

Webinar

Руководство пользователя

API

Содержание

Пользовательское API	3
Доступ к пользовательскому API	3
Ограничения пользовательского API	3
Пользовательское API: модуль Организации	4
Пользовательское API: модуль Контакты	6
Пользовательское API: Мероприятия	14
Пользовательское API: Регистрация на мероприятие	38
Пользовательское API: Статистика	55
Пользовательское API: Записи	66
Пользовательское API: Чат и вопросы	70
Пользовательское API: Файлы	80
Пользовательское API: Тесты	90

Пользовательское API

В этом разделе будут размещены описания модулей пользовательского API.

URL сервера пользовательского API: <https://userapi.webinar.ru/v3/>
Все запросы необходимо отсылать по этому адресу.

Доступ к пользовательскому API

Доступ к пользовательскому API возможен только при наличии специального заголовка в запросе: **X-Auth-Token**.
Значение заголовка должно быть равно ключу API организации.
Ключ можно посмотреть в разделе **Бизнес** → **API**.
В данном разделе можно как скопировать ключ и изменить его.

Ограничения пользовательского API

Количество запросов к пользовательскому API не может превышать 2 запросов в секунду. При получении ответа

```
{
  error:
  {
    code: 429,
    message: "I only allow 2 requests per second. Try again."
  }
}
```

следует снизить частоту запросов.

Пользовательское API: модуль Организация

Данный модуль предназначен для получения и управления данными организации. Управлять данными организации/приглашать новых участников могут только участники с правами администратора организации.

Запросы:

GET /organization/members — получение списка участников организации. Возвращаются основные контактные данные участников, их роль в организации и их ID в системе, для использования в других запросах пользовательского API. Никакие параметры в этом запросе не используются. Описание полей ответа:

1. **id** - ID пользователя в системе.
2. **role** - Роль пользователя в организации. ADMIN - администратор, LECTURER - лектор.
3. **name** - Имя участника.
4. **secondName** - Фамилия участника.
5. **patrName** - Отчество участника.
6. **nickname** - Никнейм участника.
7. **email** - Email участника.
8. **phone** - Телефон участника.
9. **position** - Должность участника.
10. **organization** - Компания участника.
11. **sex** - Пол участника: **m** - мужчина, **f** - женщина, **o** - не известно.
12. **photo** - Массив с данными аватара

Коды ответов:

1. **200** - все проверки пройдены и список участников возвращен.
2. **401** - в запросе не передан заголовок **x-auth-token**.
3. **403** - значение **x-auth-token** не верно.

Пример запроса:

```
GET /organization/members
Headers
x-auth-token: HelloWorld
```

Пример ответа:

```
[
{
  "id": 11,
  "role": "LECTURER",
  "name": "Александр",
  "secondName": "Бровко",
  "patrName": "Александрович",
  "nickname": "Александр Бровко",
  "email": "a.brovko@webinar.ru",
  "phone": "79269087306",
  "position": "Менеджер по продукту",
  "organization": "Webinar",
  "sex": "m",
  "photo":
  {
    "url": "https://events-storage.webinar.ru/api-storage/files/webinar//ejhvMd.jpg",
  }
}]
```

Другие возможности модуля “Организация”:

1. При создании шаблона вебинара (POST /events) можно выбрать владельца организации, указав параметр `ownerId`.
2. При редактировании события изменить владельца вебинара, указав в запросе новое значение параметра `ownerId`.
3. Сменить владельца вебинара отдельным запросом PUT /organization/events/{eventId}/move.

Пользовательское API: модуль Контакты

Модуль создан для предоставления доступа к функционалу управления данными контактов (адресной книги пользователя) через пользовательское API.

Запросы.

POST /users/{userId}/contacts — создание контакта для пользователя `UserId`.

Для задания аватара нужно передать в запросе параметр `photo`, значением которого является либо ссылка на картинку, либо же ID файла в файловой системе платформы.

В ответ на запрос возвращается сущность `contact`, включая связанные сущности `tag[]` и `avatar`(сущность `FileSystem` в системе)

Обязательные параметры:

1. **email** - email контакта.

Необязательные параметры:

1. **name** - Имя контакта.
2. **secondName** - Фамилия контакта.
3. **company** - Компания, в которой работает контакт.
4. **position** - Должность, которую занимает контакт.
5. **phoneMain** - Контактный телефон.
6. **tags** - Массив тегов, которые будут прикреплены к контакту при создании. Каждый тег - отдельная строка.
7. **photo** - Аватар для контакта в адресной книге пользователя. Значением может быть либо ссылка на картинку, либо же ID записи картинки в файловой системе.

Коды ответов:

1. **201** - контакт успешно создан.
2. **400** - не передан заголовок или **email**.
3. **401** - в запросе не передан заголовок **x-auth-token**.
4. **403** - значение **x-auth-token** не верно или когда пользователь, для которого создается контакт, не принадлежит к организации.
5. **404** - пользователь не найден по **userId**.
6. **409** - контакт с переданным **email** уже существует для пользователя.

Пример запроса:

```
POST /users/274/contacts
```

Headers

```
x-auth-token: HelloWorld
```

Parameters

```
email: krondor124@inbox.ru
```

```
name: New
```

```
secondName: Contact
```

```
tags[0]: Hello
```

```
photo: https://www.dropbox.com/s/eh5ntzxhtv47pwb/avatar.jpg?dl=0
```

Пример ответа:

```
{ "userId": 274,  
  "name": "New",  
  "secondName": "Contact",  
  "email": "krondor124@inbox.ru",  
  "avatar": {  
    "url":  
    "https://events-storage/api-storage/files/webinar/2016/03/01/VkxeFV3bBgTPJj  
dTSfJRd7doRa9KPGd3B2T1NLpFUITp6sNG.jpg?dl=0",  
    "thumbnails": { } },  
  "tags": [ {  
    "id": 523,  
    "isDeleted": false,  
    "createAt": "2016-02-17T18:23:15+0300",  
    "name": "Hello" } ],  
  "id": 542 }
```

PUT /contacts/{contactId} - обновление контакта по contactId. Параметры запроса аналогичны запросу на создание контакта, за исключением полей tags. Обновлять теги через этот запрос нельзя. Обновлять аватар возможно. В случае, если передан аватар, но старый аватар удаляется и заменяется на новый. В ответ на запрос возвращается сущность Contact.

Необязательные параметры:

1. **name** - имя контакта.
2. **email** - email контакта.
3. **secondName** - фамилия контакта.
4. **company** - компания, в которой работает контакт.
5. **position** - должность, которую занимает контакт.
6. **phoneMain** - контактный телефон.
7. **photo** - аватар для контакта в адресной книге пользователя. Значением может быть либо ссылка на картинку, либо же ID записи картинки в файловой системе.

Коды ответов:

1. **200** - контакт успешно обновлен.
2. **401** - в запросе не передан заголовок **x-auth-token**.
3. **403** - значение **x-auth-token** не верно или когда пользователь, которому принадлежит, не принадлежит к организации.
4. **404** - контакт не найден по **contactId**.

Пример запроса:

```
PUT /contacts/542
```

```
Headers
```

```
x-auth-token: HelloWorld
```

```
Parameters
```

```
name: New Super
```

```
secondName: Mega Contact
```

Пример ответа:

```
{ "userId": 274,
  "name": "New Super",
  "secondName": "Mega Contact",
  "email": "krondor124@inbox.ru",
  "avatar": {
    "url":
    "https://events-storage/api-storage/files/webinar/2016/03/01/VkxeFV3bBgTPJj
    dTSfJRd7doRa9KPGd3B2T1NLpFUITp6sNG.jpg?dl=0",
  },
  "tags": [
    {
      "id": 523,
      "isDeleted": false,
      "createAt": "2016-02-17T18:23:15+0300",    "name": "Hello"
    } ],
  "id": 542,
```

DELETE /contacts/{contactId} — удаление контакта по contactId. При запросе происходят стандартные проверки доступа и корректности данных. В случае успеха возвращается ответ со статусом 204 и пустым телом ответа.

Коды ответов:

1. **204** - Когда контакт успешно удален.
2. **401** - Когда в запросе не передан заголовок **x-auth-token**.
3. **403** - Когда значение **x-auth-token** не верно или когда пользователь, которому принадлежит, не принадлежит к организации.
4. **404** - Когда контакт не найден по **contactId**.

Пример запроса:

```
DELETE /contacts/542
Headers
x-auth-token: HelloWorld "isDeleted": false
```

GET /contacts/{contactId} — получение контакта по contactId. При запросе происходят стандартные проверки доступа и корректности данных. В случае успеха возвращается ответ со статусом 200 и данными контакта в теле ответа.

Коды ответов:

1. **200** - Когда контакт успешно найден и возвращен.
2. **401** - Когда в запросе не передан заголовок **x-auth-token**.
3. **403** - Когда значение **x-auth-token** не верно.
4. **404** - Когда контакт не найден по **contactId**.

Пример запроса:

```
GET /contacts/542
Headers
x-auth-token: HelloWorld
```

Пример ответа:

```
{ "userId": 274,
  "name": "New Super",
  "secondName": "Mega Contact",
  "email": "krondor124@inbox.ru",
  "avatar": {
    "url":
    "https://events-storage/api-storage/files/webinar/2016/03/01/VkxeFV3bBgTPJj
    dTSfJRd7doRa9KPGd3B2T1NLpFUITp6sNG.jpg?dl=0",
    "thumbnails": {   } },
  "tags": [   ] ,
  "id": 542,
  "isDeleted": false }
```

POST /contacts/tags/add — добавить несколько тегов к нескольким контактам.

Обязательными параметрами являются массивы `contactIds` и `tags`. Наборы ID контактов и названия тегов, соответственно. Если какие-то теги отсутствуют в БД — они добавляются. При обработке запроса теги добавляются только для тех контактов, которые принадлежат членам организации. Контакты, принадлежащие другим пользователям, даже если их ID переданы в запросе, игнорируются. В ответ на запрос возвращается количество обновленных контактов.

Обязательные параметры:

1. **contactIds** - Массив ID контактов, для которых добавляются теги.
2. **tags** - Массив тегов для добавления к контактам.

Коды ответов:

1. **201** - теги успешно добавлены к контактам.
2. **401** - в запросе не передан заголовок **x-auth-token**.
3. **403** - значение **x-auth-token** не верно.

Пример запроса:

```
POST /contacts/tags/add
```

```
Headers
```

```
x-auth-token: HelloWorld
```

```
Parameters
```

```
tags[0]: New
```

```
tags[1]: Super
```

```
tags[2]: Tag
```

```
contactIds[0]: 542
```

```
contactIds[1]: 534
```

Пример ответа:

```
{"updatedContacts":2}
```

POST /contacts/tags/remove — удалить несколько тегов от нескольких контактов.

Обязательными параметрами являются массивы `contactIds` и `tags`. Наборы ID контактов и названия тегов соответственно. При обработке запроса теги добавляются только для тех контактов, которые принадлежат членам организации. Контакты, принадлежащие другим пользователям, даже если их ID переданы в запросе, игнорируются. В ответ на запрос возвращается количество обновленных контактов.

Обязательные параметры:

1. **contactIds** - массив ID контактов, для которых добавляются теги.
2. **tags** - массив тегов для добавления к контактам.

Коды ответов:

1. **204** - теги успешно удалены из контактов.
2. **401** - в запросе не передан заголовок **x-auth-token**.
3. **403** - значение **x-auth-token** не верно.

Пример запроса:

```
POST /contacts/tags/remove
```

```
Headers
```

```
x-auth-token: HelloWorld
```

```
Parameters
```

```
tags[0]: New
```

```
tags[1]: Super
```

```
tags[2]: Tag
```

```
contactIds[0]: 542
```

```
contactIds[1]: 534
```

В качестве ответа возвращается пустота со статусом 204.

GET /contacts/search — поиск контактов по различным критериям. В ответ на запрос возвращается массив сущностей Contact

Необязательные параметры:

1. **userIds** - массив ID пользователей, контакты которых нужно получить.
2. **contactIds** - массив ID контактов, которые нужно получить.
3. **contactsData** - набор параметров контактов, по которым осуществляется поиск. Можно указать **name**, **secondName**, **company**, **email**, **phoneMain**, **position**.
4. **tags** - набор тегов, по которым осуществляется поиск.

Коды ответов:

1. **200** - контакты успешно найдены и возвращены.
2. **401** - в запросе не передан заголовок **x-auth-token**.
3. **403** - значение **x-auth-token** не верно.

Пример запроса:

```
GET /contacts/search?userIds[0]=274
```

Headers

```
x-auth-token: HelloWorld
```

Пример ответа:

```
[ {  
  "userId": 274,  
  "name": "New Super",  
  "secondName": "Mega Contact",  
  "email": "krondor124@inbox.ru",  
  "avatar": {  
    "url":  
    "https://events-storage/api-storage/files/webinar/2016/03/01/VkxeFV3bBgTPJj  
    dTSfJRd7doRa9KPGd3B2T1NLpFUITp6sNG.jpg?dl=0",  
    "thumbnails": {      } },  
  "tags": [  ],  
  "id": 542 }, ... ]
```

Пользовательское API: Мероприятия

Модуль создан для предоставления доступа к функционалу мероприятий.

Основная концепция мероприятий заключается в разделении понятия мероприятия на две сущности: **Event** и **EventSession**.

1. **Event** представляет собой шаблон мероприятия(названия, описание, лекторы, файлы и пр.), включающий так же правило повторения(определения дат/времени, когда это мероприятие будет повторяться). Также Event может называться серией одинаковых мероприятий.
2. **EventSession** представляет собой конкретное мероприятие в конкретную дату/время. EventSession всегда принадлежит какому-то Event-у. Все EventSession повторяющегося мероприятия принадлежат одному Event-у. При создании EventSession все поля заполняются соответствующими значениями из Event-a. После создания можно редактировать EventSession независимо от Event(например на конкретную EventSession добавить нового лектора или загрузить дополнительный файл) и Event независимо от всех, принадлежащих ему EventSession. Для того, чтобы запустить вебинар, необходимо сначала создать Event, потом для данного Event создать EventSession и работать уже непосредственно с EventSession.

Разные запросы пользовательского API оперируют разными сущностями. По большей части Event и EventSession имеют одинаковые поля с одинаковыми доступными значениями, но есть и различия, о которых речь пойдет дальше.

Описания полей Event и EventSession.

Поля, идентичные для обеих сущностей:

1. **id** - ID сущности в БД. ID сущностей Event и EventSession независимы друг от друга.
2. **isDeleted** - флаг, обозначающий то, что сущность удалена.
3. **updateUserId** - ID пользователя, который последним обновлял значения полей данной сущности.
4. **deleteUserId** - ID пользователя, который удалил данную сущность. До момента удаления это поле хранит значение **null**.
5. **lang** - основной язык, используемый на мероприятии. При создании сущности берется значение, указанное в профиле пользователя, который создает мероприятие.
6. **access** - Параметр, определяющий уровень доступа к мероприятию. Значение флага представляет из себя битовую маску из 4 битов, где: первый бит означает свободный вход, второй бит означает наличие пароля на вход, третий бит означает регистрацию без ручной модерации участников, а четвертый бит

означает регистрацию с последующей ручной модерацией участников. Таким образом возможны следующие значения:

- 1(0001) — свободный доступ без пароля.
 - 3(0011) — свободный доступ с паролем.
 - 4(0100) — регистрация без последующей ручной модерации участников и без пароля.
 - 6(0110) — регистрация без последующей ручной модерации участников, но с паролем.
 - 8(1000) — регистрация с последующей ручной модерацией участников и без пароля.
 - 10(1010) — регистрация с последующей ручной модерацией участников и с паролем.
7. **password** - пароль для входа на мероприятие. Никогда не отдается в ответ на какой-либо запрос.
 8. **createUserId** - ID пользователя, который создал мероприятие(по умолчанию – владелец организации). При создании EventSession: EventSession.createUserId = Event.createUserId. Пользователь, создавший мероприятие обладает административными правами на нем.
 9. **createUser** - Объект с данными пользователя, который создал мероприятие. Данный объект возвращается в ответах на запросы данного модуля вместо **createUserId**.
 10. **name** - Название мероприятия.
 11. **description** - Детальное описание мероприятия.
 12. **startsAt** - Дата/Время начала мероприятия. Для Event - это время начала самой первой EventSession. Для EventSession - фактическое время начала.
 13. **utcStartsAt** - Timestamp времени начала мероприятия. Фактически это Timestamp от значения **startsAt**
 14. **endsAt** - Дата/Время завершения мероприятия. Для Event с повторением - это верхняя граница, после которой новые EventSession уже не создаются. Для EventSession - дата/время фактического завершения мероприятия.
 15. **duration** - Заявленная длительность мероприятия. Значение данного поля должно подпадать под регулярное выражение

```
^P(((\d+D))(\d+W))(\d+W\d+D)))(T(\d+H)*(\d+M)*(\d+S*))(((\d+D))(\d+W))(\d+W\d+D))T(\d+H)*(\d+M)*(\d+S*))$
```

Примеры значений:

- **PT0H30M0S** - 30 минут
 - **PT1H0M0S** - 1 час
 - **PT1H30M0S** - 1 час 30 минут.
16. **additionalFields** - Массив объектов, хранящих дополнительные поля, которые пользователь должен заполнить при регистрации на мероприятие. Каждый объект имеет следующие поля:
 - **key** - Уникальный ключ-идентификатор дополнительного поля. Через этот идентификатор связываются

дополнительные поля мероприятия и ответы пользователей. Значение генерируется автоматически при создании дополнительного поля.

