,  
"is_unlimited": false,  
"is_free": true,  
"type": "class",  
"is_special_offer": true,  
"is_gift": false,  
"gift_credit_duration": 0,  
"gift_time_duration": 0,  
"is_dedicated": false,  
"discounted_price": 0  
},  
{  
  "id": 6,  
  "title": "service-test-2",  
  "price": 0,  
  "discount": 0,  
  "credit_duration": 1,  
  "time_duration": 1,  
  "concurrent_voice": 1,  
  "concurrent_video": 1,  
  "max_online_user": 1,
```

```
"is_unlimited": false,  
"is_free": true,  
"type": "class",  
"is_special_offer": true,  
"is_gift": false,  
"gift_credit_duration": 0,  
"gift_time_duration": 0,  
"is_dedicated": false,  
"discounted_price": 0  
},  
{  
"id": 28,  
"title": "friday-service-test-3",  
"price": 700000,  
"discount": 200000,  
"credit_duration": 1,  
"time_duration": 1,  
"concurrent_voice": 1,  
"concurrent_video": 1,  
"max_online_user": 1,  
"is_unlimited": false,  
"is_free": false,  
"type": "class",
```

```
"is_special_offer": false,  
"is_gift": false,  
"gift_credit_duration": 0,  
"gift_time_duration": 0,  
"is_dedicated": true,  
"discounted_price": 500000  
}  
]  
},
```

GET

/api/v1/agents/services/available-services Available Services

توسط این api کاربر می تواند سرویس های موجود را مشاهده کند.

نمونه Response

```
"data": {  
  "services": [  
    {  
      "id": 66,  
      "title": "Prof. Donavon Mills II",  
      "price": 0,
```

```
"discount": 0,  
"credit_duration": 1,  
"time_duration": 90,  
"concurrent_voice": 5,  
"concurrent_video": 2,  
"max_online_user": 1,  
"is_unlimited": false,  
"is_free": true,  
"type": "conference",  
"is_special_offer": false,  
"is_gift": false,  
"gift_credit_duration": 0,  
"gift_time_duration": 0,  
"is_dedicated": false,  
"discounted_price": 0  
},  
{  
"id": 65,  
"title": "Jackson Padberg",  
"price": 0,  
"discount": 0,  
"credit_duration": 1,  
"time_duration": 90,
```

```
"concurrent_voice": 5,  
"concurrent_video": 2,  
"max_online_user": 1,  
"is_unlimited": false,  
"is_free": true,  
"type": "conference",  
"is_special_offer": false,  
"is_gift": false,  
"gift_credit_duration": 0,  
"gift_time_duration": 0,  
"is_dedicated": false,  
"discounted_price": 0  
},
```

Events

POST

/api/v1/agents/events/index Agent Event Get

توسط این api برگزار کننده می تواند لیست تمامی رویداد های مربوط به خود را مشاهده کند.

نوع درخواست از نوع post است و داخل body آن مقادیر زیر وجود دارد:

Take و skip که مقادیر اجباری هستند و مابقی مقادیر اختیاری هستند که در ادامه شرح داده خواهد شد:

Service_title مقدار این فیلد عنوان سرویس را می تواند فیلتر کند و از نوع string است.

Agents مقدار این فیلد نام agent مربوط به هر رویداد را می تواند فیلتر کند این فیلد از نوع آرایه می باشد و می تواند چند مقدار را دریافت کند

Type که می تواند مقادیر business یا educational یا webinar را دریافت کند و بر اساس این فیلد فیلتر کند .

In_progress این فیلد یک فیلد Boolean است که تعیین می کند رویداد در حال برگزاری است یا خیر.

نمونه Response:

```
"data": [  
  {  
    "id": 54,  
    "agent": "محسن طهماسب پور",  
    "title": "تست جمعه 1",  
    "type": "class",  
    "service": "friday-service-test-1",  
    "duration": "زمان مصرفی",  
    "user_count": 6,  
    "status": "active",  
    "alocom_link": "https://class.kavano.org/class/mohsen-admin/testfriday  
1"  
  },  
]
```

GET

/api/v1/agents/events/{event} Agent get Event

توسط این api کاربر می تواند رویداد خاصی را دریافت کند.

داخل route باید پارامتر ID رویدادی باشد که مد نظر دارد.

نمونه Response :

```
"event": {
  "id": 53,
  "title": "service-test-8",
  "agent_id": 3,
  "agent_service_id": 22,
  "slug": "asdasdasdasw",
  "status": "expired",
  "guest_count": 0,
  "guest_code": null,
  "guest_access": 0,
  "start_by_admin": 0,
  "alocom_id": "6100df77307f0900b1eaac43",
  "alocom_link": "https://class.kavano.org/class/test-agent/asdasdasdasw",
  "agent_service": {
    "id": 22,
    "invoice_id": 31,
    "service_id": 6,
```



```
"action": "activate",
"title": "service-test-2",
"type": "class",
"properties": {
  "disk_capacity": 1,
  "time_duration": 1,
  "max_event_user": 1,
  "credit_duration": 31,
  "max_online_user": 1
},
"created_at": "2021-07-26 16:10:06",
"alocom_id": "60fea0722193840076340ef1",
"expired_at": "2021-08-26 16:10:06",
"deleted_at": null,
"status": "expired"
}
"serviceInfo": {
  "agent_service_id": 22,
  "editable": false,
  "title": "service-test-2",
  "type": "class",
  "expired_at": "1400/6/4",
  "max_event_user": 1,
```