- **label** - Текст вопроса.
 - **type** - Тип вопроса. Может быть либо *text*(ответ просто текстом), либо **radio**(выбор ответа из предложенных вариантов)
 - **values** - массив вариантов ответов для типа **radio**.
17. **isArchived** - флаг, устанавливаемый при архивации мероприятия. Заархивированное мероприятие не удаляется из БД, но все связанные с ним данные записываются в отдельный файл и в дальнейшем возвращается содержимое этого файла.
 18. **imageld** - ID изображения, используемого фоном данного мероприятия. ID из файловой системы платформы.
 19. **image** - объект изображения, используемого фоном данного мероприятия. Данный объект возвращается в ответах на запросы данного модуля вместо **imageld**.
 20. **organizationId** - ID организации, которой принадлежит данное мероприятие.
 21. **organization** - Объект организации, которой принадлежит данное мероприятие. Данный объект возвращается в ответах на запросы данного модуля вместо **eventId**.
 22. **lectors** - Список лекторов, проводящих данное мероприятие.
 23. **announceFiles** - Список файлов, доступных в анонсе(лендинг, страница на которую попадает пользователь для того, чтобы зарегистрироваться или перед тем, как непосредственно попасть на страницу мероприятия) мероприятия.
 24. **tags** - Список тегов, связанных с данным мероприятием.

Далее идут поля, уникальные (или различающиеся) для сущностей Event и EventSession.

Event:

1. **status** - Статус мероприятия. Возможен один из трех вариантов: **ACTIVE**, **STOP**, **RESERVED**
 - **ACTIVE** - Активное мероприятие, можно создавать EventSessions и запускать их. Можно редактировать мероприятие. Можно просматривать статистику.
 - **STOP** - Мероприятие уже завершено. Запрещено редактирование мероприятия и создание EventSession. Можно просматривать статистику.
 - **RESERVED** - Данный статус означает, что данное мероприятие - болванка и подлежит последующему редактированию. Для такого Event нельзя создавать EventSessions и нельзя просматривать статистику.
2. **urlAlias** - Уникальный токен, используемый при формировании ссылок на данное мероприятие в интернете, подставляемый вместо Event.id. Значение уникально в паре с значениями createUserId или organizationId.
3. **rule** - Правило определения даты/времени для повторяющихся мероприятий. По этому правилу определяется дата/время старта для EventSession. Для мероприятий без повторения значение этого поля равно **null**. Примеры значений:
 - **FREQ=DAILY** - Ежедневное повторение.
 - **FREQ=WEEKLY;INTERVAL=2** - Повторять каждые две недели.
 - **FREQ=MONTHLY** - Повторять каждый месяц.
 - **FREQ=WEEKLY;BYDAY=TU,TH,SA;INTERVAL=3** - Повторять по вторникам, четвергам и субботам каждые три недели.
4. **exceptionDates** - Список дат, которые будут игнорироваться при определении даты старта следующей EventSession, даже если они попадают под **rule**. В этот список всегда попадает дата, указанная в поле **createdAt**.
5. **isEventRegAllowed** - Флаг, означающий возможность регистрироваться сразу на всю серию мероприятий. В противном случае пользователь должен будет зарегистрироваться на каждую EventSession отдельно

EventSession:

1. **status** - Статус сессии. Возможен один из трех вариантов: **ACTIVE**, **START**, **STOP**
 - **ACTIVE** - Сессия активна, может быть отредактирована, запущена. Не возможен просмотр статистики.
 - **START** - Сессия запущена и в данный момент в эфире. Не может быть изменена. Возможен просмотр статистики.
 - **STOP** - Сессия завершена. Не может быть отредактирована. Возможен просмотр статистики. Когда завершается последняя EventSession, для Event проставляется статус **STOP**.
2. **eventId** - ID сущности Event, к которой относится данная EventSession.
3. **estimatedAt** - Запланированная Дата/Время начала мероприятия.

Запросы:

POST /events — создание мероприятия. При создании мероприятия возможно, кроме основных свойств добавить лекторов и теги к мероприятию.

Сначала создается Event, затем внутри него (обязательно) EventSession. Для серийных мероприятий требуется отдельное создание каждой EventSession в соответствии с правилом повторения.

В качестве создателя мероприятия будет установлен создатель организации. В ответ на запрос возвращается массив из двух элементов:

1. **eventId** - ID сущности Event.
2. **link** - ссылка на страницу лендинга мероприятия.

Обязательные параметры:

1. **name** - Название мероприятия.
2. **access** - Уровень доступа к мероприятию.

Необязательные параметры:

1. **status** - статус мероприятия на момент создания. может быть либо ACTIVE, либо STOP.
2. **password** - пароль для входа на мероприятие.
3. **description** - более полное описание мероприятия.
4. **lang** - язык мероприятия. определяет язык интерфейса, язык проведения мероприятия. может быть одним из значений: RU, EN, UK, TR.
5. **urlAlias** - уникальный строковый идентификатор мероприятия, подставляемый в ссылки вместо цифрового ID.

6. **startsAt** - дата/время начала мероприятия. Параметр передается в виде массива startsAt, состоящего из двух элементов: date и time. date - массив из трех элементов: year, month, day. time - массив из двух элементов: hour и minute.
7. **endsAt** - дата/время запланированного завершения мероприятия. формат данных аналогичен параметру startsAt.
8. **rule** - правило генерации дат повторения мероприятия. Подробно можно ознакомиться по ссылке: <https://tools.ietf.org/html/rfc5545#section-3.3.10>
9. **duration** - запланированная длительность мероприятия.
10. **isArchive** - отправлено ли мероприятие в архив. Булево значение.
11. **additionalFields** - массив дополнительных полей, которые пользователь должен будет ввести при регистрации на мероприятие. каждое поле задается массивом из элементов: label - название поля, type - тип поля(поле со свободным ответом(text) либо же выбор из предустановленных вариантов(radio)), values - массив предустановленных вариантов ответа, placeholder - значение, подставляемое в ответ по умолчанию.
12. **image** - ID файла в файловой системе, используемое в качестве фонового изображения для мероприятия.
13. **lectorIds** - массив ID пользователей, которые будут добавлены на мероприятие в качестве лекторов.
14. **tags** - набор тегов, которые будут добавлены к мероприятию.
15. **ownerId** - сотрудник организации, который будет назначен владельцем мероприятия. По умолчанию владелец - создатель организации. Получить идентификатор сотрудников можно запросом GET /organization/members (подробнее в разделе Пользовательское API: модуль Организация).
16. **type** - тип мероприятия. Может принимать значения webinar или meeting. По умолчанию: webinar.
17. **defaultRemindersEnabled** - напоминания по умолчанию. Может принимать значения false (напоминаний не будет) или true (стандартные напоминания за 1 час и за 15 минут до мероприятия будут созданы). По умолчанию: true.

Коды ответов:

1. **201** - мероприятие успешно создано.
2. **400** - не переданы какие-то из обязательных параметров, либо же переданные значения не соответствуют типу поля.
3. **401** - в запросе не передан заголовок **x-auth-token**.
4. **403** - значение **x-auth-token** не верно.

Пример запроса:

```
POST /events
Headers
x-auth-token: HelloWorld
Parameters:
name: New event
access: 4
description: new event for demo
lectorIds[0]: 181
lectorIds[1]: 274
additionalFields[0][label]: Просто поле
additionalFields[0][type]: text
additionalFields[1][label]: Вопрос
additionalFields[1][type]: radio
additionalFields[1][values][0]: Ответ 1
additionalFields[1][values][1]: Ответ 2
startsAt[date][year]: 2016
startsAt[date][month]: 3
startsAt[date][day]: 20
startsAt[time][hour]: 12
startsAt[time][minute]: 0
tags[0]: Just
tags[1]: Tag
```

Пример ответа:

```
{
  "eventId": 1489,
  "link": "https://events.webinar.ru/lzd2/1489" }
```

GET /organization/events/schedule — получить информацию о мероприятиях организации.

По умолчанию запрос отдает запланированные мероприятия (ACTIVE). Можно запросить завершённые (STOP) или идущие (START), указав параметр “status”.

Возвращаются только те мероприятия, которые попадают в промежуток **from** - **to** параметров запроса.

Если параметр **from** не указан, значит выборка будет осуществляться от текущей даты и времени.

Данные возвращаются в виде массива Event.

Если для серийного мероприятия у eventSessions есть только даты, это означает, что на указанные даты не были созданы eventSessions

Необязательные параметры:

1. **name** - название мероприятия. Возвращаются только те мероприятия, в названиях которых содержится значение параметра name.
2. **status** - массив статусов. По умолчанию отображаются только активные ACTIVE (активные мероприятия). Можно запросить завершённые (STOP) или идущие (START). Можно два или три сразу.
3. **access** - массив чисел(битовых масок), определяющих правила доступа на мероприятие. Возвращаются те мероприятия, для которых модификатор доступа входит в переданный массив.
4. **from** - дата начала периода выборки. По умолчанию - сейчас.
5. **to** - дата окончания периода выборки. По умолчанию - значение from плюс 1 год.
6. **page** - номер страницы для пагинации выборки. По умолчанию - 1.
7. **perPage** - количество элементов на странице выборки. Допустимые значения: 10, 50, 100, 250, 500. По умолчанию - 10.

Коды ответов:

1. **200** - мероприятия успешно найдены и возвращены.
2. **401** - в запросе не передан заголовок **x-auth-token**.
3. **403** - значение **x-auth-token** не верно.

Пример запроса:

```
GET
/organization/events/schedule?status[0]=stop&status[1]=start&status[2]=active&from=2019-01-01&to=2019-02-15&page=1

Headers
x-auth-token: HelloWorld
```

Пример ответа:

```
[ {
  "id": "1",
  "status": "ACTIVE",
  "rule": "FREQ=WEEKLY;INTERVAL=2",
  "name": "sdsd фвафыва",
  "description": "RESERVED ывапывап",
  "startsAt": "2015-08-10T15:26:00+0300",
  "isArchive": "0",
  "createUserId": "181",
  "timezoneName": "Europe/Moscow",
  "image": "image_url.png",
  "eventSessions": [
    { "startsAt": "2016-03-21 15:26" },
  ],
  "tags": [],
  "files": [],
  "lectors": [
    { "name": "sdf",
      "secondName": "1" },
  ]
}
```

GET /users/userID/events/schedule — получить информацию о мероприятиях участника организации. Можно получить информацию как по всем мероприятиям участника, так и фильтровать результат при помощи необязательных параметров запроса.

Данные возвращаются в виде массива Event-ов, имеющих поле eventSessions, в котором находятся все, подходящие по периоду, сессии. Если в массиве eventSessions находятся только даты - это означает, что данное мероприятие имеет правило повторения и на указанные даты как раз приходятся запланированные сессии, но сами записи в БД об этих сессиях еще не созданы. Возвращаются только те запланированные сессии, которые попадают в промежуток **from** - **to** параметров запроса. Вместе с данными мероприятия возвращаются также данные по назначенным лекторам, прикрепленным к описанию мероприятия файлам и тегам мероприятия.

Необязательные параметры:

1. **name** - название мероприятия. Возвращаются только те мероприятия, в названиях которых содержится значение параметра name.
2. **status** - массив статусов. Возвращаются только те мероприятия, статус которых совпадает с одним из переданных значений. Допустимые значения - ACTIVE(активные мероприятия), STOP(завершенные мероприятия), START(запущенные в данный момент мероприятия).
3. **access** - массив чисел(битовых масок), определяющих правила доступа на мероприятие. Возвращаются те мероприятия, для которых модификатор доступа входит в переданный массив.
4. **from** - дата начала периода выборки. По умолчанию - сейчас.
5. **to** - дата окончания периода выборки. По умолчанию - значение from плюс 1 год.
6. **page** - номер страницы для пагинации выборки. По умолчанию - 1.
7. **perPage** - количество элементов на странице выборки. Допустимые значения: 10, 50, 100, 250, 500. По умолчанию - 10.

Коды ответов:

1. **200** - мероприятия успешно найдены и возвращены.
2. **401** - в запросе не передан заголовок **x-auth-token**.
3. **403** - значение **x-auth-token** не верно.

Пример запроса:

```
GET
/users/274/events/schedule?page=1&perPage=50&status[]=ACTIVE&from=2016-03-11&to=2016-06-18

Headers
x-auth-token: HelloWorld
```

Пример ответа:

```
[ {
  "id": "260",
  "status": "ACTIVE",
  "rule": "FREQ=WEEKLY",
  "name": "Мое новое мероприятие",
  "description": "пускай тут будет тема",
  "startsAt": "2015-08-07T02:00:00+0300",
  "isArchive": "0",
  "createUserId": "274",
  "timezoneName": "Europe/Moscow",
  "image":
  "https://events-storage/api-storage/files/webinar/2016/03/03/eh3NActdCZxPuxFFUvaKGd1Lyb2ACAtapj2ScJ7PnATIITZt.png",
  "eventSessions": [
    {"startsAt": "2016-03-21 15:26" },
    {"startsAt": "2016-04-04 15:26"},
    {"startsAt": "2016-04-18 15:26" },
  ],
  "tags": [
    "просто",
    "мероприятие"
  ],
  "files": [ ],
  "lectors": [ ]
}, ... ]
```

GET /organization/events/{eventId} — получить информацию по конкретному мероприятию. Мероприятие должно принадлежать организации.

Обязательные параметры:

1. **eventId** - ID мероприятия. Указывается в заголовке.

Коды ответов:

1. **200** - мероприятие успешно найдено и возвращено.
2. **401** - в запросе не передан заголовок **x-auth-token**.
3. **403** - значение **x-auth-token** не верно.
4. **404** - мероприятие не найдено по ID.

Пример запроса:

```
GET /organization/events/3631
```

```
Headers
```

```
x-auth-token: HelloWorld
```

Пример ответа:

```
{
  "id": 2186015,
  "name": "Новое мероприятие сегодня",
  "status": "ACTIVE",
  "access": 1,
  "lang": "RU",
  "startsAt": "2019-03-18T11:41:00+0300",
  "utcStartsAt": 1552898460,
  "createUserId": 825893,
  "timezoneId": 1,
  "endsAt": "2019-04-18T23:59:00+0300",
  "organizationId": 43137,
  "type": "webinar",
  "createUser": {
    "id": 825893,
    "name": "Мария",
    "secondName": "Финогеева",
    "email": "m.finogeeva@webinar.ru"
  },
  "rule": "FREQ=DAILY;COUNT=1",
  "lectors": [ ],
  "tags": [],
  "announceFiles": [],
  "files": [],
  "eventSessions": [
    {
      "id": 2237595,
      "name": "Новое мероприятие сегодня",
      "status": "ACTIVE",
      "access": 1,
```

```
"lang": "RU",
"startsAt": "2019-03-18T11:41:00+0300",
"utcStartsAt": 1552898460,
"createUserId": 825893,
"timezoneId": 1,
"organizationId": 43137,
"type": "webinar",
"createUser": {
  "id": 825893,
  "name": "Мария",
  "secondName": "Финогеева",
  "email": "m.finogeeva@webinar.ru  },
"lectors": [
],
"tags": [],
"announceFiles": [],
"files": []
}
]
}
```

PUT /organization/events/{eventId} — редактирование мероприятия. При редактировании мероприятия возможно изменять только поля, относящиеся непосредственно к сущности мероприятия. Поля, относящиеся к связанным сущностям(лекторы, файлы, теги) редактировать нельзя. Дополнительные поля редактировать можно. При успешном обновлении сущности сервер возвращает код ответа 204 и пустое тело.

Необязательные параметры:

1. **name** - Название мероприятия.
2. **access** - Уровень доступа к мероприятию.
3. **status** - статус мероприятия на момент создания. может быть либо ACTIVE, либо STOP.
4. **password** - пароль для входа на мероприятие.
5. **description** - более полное описание мероприятия.
6. **lang** - язык мероприятия. определяет язык интерфейса, язык проведения мероприятия. может быть одним из значений: RU, EN, UK, TR.
7. **urlAlias** - уникальный строковый идентификатор мероприятия, подставляемый в ссылки вместо цифрового ID.
8. **startsAt** - дата/время начала мероприятия. Параметр передается в виде массива startsAt, состоящего из двух элементов: date и time. date - массив из трех элементов: year, month, day. time - массив из двух элементов: hour и minute.
9. **endsAt** - дата/время запланированного завершения мероприятия. формат данных аналогичен параметру startsAt. Параметр передается в виде массива, состоящего из двух элементов: date и time. date - массив из трех элементов: year, month, day. time - массив из двух элементов: hour и minute.
10. **rule** - правило генерации дат повторения мероприятия. Подробно можно ознакомиться по ссылке:
<https://tools.ietf.org/html/rfc5545#section-3.3.10>
11. **duration** - запланированная длительность мероприятия.
12. **additionalFields** - массив дополнительных полей, которые пользователь должен будет ввести при регистрации на мероприятие. каждое поле задается массивом из элементов: label - название поля, type - тип поля(поле со свободным ответом(text) либо же выбор из предустановленных вариантов(radio)), values - массив предустановленных вариантов ответа, placeholder - значение, подставляемое в ответ по умолчанию.
13. **image** - ID файла в файловой системе, используемое в качестве фонового изображения для мероприятия.
14. **ownerId** - сотрудник организации, который будет назначен владельцем мероприятия. Если не передать, владелец не изменится.

Коды ответов:

1. **204** - мероприятие успешно обновлено.
2. **401** - в запросе не передан заголовок **x-auth-token**.
3. **403** - значение **x-auth-token** не верно.
4. **404** - **мероприятие** не найдено по ID.

Пример запроса:

```
PUT /organization/events/4958
```

Headers

```
x-auth-token: HelloWorld
```

Parameters

```
name: Очень новое мероприятие
```

```
access: 1
```

PUT /organization/events/{eventId}/move — сменить владельца мероприятия. При успешном обновлении сущности сервер возвращает код ответа 204 и пустое тело.

Обязательные параметры:

1. **userId** - ID пользователя (сотрудника организации), которому будет перенесено мероприятия.