```
"max_online_user": 1
},
"activeServices": [
  {
    "disk_capacity": 500,
    "time_duration": 12,
    "max_event_user": 100,
    "credit_duration": 155,
    "max_online_user": 100,
    "service_id": 7,
    "title": "سرویس کنندگان ۲",
    "type": "class",
    "agent_service_id": 14,
    "expired_at": "1400/10/2"
  },
"enrolledUsers": [
  {
    "id": 295,
    "userid": 148,
    "name": "احمد",
    "surname": "جعفری",
    "role": "participant"
  }
]
```

DELETE**/api/v1/agents/events/{event}** Agent Event delete

توسط این api کاربر می تواند رویداد خاصی را حذف کند.

داخل route باید ID رویدادی باشد که مد نظر دارد.

نمونه Response :

200 No links

Media type
application/json

Controls Accept header.

Example Value | Schema

```
{  
  "message": "رویداد با موفقیت حذف شد"  
}
```

لازم به ذکر است که در صورتی که رویداد در حال برگزاری باشد و قابل حذف نباشد این ارور باز خواهد گشت :
امکان حذف رویداد وجود ندارد .

POST**/api/v1/agents/events** Agent event store

توسط این api کاربر می تواند رویداد جدیدی را ثبت کند.

مقادیر ارسالی در body شامل موارد زیر است که در ادامه توضیح داده خواهد شد:

● title که عنوان رویداد می باشد و این فیلد اجباری است.

- **agent_service_id** که ID مربوط به سرویس خریداری شده توسط این برگزار کننده است این فیلد هم اجباری است. (این مقدار را از api های قبلی می توان استخراج کرد و مقدار صحیح را در **body** درخواست قرار داد .)
- **start_by_admin** که یک **Boolean** می باشد که مشخص می کند که امکان ورود کاربران پیش از مدیر وجود دارد یا خیر . اگر مقدار این فیلد 1 باشد امکان ورود به رویداد برای کاربران پیش از مدیر نیست. این فیلد هم اجباری است.
- **status** که وضعیت رویداد را مشخص می کند که فعال یا غیر فعال باشد و این فیلد هم یک فیلد **boolean** می باشد و مقادیر 0 یا 1 را دریافت می کند. این فیلد هم اجباری است.
- **duration_time** که زمان یک رویداد را مشخص می کند و براساس دقیقه مقدار می گیرد و یک فیلد اختیاری است. در مورد این فیلد باید توجه شود که از آغاز رویداد این زمان شروع به محاسبه می شود و پس از طی زمان تعیین شده رویداد بسته می شود. عملکرد این تابع مشابه بسته شدن رویداد توسط مدیر است و به این ترتیب چت ها و وایت برد پاک می شود. در صورتی که مدیر زودتر اقدام به بستن رویداد کند . محاسبه مربوطه ریست می شود.
- **slug** نام لاتین مربوط به رویداد است که از نوع **string** هم باید باشد . این فیلد اجباری است.
- **Back_link** فیلدی است که در صورت خروج کاربر از رویداد کاربر به صفحه ای به جز صفحه لاگین هدایت شود (همان آدرس **back_link**) این فیلد اختیاری است و بیشتر مناسب برگزار کنندگانی است که توسط **api** و وب سرویس از صفحه سایت خود وارد رویداد می شوند و مایلند بعد از خروج کاربر از رویداد به همان صفحه ورود به کلاس که در سایت خودشان است هدایت شوند.
- **users** به صورت آرایه ای از آبجکت ها ساختار داده ای شامل آی دی کاربر و نقش آن ها است که می بایست ارسال شود.

Example :

```
Users : {  
  
    [  
  
        'User_id' : 1,  
  
        'Role' : 'participant'  
  
    ],  
  
    [  
  
        'User_id' : 2,  
  
        'Role' : 'teacher'  
  
    ]  
  
}
```

نکته : هر بار که لازم است اطلاعات دانش آموزان یا معلمان تغییر کند از جمله تغییر نقش و حذف، مقدار users در این متد آپدیت می شود.

userid و role که شامل ID کاربران و نقش آن ها در رویداد می باشد که نقش شامل participant, Assistant, teacher می باشد.

لازم به ذکر است فیلد users اجباری نیست .

- **guest_access** این فیلد یک Boolean است که مشخص میکند که رویداد کاربر مهمان را بپذیرد یا خیر

(این فیلد اختیاری است)

- **guest_code** این فیلد به صورت آرایه است برای تعیین رمز عبور کاربران مهمان می باشد این فیلد اختیاری است و در صورت تمایل می توانید چند رمز عبور را تعیین کنید.

- **guest_count** این فیلد تعداد کاربران مهمان برای رویداد را تعیین می کند (این فیلد اختیاری است)

نمونه Response:

```
{  
  "message": "رویداد با موفقیت ایجاد شد"  
}
```

PATCH

`/api/v1/agents/events/{event}/` Agent Event update

این api برای آپدیت یک رویداد می باشد.

پارامتر **event** در حقیقت ID رویداد مورد نظر است.

فیلدهایی که داخل **body** باید برای آپدیت ارسال شوند به شرح زیر می باشند:

title که عنوان رویداد می باشد و این فیلد اجباری است.

`start_by_admin` که یک `Boolean` می باشد که مشخص میکند حتما باید ابتدا معلم وارد رویداد شود سپس سایر کاربران با نقش شرکت کننده . اگر مقدار این فیلد 1 باشد یعنی قبل از ورود مدیر کاربر ها اجازه ورود به رویداد را ندارند.
این فیلد هم اجباری است.

`status` که وضعیت رویداد را مشخص میکند که فعال یا غیر فعال باشد و این فیلد هم یک فیلد `boolean` می باشد و مقادیر 0 یا 1 را دریافت می کند. این فیلد هم اجباری است.

`duration_time` که زمان یک رویداد را مشخص می کند و براساس دقیقه مقدار می گیرد و یک فیلد اختیاری است. در صورتی که رویداد در حال برگزاری ویرایش شود و این فیلد مقدار دهی شود ، مقدار داده شده پس از اتمام رویداد و آغاز مجدد رویداد اعمال می شود.

به همراه اطلاعات رویداد ، اطلاعات شامل آی دی کاربر و نقش آن در مقدار `users` می بایست ارسال شود:

Example :