Коды ответов:

1. **204** - мероприятие успешно обновлено.
2. **401** - в запросе не передан заголовок x-auth-token.
3. **403** - значение x-auth-token неверно.
4. **404** - когда мероприятие не найдено по ID.

Пример запроса:

```
PUT /organization/events/4958/move
```

Headers

```
x-auth-token: HelloWorld
```

Parameters

```
userId: 4321
```

DELETE /organization/events/{eventId} - Удаление серии мероприятий по ID, Удаляется каждая сессия, интерактивные записи. По данной серии больше нельзя получать статистику. В ответ на запрос возвращается статус 204 и пустое тело ответа.

Коды ответов:

1. **200** - Когда мероприятие успешно найдено и возвращено.
2. **401** - Когда в запросе не передан заголовок **x-auth-token**.
3. **403** - Когда значение **x-auth-token** не верно.
4. **404** - Когда мероприятие не найдено по ID.

Пример запроса:

```
DELETE /system/organization/events/4958
```

Headers

```
x-auth-token: HelloWorld
```

POST /events/{eventId}/sessions — создать мероприятие (EventSession) для указанного шаблона или серии (Event).

По умолчанию все значимые поля EventSession заполняются из соответствующих полей Event. Но так же можно передать в этом запросе параметры для значений, которые будут индивидуальны для этой сессии. В ответ на запрос приходит массив из двух элементов:

1. **eventSessionId** - ID сущности EventSession.
2. **link** - ссылка на лендинг мероприятия для конкретной EventSession.

Необязательные параметры:

1. **name** - Название мероприятия.
2. **description** - Описание мероприятия.
3. **access** - Уровень доступа к мероприятию.
4. **lang** - язык мероприятия. определяет язык интерфейса, язык проведения мероприятия. может быть одним из значений: RU, EN, UK, TR.
5. **timezone** - ID временной зоны, по времени которой считается время проведения мероприятия. подразумевается ID временной зоны в нашей системе.
6. **startsAt** - дата/время начала мероприятия. Параметр передается в виде массива startsAt, состоящего из двух элементов: date и time. date - массив из трех элементов: year, month, day. time - массив из двух элементов: hour и minute.
7. **additionalFields** - массив дополнительных полей, которые пользователь должен будет ввести при регистрации на мероприятие. каждое поле задается массивом из элементов: label - название поля, type - тип поля(поле со свободным ответом(text) либо же выбор из предустановленных вариантов(radio)), values - массив предустановленных вариантов ответа, placeholder - значение, подставляемое в ответ по умолчанию.
8. **image** - ID файла в файловой системе, используемое в качестве фонового изображения для мероприятия.
9. **lectorIds** - массив ID пользователей, которые будут добавлены на мероприятие в качестве лекторов. Данные лекторы заменят тех, которые установлены в Event.
10. **tags** - набор тегов, которые будут добавлены к мероприятию. Данные теги заменят те, которые установлены в Event.

Коды ответа:

1. **201** - eventsession успешно создана.
2. **401** - в запросе не передан заголовок **x-auth-token**.
3. **403** - значение **x-auth-token** не верно.
4. **404** - Event не найден в организации.
5. **409** - eventsession не может быть создана.
Для несерийного мероприятия: уже существует eventsession.
Для серийного мероприятия: уже есть eventsession на указанную дату и время.

Пример запроса:

```
POST /events/5866/sessions
```

Headers

```
x-auth-token: HelloWorld
```

Parameters

```
name:new event session name
```

```
access:1
```

```
tags[0]:a tag
```

```
lectorIds[0]:8
```

Пример ответа:

```
{  
  "eventSessionId": 6884,  
  "link": "http://events.webinar.ru/lsd2/5858/session/6884" }
```

PUT /eventsessions/{eventSessionId} — обновить данные мероприятия (EventSession)

Запросом обновляются только данные EventSession.

Данные связанных сущностей(теги, файлы, лекторы) не обновляются.

Обновлять можно только сессии со статусом ACTIVE и START.

Сессию со статусом STOP обновлять нельзя.

При успешном обновлении сущности в ответ на запрос возвращается код ответа 204 и пустое тело ответа.

Необязательные параметры:

1. **name** - Название мероприятия.
2. **description** - Описание мероприятия.
3. **access** - Уровень доступа к мероприятию.
4. **lang** - язык мероприятия. определяет язык интерфейса, язык проведения мероприятия. может быть одним из значений: RU, EN, UK, TR.
5. **startsAt** - дата/время начала мероприятия. Параметр передается в виде массива startsAt, состоящего из двух элементов: date и time. date - массив из трех элементов: year, month, day. time - массив из двух элементов: hour и minute.
6. **additionalFields** - массив дополнительных полей, которые пользователь должен будет внести при регистрации на мероприятие. каждое поле задается массивом из элементов: label - название поля, type - тип поля(поле со свободным ответом(text) либо же выбор из предустановленных вариантов(radio)), values - массив предустановленных вариантов ответа, placeholder - значение, подставляемое в ответ по умолчанию.
7. **image** - ID файла в файловой системе, используемое в качестве фонового изображения для мероприятия.

Коды ответов:

1. **204** - сессия успешно обновлена.
2. **401** - в запросе не передан заголовок **x-auth-token**.
3. **403** - значение **x-auth-token** не верно.
4. **404** - сессия не найдена в организации.

Пример запроса:

```
PUT /eventsessions/6884
Headers
x-auth-token: HelloWorld
Parameters
description:just a description
```

GET /eventsessions/{eventSessionId} — получить данные о мероприятии (EventSession).

Коды ответов:

1. **200** - Когда сессия успешно найдена.
2. **401** - Когда в запросе не передан заголовок **x-auth-token**.
3. **403** - Когда значение **x-auth-token** не верно.
4. **404** - Когда сессия не найдена в организации.

Пример запроса:

```
GET /eventsessions/4304
```

Headers

```
x-auth-token: Hello World
```

Пример ответа:

```
{
  "id": 4304,
  "name": "Тест приглашения",
  "status": "ACTIVE",
  "access": 3,
  "lang": "RU",
  "password": "123123",
  "startsAt": "2015-12-07T12:40:00+0300",
  "utcStartsAt": 1449481200,
  "createUserId": 274,
  "timezoneld": 1,
  "additionalFields": [],
  "organizationId": 1,
  "lectors": [
    {
      "name": "Юрий",
      "secondName": "Барсуков",
      "tags": [],
      "announceFiles": [],
      "files": []
    }
  ]
}
```

DELETE /eventsessions/{eventSessionId} - Удаление сессии мероприятия. После удаления также становится недоступной интерактивная запись мероприятия. При успешном удалении сессии в ответ на запрос ничего не возвращается, только 204 статус.

Обязательные параметры:

1. **eventSessionId** - ID сессии мероприятия.

Коды ответов:

1. **204** - сессия успешно удалена.
2. **401** - в запросе не передан заголовок **x-auth-token**.
3. **403** - значение **x-auth-token** не верно.
4. **404** - сессия не найдена в организации.

Пример запроса:

```
DELETE /eventsessions/6784
```

```
Headers
```

```
x-auth-token: HelloWorld
```

PUT /eventsessions/{eventSessionId}/start — запуск (старт) мероприятия.

Стартовать можно только сессию со статусом ACTIVE.

Сессия должна принадлежать организации.

В ответ на запрос возвращается код ответа 204 и пустое тело ответа.

Коды ответов:

1. **204** - Когда сессия успешно запущена.
2. **401** - Когда в запросе не передан заголовок **x-auth-token**.
3. **403** - Когда значение **x-auth-token** не верно.
4. **404** - Когда сессия не найдена в организации или не имеет статус **ACTIVE**

Пример запроса:

```
PUT /eventsessions/6827/start
```

```
Headers
```

```
x-auth-token: HelloWorld
```

PUT /eventsessions/{eventSessionId}/stop — завершить (остановить) мероприятие. Остановить можно только сессию со статусом **START**. Сессия должна принадлежать организации. В ответ на запрос возвращается код ответа 204 и пустое тело ответа.

Коды ответов:

1. **204** - Когда сессия успешно остановлена.
2. **401** - Когда в запросе не передан заголовок **x-auth-token**.
3. **403** - Когда значение **x-auth-token** не верно.
4. **404** - Когда сессия не найдена в организации или не имеет статус **START**

Пример запроса:

```
PUT /eventsessions/6827/stop
```

```
Headers
```

```
x-auth-token: HelloWorld
```

Пользовательское API: Регистрация на мероприятие

Данный модуль позволяет зарегистрировать участника на мероприятие целиком, либо же на одну конкретную сессию мероприятия.

Основной сущностью, которой оперируют методы данного модуля, является сущность **Participation**, которая представляет собой регистрацию конкретного пользователя на мероприятие(**EventSession**) или же серию мероприятий(**Event**).

Поля данной сущности:

1. **id** - ID данной сущности в БД.
2. **userId** - ID пользователя в системе, который зарегистрирован на мероприятие.
3. **eventId** - ID серии мероприятий(**Event**), на которую зарегистрирован пользователь.
4. **eventSessionId** - ID сессии мероприятия(**EventSession**), на которую зарегистрирован пользователь. Если пользователь зарегистрирован на всю серию целиком - это поле будет отсутствовать в выдаче.
5. **isAccepted** - Флаг, показывающий, подтвердил ли ведущий участие пользователя в мероприятии. Для мероприятий без ручной модерации зарегистрированных - всегда true.
6. **isOnline** - Флаг, показывающий, что пользователь находится в данный момент online во время запущенного мероприятия (0 offline, 1 online).
7. **role** - Роль пользователя в данном мероприятии. Возможен один из трех вариантов: ADMIN, LECTURER, GUEST.
 - **GUEST** - Гость, обладает минимальным набором привилегий. Только чтение/получение данных и отправка сообщений в чат или вопросов. Не может начать мероприятие, изменять свойства мероприятия или список участников.
 - **LECTURER** - Лектор - Обладает расширенными правами, может изменять список участников, начинать/завершать мероприятие. Использовать дополнительный функционал: презентер, указка, рисовалка. Может отвечать на вопросы. Не может изменять свойства мероприятия.
 - **ADMIN** - Администратор - обладает максимально возможными правами. Может изменять свойства мероприятия.
8. **inviteUser** - Объект с данными пользователя, который пригласил участника на мероприятие. Если участник зарегистрировался самостоятельно - это поле отсутствует в выборке.
9. **additionalFields** - Массив ответов пользователя на дополнительные поля, которые необходимо указать при

регистрации/входе(по приглашению) на мероприятие. Массив содержит объекты, имеющие следующие поля:

- **key** - Уникальный ключ-идентификатор дополнительного поля. Через этот идентификатор связываются дополнительные поля мероприятия и ответы пользователей. Значение генерируется автоматически при создании дополнительного поля.
 - **label** - Текст вопроса.
 - **type** - Тип вопроса. Может быть либо *text*(ответ просто текстом), либо **radio**(выбор ответа из предложенных вариантов)
 - **value** - ответ пользователя.
10. **isSeen** - Флаг, отмечающий то, что пользователь видел приглашение на данное мероприятие.
11. **agreementStatus** - Поле, содержащее реакцию пользователя на приглашение на данное мероприятие. Возможные значения: DECLINED(отказ пользователя), WAITING(пользователь еще не дал ответ), AGREED(согласие пользователя)
12. **referrer** - URL, с которого пользователь перешел на страницу для регистрации или входа на мероприятие.

Запросы:

POST /events/{eventId}/register — зарегистрировать участника на серию вебинаров. При обработке запроса проверяется, является ли создатель мероприятия участником организации. В ответ на запрос возвращается массив из трех элементов:

1. **participationId** - ID сущности Participation, созданной при регистрации.
2. **link** - персональная ссылка, для участника, по которой он может войти на мероприятие.
3. **contactId** - ID контакта, в случае, если при регистрации участника был создан контакт.

Для регистрации на вебинар без повторений следует использовать запрос **POST /eventsessions/{eventSessionId}/register**

Обязательные параметры:

1. **email** - email участника. По этому полю проверяется, зарегистрирован ли участник уже на нашей платформе или нет и существует ли у создателя мероприятия контакт данного участника. Если участника или контакта нет - он(и) создается.

Необязательные параметры:

1. **role** - роль пользователя на мероприятии. возможные значения ADMIN, LECTURER, GUEST. по умолчанию GUEST.
2. **isAutoEnter** - флаг, определяющий, нужно ли сразу входить на страницу мероприятия после попадания на лендинг или же нужно подтверждение имени пользователя. по умолчанию false.
3. **sendEmail** - флаг, определяющий, нужно ли отправлять приглашенному пользователю письмо с персональной ссылкой. По умолчанию true.
4. **date** - Дата проведения сессии мероприятия, на которую хочет зарегистрироваться пользователь. Дата в формате: уууу-мм-dd. Любой другой формат вызовет 400 ошибку. Данный параметр имеет смысл только для повторяющихся мероприятий: Event.rule != null. Если данный параметр присутствует в запросе так же проверяется, что указанная дата подходит под правило повторения мероприятия. В противном случае возвращается 400 ошибка. Если дата верна, то пользователь регистрируется только на указанную сессию, а не на всё мероприятие целиком.
5. **nickname** - никнейм пользователя.
6. **name** - имя пользователя.
7. **secondName** - фамилия пользователя.
8. **pattnName** - отчество пользователя.
9. **phone** - телефон пользователя.
10. **description** - описание пользователя.
11. **organization** - организация, в которой работает пользователь.
12. **position** - должность пользователя.
13. **sex** - пол пользователя.
14. **avatar** - URL картинки для загрузки в ФС и установки в качестве аватара пользователя/контакта.
15. **additionalFields** - дополнительные поля, необходимые для регистрации на мероприятие, если они объявлены(Получить список доп полей можно по запросу на получение данных по мероприятию). additionalFields является массивом пар «ключ»⇒«значение», в которых «ключ» - поле key из объекта additionalField, а «значение» - введенный/выбранный пользователем ответ.

Если в запросе переданы поля с данными существующего контакта (идентификатор - **email**), то переданные данные будут проигнорированы. Использоваться будут существующие данные.

Коды ответов:

1. **201** - пользователь успешно зарегистрирован.
2. **401** - в запросе не передан заголовок **x-auth-token**.
3. **403** - значение **x-auth-token** не верно или когда создатель мероприятия не является членом организации.
4. **404** - мероприятие не найдено по eventId.
5. **409** - участник с таким **email** уже зарегистрирован.
6. **ERROR_BLANK** - Когда одно или несколько обязательных полей регистрации не указаны для участника.

Пример запроса:

```
POST /events/5270/register
```

Headers

```
x-auth-token: HelloWorld
```

FormData

```
email: newtestuser9@mailinator.com
```

```
isAutoEnter: false
```

```
name: New
```

```
pattrName: Test
```

```
secondName: User 3
```

```
additionalFields[46f065102882c8d846621d59e14a8369]: просто текст
```

```
additionalFields[ad0a0bf18cd3bccfb02947a39339be57]: ответ 1
```

```
additionalFields[9fdc3f131f7923e7bdd4ec60d465ae87]: 89296629667
```

Пример ответа:

```
{ "participationId": 24949,  
  "link":  
  "http://events.webinar.ru/6618/5270/c1bb3c979356f404b8f6bdfa98fc57fe,  
  "contactId": 1541 }
```

POST /eventsessions/{eventSessionId}/register — зарегистрировать участника на единоразовое мероприятие.

При обработке запроса проверяется, является ли создатель мероприятия участником организации.

В ответ на запрос возвращается массив из трех элементов:

1. **participationId** - ID сущности Participation, созданная при регистрации пользователя.
2. **link** - персональная ссылка, для участника, по которой он может войти на мероприятие.
3. **contactId** - ID контакта, в случае, если при регистрации участника был создан контакт.

Обязательные параметры:

1. **email** - email участника. По этому полю проверяется, зарегистрирован ли участник уже на нашей платформе или нет и существует ли у создателя мероприятия контакт данного участника. Если участника или контакта нет - он(и) создается.

Необязательные параметры:

1. **role** - роль пользователя на мероприятии. возможные значения ADMIN, LECTURER, GUEST. по умолчанию GUEST.
2. **isAutoEnter** - флаг, определяющий, нужно ли сразу заходить на страницу мероприятия после попадания на лендинг или же нужно подтверждение имени пользователя. по умолчанию false.
3. **sendEmail** - флаг, определяющий, нужно ли отправлять приглашенному пользователю письмо с персональной ссылкой. По умолчанию true.
4. **nickname** - никнейм пользователя.
5. **name** - имя пользователя.
6. **secondName** - фамилия пользователя.
7. **pattnName** - отчество пользователя.
8. **phone** - телефон пользователя.
9. **description** - описание пользователя.
10. **organization** - организация, в которой работает пользователь.
11. **position** - должность пользователя.
12. **sex** - пол пользователя.
13. **avatar** - URL картинки для загрузки в ФС и установки в качестве аватара пользователя/контакта.
14. **additionalFields** - дополнительные поля, необходимые для регистрации на мероприятие, если они объявлены (Получить список доп полей можно по запросу на получение данных по мероприятию). additionalFields является массивом пар «ключ» ⇒ «значение», в которых «ключ» - поле key из объекта additionalField, а «значение» - введенный/выбранный пользователем ответ.

Если в запросе переданы поля с данными существующего контакта (идентификатор - **email**), то переданные данные будут проигнорированы. Использоваться будут существующие данные.

Коды ответов:

1. **201** - пользователь успешно зарегистрирован.
2. **401** - в запросе не передан заголовок **x-auth-token**.
3. **403** - значение **x-auth-token** не верно или когда создатель мероприятия не является членом организации.
4. **404** - мероприятие не найдено по eventId.
5. **409** - участник с таким **email** уже зарегистрирован или когда участник не может быть зарегистрирован на сессию, поскольку вебинар предусматривает регистрацию **только** на весь **event**.
6. **ERROR_BLANK** - одно или несколько обязательных полей регистрации не указано для участника.

Пример запроса:

```
POST /eventsessions/6214/register
```

Headers

```
x-auth-token: HelloWorld
```

FormData

```
email: newtestuser9@mailinator.com
```

```
isAutoEnter: false
```

```
name: New
```

```
pattnName: Test
```

```
secondName: User 3
```

```
additionalFields[46f065102882c8d846621d59e14a8369]: просто текст
```

```
additionalFields[ad0a0bf18cd3bccfb02947a39339be57]: ответ 1
```

```
additionalFields[9fdc3f131f7923e7bdd4ec60d465ae87]: 89296629667
```

Пример ответа:

```
{ "participationId": 24949,  
  "link":  
  "http://events.webinar.ru/6618/5270/c1bb3c979356f404b8f6bdfa98fc57fe,  
  "contactId": 1542 }
```

PUT /events/{eventId}/moderate — модерировать самостоятельно зарегистрированных на серию мероприятий участников. С помощью этого запроса можно в любой момент дать/запретить самостоятельно зарегистрированному участнику доступ на мероприятие. В ответ на запрос возвращается количество регистраций, прошедших модерацию.

Обязательные параметры:

1. **participationIds** - массив ID сущностей Participation(регистрация участника на мероприятие), которые модерируются. Все Participation должны принадлежать мероприятию, ID которого передан.
2. **isAccepted** - флаг, обозначающий подтверждение/отказ пользователю заходить на мероприятие.

Коды ответов:

1. **200** - модерация прошла успешно.
2. **401** - в запросе не передан заголовок **x-auth-token**.
3. **403** - значение **x-auth-token** не верно.
4. **404** - не найдено мероприятие или ни одного Participation по ID.

Пример запроса:

```
PUT /events/5270/moderate
```

Headers:

```
x-auth-token: HelloWorld
```

Parameters:

```
isAccepted:true
```

```
participationIds[0]:24948
```

```
participationIds[1]:24947
```

Пример ответа:

```
{ "count": 2 }
```

GET /events/{eventId}/participations — получить список участников, зарегистрированных на всю серию мероприятий(Event). Участники, зарегистрированные на любую конкретную сессию данного Event-а не возвращаются. Только те, кто зарегистрирован на всю серию в целом.

Необязательные параметры:

1. **page** - Номер страницы для пагинации результата. По умолчанию 1.
2. **perPage** - Количество элементов на странице при пагинации результата. По умолчанию 50.

Коды ответов:

1. **200** - все проверки пройдены и возвращен список.
2. **401** - в запросе не передан заголовок **x-auth-token**.
3. **403** - значение **x-auth-token** не верно.
4. **404** - мероприятие не найдено.

Пример запроса:

```
GET /events/5696/participations?perPage=10 Headers
x-auth-token:HelloWorld
```

Пример ответа:

```
[ {  "id": 21862483,
     "name": "Участник",
     "secondName": "Вебинара",
     "email": "ychastnick@webinar.ru",
     "avatar": { "url": "image.jpg", "thumbnails": "{...}", }
     "eventId": 123456,
     "isAccepted": 1,
     "role": "GUEST",
     "isSeen": 1,
     "agreementStatus": "AGREED",
     "referrer": "https://mail.google.com/mail/u/1/",
     "userId": 1234567,
     "isOnline": "0",
     "registerStatus": "REGISTERED",
     "registerDate": "2018-03-02 15:35:48",
     "paymentStatus": "NOT_PAID",
     "url": "ab5dfdfg00a296da5c7de8cc3fgfjd474dbb" }, ]
```

GET /eventsessions/{eventSessionId}/participations — получить список участников, зарегистрированных на мероприятие (EventSession). Возвращается список пользователей, зарегистрированных либо на данную сессию, либо же на всю серию мероприятий в целом(что дает право посещать любую сессию данного мероприятия. В каждом элементе возвращаемого списка присутствует специальный параметр **visited** отвечающий за то, что пользователь посетил данное мероприятие.

Необязательные параметры:

1. **page** - Номер страницы для пагинации результата. По умолчанию 1.
2. **perPage** - Количество элементов на странице при пагинации результата. По умолчанию 50.

Коды ответов:

1. **200** - когда все проверки пройдены и возвращен список.
2. **401** - Когда в запросе не передан заголовок **x-auth-token**.
3. **403** - Когда значение **x-auth-token** не верно.
4. **404** - когда мероприятие не найдено.

Пример запроса:

```
GET /eventsessions/6726/participations?perPage=10
Headers
x-auth-token>HelloWorld
```

Пример ответа:

```
[ { "id": 21862483,
  "name": "Участник",

  "secondName": "Вебинара",
  "email": "ychastnick@webinar.ru",
  "avatar": { "url": "image.jpg", "thumbnails": "{...}", }
  "eventId": 123456,
  "isAccepted": 1,
  "role": "GUEST",
  "isSeen": 1,
  "agreementStatus": "AGREED",
  "referrer": "https://mail.google.com/mail/u/0/",
  "userId": 319,
  "isOnline": "0",
  "registerStatus": "REGISTERED_BY_URL",
  "registerDate": "2018-03-16 15:14:58",
  "paymentStatus": "NOT_PAID",
  "url": "3a7e8a6162e47bf2ee8cc3fgfjd44895b7", } ]
```

PUT /participations/update — обновить данных регистрации участников. На данный момент значимым для изменения параметром является только роль пользователя на мероприятии и только роль можно обновлять этим запросом. Обновлять можно только те регистрации, которые принадлежат мероприятиям организации. В ответ возвращается количество обновленных регистраций.

Обязательные параметры:

1. **participationIds** — массив из ID регистрации(**participationId**).
2. **role** — роль, которая будет установлена всем участникам по переданным **participationIds**,

Коды ответов:

1. **200** - регистрации успешно обновлены.
2. **400** - какой-то из обязательных параметров не передан или имеет неверный формат.
3. **401** - в запросе не передан заголовок **x-auth-token**.
4. **403** - значение **x-auth-token** не верно.
5. **404** - ни одной регистрации не найдено.

Пример запроса:

```
PUT /participations/update
```

```
Headers
```

```
x-auth-token: HelloWorld
```

```
Parameters
```

```
role:ADMIN
```

```
participationIds[0]:26769
```

```
participationIds[1]:26770
```

Пример ответа:

```
{ "count": 2 }
```

PUT /participations/kick — выгнать участников с мероприятия. При выполнении этого запроса запись о регистрации (Participation) фактически не удаляется из БД. Вместо этого флаг **Participation.isAccepted** устанавливается в значение **0**.

На странице мероприятия в списке участников перемещается в группу «Участники на модерации». Как мера снижения нагрузки - удалять можно только те регистрации(Participation), у которых значение isAccepted = 1. Удалять можно только те регистрации, которые принадлежат мероприятиям организации. В ответ возвращается количество обновленных регистраций.

Обязательные параметры:

1. **participationIds** - Массив из ID регистрации(**participationId**).

Коды ответов:

1. **200** - регистрации успешно обновлены.
2. **400** - какой-то из обязательных параметров не передан или имеет неверный формат.
3. **401** - в запросе не передан заголовок **x-auth-token**.
4. **403** - значение **x-auth-token** не верно.
5. **404** - ни одной регистрации, которую можно удалить, не найдено.

Пример запроса:

```
PUT /participations/kick
Headers
x-auth-token: HelloWorld
Parameters
participationIds[0]:26769
participationIds[1]:26770
```

Пример ответа:

```
{ "count": 2 }
```

POST /participations/delete — удалить участников из мероприятия.

Данный метод полностью удаляет регистрации из БД.

После этого участнику необходимо заново зарегистрироваться на мероприятие. Удалять можно только те регистрации, которые принадлежат мероприятиям организации. В ответ возвращается количество обновленных регистраций.

Обязательные параметры:

1. **participationIds** - Массив из ID регистрации(**participationId**).

Коды ответов:

1. **200** - регистрации успешно обновлены.
2. **400** - какой-то из обязательных параметров не передан или имеет неверный формат.
3. **401** - в запросе не передан заголовок **x-auth-token**.
4. **403** - значение **x-auth-token** не верно.
5. **404** - ни одной регистрации, которую можно удалить, не найдено.

Пример запроса:

```
POST /participations/delete
```

```
Headers
```

```
x-auth-token: HelloWorld
```

```
Parameters
```

```
participationIds[0]:26769
```

```
participationIds[1]:26770
```

Пример ответа:

```
{ "count": 2 }
```

POST /events/{eventId}/invite - массовая регистрация/приглашение участников на серию мероприятий (Event).

Метод позволяет одним запросом зарегистрировать/пригласить до **50** участников. В ответ на запрос возвращается массив сущностей, содержащих следующие элементы:

1. **participationId** - ID созданной сущности Participation.
2. **email** - Email, для которого создана данная сущность Participation.
3. **link** - Персональная ссылка для данного участника на лендинг мероприятия.
4. **contactId** - В случае, если в ходе регистрации данного участника был создан контакт для создателя мероприятия, возвращается ID этого контакта.

В ходе запроса проверяется корректность данных по каждому участнику и если данные по одному/нескольким участникам некорректных - они игнорируются. Также проверяется, был ли участник уже зарегистрирован на мероприятие и если участник уже зарегистрирован, попытка регистрации так же игнорируется. Таким образом в ответ на запрос возвращаются только те участники, которые были фактически зарегистрированы. Если в ходе запроса ни одного участника не было зарегистрировано - возвращается ошибка со статусом **409**.

Обязательные параметры:

1. **eventId** - ID мероприятия(Event), на которое происходит регистрация.
2. **isAutoEnter** - Должны ли зарегистрированные участники сразу попадать на страницу мероприятия, минуя страницу лендинга.
3. **sendEmail** - Отсылать ли приглашения зарегистрированным участникам.
4. **users** - Массив регистрационных данных пользователей. Поля, содержащиеся в каждом элементе этого массива идентичны полям для запроса **POST /events/{eventId}/register**. Элемент **email** так же является обязательным, остальные необязательны.

Коды ответов:

1. **201** - участники успешно зарегистрированы.
2. **400** - какой-то из обязательных параметров не передан или имеет неверный формат.
3. **401** - в запросе не передан заголовок **x-auth-token**.
4. **403** - значение **x-auth-token** не верно.
5. **404** - мероприятие найдено в организации.
6. **409** - все участники, корректные регистрационные данные которых переданы, уже зарегистрированы на мероприятие.

Пример запроса:

```
POST /events/5866/massregister
Headers
x-auth-token: HelloWorld
Parameters
isAutoEnter:true
sendEmail:false
users[0][email]:krondor12313@inbox.ru
users[0][name]:New
users[0][secondName]:User
users[0][role]:ADMIN
users[0][additionalFields][2c61404c6699d18f662798198aa55808]:просто
ОТВЕТ
users[1][email]:krondor12316@inbox.ru
users[1][name]:New 1
users[1][secondName]:User 2
users[1][additionalFields][2c61404c6699d18f662798198aa55808]:еще ответ
```

Пример ответа:

```
[ {
  "participationId": 27999,
  "email": "krondor12313@inbox.ru",
  "link":
  "http://events.webinar.ru/lsd2/5866/440b7125125fc6e95c5e31dcf6b2c3fb",
  "contactId": 17757  },
{
  "participationId": 28000,
  "email": "krondor12316@inbox.ru",
  "link":
  "http://events.webinar.ru/lsd2/5866/d145ca3accf4eaa58a01f004d2ee1822",
  "contactId": 17758  } ]
```

POST /eventsessions/{eventSessionId}/invite - массовая регистрация/приглашение участников на мероприятие (EventSession). Метод позволяет одним запросом зарегистрировать/пригласить до **50** участников. В ответ на запрос возвращается массив сущностей, содержащих следующие элементы:

1. **participationId** - ID созданной сущности Participation.
2. **email** - Email, для которого создана данная сущность Participation.
3. **link** - Персональная ссылка для данного участника на лендинг мероприятия.
4. **contactId** - В случае, если в ходе регистрации данного участника был создан контакт для создателя мероприятия, возвращается ID этого контакта.

В ходе запроса проверяется корректность данных по каждому участнику и если данные по одному/нескольким участникам некорректных - они игнорируются. Также проверяется, был ли участник уже зарегистрирован на мероприятие и если участник уже зарегистрирован, попытка регистрации так же игнорируется. Таким образом в ответ на запрос возвращаются только те участники, которые были фактически зарегистрированы.

Обязательные параметры:

1. **eventSessionId** - ID сессии мероприятия(EventSession), на которое происходит регистрация.
2. **isAutoEnter** - Должны ли зарегистрированные участники сразу попадать на страницу мероприятия, минуя страницу лендинга.
3. **sendEmail** - Отсылать ли приглашения зарегистрированным участникам.
4. **users** - Массив регистрационных данных пользователей. Поля, содержащиеся в каждом элементе этого массива идентичны полям для запроса **POST /eventsessions/{eventSessionId}/register**. Элемент **email** так же является обязательным, остальные необязательны.

Коды ответов:

1. **201** - участники успешно зарегистрированы.
2. **400** - какой-то из обязательных параметров не передан или имеет неверный формат.
3. **401** - в запросе не передан заголовок **x-auth-token**.
4. **403** - значение **x-auth-token** не верно.
5. **404** - мероприятие найдено в организации.
6. **409** - все участники, корректные регистрационные данные которых переданы, уже зарегистрированы на мероприятие.

Пример запроса:

```
POST /eventsessions/6949/massregister
Headers
x-auth-token: HelloWorld
Parameters
isAutoEnter:true
sendEmail:false
users[0][email]:krondor1233@inbox.ru
users[0][name]:New
users[0][secondName]:User
users[0][role]:ADMIN
users[0][additionalFields][2c61404c6699d18f662798198aa55808]:просто
ОТВЕТ
users[1][email]:krondor1236@inbox.ru
users[1][name]:New 1
users[1][secondName]:User 2
users[1][additionalFields][2c61404c6699d18f662798198aa55808]:еще ответ
```

Пример ответа:

```
[
{
"participationId": 27997,
"email": "krondor1233@inbox.ru",
"link":
"http://events.webinar.ru/lzd2/5866/6c17118c8ca4cdb824734e383ffaefa0"
},
{
"participationId": 27998,
"email": "krondor1236@inbox.ru",
"link":
"http://events.webinar.ru/lzd2/5866/128a52dc07993c9eb461daf97079d796"
}]
```

Пользовательское API: Статистика

Модуль предназначен для получения статистики по мероприятиям и по участникам мероприятий. При выборке мероприятий возвращаются данные по каждому мероприятию и сводные данные по участникам этих мероприятий. При выборке участников, возвращается список участников, которые участвовали во всех мероприятиях за указанный период и сводные данные по активности участника в каждом мероприятии. Под мероприятием везде понимается EventSession. В статистике учитываются только завершённые или мероприятия, которые в данный момент в эфире.

Как и все запросы к пользовательскому API, доступ к функционалу статистики возможен только при наличии в запросе специального заголовка `x-auth-token` равного `apiKey` организации.

Запросы:

GET /stats/events — получить статистику по мероприятиям. Возвращается список мероприятий в данными мероприятия и сводными данными по пользователям в этом мероприятии.

Необязательные параметры:

1. **from** - дата начала периода выборки. По умолчанию - сейчас.
2. **to** - дата начала периода выборки. По умолчанию - плюс 1 год от значения **from**.
3. **userId** - ID пользователя для получения статистики по мероприятиям только одного пользователя.
4. **eventId** - ID повторяющегося вебинара(Event), для получения статистики по всем сессиям выбранного вебинара.

Коды ответов:

1. **200** - мероприятия успешно найдены и возвращены.
2. **400** - не передан заголовок **x-auth-token** или передана дата в неверном формате.
3. **401** - передано некорректное значение **x-auth-token**.
4. **403** - пользователь, **userId** которого передан, не является членом организации.
5. **404** - вебинар, **eventId** которого передан, не найден в организации или же в случае, когда за указанный период выборки не найдено ни одного мероприятия.

Список полей возвращаемых сущностей:

1. **id** - ID мероприятия.
2. **name** - Название мероприятия.
3. **startsAt** - Время начала мероприятия.
4. **endsAt** - Время окончания мероприятия. Может иметь значение 0, что означает что время окончания отсутствует, следовательно мероприятие в данный момент в эфире.
5. **duration** - Длительность мероприятия в секундах. Если мероприятие в данный момент в эфире - 0.
6. **invitedCount** - Количество пользователей, которым было отправлено приглашение на данное мероприятие.
7. **invitedVisitedCount** - Количество приглашенных пользователей, которые посетили мероприятие.
8. **registeredCount** - Общее количество пользователей, которые зарегистрированы на данное мероприятие, включая тех, которым были отправлены приглашения.
9. **registeredVisitedCount** - Количество зарегистрированных пользователей, которые посетили мероприятие, включая тех, которым были отправлены приглашения.
10. **attendance** - массив, содержащий информацию о том, сколько из посетивших пользователей были новыми, постоянными или другими. Например:"attendance":

```
{  "NEW": 1,  
  "REGULAR": 1,  
  "OTHER": 0  }
```

11. **country** - массив, содержащий информацию по географическому распределению пользователей мероприятия. Массив имеет структуру: **название страны** → **список городов внутри данной страны** : **количество участников из данного города**.
12. **Также** внутри элемента **название страны** имеется специальное поле **countryCount** содержащее суммарное количество участников из данной страны. Например:

```
"country": {  
  "Russia": {  
    "countryCount": 2,  
    "Moscow": 2  
  }  
}
```

13. **platform** - массив, содержащий информацию по распределению пользователей по различным платформам. Доступные платформы: Web(просмотр через веб-браузер), Android(просмотр через мобильное приложение для Android), iOS(просмотр через мобильное приложение для iOS). Например:

```
"platform": {  
  "iOS": 0,  
  "Android": 0,  
  "Web": 2  
}
```

14. **referrer** - массив, содержащий список уникальных источников, с которых пользователи переходили на данное мероприятие. Для определения уникальности используются всевозможные части URL, так что одинаковые URLы, содержащие одинаковые адреса, но имеющие разные протоколы(например http и https) считаются разными. Например:

```
"referrer": {  
  "http://events.webinar.ru/lzd2/4143/stream": 1,  
  "https://events.webinar.ru/lzd2/4143/stream": 1  
}
```

Пример запроса:

```
GET
/stats/events?from=2016-01-01+16:20:00&to=2016-05-01+23:59:59&userId=
274 Headers x-auth-token:HelloWorld
```

Пример ответа:

```
[ {
  "id": "6214",
  "name": "Тест допполей.",
  "startsAt": "2016-03-03T16:53:28+0300",
  "endsAt": "2016-03-03T16:56:39+0300",
  "duration": 191,
  "attendance": {
    "NEW": 0,
    "REGULAR": 1,
    "OTHER": 1
  },
  "country": {
    "Russia": {
      "countryCount": 2,
      "Moscow": 2
    }
  },
  "platform": {
    "iOS": 0,
    "Android": 0,
    "Web": 2
  },
}]
```

GET /stats/users — получить информацию по пользователям, посетившим мероприятия за указанный период. Статистика выдается только по тем пользователям, которые посетили мероприятия, принадлежащие организации. Есть возможность сделать выборку только по одному конкретному участнику или по одному конкретному вебинару(Event).