```
Users : {  
  
    [  
  
        'User_id' : 1,  
  
        'Role' : 'participant'  
  
    ],
```

```
[
    'User_id' : 2,
    'Role' : 'teacher'
]
```

نکته : هربار که لازم است اطلاعات دانش آموزان یا مدیران تغییر کند از جمله تغییر نقش و حذف، مقدار users در این متد آپدیت می شود.

userid و role که شامل ID کاربران و نقش آن ها در رویداد می باشد که شامل نقش های participant, assistant, teacher می باشد. participant نقش شرکت کننده با دسترسی های محدود را دارد و teacher , assistant به ترتیب دستیار و مدیر با دسترسی های کامل هستند.

- slug این فیلد slug مربوط به رویداد را وارد کنید که از نوع string می باشد این فیلد هم اجباری است.

- guest_access این فیلد یک Boolean است که مشخص میکند که رویداد کاربر مهمان را بپذیرد یا خیر

(این فیلد اختیاری است)

- guest_code این فیلد برای تعیین رمز عبور کاربران مهمان می باشد این فیلد اختیاری است و در صورت تمایل می توانید رمز عبوری را تعیین کنید که کاربر مهمان در هنگام ورود به رویداد ملزم به وارد کردن آن باشد.

● **guest_count** این فیلد تعداد کاربران مهمان برای رویداد را تعیین می کند (این فیلد اختیاری

است)

نمونه Response

```
{  
  "message": "رویداد با موفقیت ویرایش شد"  
}
```

ثبت کاربر در رویداد به همراه دریافت توکن ورود به رویداد:

POST /api/v1/agents/events/{event}/enroll-user-with-token Agent event enroll-user-with-token

این api برای ثبت نام کاربران در رویداد استفاده می شود که پس از ارسال در پاسخ سرور ، توکن مربوط به کاربر به همراه لینک ورود به رویداد به رویداد نمایش داده می شود. از این api برای دریافت لینک مستقیم ورود به رویداد غالباً استفاده می شود. به طوری که تنها لازم است اطلاعات کاربر به همراه شناسه رویداد به وب سرویس ارسال شود و عملیات اضافه شدن به رویداد در این درخواست انجام می شود .

پارامتر داخل داخل route آیدی رویداد {event} است که از طریق گرفتن لیست کلاس ها قابل دسترسی است.

برای دریافت توکن هم کما فی السابق از درخواست login استفاده شود .

برای تنظیم توکن باید در header با کلید واژه Authorization به شیوه زیر تنظیم شود.

Authorization: Bearer eyJ0eXAiOiJKV1QiLCJ...

در بدنه درخواست (body) مولفه هایی گرفته می شود که در انتها به صورت مثال آمده است .

Name: نام کاربر (مثلا علی)

surname: اسم مستعار یا نام خانوادگی (مثلا هاشمی)

username: نام کاربری (که باید یکتا باشد)

role: نقش کاربر است که باید یکی از ۳ نوع participant (شرکت کننده در رویداد)، teacher (مدیر رویداد) و assistant (دستیار رویداد) هستند.

توجه: اگر کاربری با مشخصات ارسالی از قبل ثبت نام شده باشد لینک به همراه توکن ورود به رویداد را در خروجی نمایش میدهد .

نکته: در صورتی که کاربر موجود در کلاس مجدد بخواهد به کلاس وارد شود تعداد کاربران حاضر در کلاس تغییر نخواهند کرد و فقط توکن کاربر بروزرسانی می گردد.

نمونه Body:

```
{
  "name": "Sina",
  "surname": "Moradi",
  "username": "sinamoradi",
  "role": "participant"
}
```

توضیح پارامترهای ارسالی در پاسخ:

ویژگی	نوع	توضیحات
eventLink	string	لینک رویداد به همراه توکن برای ورود مستقیم به رویداد

توکن ورود به رویداد برای کاربر ثبت نام شده	string	token
ای دی کاربر ثبت نام شده	string	userId

نمونه Response:

```
{
  "message": "sign in successfully",
  "eventLink": "https://class.kavano.org/class/massooti/3908c1be?token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjYyNjc5MTRmYzBkYmQzMDAyOTVmNGViMiIsInVzZXJyYWI1IjoibWZoZGlqYWZhc2Iiwicm9sZSI6InVzZXIiLCJmaXJzde5hbWUiOiJtYWkaSIsImxhc3R0YW11IjoiamFmYXJpIiwiaXNhZdWVzdCI6ZmFsc2UsInNlc3Npb24iOiI2MjY2OTcwODQ4YTU1MDAwZmQ3NzZiMmIiLCJpYXQiOjE2NTA5NQ5NjZ9.AaeyXN1SS93LN6I_Ozgg8jyA16y4p_YCEmtZqYwkg4E",
  "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOlwvXC9hbG9jb21tLWFkbWlucGFuZwZlLmVvY2FsaG9zdFVvYXBPXC92MVVvYXV0aFwvbG9naW4iLCJpYXQiOjE2MjY2OTcwODQ4YTU1MDAwZmQ3NzZiMmIiwicm9sZSI6InVzZXIiLCJmaXJzde5hbWUiOiJtYWkaSIsImxhc3R0YW11IjoiamFmYXJpIiwiaXNhZdWVzdCI6ZmFsc2UsInNlc3Npb24iOiI2MjY2OTcwODQ4YTU1MDAwZmQ3NzZiMmIiLCJpYXQiOjE2NTA5NQ5NjZ9.AaeyXN1SS93LN6I_Ozgg8jyA16y4p_YCEmtZqYwkg4E",
  "userId": "6267914fc0dbd300295f4eb2"
}
```

نمایش فایل ضبط شده و آپلود شده بر اساس سرویس