Необязательные параметры:

1. **from** - дата начала периода выборки. По умолчанию - сейчас.
2. **to** - дата начала периода выборки. По умолчанию - плюс 1 год от значения **from**.
3. **userId** - ID пользователя для получения статистики по одному пользователю.
4. **eventId** - ID повторяющегося вебинара(Event), для получения статистики по всем пользователям, посетившим любую из сессий данного вебинара.

Коды ответов:

1. **200** - пользователи успешно найдены и возвращены.
2. **400** - не передан заголовок **x-auth-token** или передана дата в неверном формате.
3. **401** - передано некорректное значение **x-auth-token**.
4. **403** - пользователь, **userId** которого передан, не является членом организации.
5. **404** - вебинар, **eventId** которого передан, не найден в организации или же в случае, когда за указанный период выборки не найдено ни одного мероприятия, которое посетил хотя бы один пользователь.

Список полей возвращаемых сущностей:

1. **id** - ID пользователя в системе.
2. **email** - Email пользователя.
3. **name** - Имя пользователя.
4. **secondName** - Фамилия пользователя.
5. **patrnName** - Отчество пользователя.
6. **sex** - пол пользователя. Варианты: m - мужчина, f - женщина, o - не известно.
7. **phone** - Телефон пользователя.
8. **organization** - Компания, в которой работает пользователь.
9. **position** - Должность, занимаемая пользователем.
10. **eventSessions** - список мероприятий, которые посетил пользователь. Каждый элемент содержит в себе следующие данные:
 - id** - ID мероприятия.
 - name** - Название мероприятия.
 - startsAt** - Время начала мероприятия.

endsAt - Время окончания мероприятия. Может принимать значение 0, если мероприятие в эфире в данный момент.

duration - Длительность мероприятия в секундах. Может принимать значение 0, если мероприятие в эфире в данный момент.

questionCount - Общее количество вопросов в данном мероприятии.

userQuestionCount - Количество вопросов, заданное данным пользователем.

chatMessageCount - Общее количество сообщений в чате данного мероприятия.

userChatMessageCount - Количество сообщений в чате мероприятия, написанного данным пользователем.

additionalFieldValues - в данном поле находится список ответов данного пользователя в виде набора элементов, содержащих два параметра: **label** - название дополнительного поля и **value** - ответ, данный пользователем.

connections - массив подключений данного пользователя к данному мероприятию. Каждый элемент массива **connections** содержит следующие поля:

- joined** - время подключения пользователя к мероприятию.
- leaved** - время отключения пользователя от мероприятия.
- duration** - продолжительность присутствия пользователя на мероприятии(в рамках текущего connection) в секундах.
- country** - страна, из которой пользователь зашел на мероприятие.
- city** - город, из которого пользователь зашел на мероприятие.
- referrer** - источник, с которого пользователь перешел на страницу мероприятия.
- platform** - платформа, с которой пользователь просматривал мероприятие. может быть одной из: Web, iOS, Android.

Пример данных, содержащихся в **connections**:

```
{ "joined": "2016-03-03T16:57:59+0300",  
  "leaved": "2016-03-03T18:40:49+0300",  
  "duration": 6170,  
  "country": "Russia",  
  "city": "Moscow",  
  "referrer":  
  "http://events.webinar.ru/lsd2/5270/stream/6214",  
  "platform": "Web" }
```

Пример запроса:

```
GET
/stats/users?from=2016-01-01+16:20:00&to=2016-04-01+00:00:00&eventId=5270

Headers

x-auth-token:HelloWorld
```

Пример ответа:

```
[ { "id": "274",
  "email": "krondor@inbox.ru",
  "name": "Юрий",
  "secondName": "Барсуков",
  "patrName": "",
  "phone": "89296629667",
  "sex": "o",
  "eventSessions": [
    { "id": "6214",
      "name": "Тест допполей.",
      "startsAt": "2016-03-03T16:53:28+0300",
      "endsAt": "2016-03-03T16:56:39+0300",
      "duration": 191,
      "questionCount": 0,
      "chatMessageCount": 1,
      "connections": [
        { "joined": "2016-03-03T16:53:28+0300",
          "leaved": "2016-03-03T16:56:39+0300",
          "duration": 191,
          "country": "Russia",
          "city": "Moscow",
          "referrer": "http://events.webinar.ru/lsd2/5270/stream/6214",
          "platform": "Web"
        }
      ],
      "userChatMessageCount": 0, "userQuestionCount": 0,
      "additionalFieldValues": [] } ] }, { .... }
```

GET /stats/users/visits - Получение статистики по посещению пользователем мероприятий за указанный период. Учитываются только мероприятия, принадлежащие организации. Запрос возвращает массив из трех элементов:

1. **registered** - Список сессий(EventSession), на которые пользователь зарегистрировался. Если пользователь зарегистрировался на всю серию(Event) целиком, то информация по каждой отдельной сессии также будет содержаться в этом массиве.
2. **not_visited** - Список сессий, на которые пользователь зарегистрировался, но не посетил. Если пользователь зарегистрирован на всю серию мероприятий, но присутствовал только на одной, система посчитает, что на все остальные сессии он не пришел.
3. **visited** - Список сессий, на которые пользователь зарегистрировался и посетил.

Каждый элемент списков содержит следующие поля:

1. **id** - ID сессии мероприятия.
2. **name** - Название сессии мероприятия.
3. **additionalFieldValues** - Если для регистрации требовалось ввести дополнительную информацию, то в этом элементе содержится массив вопросов/ответов пользователя в виде пар **label/value**, где **label** - вопрос, а **value** - ответ пользователя.
4. **referrer** - Источник, с которого пользователь перешел на страницу регистрации.

Элемент списка **visited** содержит следующие элементы:

1. **userQuestionCount** - Количество вопросов, заданные выбранным пользователем на данной сессии.
2. **userChatMessageCount** - Количество сообщений, написанных в общем чате данной сессии выбранным пользователем.
3. **connections** - массив подключений данного пользователя к данному мероприятию. Каждый элемент массива **connections** содержит следующие поля:
 - joined** - время подключения пользователя к мероприятию.
 - leaved** - время отключения пользователя от мероприятия.
 - duration** - продолжительность присутствия пользователя на мероприятии(в рамках текущего connection) в секундах.
 - country** - страна, из которой пользователь зашел на мероприятие.
 - city** - город, из которого пользователь зашел на мероприятие.
 - referrer** - источник, с которого пользователь перешел на страницу мероприятия.

platform - платформа, с которой пользователь просматривал мероприятие. может быть одной из: Web, iOS, Android.

Пример данных, содержащихся в **connections**:

```
"joined": "2016-03-03T16:57:59+0300",  
{  
  "leaved": "2016-03-03T18:40:49+0300",  
  "duration": 6170,  
  "country": "Russia",  
  "city": "Moscow",  
  "referrer":  
  "http://events.webinar.ru/lsd2/5270/stream/6214",  
  "platform": "Web" }
```

Необязательные параметры:

1. **from** - дата начала периода выборки. По умолчанию - сейчас.
2. **to** - дата начала периода выборки. По умолчанию - плюс 1 год от значения **from**.
3. **email** - Email пользователя, статистику которого необходимо получить.
4. **contactId** - ID контакта, статистику которого необходимо получить.

В запросе обязан присутствовать хотя бы один из двух параметров: **email**, **contactId**. Если присутствуют оба, предпочтение отдается **contactId**.

Коды ответов:

1. **200** - Когда пользователи успешно найдены и возвращены.
2. **400** - Когда не передан заголовок **x-auth-token** или не переданы какие-либо необходимые параметры.
3. **401** - Когда передано некорректное значение **x-auth-token**.

Пример запроса:

```
GET /stats/users/visits?from=2016-05-22&to=2016-05-24&contactId=4810
Headers x-auth-token:HelloWorld
```

Пример ответа:

```
{ "registered":
[
{ "id": "11480",
"name": "Тест посещений 4 серия.",
referrer": "http://some.page.com/marketing_page_webinar",
"additionalFieldValues":
[ {
"id": "11482",
"name": "Тест посещений 5. серия",
"additionalFieldValues": [
{ "label": "раз",
"value": "три" },
{ "label": "два",
"value": "четыре" }
] } ],
"not_visited": [
{
"id": "11482",
"name": "Тест посещений 5. серия",
"additionalFieldValues": [
{
"label": "раз",
"value": "три" },
] }
],
"visited": [
```

```
{
  "id": "11480",
  "name": "Тест посещений 4 серия.",
  "referrer": "http://some.page.com/marketing_page_webinar",
  "additionalFieldValues": [
    {
      "label": "Кто ты?",
      "value": "зомби"
    },
    {
      "label": "Чего ты хочешь?",
      "value": "мозги"
    }
  ],
  "userQuestionCount": 0,
  "userChatMessageCount": 1,
  "connections": [
    {
      "joined": "2016-05-23T13:01:23+0300",
      "leaved": "2016-05-23T13:01:46+0300",
      "duration": 23,
      "country": "Russia",
      "city": "Moscow",
      "referrer": "http://events.webinar.ru/lsd2/7175/stream/11480",
      "platform": "Web" } ] ] }
```

Пользовательское API: Записи

Модуль создан для предоставления доступа к функционалу записей мероприятий.

Запросы:

GET /records — получить список записей мероприятий.

Пример запроса:

```
GET /records
```

```
Headers
```

```
x-auth-token: HelloWorld
```

Также существует возможность передать в URL один или несколько необязательных параметров:

1. **id** - ID конкретной записи
2. **period** - Период выборки, может быть один из следующих вариантов: **day** - день, **week** - неделя, **month** - месяц, **year** - год
3. **from** - Фильтр начальной даты, например **2015-03-01**
4. **to** - Фильтр конечной даты, например **2017-03-01**
5. **userId** - Фильтр записей по конкретному пользователю по его ID

Пример ответа:

```
[ {
  "id": 812117,
  "name": "Новое мероприятие сегодня",
  "link": "http://events.webinar.ru/1929/4290/record/5043", "isViewable": false,
  "hasPassword": false,
  "size": 0,
  "createAt": "2016-01-20 13:11:33"
}, {
  "id": 814668,
  "name": "слайды",
  "link": "http://events.webinar.ru/1929/4333/record/5097", "isViewable": false,
  "hasPassword": false,
  "size": 0,
  "createAt": "2016-01-22 15:08:51"
}]
```

POST /records/{recordID}/conversions — поставить запись на конвертацию с использованием recordId. Его значение можно получить запросом GET /records.

Необязательные параметры:

- **quality** - Качество сконвертированного видео. Может быть одним из: **720** и **1080**. По умолчанию **720**.
- **view** - Какую вкладку мероприятия отображать в видеофайле. Может быть одним из: **questions** - отображать вкладку вопросов, **chat** - отображать вкладку чата. По умолчанию **chat**.

Коды ответов:

1. **201** - запись поставлена на конвертацию. Ответ содержит conversionID — идентификатор видеофайла, который будет доступен по окончании конвертации.
2. **401** - в запросе не передан заголовок **x-auth-token**.
3. **403** - значение **x-auth-token** не верно.
4. **403 ("message": "Simultaneous conversions quantity exceeded")** — превышен лимит одновременных конвертаций для этой организации.
5. **404 ("message": "Event session not found")** — не найдена запись с таким recordID.

Пример запроса:

```
POST /records/10/conversions
Headers
x-auth-token: HelloWorld
```

Пример ответа:

```
{ "id": 34 }
```

POST eventsessions/eventsessionID/records/conversions — поставить запись на конвертацию с использованием уникального идентификатора мероприятия — eventsessionID.

Необязательные параметры:

- **quality** - Качество сконвертированного видео. Может быть: **720**, **1080**. По умолчанию **720**.
- **view** - Какую вкладку мероприятия отображать в видеофайле. Может быть одним из: **questions** - отображать вкладку вопросов, **chat** - отображать вкладку чата. По умолчанию **chat**.

Коды ответов:

6. **201** — запись поставлена на конвертацию. Ответ содержит `conversionID` — идентификатор видеофайла, который будет доступен по окончании конвертации.
7. **401** — в запросе не передан заголовок `x-auth-token`.
8. **403** — значение `x-auth-token` не верно.
9. **403 ("message": "Simultaneous conversions quantity exceeded")** — превышен лимит одновременных конвертаций для этой организации.
10. **404 ("message": "Event session not found")** — не найдена запись с таким `eventsessionID`.

Пример запроса:

```
POST /eventsessions/23581/records/conversions?view=questions
```

Headers

```
x-auth-token: HelloWorld
```

Пример ответа:

```
{ "id": 1350371 }
```

GET /records/conversions/{conversionId} — проверить статус конвертации.

Параметры URL:

- `conversionId` - ID конвертируемого видеофайла

Ответ содержит статус конвертации видеофайла.

Пример запроса:

```
GET /records/conversions/34
```

Headers

```
x-auth-token: HelloWorld
```

Пример ответа:

```
{ "state": "waiting" }
```

Статусы конвертации:

- **waiting** - Ожидает начала конвертации
- **processing** - Конвертация в процессе
- **failed** - Конвертация завершена с ошибкой
- **completed** - Конвертация успешно завершена.

POST /records/{recordId}/share - отправить участникам мероприятия ссылку на запись по e-mail. Значение RecordId можно получить запросом GET /records.

Пример запроса:

```
POST /records/10/share
Headers
x-auth-token: HelloWorld
```

PUT /records/{recordId} — изменить параметры записи мероприятия. Значение RecordId можно получить запросом GET /records.

Необязательные параметры:

1. **isViewable** - флаг, который определяет возможность просмотра записи мероприятия по общей ссылке. Может быть **true** или **false**.
2. **password** - пароль на просмотр записи. Для отключения пароля, передайте пустую строку.

Коды ответов:

1. **204** - параметры записи успешно изменены.
2. **401** - в запросе не передан заголовок **x-auth-token**.
3. **403** - значение **x-auth-token** не верно.
4. **404** - запись с таким идентификатором не найдена в организации.

Пример запроса:

```
PUT /records/10
Headers:
x-auth-token: HelloWorld
Parameters:
isViewable=true
password=123
```

PUT /eventsessions/{eventSessionId}/records — изменить параметры записи мероприятия по идентификатору этого события.

Необязательные параметры:

1. **isViewable** — флаг, который определяет возможность просмотра записи мероприятия по общей ссылке. Может быть **true** или **false**.
2. **password** — пароль на просмотр записи. Если этот параметр не передать, он будет проигнорирован. Чтобы сбросить текущий пароль, нужно передать значение “null”.

Коды ответов:

5. **204** - параметры записи успешно изменены.
6. **401** - в запросе не передан заголовок **x-auth-token**.
7. **403** - значение **x-auth-token** не верно.
8. **404** - мероприятие с таким идентификатором не найдено в организации.

Пример запроса:

```
PUT /eventsessions/123456/records
```

Headers:

```
x-auth-token: HelloWorld
```

Parameters:

```
isViewable=true
```

```
password=123
```

DELETE /records/{recordId} — удалить запись мероприятия.
Дополнительных параметров нет

Пример запроса:

```
DELETE /records/10
```

Headers

```
x-auth-token: HelloWorld
```

Пользовательское API: Чат и вопросы

Чат-бот организации

Данный модуль представляет собой сущность, от имени которой можно отправлять сообщения в чат, задавать и отвечать на вопросы. Свойства сущности:

- name — обязательный параметр, имя бота
- secondName — необязательный параметр, фамилия
- position — должность
- organization — организация, которой бот принадлежит (у организации может быть только 1 бот)
- avatar — аватар бота (id сущности FileSystem)

Запрос: **GET /organization/bot** получить информацию о боте организации

Коды ответа:

- 200 — Когда бот получен.
- 401 — Когда x-auth-token неверный.
- 404 — Когда бот не найден.

Пример запроса:

```
GET /organization/bot
Headers
x-auth-token: HelloWorld
```

Пример ответа:

```
{
  "id": 26473,
  "name": "Terminator",
  "secondName": "T-2000",

  "organization": {
    "id": 1,
    "name": "Organization #1",  }}
```

POST /organization/bot - создать бота организации.

Параметры:

- name — обязательный параметр, имя бота
- secondName — необязательный параметр, фамилия
- position — должность
- avatar — аватар бота, id сущности FileSystem

Статус коды:

- 201 - Когда бот создан.
- 400 - Когда заданы неверные параметры (не задан обязательный параметр — имя).
- 401 - Когда x-auth-token неверный.
- 409 - Когда бот уже существует.