```
POST /api/v1/agents/agent-services/{agentservice}/files get Service Files
```

توسط این api کاربر می تواند لیست تمام فایل های ضبط شده و یا آپلود شده در سرویس های مربوط به خود را مشاهده کند .

فیلد take و skip فیلد های اجباری می باشند.

Take و skip چه هستند؟

Take به معنی تعداد رکورد هایی است که شما از دیتابیس دریافت میکنید

Skip را با یک مثال توضیح خواهیم داد، فرض کنید ما می خواهیم 5 رکورد اول را دریافت نکنیم و از رکورد 6 به بعد را دریافت کنیم یعنی باید 5 رکورد را رد (skip) کنیم در صورت صفر بودن این مقدار از اولین رکورد دریافت انجام خواهد شد.

و مابقی فیلدها اختیاری هستند که شامل:

Title که عنوان فایلی است و می تواند داخل body به عنوان فیلتر ارسال شود.

Flag اگر این مقدار به صورت record ارسال شود تنها فایلی هایی ضبط شده را نشان میدهد و اگر ارسال نگردد همه فایل ها را نمایش میدهد

مثال:

Request body **required**

Example Value | Schema

```
{
  "flag": "record",
  "title": "testEvent1",
  "take": 100,
  "skip": 0
}
```

نوع response نمونه را می توانید مشاهده کنید:

```
[
  {
    "uploader": {
      "_id": "60f55725833367001d639641",
      "role": "admin",
      "firstName": "",
      "lastName": "كاوانو"
    },
    "remover": {
      "_id": null,
      "username": null,
      "firstName": null,
      "lastName": null,
      "role": null
    },
    "flag": "record",
    "converted": false,
    "isDeleted": false,
    "deleted_at": null,
    "_id": "65f809d85022e0944ad15530",
    "title": "kavano-meetingback-2024-2-18-13:0",
    "size": 87001,
    "serviceId": "60f558e2e99ed5002f24b8c7",
    "session": "612494ec303a6701848eceed",
    "format": ".mp4",
    "downloadable": false,
    "publicUrl": "https://download.alocom.co/rocket-api/files/records/kavano-meetingback-2024-2-18-13:0.mp4",
    "created_at": "2024-03-18T09:31:04.906Z"
  },
  {
    "uploader": {
      "_id": "60f55725833367001d639641",
      "role": "admin",
      "firstName": "",
      "lastName": "كاوانو"
    },
    "remover": {
      "_id": null,
      "username": null,
      "firstName": null,
      "lastName": null,
      "role": null
    },
    "flag": "record",
    "converted": false,
    "isDeleted": false,
    "deleted_at": null,
    "_id": "65f80a402b8e3530747beddd",
    "title": "kavano-meetingback-2024-2-18-13:1",
    "size": 380530,
    "serviceId": "60f558e2e99ed5002f24b8c7",
    "session": "612494ec303a6701848eceed",
    "format": ".mp4",
    "downloadable": false,
    "publicUrl": "https://download.alocom.co/rocket-api/files/records/kavano-meetingback-2024-2-18-13:1.mp4",
    "created_at": "2024-03-18T09:32:48.837Z"
  }
]
]
```

اتمام رویداد:

DELETE /api/v1/agents/events/{event}/terminate Agent Event terminate

توسط این api کاربر برگزارکننده می تواند یک رویداد را به اتمام برساند که تمامی نفرات داخل رویداد خارج می شوند. با اتمام رویداد تمامی اطلاعات چت و وایت برد حذف می شود.

داخل route باید ID رویدادی باشد که مد نظر دارد.

نمونه Response :

Responses	
Code	Description
200	<p>Media type</p> <p><input type="text" value="application/json"/></p> <p>Controls Accept header.</p> <p>Example Value Schema</p> <pre>{ "message": "رویداد با موفقیت انجام یافت" }</pre>

Users

POST	<code>/api/v1/agents/users/index</code> Agent users list
-------------	--

توسط این api کاربر برگزارکننده می تواند لیست کاربرانی که مربوط به او هستند را مشاهده کند.

این بخش شامل فیلتر هایی میباشد که در ادامه شرح داده خواهند شد:

داخل body می توانید مقادیر زیر را ارسال کنید:

take , skip که اجباری بوده و حتما باید ارسال شوند.

مابقی فیلتر های همگی اختیاری هستند و شامل موارد زیر است.

username که نام کاربری است.

name که اسم کاربر است.

surname که نام خانوادگی کاربر است.

labels که برچسب های مربوط به کاربران را فیلتر میکند .

نمونه Response :