Пример запроса:

```
POST /organization/bot
```

```
Headers:
```

```
x-auth-token: HelloWorld
```

```
Parameters:
```

```
name: Бот
```

```
secondName: Поддержки
```

```
position: Помощник
```

```
avatar: 1347761
```

PUT /organization/bot - изменить параметры для бота организации.

Параметры аналогичны **POST /organization/bot**

Если поля отсутствуют в запросе, они не меняются. Чтобы удалить поле, нужно отправить его с пустым значением

Статус коды:

- 201- Успешное обновление бота.
- 400 - Неверные параметры запроса (поле name пустое).
- 401 - Когда x-auth-token неверный.
- 404 - Когда бот не найден

Работа с чатом

Чат представляет собой основную комнату, сообщения которой видны всем участникам мероприятия, и приватные комнаты, которые создаются между двумя пользователями.

GET /eventsessions/{eventSessionId}/chat - Получает все сообщения из главного чата по eventSessionId

Параметры:

- **isModerated:** true/false (получить только отмодерированные/неотмодерированные сообщения (по умолчанию получает отмодерированные сообщения));
- **limit:** количество сообщений (по умолчанию последние 100, отсортированные по времени по убыванию)
- **authorId:** id пользователя (сообщения конкретного пользователя в данной EventSession)
- **privateChat:** true/false получает только все комнаты пользователя (если указан пользователь, в противном случае параметр игнорируется) в данной EventSession (по умолчанию false).

Статус коды ответа:

- 200 - ОК.
- 401- Неверный токен.
- 403- Недостаточно прав.
- 404- Когда EventSession не существует или удалена.

Пример запроса:

```
GET /eventsessions/11313/chat
```

```
Headers
```

```
x-auth-token: HelloWorld
```

Пример ответа:

```
{
  "id": 17475,
  "authorName": "Василий Петров",
  "text": "blablbablabla",
  "isModerated": true,
  "sentByAdmin": false,
  "authorId": 9689
},
{
  "id": 17495,
  "authorName": "name secondName",
  "text": "afsfasdfasdfasfd",
  "isModerated": true,
  "sentByAdmin": true,
  "avatarUrl":
  "https://events-storage/api-storage/files/webinar/2016/04/27/IkiLZcChkjPYKZ6
  K3leF7V3uKbuXoU8k06BGNx74PhJuKZ8x.png", "thumbnails":
  "authorId": 3209
}, ]
```

Описание полей:

- «id»: идентификатор сообщения;
- «authorName»: имя автора сообщения;
- «text»: текст сообщения;
- «isModerated»: флаг модерации сообщения (зависит от настроек мероприятия, по умолчанию модерация отключена);
- «sentByAdmin»: отправлено ли сообщение админом мероприятия (либо ботом);
- «avatarUrl»: url аватара отправителя;
- «thumbnails»: аватар отправителя в различных разрешениях;
- «authorId»: id отправителя.

POST eventsessions/{eventSessionId}/chat - Отправляет сообщение в чат (по умолчанию отправляет в общий чат от имени бота организации)

Поля:

- **fromUserId** — id отправителя (по умолчанию бот организации, при его отсутствии создатель организации)
- **toUserId** — id получателя (для отправки в личные сообщения) (по умолчанию общий чат)
- **text** –текст сообщения

Пример запроса:

```
POST /eventsessions/11313/chat
```

Headers

```
x-auth-token: HelloWorld
```

Parameters

```
text: Привет
```

```
fromUserId: 8441
```

Пример ответа:

```
{  
  "id": 17492,  
  "authorName": "Михаил Дорош",  
  "text": "text",  
  "isModerated": true,  
  "sentByAdmin": true,  
  "avatarUrl":  
  "https://events-storage/api-storage/files/webinar/2016/04/27/lkiLZcChkjPYKZ6  
  K3leF7V3uKbuXoU8k06BGNx74PhJuKZ8x.png", "thumbnails": {},  
  "authorId": 8441 }  
}
```

Статус-коды:

- 201 - Сообщение отправлено.
- 400 - Не указан текст сообщения.
- 401 -Неверный токен.
- 403 - Недостаточно прав.

DELETE /eventSessions/{eventSessionId}/chat

Удаляет сообщения из чата (по умолчанию все сообщения данной EventSession, которые не относятся к личным)

Параметры:

- **messageId**: удаляет конкретное сообщение по его id
- **userId**: удаляет все сообщения пользователя по его id
- **onlyNotModerated**: удаляет только неотмодерированные сообщения в данной EventSession

Параметры работают независимо, т.е. если указать и messageId и userId, то будут удалены и сообщения пользователя и сообщение по его id.

Ответ:

- 204 — в случае успешного удаления сообщения
- 401 — неверный токен
- 403 — недостаточно прав
- 404 — не найден пользователь, чьи сообщения удалять, либо не найдено сообщение, которое необходимо удалить (либо сущности уже удалены)

Пример запроса:

```
DELETE /eventSessions/11313/chat
```

Headers

```
x-auth-token: HelloWorld
```

```
Ответ: 204 No content
```

PUT /eventSessions/{eventSessionId}/chat/messages/moderate

Модерирует сообщения в данной eventSession (по умолчанию все, если заданы параметры messageIds[] и userId, то модерирует по всем параметрам одновременно) Поля:

- **isModerated**: true/false (Разрешить/Запретить публикацию сообщений) (по умолчанию false)
- **messageIds[]**: массив, модерирует сообщения по их id
- **userId**: модерирует сообщения по конкретному юзеру

Статус коды ответа:

- 200 - ОК. В ответе массив из прошедших модерацию сообщений. Поля аналогичны /eventSessions/{eventSessionId}/chat.
- 401 - Неверный токен.
- 403 - Недостаточно прав.
- 404- Не найден пользователь или сообщения по id.

Работа с вопросами

Данный модуль представляет собой цепочку вопросов и ответов, без возможности задавать приватные вопросы

GET /eventsessions/{eventSessionId}/questions — получает вопросы из данной EventSession (по умолчанию все)

Параметры:

- **questionId** : получает только ответы конкретного вопроса по его Id
- **userId** : получает только вопросы конкретного пользователя
- **onlyOnModeration** : получает только все вопросы, которые не прошли модерацию.

Параметры используются **раздельно!**

Если задать одновременно `userId` и `onlyOnModeration`, то вернет только вопросы, которые были отобраны по параметру `onlyOnModeration`.

Список полей ответа:

- **id**: id вопроса;
- **createAt**: дата создания вопроса;
- **text**: текст вопроса;
- **user**: данные пользователя;
- **answers**: массив вопросов;
- **isModerated**: промодерирован ли вопрос;
- **isAnswered**: есть ли ответ на вопрос.

Статус-коды:

- 200=«ОК»,
- 401=«Неверный токен»
- 403=«Недостаточно прав»
- 404=«Когда EventSession не существует или удалена, либо не найдены вопросы по id»

Пример запроса:

```
GET /eventsessions/11313/questions?userId=685
```

```
Headrs
```

```
x-auth-token: HelloWorld
```

Пример ответа:

```
{
  "id": 2363,
  "createAt": "2016-04-06T16:34:54+0300",
  "text": "qwe",
  "user": {
    "id": 685,
    "createAt": "2015-08-05T12:54:51+0300",
    "nickname": "Андрей Морозов",
    "photo": {
      "id": 878313,
      "createAt": "2016-04-12T16:23:11+0300",
      "url":
        "https://events-storage/api-storage/files/webinar/2016/04/12/gilHFoAcSFUIPV
        KarjXCR3VsVDhefmNos4nJ9KvZpX7u15Jg.jpg",
      "thumbnails": { } },
      "answers": [],
      "isModerated": true,
      "isAnswered": false }
}
```

Post /eventsessions/{eventSessionId}/questions — отправляет вопрос (либо ответ, если указан questionId) Параметры:

- questionId — отправляет ответ на вопрос по его Id
- userId — указать id пользователя, задавшего вопрос (либо давшего ответ), по умолчанию бот организации
- text — текст вопроса/ответа

Статус-коды:

- 201=«Сообщение (вопрос или ответ) отправлено»
- 400=«Не указан текст сообщения (вопроса или ответа)»
- 401=«Неверный токен»
- 403=«Недостаточно прав»

Пример запроса:

```
POST /eventsessions/11313/questions x-auth-token: HelloWorld
Content-Type: multipart/form-data; "text": text "userId": 26473
```

Пример ответа:

```
{ "id": 2394, "createAt": "2016-05-23T17:07:10+0300", "text": "text",
"user": { "id": 26473 }, "isModerated": true, "isAnswered": false }
```

DELETE /eventsessions/{eventSessionId}/questions удаляет вопросы из EventSession (по умолчанию все):

- userId — удаляет только вопросы конкретного пользователя
- onlyOnModeration — удаляет только все вопросы, которые не были отмодерированы (isModerated: false)
- questionId — удаляет вопрос по его id

Запрос:

```
DELETE /eventsessions/11313/questions?userId=685 x-auth-token:
HelloWorld
```

Ответ:

- 204 — в случае успешного удаления вопроса;
- 401 — неверный токен;
- 403 — недостаточно прав;
- 404 — не найдены вопросы по заданным параметрам, либо они уже удалены.

PUT /eventsessions/{eventSessionId}/questions/moderate

отмодерировать вопросы в зависимости от параметра `isModerated` (по умолчанию модерирует все вопросы, запрещая их публикацию)

Поля:

- **isModerated** : true/false (Разрешить/Запретить публикацию вопросов) (по умолчанию false);
- **questionIds[]** : массив, модерирует вопросы по их id;
- **userId** : модерирует вопросы по конкретному юзеру.

Статус-коды:

- 200=«ОК»;
- 401=«Неверный токен»;
- 403=«Недостаточно прав»;
- 404=«Не найден пользователь или сообщения по id».

Ответ аналогичен запросу **GET /eventsessions/{eventSessionId}/questions**.

Пользовательское API: Файлы

Файлы в файловом менеджере

Функциональность позволяет работать с файлами и папками файлового менеджера, загружать, удалять, получать информацию о них.

Типы файлов:

- «video» = «Видеофайл»;
- «presentation» = «Презентация»;
- «slide» = «Слайд»;
- «test» = «Файл теста»;
- «testResult» = «Файл результатов теста»;
- «record» = «Файл записи»;
- «convertedRecord» = «Сконвертированный файл записи».

Список полей:

- «id»: идентификатор файла/папки;
- «parent»: папка, в которой находится файл/папка. Если значение NULL, то файл/папка находятся в корневой папке;
- «isDeleted»: флаг, удален ли файл/папка;
- «createAt»: дата создания/загрузки файла/папки;
- «name»: имя файла/папки;
- «type»: определяет, является ли сущность файлом (file) или папкой (folder);
- «user»: пользователь, загрузивший/создавший файл/папку;
- «organization»: если файл является общим, то файл имеет данное поле, указывающее принадлежность к организации;
- «path»: относительный путь файла;
- «url»: полный путь файла;
- «thumbnails»: если файл является картинкой, то в данном поле хранятся миниатюры картинки в следующих размерах:
- [«640×1920», «1920×1080», «1333×1000», «224×199», «50×50», «34×34», «148×112»];
- «size»: размер файла в байтах;
- «format»: расширение файла;
- «isHidden»: является ли файл скрытым;
- «isSystem»: является ли файл системным;
- «mimeType»: MIME тип файла;
- «typeFile»: если файл относится к одному из известных форматов, указывается его тип (video, slide и т.д.). Если нет, то передается file;
- «uri»: uri файла.

В зависимости от типа файла, отдаются следующие поля:

Для video:

- duration — длительность видео;
- description — описание.

Для видео с youtube или vimeo:

- src — ссылка на видео;
- author — имя автора видео;
- authorUrl - канал автора на youtube или vimeo;
- videoid - id видео на vimeo;

Для Presentation:

- state (completed,failed,loaded,waiting) - состояние конвертации;
- slides - набор слайдов (доступны после конвертации), представляют собой сущность filesystem:slide. Содержат дополнительные поля;
- number - порядок слайда и rotate - угол поворота;
- progress - прогресс конвертации;
- isConvert - флаг, сконвертирована ли презентация.

Для Test:

- «duration»: - длительность теста/голосования;
- «minAnswers»: минимальное количество ответов для того чтобы пройти тест;
- «minPoints»: минимальное количество баллов;
- «assessType»: определяет, по какому критерию судить прохождение теста: minAnswers или minPoints;
- «contextType»: тест/голосование;
- «questions»: вопросы и ответы на них, либо голосование с вариантами ответа.

Для Record:

- duration — длительность видео;
- description — описание;
- cuts - поле, которое показывает вырезанные отрезки видео в записи, которые определяются по полям start – end;
- password - пароль на запись;
- isViewable - открыта ли запись для общего доступа;
- eventSession - указывает на сессию, которой принадлежит запись;
- state - готова ли запись Если запись поставлена на конвертацию/сконвертирована, для неё существует сущность (или несколько)

Для ConvertedRecord:

- convertedAt - дата конвертации
- state - состояние конвертации
- progress - состояние конвертации в %

Запросы:

GET /fileSystem/file/{id} - получает файл по его ID

Дополнительные параметры:

- name — имя файла.

Пример запроса:

```
GET /fileSystem/file/913440
```

```
x-auth-token: HelloWorld
```

Пример ответа:

```
{ "id": 912149,
  "isDeleted": false,
  "createAt": "2016-05-24T11:31:42+0300",
  "name": "calendar (6).ics",
  "type": "file",
  "user": {
    "id": 8441,
    "nickname": "Михаил Дорош",
    "type": "USER",
    "name": "Михаил",
    "secondName": "Дорош",
    "photo": {
      "url":
        "https://events-storage/api-storage/files/webinar/2016/05/20/4HVdza7db2zB0rmmrMvmLM7aez2CJvE1t4vBPnjrfgiGJh7m.jpg",
      "uri":
        "//events-storage/api-storage/files/webinar/2016/05/20/4HVdza7db2zB0rmmrMvmLM7aez2CJvE1t4vBPnjrfgiGJh7m.jpg",
      "thumbnailUri": "",
      "thumbnails": { } } },
  "organization": {
    "id": 1534,
    "isDeleted": false,
```

```
"createAt": "2016-04-05T17:13:54+0300",
"name": "BLABLABLA",
"branding": [
{
"id": 191,
"isDeleted": false,
"createAt": "2016-04-22T14:43:49+0300"
}
],
},
"path":
"./files/webinar/2016/05/24/5PetviPJFiXjETzg7Z0MFCMEFAsUJCA1fInBunYN4HBYZHB5.ics",
"url":
"https://events-storage/api-storage/files/webinar/2016/05/24/5PetviPJFiXjETzg7Z0MFCMEFAsUJCA1fInBunYN4HBYZHB5.ics",
"thumbnails": [],
"uploadUrl":
"https://events-storage/api-storage/upload?token=YhtYXE3n9LHgixMFccmkvPH64feH8kMj&extension={extension}",
"size": 1188,
"format": "ics",
"isHidden": false,
"isSystem": false,
"mimeType": "text/plain",
"tags": [],
"typeFile": "file",
"uri":
"//events-storage/api-storage/files/webinar/2016/05/24/5PetviPJFiXjETzg7Z0MFCMEFAsUJCA1fInBunYN4HBYZHB5.ics" }
```

POST /fileSystem/file — создает запись о файле в базе данных. Ограничение по объему файла составляет 100 МБ. Для видеофайлов с расширением .mp4 до 500 МБ.

Параметры:

- name — имя файла.

POST /fileSystem/folder - создает папку

Параметры аналогичны для данных методов, за исключением name (только для папки) и url (только для файла):

- parent — id родительской папки;
- url — если требуется загрузить файл по его url;
- isShared — если true, то общий файл для организации;
- user — id пользователя, под чьим именем загрузить файл;
- name - название папки.

Если не указаны никакие параметры, то файл загружается в корневую папку организации от имени создателя.

Если Вы создаете папку, то Вы должны учесть, что в папке организации нельзя создать собственную папку (и наоборот, в собственной папке нельзя создать общую папку).

В описанных случаях сервер вернет 400 (неверные параметры).

Пример запроса: (для POST /fileSystem/file)

```
POST /fileSystem/file
HTTP/1.1
Headers:
x-auth-token: d0c88dc2f9ecacba239487efcc2ae074
Body:
Content-Type: multipart/form-data;
user:8354
```

Коды ответа:

```
201 - Запись о файле создана;
403 - неверный токен, или нет доступа к пользователю или папке, закончилось место на диске;
404 - пользователь или папка не найдены.
```

Ответ:

```
"id": 924389,  
"token": "Xb3pEsC*****js7GJhGa9"
```

После того, как получен токен, его необходимо использовать в запросе

POST <https://events-storage.webinar.ru/api-storage/slowupload>

Данный метод загружает файл в файловое хранилище и записывает метаданные файла в базу.

Параметры:

- token - токен, полученный через метод, создавший запись о файле;
- file - файл, который необходимо загрузить.