```
"data": [
```

```
{
```

```
"id": 159,  
"name": "محسن3",  
"username": "mohsen-user-test-friday-3",  
"surname": "طهماسب پور3",  
"cellphone": null,  
"labels": []  
},  
{  
"id": 158,  
"name": "محسن2",  
"username": "mohsen-user-test-friday-2",  
"surname": "طهماسب پور2",  
"cellphone": null,  
"labels": []  
},  
{  
"id": 157,  
"name": "محسن1",  
"username": "mohsen-user-test-friday-1",  
"surname": "طهماسب پور1",  
"cellphone": null,  
"labels": []  
},
```

GET

/api/v1/agents/users/{agentUser} Agent users get

توسط این api برگزار کننده می تواند مشخصات یک کاربر خاص را دریافت کند.

لازم به ذکر است که agentUser یک ID مربوط به کاربر مورد نظر است که میخواهیم مشخصات اش را دریافت کنیم.

نمونه Response :

```
"id": 159,  
"agent_id": 20,  
"name": "محسن3",  
"surname": "طهماسب پور3",  
"username": "mohsen-user-test-friday-3",  
"cellphone": null,  
"email": null,  
"status": 1,  
"unlimited": null,  
"alocom_id": "6103c7b2eae0d010e6a66b9",  
"labels": [],
```


DELETE**/api/v1/agents/users/{agentUser}** Agent users delete

توسط این api برگزارکننده می تواند کاربر خاصی را از سیستم حذف کند.
لازم به ذکر است که agentUser یک ID مربوط به کاربر مورد نظر است .

حالت های مختلف Response :

Code	Description	Links
200	<p>Media type application/json</p> <p>Controls Accept header.</p> <p>Example Value Schema</p> <pre>{ "message": "کاربر با موفقیت حذف شد", "username": "mah777" }</pre>	No links
400	<p>Media type application/json</p> <p>Example Value Schema</p> <pre>{ "message": "امکان حذف کاربر وجود ندارد" }</pre>	No links

ممکن است به دلایلی مانند اینکه کاربر در حال حاضر در یک رویداد باشد امکان حذف آن وجود نداشته باشد.

PATCH

/api/v1/agents/users/{agentUser} Agent users update

این api برای بروزرسانی مشخصات کاربر است که برگزار کننده می تواند توسط این api مشخصات کاربر خاصی را ویرایش کند.

لازم به ذکر است که agentUser یک ID است مربوط به کاربر مورد نظر.

فیلدهای بخش body شامل موارد زیر می باشند:

name که نام کاربر می باشد این فیلد اجباری است.

surname نام خانوادگی کاربر می باشد این فیلد اجباری است.

username که نام کاربری می باشد (این فیلد اختیاری است).

email که ایمیل مربوط به کاربر می باشد و باید فرمت صحیح ایمیل را داشته باشد

cellphone که موبایل مربوط به کاربر می باشد و باید فرمت صحیح شماره موبایل های معمول را داشته باشد.

status که وضعیت کاربر می باشد که از نوع Boolean است و می تواند صفر یا یک باشد این فیلد اجباری است.

Labels که آرایه ای از برچسب های مربوط به کاربر می تواند باشد.

نمونه Response :

```
{  
  "message": "اطلاعات کاربر با موفقیت ویرایش شد"  
}
```

POST

/api/v1/agents/users Agent users store

با این api برگزار کننده می تواند کاربری را به سیستم اضافه کند.

فیلدهای مربوطه شامل موارد زیر می شود:

name که نام کاربر می باشد این فیلد اجباری است.

surname نام خانوادگی کاربر می باشد این فیلد اجباری است .

username که نام کاربری می باشد این فیلد اجباری است.

email که ایمیل مربوط به کاربر می باشد و باید فرمت صحیح ایمیل را داشته باشد

cellphone که موبایل مربوط به کاربر می باشد و باید فرمت صحیح شماره موبایل های معمول را داشته باشد.

status که وضعیت کاربر می باشد که از نوع Boolean است و می تواند صفر یا یک باشد این فیلد اجباری است .

Labels که آرایه ای از برچسب های مربوط به کاربر می تواند باشد.

نمونه Response :

```
"message": "کاربر جدید توسط برگزار کننده ایجاد شد"
```

```
"username": "cccc123",  
"password": "cccc123@123",
```

POST /api/v1/agents/users/import-by-excel Agent users import via excel

توسط این api می توانید گروهی از کاربران را توسط فایل اکسل به سیستم وارد کنید تا برای آنها اکانت ساخته شود.

body این درخواست شامل یک file است که باید فایل مورد نظر را وارد کند و فرمت های قابل پشتیبانی شامل موارد زیر می باشند:

Xlsx,xls,csv

نمونه Response مانند مورد زیر می باشد:

```
{  
  "response": [  
    {  
      "total": 2,  
      "invalid": 0,  
      "duplicate": 0,  
      "new": 2,  
      "users": "array of json objects that each object contains user parameters, agent_id , name , username , surname , cellphone , email , password ,  
status "  
    }  
  ]  
}
```

POST /api/v1/agents/users/confirm-import-by-excel Agent users complete import

در این بخش ثبت نام کاربران به صورت گروهی نهایی می شود.

در قسمت body این درخواست می توانید برچسب های مربوط به کاربران را با عنوان label و به صورت آرایه ای وارد نمایید.

نمونه Response