Пример запроса:

```
Content-Disposition: form-data;  
name="token": 11b9aV*****vBbkhfMz  
Content-Disposition: form-data;  
name="file";  
filename="467_p (1).docx"  
Content-Type:  
application/vnd.openxmlformats-officedocument.wordprocessingml.document
```

Пример ответа:

```
{ "createAt": "2016-05-31T15:09:32+0300",  
  "updateAt": "2016-05-31T15:09:33+0300",  
  "file": {  
    "id": 1337829,  
    "isDeleted": false,  
    "createAt": "2016-05-31T15:09:32+0300",  
    "updateAt": "2016-05-31T15:09:33+0300",  
    "name": "467_p (1).docx",  
    "type": "file",  
    "user": {  
      "id": 314873,  
      "nickname": "Михаил Дорош",
```

```
"type": "USER",
"name": "Михаил",
"secondName": "Дорош"
},
"path": "./files/webinar...docx",
"url": "https://events-storage/...docx",
"thumbnails": {},
"uploadUrl":
"https://events-storage.webinar.ru/api-storage/app.php/upload?token=11b9aV
062D0xLTEfSoBr3hAavBbkhfMz&extension={extension}",
"size": 828643,
"format": "docx",
"isHidden": false,
"isSystem": false,
"mimeType":
"application/vnd.openxmlformats-officedocument.wordprocessingml.document",
"tags": [],
"typeFile": "presentation",
"uri": "//events-storage/....docx",
"thumbnailUri": "",
"slides": [],
"state": "waiting",
"progress": 0,
"isConvert": true,
"rotate": 0 },
"id": 2772711,
"path": "./files....docx",
"url": "https://events-storage/....docx",
"token": {
"createAt": "2016-05-31T15:09:32+0300",
"id": 1433733,
"word": "11b9aV062D0xLTEfSoBr3hAavBbkhfMz"
```

```
},
"metadata": {"userId":314873,"type":"UPLOAD_TYPE_USER"},
"callback":
"https://events.webinar.ru/api/app.php/fileSystem/callbackFileUpload",
"thumbnailsList": { }, "state": "AVAILABLE" }
```

Пример запроса (для POST /fileSystem/folder):

```
POST /userapi/fileSystem/folder
HTTP/1.1 Host: userapi.webinar.ru/v3
x-auth-token: key
Content-Type: multipart/form-data; name: FirstFolder isShared:true
```

Ответ:

```
{
"id": 914014,
"name": "FirstFolder",
"type": "folder",
"thumbnails": [],
"uri": ""
}
```

GET /fileSystem/files - возвращает список файлов по определенным параметрам:

- user - id пользователя, чьи файлы нужно показать;
- parent - папка, в которую требуется загрузить файл;
- format - расширение файла (pdf,png,jpg и т.д.);
- isShared - общие файлы (true).

Если не указан ни один параметр, либо только параметр format, то ответ будет состоять из общих файлов организации (если указан format, то отобранных по этому полю).

Если указать несколько параметров, то отбор будет вестись по нескольким, например если указать user 12345 и isShared true - то ответом будут все общие файлы, загруженные пользователем 12345

Пример запроса:

GET /fileSystem/files?format=jpg&user=8441&isShared=true

Пример ответа:

```
{
  "id": 887182,
  "name": "desert.jpg",
  "type": "file",
  "url": "https://events-storage/...jpg",
  "thumbnailUrl": "https://events-storage/...jpg",
  "thumbnails": {  },
  "format": "jpg",
  "uri": "//events-storage/...jpg"},
{
  "id": 887353,
  "name": "tiger.jpg",
  "type": "file",
  "url": "...jpg",
  "thumbnailUrl": "...jpg",
  "thumbnails": [],
  "format": "jpg",
  "uri": "...jpg"  },
```

DELETE /fileSystem/file/{id} - удаляет файл по id

Параметры не принимает.

Ответ:

- 204 - файл удален
- 403 - нет доступа к файлу
- 404 - файл не найден.

Управление файлами вебинара

К мероприятиям можно прикреплять файлы.

Концепция следующая: можно прикрепить уже существующий файл/либо загрузить файл и сразу прикрепить его к /Event/EventSession (При использовании методов, связанных с Event, совершаются аналогичные операции с EventSession, относящимися к event).

Методы:

POST/events/{eventId}/files

POST /eventsessions/{eventSessionId}/files

Параметры:

- fileId - id файла, который необходимо прикрепить;

При отсутствии fileId для данной сущности создается прикрепление файла и возвращает токен для загрузки файла в storage.

GET/events/{eventId}/files/{fileId}

GET /eventsessions/{eventSessionId}/files/{fileId}

Получает данные о прикрепленном файле к Event(EventSession) по id файла

GET /events/{eventId}/files /eventsessions/{eventSessionId}/files

Получает список файлов, прикрепленных к Event(EventSession)

DELETE /events/{eventId}/files

DELETE /events/{eventId}/files/{fileId}

DELETE /eventsessions/{eventSessionId}/files

DELETE /eventsessions/{eventSessionId}/files/{fileId}

Удаляет файл по id, либо все файлы для Event/EventSession

PUT /eventsessions/{eventSessionId}/files/{fileReferenceId}

Делает файл доступным/недоступным для всех посетителей данной EventSession

Параметры:

- isShared = true/false разрешает/запрещает просмотр файла.

Пользовательское API: Тесты

Сущности модуля

TestFile - Сущность теста, содержит в себе все настройки. Данная сущность так же является наследником сущности файла и это позволяет, после создания, применять к ней операции, доступные другим файлам: перемещать между папками, прикреплять к мероприятию.

Список полей сущности **TestFile**:

1. **id** - ID записи в БД. Уникальный идентификатор сущности.
2. **userId** - ID пользователя, которому данный тест принадлежит.
3. **name** - Имя теста
4. **assesType** - Тип оценки результатов. По количеству баллов(**pointsSum**) или по количеству правильных ответов(**answerCount**). Не обязательное поле. По умолчанию: **answerCount**.
5. **minPoints** - Минимальное количество баллов, необходимое для успешного прохождения теста. Не обязательное поле. По умолчанию 0.
6. **minAnswers** - Минимальное количество правильных ответов, необходимое для успешного прохождения теста. Не обязательное поле. По умолчанию 0.
7. **duration** - Заявленная длительность теста в секундах. Используется для тестирования в режиме реального времени. Не обязательное поле. По умолчанию 0(без ограничения времени).
8. **questions** - список вопросов теста.

TestQuestion - Сущность вопрос теста. Имеет следующие поля:

1. **id** - ID записи в БД. Уникальный идентификатор сущности.
2. **questionData** - Текст вопроса. Обязательное поле.
3. **questionImage** - В случае, если в вопросе используется изображение, в этом поле находится массив данных для изображения:
 - **id** - ID файла, используемого для изображения. Файлы, используемые в тесте должны быть обязательно загружены в нашу файловую систему.
 - **url** - ссылка на скачивание изображения.
 - **thumbnails** - массив со ссылками на миниатюры изображения.
4. **allowCustomAnswer** - флаг, обозначающий возможность свободного ответа на данный вопрос. По умолчанию - false.
5. **maxPointsAnswers** - максимальное возможное количество ответов с баллами, доступное в данном вопросе. Имеет смысл

- только в случае, если возможно выбрать одновременно несколько ответов. По умолчанию 1.
6. **order** - порядковый номер вопроса в списке.
 7. **answers** - список ответов на данный вопрос.

TestQuestionAnswer - сущность ответа на вопрос теста. Имеет следующие поля:

1. **id** - ID записи в БД. Уникальный идентификатор.
2. **answerData** - текст ответа на вопрос.
3. **answerImage** - Если ответом на вопрос является изображение, то в этом поле находится массив с данными изображения:
 - **id** - ID файла, используемого для изображения. Файлы, используемые в тесте должны быть обязательно загружены в нашу файловую систему.
 - **url** - ссылка на скачивание изображения.
 - **thumbnails** - массив со ссылками на миниатюры изображения.
4. **isCorrect** - флаг, отвечающий за правильность ответа. Не обязательное поле. По умолчанию - false,
5. **points** - количество баллов, получаемое за выбор данного ответа. Не обязательное поле. По умолчанию - 0
6. **order** - порядковый номер ответа в списке.

TestSession - Данная сущность представляет собой одиночный запуск теста. Всякий раз, когда запускается тест, создается сущность **TestSession**. В рамках этой сущности каждый пользователь отвечает на вопросы теста и в рамках этой сущности, в случае, если тест был проведен в рамках мероприятия(**EventSession**), лектор может поделиться результатами, как общими, так и индивидуальными. Если тест был проведен в рамках мероприятия(**EventSession**), так же добавляется связь между **TestSession** и **EventSession**.

Список полей сущности:

1. **id** - ID записи в БД. Уникальный идентификатор.
2. **eventSessionId** - ID **eventSession**, в случае, если тестирование проводилось в рамках мероприятия.
3. **startTime** - время старта **testSession**. Формат: YYYY-MM-DD HH:MM;SS.
4. **endTime** - время завершения **testSession**. Формат: YYYY-MM-DD HH:MM;SS.
5. **isComplexResultsShared** - флаг, отвечающий за то, разрешено ли участникам тестирования получить сводные результаты теста. Используется при тестировании в рамках мероприятия.
6. **isIndividualResultsShared** - флаг, отвечающий за то, разрешено ли участникам тестирования получить индивидуальные результаты теста. Используется при тестировании в рамках мероприятия.

UserTestPassing - сущность, представляющая собой прохождение теста одним пользователем в рамках TestSession. Именно в рамках этой сущности будут агрегироваться ответы и все результаты тестирования конкретного пользователя. Данная сущность создается автоматически при отправке ответов пользователя.

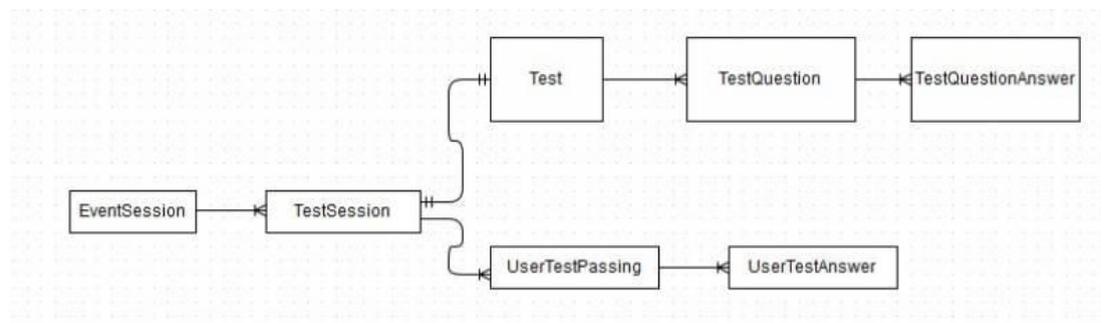
Список полей сущности:

1. **id** - ID записи в БД. Уникальный идентификатор.
2. **questionCount** - Количество вопросов теста.
3. **minPointsToPass** - минимальное количество баллов, необходимое пользователю для успешного прохождения тестирования.
4. **minCorrectAnswersToPass** - минимальное количество правильных ответов, необходимое пользователю для успешного прохождения тестирования.
5. **startTime** - дата/время начала прохождения пользователем тестирования.
6. **endTime** - дата/время окончания прохождения пользователем тестирования.
7. **isTestPassed** - успешно ли пользователь прошел тестирование?
8. **answerCount** - количество ответов, данных пользователем.
9. **pointsSum** - сумма баллов, набранная пользователем.
10. **correctAnswerCount** - количество вопросов, на которые пользователь ответил правильно. Правильный ответ на вопрос подразумевает, что пользователь выбрал все ответы, которые отмечены как правильные в вопросе.
11. **testSessionId** - ID TestSession, в рамках которой пользователь проходил тестирование.
12. **userId** - ID пользователя, который проходил тестирование.
13. **eventSessionId** - ID мероприятия(EventSession), в рамках которого было проведено тестирование. Может быть null, если тестирование проходило без привязки к мероприятию.

UserTestQuestionAnswer - сущность, представляющая собой один ответ пользователя на один вопрос теста. Имеет следующие поля:

1. **id** - ID записи в БД. Уникальный идентификатор.
2. **questionData** - текст вопроса, на который отвечает пользователь.
3. **questionImage** - Если вопрос был с изображением, то в этом поле находится массив с данными изображения:
 - **id** - ID файла, используемого для изображения. Файлы, используемые в тесте должны быть обязательно загружены в нашу файловую систему.
 - **url** - ссылка на скачивание изображения.
 - **thumbnails** - массив со ссылками на миниатюры изображения.
4. **answerData** - текст ответа пользователя.
5. **answerImage** - Если в качестве ответа было выбрано изображение, то в этом поле находится массив с данными изображения:
 - **id** - ID файла, используемого для изображения. Файлы, используемые в тесте должны быть обязательно загружены в нашу файловую систему.
 - **url** - ссылка на скачивание изображения.
 - **thumbnails** - массив со ссылками на миниатюры изображения.
6. **isCorrect** - флаг, указывающий, правильный ли это ответ или нет?
7. **answerPoints** - количество баллов, полученное пользователем за выбор ответа.
8. **isCustomAnswerSubmitted** - флаг, показывающий, что ответ, данный пользователем является свободным, а не выбранным из предложенных.
9. **isCustomAnswerAssessed** - флаг, показывающий, что свободный ответ был оценен ведущим и ответу проставлены значения **isCorrect** и/или **answerPoints**.
10. **userId** - ID пользователя, который дал ответ.
11. **userTestPassingId** - ID сущности **UserTestPassing**, в рамках которой пользователем был дан ответ.

Наглядно представить себе взаимосвязь всех сущностей можно на следующем рисунке:



Запросы:

POST /tests - Создать тест. С помощью данного запроса создается тест целиком. Возможно закрепить тест за определенным участником организации, либо за создателем организации(по умолчанию). Для выбранного пользователя делается проверка, доступен ли ему модуль тестирования по тарифу. В ответ на запрос возвращается JSON-массив с одним элементом: id созданного теста.

Параметры запроса:

1. **testData** - массив с данными теста. Обязательный параметр. Поля массива соответствуют полям сущностей(тест, вопрос, ответ), описанными выше. Если для вопроса или ответа предполагается использование изображений, то возможно, как использование изображений, уже загруженных в файловую систему(используя поле `imageId`), либо передать URL изображения, которое будет загружено в файловую систему и прикреплено(используя поле `imageUrl`)
2. **userId** - ID участника организации, за которым закрепляется тест.

Коды ответов:

1. **201** - тест успешно создан.
2. **400** - данные теста не корректны(см. правила валидации теста)
3. **401** - отсутствует заголовок **x-auth-token** или его значение не верно.
4. **403** - если передан параметр **userId** и пользователь не является членом организации или не имеет доступа к модулю Тестирование по тарифу.
5. **404** - если передан параметр **userId** и по его значению в организации не найдено пользователя, или файл с `imageId`, не найден

Пример запроса:

```
POST /tests
Headers:
x-auth-token: HelloWorld
Parameters:
userId:274
testData[name]:api test with image
testData[assessType]:answerCount
testData[questions][0][questionData]:first question with images of api test
testData[questions][0][answers][0][imageId]:941697
testData[questions][0][answers][0][isCorrect]:1
testData[questions][0][answers][1][imageUrl]:http://nourriture.ru/upload/iblock/8fd/frukt-salat-oreh.jpeg
testData[questions][0][answers][1][isCorrect]:0
testData[questions][1][questionData]:first question of api test
testData[questions][1][allowCustomAnswer]:true
testData[questions][1][answers][0][answerData]:first answer second question of api test
testData[questions][1][answers][0][isCorrect]:0
testData[questions][1][answers][1][answerData]:second answer second question of api test
testData[questions][1][answers][1][isCorrect]:0
```

Пример ответа:

```
{ "id": 941728 }
```

PUT /tests/{testId} - Обновить информацию по тесту. В запросе необходимо передавать только те данные, которые подлежат изменению. Формат данных почти такой же, как и в запросе на создание теста, за небольшими исключениями. В массиве данных по вопросу или ответу обязательно должен присутствовать элемент ID вопроса/ответа. В случае, если необходимо удалить какой-то вопрос/ответ, то вместе с ID сущности передается параметр `_destroy=true`. Если необходимо удалить только изображение, а не весь вопрос/ответ целиком, то нужно в запрос добавить параметр, `[image][_destroy]:true`. Сменить пользователя, которому принадлежит тест нельзя. Это возможно через запрос на изменение свойств файла. В ответ на запрос возвращается статус 204 No Content.\

Параметры запроса:\

1. **testData** - массив с данными теста, которые подлежат изменению.

Коды ответов:

1. **204** - тест успешно обновлен.
2. **400** - данные теста не корректны(см. правила валидации теста)
3. **401** - отсутствует заголовок **x-auth-token** или его значение не верно.
4. **404** - тест не найден, либо в случае, если в тесте изменяется изображение(передан `imageId`) и файл по этому `imageId` не найден.
5. **409** - тест не может быть обновлен. Данная ситуация возможная только в случае, когда данный тест имеет хотя бы одну незавершенную в данный момент `TestSession`.

Пример запроса:

```
PUT /tests/941728
```

Headers:

```
x-auth-token: HelloWorld
```

```
Parameters: testData[questions][0][questionId]:868
```

```
testData[questions][0][answers][0][answerId]:2023
```

```
testData[questions][0][answers][0][isCorrect]:false
```

```
testData[questions][0][answers][0][answerData]:answer
```

```
test instead of image
```

```
testData[questions][0][answers][0][image][_destroy]:true
```

```
testData[questions][0][answers][1][answerId]:2024
```

```
testData[questions][0][answers][1][_destroy]:true
```

```
testData[questions][0][answers][2][answerData]:first answer added by update
```

```
testData[questions][0][answers][2][isCorrect]:true
```

```
testData[questions][0][answers][3][answerData]:second answer added by  
update
```

```
testData[questions][0][answers][3][isCorrect]:false
```

```
testData[name]:updated api test with no image anymore
```

GET /tests/{testId} - получение данных теста. В ответ на запрос возвращается сущность **TestFile**. Среди полей сущности так же присутствует массив **testSessions**, содержащий данные по запускам данного теста, без данных по результатам. Если в поле **testSession** отсутствует поле **endTime**, значит данная **testSession** в данный момент еще не завершена.

Параметры запроса:

1. **testSessionId** - не обязательный параметр. позволяет ограничить выборку **testSession** только одним элементом.