```
{  
  "message": "نُبتنام کاربران بصورت گروهی با موفقیت انجام شد"  
}
```

PATCH /api/v1/agents/users/{agentUser}/password-change Agent users change password

توسط این api برگزارکننده می تواند رمز عبور کاربر خاصی را تغییر دهد.

لازم به ذکر است که agentUser یک ID مربوط به کاربر مورد نظر است .

body درخواست مورد نظر شامل دو فیلد password و password_confitmation است که باید هر دو یکسان باشند و رمز عبور باید حداقل 6 کاراکتر باشد.

GET /api/v1/agents/users/download-sample Download Sample Excel

با فراخوانی این api فایل نمونه اکسل لینک دانلود فایل نمونه اکسل بر می گردد

نمونه Response :

```
"link": "https://cloud.alocom.co/s/9S6YbK2Q9sN3hWS/download",
```

موارد این بخش عمدتاً ناظر به اضافه یا حذف و یا ویرایش کاربران ساخته شده بود. برای اینکه کاربر یا کاربرانی به یک رویداد اضافه یا از یک رویداد حذف شوند از این دو api استفاده شود:

POST /api/v1/agents/events/{event}/add-users Agent event add-users

PATCH /api/v1/agents/events/{event}/remove-users Agent event remove-users

Labels

برچسب ها

به منظور دسته بندی و مدیریت کاربران در پنل ادمین در حین ساخت رویداد و جستجو کاربر، کاربرد دارد.

POST /api/v1/agents/labels/index Label index

توسط این api برگزار کننده می تواند لیست تمام برچسب ها را مشاهده نماید

body این درخواست شامل موارد زیر می باشد:

Take, skip که اجباری هستند.

title که عنوان برچسب است و می تواند فیلتر شود (این فیلد اختیاری است)

نمونه Response :

```
{
  "data": [
    {
      "id": 2,
      "title": "test title 2"
    },
    {
      "id": 1,
      "title": "test title 1"
    }
  ],
  "meta": {
    "total_count": 0
  }
}
```

GET

/api/v1/agents/labels/{label} Agent Label get

توسط این api برگزار کننده می تواند مشخصات یک برچسب خاص را دریافت کند

این مشخصات شامل نام برچسب، ID برگزارکننده ایجاد کننده برچسب و ID خود برچسب می باشند

لازم به ذکر است که label که داخل route ارسال می شود یک ID است مربوط به برچسب مورد نظر

نمونه Response :

```
{
  "data": [
    {
      "id": 9,
      "title": "55555555555555",
      "agent_id": 3,
      "deleted_at": null
    }
  ]
}
```

DELETE**/api/v1/agents/labels/{label}** Agent Label delete

توسط این api برگزارکننده می تواند برچسب خاصی را حذف کند

لازم به ذکر است که label که داخل route ارسال می شود یک ID است مربوط به برچسب مورد نظر

نمونه Response :

همانطور که در تصویر زیر میبینید در صورتی که برچسب در حال استفاده باشد امکان حذف آن وجود ندارد.

Code	Description	Links
200	<p>Media type application/json</p> <p>Controls Accept header.</p> <p>Example Value Schema</p> <pre>{ "message": "برچسب با موفقیت حذف شد." }</pre>	No links
400	<p>Media type application/json</p> <p>Example Value Schema</p> <pre>{ "message": "برچسب در حال استفاده است." }</pre>	No links

PATCH**/api/v1/agents/labels/{label}** Agent Label update

توسط این api برگزارکننده می تواند برچسب خاصی را بروزرسانی نماید.

لازم به ذکر است که label که داخل route ارسال می شود یک ID است مربوط به برچسب مورد نظر

body درخواست شامل title است که عنوان برچسب است و اجباری است .

نمونه Response :


```
{  
  "message": "برچسب با موفقیت ویرایش شد"  
}
```

POST

/api/v1/agents/labels/ Agent Label store

توسط این api برگزارکننده می تواند برچسبی را ایجاد نماید.

فیلدهای body درخواست شامل موارد زیر می باشند:

body درخواست شامل title است که عنوان برچسب است و اجباری است.

نمونه Response :

```
{  
  "message": "برچسب با موفقیت ایجاد شد"  
}
```

GET

/api/v1/agents/labels/list Agent Label list

. برگزار کننده می تواند لیست تمام برچسب های ایجاد شده توسط خود را دریافت کند api توسط این

نمونه Response :

```
{
  "data": [
    {
      "id": 2,
      "title": "test label 2"
    },
    {
      "id": 3,
      "title": "test label 3"
    }
  ]
}
```

Profile

GET `/api/v1/agents/profile` Agent Show Information By Agent

توسط این api کاربر می تواند مشخصات profile خود را مشاهده نماید که این مشخصات داخل Response به صورت زیر نمایش داده می شود و شامل اطلاعات شخصی کاربر و سرویس های فعال وی می باشد.

PATCH `/api/v1/agents/profile/update` This endpoint update agents information.

توسط این api کاربر می تواند profile خود را ویرایش کند.

مقادیر قابل ویرایش در زیر نشان داده شده اند:

فیلد name برای ویرایش نام کاربر می باشد و از نوع string می باشد.

فیلد `username` برای نام کاربری می باشد و از نوع `string` می باشد.

فیلد `email` حتما باید فرمت صحیح ایمیل را داشته باشد.

فیلد `cellphone` حتما باید از نوع عددی و فرمت صحیح شماره موبایل رایج باشد همراه با 0 اول

فیلد `status` وضعیت فعال یا غیر فعال بودن اکانت کاربر را مشخص میکند که یک `bool` است و

می تواند مقادیر 0 یا 1 را داشته باشد .

فیلد رمز عبور باید شامل حداقل 6 کاراکتر باشد.

فیلد `password_confirmation` باید دقیقا با فیلد رمز عبور یکی باشد.

فیلد `surname` هم از نوع `string` و نام خانوادگی کاربر می باشد.

لازم به ذکر است در این `api` هیچ مقدار `Required` وجود ندارد و تمامی فیلدها اختیاری هستند جز

اینکه اگر رمز عبور بخواهد عوض شود فیلد تکرار رمز عبور اجباری می شود.

Request body application/json

Pass user credentials

Example Value | Schema

```
{
  "name": "مهندس",
  "username": "mahyars777",
  "cellphone": "09126913637",
  "email": "mj_r_aipccssccro@gmail.com",
  "surname": "dsfsdfdsfsdf",
  "password": "mohsen-admin@123",
  "password_confirmation": "mohsen-admin@123",
  "status": 1
}
```

و `Response` به صورت زیر خواهد بود:

Responses		
Code	Description	Links
200	Media type <input type="text" value="application/json"/> Controls Accept header Example Value Schema <pre>{ "message": "برگزار کننده با موفقیت توسط ادمین ویرایش شد." }</pre>	No links
401	authentication is required and has failed or has not yet been provided.	No links
422	اطلاعات وارد شده صحیح نیست	No links
500	Server Error	No links

PATCH /api/v1/agents/profile/change-password Agent change password

توسط این api کاربر می تواند رمز عبور خود را تغییر دهد.

داخل body این api به صورت زیر می باشد:

مقدار فیلد current_password که باید رمز عبور فعلی کاربر وارد شود.

مقدار فیلد password که باید مقدار جدید مورد نظر کاربر وارد شود و حتما باید حداقل از 6 کاراکتر

و تشکیل شود همچنین password_confirmation باید عینا با فیلد password یکی باشد.

Request body required

application/json

Pass user credentials

Example Value | Schema

```
{
  "current_password": "123456df",
  "password": "123456df",
  "password_confirmation": "123456df"
}
```

و مقدار Response بدین صورت می باشد:

Code	Description	Links
200	<p>Media type</p> <p><input type="text" value="application/json"/></p> <p><small>Controls Accept header.</small></p> <p>Example Value Schema</p> <pre>{ "message": "رمز کاربر یا موفقیت تغییر کرد" }</pre>	No links
401	authentication is required and has failed or has not yet been provided.	No links
422	اطلاعات وارد شده صحیح نیست	No links
500	Server Error	No links

POST

/api/v1/agents/profile/upload-avatar Agent upload avatar

توسط این api کاربر می تواند اواتار جدید برای پروفایل خود آپلود کند.

فایل آواتار مورد نظر را با عنوان image باید داخل body ارسال کنید و فرمت های قابل قبول شامل

موارد زیر می باشد:

Jpeg,jpg,png

Request body required

application/json

Example Value | Schema

```
{  
  "image": "ubuntu.jpg"  
}
```

مقدار Response بدین صورت می باشد:

Responses

Code	Description	Links
200	<p>Media type</p> <p><input type="text" value="application/json"/></p> <p>Controls Accept header.</p> <p>Example Value Schema</p> <pre>{ "path": "filepath" }</pre>	No links
401	authentication is required and has failed or has not yet been provided.	No links
422	image only can jpg, jpeg, png	No links
500	Server Error	No links

گزارش حضور و غیاب :

توسط این api برگزارکننده می تواند لیست حضور و غیاب های رویداد ها را تهیه کند. لازم به توضیح است در خواست ارسال گزارش حضور و غیاب از رویداد و داخل محیط آن امکان پذیر است و این api تنها برای خواندن گزارش ها استفاده می شود.

POST

/api/v1/agents/events/{event}/attendance Agent event get-event-attendance-list

فیلد های body درخواست شامل موارد زیر می باشند:

body درخواست شامل date است که به عنوان تاریخ برگزاری رویداد است و اجباری است.

نمونه Response :

```
{
  "data": [
    {
      "id": "61d2dfc0578ada0062f31c08",
      "date": "2022-01-03T11:36:32.728Z",
      "teacherName": "jhon"
    },
    {
      "id": "61d2dfc0578ada0062f31c08",
      "date": "2022-01-03T11:36:32.728Z",
      "teacherName": "jhon"
    }
  ]
}
```


با توجه به پاسخی که از سمت سرور دریافت می شود به عنوان مثال مشاهده می شود که دو مرتبه حضور و غیاب در تاریخ یاد شده انجام شده است. که ID هر کدام از آن ها در response مشخص است.

اکنون برای دریافت جزئیات هر گزارش به api زیر مراجعه می شود:

```
POST /api/v1/agents/events/{event}/attendance/detail Agent event get-event-attendance-detail
```

فیلدهای body درخواست شامل موارد زیر می باشند:

body درخواست شامل ID حضور و غیاب مد نظر است که در api قبل در پاسخ به دست آمد:

نمونه Body:

```
{
  "attendanceId": "61d2dfc0578ada0062f31c9"
}
```

در پاسخ درخواست داده شده هم در فیلد دیتا دو مقدار دریافت می شود که آرایه آبجکت حاضران و غایبین به تفکیک است که به ترتیب در فیلدهای presents و absents مشاهده می شود.

نمونه Response:

```
{
  "data": [
    {
      "presents": {
        "firstName": "jhon",
        "lastName": "edu",
        "username": "jhon-edu"
      },
      "absents": {
        "firstName": "jhon",
        "lastName": "edu",
        "username": "jhon-edu"
      },
      "id": "61d2dfc0578ada0062f31c9"
    }
  ]
}
```

گزارش ورود و خروج :

ممکن است برگزار کننده ای بخواهد که بدون اینکه در لحظه خاصی گزارش حضور و غیاب را ثبت کرده باشد داده های مرتبط با ورود و خروج کاربران را در رویداد های خود مشاهده کند . این api برای این منظور تهیه شده است که با ارسال و امکان فیلتر داده ها این نیاز پاسخ داده شود .

فیلد های body درخواست شامل موارد زیر می باشند:

body درخواست شامل take هست که به عنوان تعداد رکورد های دریافتی و اجباری است .

body درخواست شامل skip هست که به عنوان موقعیت شروع رکورد هستش اجباری است.

نمونه Body:

```
{
  "take": 15,
  "skip": 0,
  "firstName": "jhon",
  "lastName": "due",
  "username": "jhon-due",
  "from": "2021-10-2",
  "to": "2021-10-3",
  "eventName": "class-math",
  "addressIp": "127.0.0.1",
  "sortColumn": "browser",
  "sortBy": 1,
  "deviceName": [
    "desktop",
    "mobile",
    "tablet"
  ],
  "eventType": [
    "class",
    "conference"
  ],
  "browser": [
    "safari",
    "chrome",
    "mozilla"
  ],
  "os": [
    "Mac OS",
    "Ubuntu",
    "Ios",
    "Android",
    "Windows",
    "Fedora"
  ]
}
```

نمونه Response:

```
{
  "data": [
    {
      "firstName": "jhon",
      "lastName": "jhon-due",
      "username": "jhon-due",
      "role": "user",
      "eventType": "class",
      "eventName": "sdfdsfsfs",
      "addressIp": "127.0.0.1",
      "deviceName": "desktop",
      "os": "Linux",
      "browser": "Chrome",
      "login_date": "2022-01-02T14:56:08.482000Z",
      "login_time": "2022-01-02T14:56:08.482000Z",
      "exit": "2022-01-02T14:56:23.175000Z"
    },
    {
      "firstName": "jhon",
      "lastName": "jhon-due",
      "username": "jhon-due",
      "role": "user",
      "eventType": "class",
      "eventName": "sdfdsfsfs",
      "addressIp": "127.0.0.1",
      "deviceName": "desktop",
      "os": "Linux",
      "browser": "Chrome",
      "login_date": "2022-01-02T14:56:08.482000Z",
      "login_time": "2022-01-02T14:56:08.482000Z",
      "exit": "2022-01-02T14:56:23.175000Z"
    }
  ]
}
```

نمایش فایل ضبط شده و آپلود شده بر اساس رویداد

POST

[/api/v1/agents/events/{event}/files](#) Agent Event Files

توسط این api کاربر می تواند لیست تمام فایل های ضبط شده و یا آپلود شده در رویداد های مربوط به خود را مشاهده کند.

فیلد take و skip فیلد های اجباری می باشند.

Take و skip چه هستند؟

Take به معنی تعداد رکورد هایی است که شما از دیتابیس دریافت میکنید

Skip را با یک مثال توضیح خواهم داد، فرض کنید ما می خواهیم 5 رکورد اول را دریافت نکنیم و از رکورد 6 به بعد را دریافت کنیم یعنی باید 5 رکورد را رد (skip) کنیم در صورت صفر بودن این مقدار از اولین رکورد دریافت انجام خواهد شد.

و مابقی فیلدها اختیاری هستند که شامل:

Title که عنوان فایلی است و می تواند داخل body به عنوان فیلتر ارسال شود.

Flag اگر این مقدار به صورت record ارسال شود تنها فایلی هایی ضبط شده را نشان میدهد و اگر ارسال نگردد همه فایل ها را نمایش میدهد

مثال:

Request body **required**

Example Value | Schema

```
{
  "flag": "record",
  "title": "testEvent1",
  "take": 100,
  "skip": 0
}
```

نوع response نمونه را می توانید مشاهده کنید:

```
[
  {
    "uploader": {
      "_id": "60f55725833367001d639641",
      "role": "admin",
      "firstName": "",
      "lastName": "كاوانو"
    },
    "remover": {
      "_id": null,
      "username": null,
      "firstName": null,
      "lastName": null,
      "role": null
    },
    "flag": "record",
    "converted": false,
    "isDeleted": false,
    "deleted_at": null,
    "_id": "65f809d85022e0944ad15530",
    "title": "kavano-meetingback-2024-2-18-13:0",
    "size": 87001,
    "serviceId": "60f558e2e99ed5002f24b8c7",
    "session": "612494ec303a6701848eceed",
    "format": ".mp4",
    "downloadable": false,
    "publicUrl": "https://download.alocom.co/rocket-api/files/records/kavano-meetingback-2024-2-18-13:0.mp4",
    "created_at": "2024-03-18T09:31:04.906Z"
  },
  {
    "uploader": {
      "_id": "60f55725833367001d639641",
      "role": "admin",
      "firstName": "",
      "lastName": "كاوانو"
    },
    "remover": {
      "_id": null,
      "username": null,
      "firstName": null,
      "lastName": null,
      "role": null
    },
    "flag": "record",
    "converted": false,
    "isDeleted": false,
    "deleted_at": null,
    "_id": "65f80a402b8e3530747beddd",
    "title": "kavano-meetingback-2024-2-18-13:1",
    "size": 380530,
    "serviceId": "60f558e2e99ed5002f24b8c7",
    "session": "612494ec303a6701848eceed",
    "format": ".mp4",
    "downloadable": false,
    "publicUrl": "https://download.alocom.co/rocket-api/files/records/kavano-meetingback-2024-2-18-13:1.mp4",
    "created_at": "2024-03-18T09:32:48.837Z"
  }
]
```