Коды ответов:

1. **200** - тест найден.
2. **401** - в запросе отсутствует заголовок **x-auth-token** или он имеет не верное значение.
3. **403** - заголовок **x-auth-token** имеет не верное значение или ни у одного сотрудника организации нет доступа к модулю API.
4. **404** - тест не найден по **testId**.

Пример запроса:

```
GET /tests/941728
```

Headers:

```
x-auth-token:HelloWorld
```

Пример ответа:

```
{ "id": 941728,
  "userId": 274,
  "name": "updated api test with no image anymore",
  "type": "test",
  "assessType": "answerCount",
  "minPoints": 0,
  "minAnswers": 0,
  "duration": 0,
  "questions":
  [ { "id": 868,
      "allowCustomAnswer": false,
      "order": 1,
      "questionData": "first question with images of api test",
      "answers":
      [ { "id": 2023,
          "isCorrect": false,
          "points": 0,
          "order": 1,
          "answerData": "answer test instead of image"
        },
        { ...},
      ]
    }
  ]
  "testSessions":
  [ { "id": "622",
      "startTime": "2016-06-29 12:54:59",
      "endTime": "2016-06-29 12:57:56",
      "complexResultsShared": false,
      "individualResultsShared": false
    } ] }
```

DELETE /tests/{testId} - удалить тест.

Коды ответа:

1. **204** - тест успешно удален.
2. **401** - в запросе отсутствует заголовок x-auth-token.
3. **403** - заголовок x-auth-token имеет неверное значение.
4. **404** - тест не найден.

Пример запроса:

```
DELETE /tests/941728
```

Headers:

```
x-auth-token: HelloWorld
```

POST /tests/{testId}/start - запустить тест. По этому запросу создается TestSession, в рамках которой пользователи будут отвечать на вопросы. Есть возможность запустить тест в рамках мероприятия, указав в параметрах eventId. Можно так же задать время начала TestSession, указав в параметрах startTime.

Параметры запроса:

1. **eventId** - ID мероприятия(EventSession), в рамках которого будет проходить тест. Не обязательный параметр. Тесты можно запускать и без привязки к мероприятию. Если тест проводится в рамках мероприятия так же проверяется, что в данном мероприятии уже не запущен тест, т.к. в рамках мероприятия в одно время может быть запущен только один тест.
2. **startTime** - дата/время начала теста. Не обязательный параметр. по умолчанию - сейчас.

Коды ответов:

1. **201** - тест успешно запущен. создана TestSession.
2. **401** - в запросе отсутствует заголовок x-auth-token.
3. **403** - значение x-auth-token не верно или ни у одного пользователя организации нет модуля API в тарифе.
4. **404** - тест не найден в организации или EventSession не найдена(если передан параметр) или тест не прикреплен к мероприятию(если тест запускается в рамках мероприятия)
5. **409** - Мероприятие не запущено или в мероприятии уже запущен тест(если тестирование проводится в рамках мероприятия),

Пример запроса:

```
POST /tests/941728/start
Headers:
x-auth-token:HelloWorld
Parameters:
startTime: 2016-06-29 15:00:00
```

Пример ответа:

```
{ "testSessionId": 623 }
```

PUT /testsessions/{testSessionId}/stop - остановить тестирование. закрыть TestSession. После этого пользователи больше не могут отправлять свои ответы. Ведущие могут получить список свободных ответов и оценивать их.

Коды ответов:

1. **204** - TestSession успешно завершена.
2. **401** - в запросе отсутствует заголовок x-auth-token
3. **403** - значение x-auth-token не верное или ни у одного из членов организации нет доступа к модулю API в тарифе или тест, к которому относится TestSession не принадлежит организации или у пользователя, который владеет тестом нет доступа к модулю Тестирование в тарифе.
4. **404** - TestSession не найдена или уже завершена.

Пример запроса:

```
PUT /testsessions/622/stop
Headers
x-auth-token:HelloWorld
```

POST /testsessions/{testSessionId}/answers - отправить ответы пользователя на тест. В этот момент создается сущность UserTestPassing и набор сущностей UserTestQuestionAnswer. Все ответы пользователя записываются сразу - сущность UserTestPassing сразу же закрывается и данный пользователь больше не может отправлять свои ответы - сразу же происходит оценка прохождения пользователем теста. В качестве параметров запроса отправляется массив с данными пользователя(userData. элемент userData[email] является обязательным. другие контактные данные - nickname, name, secondName, pattnName, phone, description, position, organization, avatar(URL изображения или ID файла в файловой системе - не обязательные), массив с ответами пользователя(answers. в этом массиве два поля: questionId - обязательное и одно из двух - answerId(если выбран один из предложенных ответов) или customAnswer(если пользователь дал свободный ответ.)). Можно так же отправить данные о том, когда пользователь начал и завершил проходить тест(параметры startTime и endTime). В ответ на запрос возвращается JSON-объект вида {«status»: «ok», «userId»: <ID пользователя в нашей системе>}. Так же для владельца теста создается контакт пользователя, который отвечает на вопросы теста.

Параметры запроса:

1. **userData** - массив с данными пользователя, который отвечал на вопрос. поле email обязательно. другие контактные данные - nickname, name, secondName, pattnName, phone, description, position, organization, avatar(URL изображения или ID файла в файловой системе - не обязательны.
2. **answers** - массив с ответами пользователя. каждый элемент этого массива является массивом из двух элементов: questionId - ID вопроса(TestQuestion), на который пользователь отвечает и либо answerIds - массив ID ответов(TestQuestionAnswer) - либо customAnswer, если участником дан свободный ответ.
3. **startTime** - время, когда пользователь начал проходить тест. Не обязательное поле. По умолчанию - сейчас.
4. **endTime** - время, когда пользователь закончил проходить тест. Не обязательное поле. По умолчанию - сейчас.

Коды ответов:

1. **201** - Ответы успешно записаны.
2. **400** - Данные имеют не верный формат.
3. **401** - В запросе отсутствует заголовок x-auth-token.
4. **403** - Заголовок x-auth-token имеет не верное значение или ни у одного из членов организации нет доступа к модулю API по тарифу или тест, к которому относится данная TestSession не принадлежит организации или пользователь не имеет прав проходить тест в рамках мероприятия(если тест проводится в рамках мероприятия)

5. **404** - TestSession не найдена или уже завершена, вопрос(TestQuestion) или ответ(TestQuestionAnswer) не найден.
6. **409** - Пользователь уже отвечал на вопросы теста в рамках данной TestSession.

Пример запроса:

```
POST /testsessions/623/answers Headers: x-auth-token:HelloWorld
Parameters: userData[email]:testUser@mailinator.com
answers[0][questionId]:868 answers[0][answerIds][0]:2027
answers[1][questionId]:869 answers[1][customAnswer]:this is custom answer
startTime:2016-06-29 15:01:00 endTime:2016-06-29 15:05:00
```

Пример ответа:

```
{ "status": "ok", "userId": 6677 }
```

GET /tests/{testId}/customanswers - Получить свободные ответы на тест. Все параметры запроса являются не обязательными. В ответ возвращается надо сущностей UserTestQuestionAnswer. Возвращаются даже те ответы, которые уже были оценены.

Параметры запроса:

1. **eventId** - ID мероприятия(Event) в рамках которого проводилось тестирование.
2. **eventSessionId** - ID сессии мероприятия(EventSession) в рамках которой проводилось тестирование.
3. **testSessionId** - ID TestSession в рамках которой пользователи отвечали на вопросы теста.
4. **notAssessedOnly** - Флаг, при наличии которого возвращаются только те ответы, которые еще не были оценены.
5. **startTime** - Начало периода, в котором пользователи отвечали на вопросы теста. По умолчанию: -1 месяц от текущей даты/времени.
6. **endTime** - Конец периода, в котором пользователи отвечали на вопросы теста. По умолчанию: сейчас.

Коды ответов:

1. **200** - Все ок, даже если ни одного ответа не найдено.
2. **400** - Не верный формат параметров.
3. **401** - В запросе отсутствует заголовок x-auth-token.
4. **403** - Значение x-auth-token не верное или пользователь, которому принадлежит тест не имеет доступа к модулю Тестирование по тарифу.
5. **404** - Тест не найден или не принадлежит ни одному из участников организации.

Пример запроса:

```
GET
/userapi/tests/941728/customanswers?testSessionId=623&startTime=2016-06-01+00%3A00%3A002016-06-01+00:00:00&endTime=2016-06-30+23%3A59%3A59&notAssessedOnly=true
```

Headers:

```
x-auth-token:HelloWorld
```

Пример ответа:

```
[ {
  "id": "1504",
  "userId": "6677",
  "testSessionId": "623",
  "answerData": "this is custom answer",
  "answerPoints": "0",
  "isCorrect": "0",
  "isCustomAnswerAssessed": "0",
  "questionData": "first question of api test"
}]
```

PUT /testsessions/{testSessionId}/assessanswers - оценить свободные ответы пользователей. После того, как свободные ответы оценены, для всех пользователей, чьи ответы были оценены происходит пересчет результатов.

Параметры запроса::

1. **data** - массив с данными об оценках ответов пользователей. каждый элемент массива состоит из двух элементов: `userAnswerId` - ID сущности `UserTestQuestionAnswer` - и `isCorrect`(если тест оценивается по количеству правильных ответов) или `answerPoints`(если тест оценивается по количеству баллов).

Коды ответов:

1. **204** - ответы успешно оценены.
2. **400** - неверный формат данных об ответах.
3. **401** - В запросе отсутствует заголовок `x-auth-token`.
4. **403** - Значение `x-auth-token` не верное или ни один из пользователей организации не имеет доступ к модулю API по тарифу или пользователь, которому принадлежит тест не имеет доступа к модулю Тестирование по тарифу.
5. **404** - `TestSession` не найдена.

Пример запроса:

```
PUT /testsessions/623/assessanswers
```

```
Headers: x-auth-token:HelloWorld
```

```
Parameters:
```

```
data[0][userAnswerId]:1504
```

```
data[0][isCorrect]:true
```

GET /tests/list - получить информацию по тестам, принадлежащим организации, вместе с информацией о времени запуска теста.

Необязательные параметры:

1. **eventId** - фильтр по eventId. При фильтрации по eventId возвращается список тестов, которые были запущены в рамках Event с указанным eventId.
2. **eventSessionId** - фильтр по eventSessionId. При фильтрации по eventSessionId возвращается список тестов, которые были запущены в рамках EventSession с указанным eventSessionId.
3. **testSessionId** - фильтр по testSessionId. При фильтрации по testSessionId возвращается список тестов, которые были запущены в рамках TestSession с указанным testSessionId.
4. **userId** - фильтр по userId. При фильтрации по userId, возвращается список тестов, которые были созданы участником организации с ID равным userId.
5. **startTime** - начало периода выборки. Формат - YYYY-MM-DD HH:MM:SS. По умолчанию: «минус 6 месяцев от текущего времени.».
6. **endTime** - окончание периода выборки. Формат - YYYY-MM-DD HH:MM:SS. По умолчанию: «сейчас».

Коды ответов:

1. **200** - Все хорошо.
2. **400** - Какой-то из параметров имеет неверный формат.
3. **401** - В запросе отсутствует заголовок **x-auth-token**.
4. **403** - Значение заголовка **x-auth-token** не корректно.

Пример запроса:

```
GET
/tests/list?userId=274&eventId=6567&startTime=2016-04-00+00:00:00&endTime=2016-08-01+00:00:00

Headers:
x-auth-token: HelloWorld
```

Пример ответа:

```
[ {
  "id": "8275",
  "userId": "274",
  "name": "asdfasdfasd",
  "createAt": "2015-08-27 01:29:52",
  "lastStart": "2016-04-21 18:11:00",
  "startsCount": 3,
  "testSessions": [
    {
      "id": "559",
      "eventSessionId": "7643",
      "startTime": "2016-04-21 18:11:00"    },
    {
      "id": "558",
      "eventSessionId": "7643",
      "startTime": "2016-04-21 18:00:27",
      "endTime": "2016-04-21 18:01:13"
    },
    {
      "id": "557",
      "eventSessionId": "7643",
      "startTime": "2016-04-21 17:59:36",
      "endTime": "2016-04-21 17:59:48"    }  ] ] }
```

GET /tests/{testId}/results - получить результаты теста. Данный запрос возвращает массив элементов TestSession с несколькими дополнительными полями:

1. **usersCount** - количество человек, начавших проходить тест,
2. **endedCount** - количество человек, приславших ответы на вопросы теста,
3. **passedCount** - количество человек, успешно прошедших тест в рамках данной TestSession,
4. **users** - массив с данными по каждому участнику, проходящему тест в рамках данной TestSession.

Необязательные параметры:

1. **eventId** - фильтр по eventId. При фильтрации по eventId возвращается список TestSession, которые были запущены в рамках Event с указанным eventId.
2. **eventSessionId** - фильтр по eventSessionId. При фильтрации по eventSessionId возвращается список TestSession, которые были запущены в рамках EventSession с указанным eventSessionId.
3. **testSessionId** - фильтр по testSessionId. При фильтрации по testSessionId возвращается результаты только по указанной TestSession.
4. **startTime** - начало периода выборки. Формат - YYYY-MM-DD HH:MM:SS. По умолчанию: «минус 6 месяцев от текущего времени.».
5. **endTime** - окончание периода выборки. Формат - YYYY-MM-DD HH:MM:SS. По умолчанию: «сейчас».

Коды ответов:

1. **200** - Все хорошо.
2. **400** - Какой-то из параметров имеет не верный формат.
3. **401** - В запросе отсутствует заголовок **x-auth-token**.
4. **403** - Значение заголовка **x-auth-token** не корректно.

Пример запроса:

```
GET
/tests/941728/results?testSessionId=628&startTime=2016-04-01+00:00:00&e
ndTime=2016-08-01+00:00:00
Headers:
x-auth-token: HelloWorld
```

Пример ответа:

```
[ {
  "id": 628,
  "startTime": "2016-07-14 18:03:37",
  "endTime": "2016-07-14 18:03:44",
  "usersCount": 3,
  "endedCount": 3,
  "passedCount": 1,
  "users": [
    {
      "id": 75666,
      "points": 0,
      "correctlyAnsweredQuestions": 1,
      "isPassed": false,
      "nickname": "user1",
      "answers": [
        {
          "id": 1528,
          "questionData": "Вопрос 1",
          "answerData": "правильный ответ 1",
          "isCorrect": true,
          "answerPoints": 0,
          "isCustomAnswerSubmitted": false,
          "isCustomAnswerAssessed": false
        },
        { ..... }
      ]
    },
    ...
  ]
}
```

Правила валидации тестов:

1. У теста обязательно должно быть название.
2. У каждого теста обязательно должен быть минимум 1 вопрос.
3. У каждого вопроса обязательно должен быть либо текст, либо изображение.
4. Если для теста выбран метод оценки «по количеству баллов», то для каждого вопроса проверяется, что значение поля **maxPointsAnswers** не превышает количество ответов на вопрос(+1 в случае возможности свободного ответа)
5. У каждого вопроса должно быть минимум 2 ответа(1 в случае возможности свободного ответа)
6. У каждого ответа должен быть либо текст, либо изображение.
7. Если выбран метод оценки «по количеству правильных ответов», то у каждого вопроса должен быть минимум 1 правильный ответ(0 в случае возможности свободного ответа)
8. Если выбран метод оценки «по количеству баллов», то у каждого вопроса должен быть минимум один ответ, с количеством баллов, не равным нулю(0 в случае возможности свободного ответа)
9. Суммарное количество ответов во всех вопросах теста должно быть больше или равно параметру **minAnswers**

GET /users/{userId}/tests/stats - получить статистику пользователя по тестам. Подразумевается статистика пользователя, который отвечает на вопросы теста, а не статистика пользователя, который создает и запускает тесты.

Необязательные параметры:

1. **testId** - фильтр по testId. При фильтрации по testId возвращается статистика по выбранному тесту.
2. **eventId** - фильтр по eventId. При фильтрации по eventId возвращается статистика по тестам, запущенным в рамках Event.
3. **eventSessionId** - фильтр по eventSessionId. При фильтрации по eventSessionId возвращается статистика по тестам, запущенным в рамках EventSession.
4. **testSessionId** - фильтр по testSessionId. При фильтрации по testSessionId возвращается статистикаа вот только по указанной TestSession.
5. **startTime** - начало периода выборки. Формат - YYYY-MM-DD HH:MM:SS. По умолчанию: «минус 6 месяцев от текущего времени.».
6. **endTime** - окончание периода выборки. Формат - YYYY-MM-DD HH:MM:SS. По умолчанию: «сейчас».

Коды ответов:

1. **200** - Все хорошо.
2. **400** - Какой-то из параметров имеет неверный формат.
3. **401** - В запросе отсутствует заголовок **x-auth-token**.
4. **403** - Значение заголовка **x-auth-token** не корректно.

Пример запроса:

```
GET
/users/75666/tests/stats?testSessionId=628&startTime=2016-04-01+00:00:00
&endTime=2016-08-01+00:00:00 Headers: x-auth-token: HelloWorld
```

Пример ответа:

```
[ {
  "testId": 952678,
  "testPassings": [
    {
      "testSessionId": 628,
      "startTime": "2016-07-14 18:03:37",
      "endTime": "2016-07-14 18:03:44",
      "isResultSubmitted": true,
      "isTestPassed": false,
      "pointsSum": 0,
      "correctlyAnsweredQuestions": 1,
      "questions": [
        {
          "questionId": 882,
          "questionData": "Вопрос 1",
          "correctAnswersCount": 3,
          "answers": [
            {
              "answerId": 2053,
              "answerData": "правильный ответ 1",
              "isCorrect": true,
              "answerPoints": 0,
              "isCustomAnswerSubmitted": false,
              "isCustomAnswerAssessed": false
            },
            {
              "answerId": 2055,
```


В приведенном примере:

В массиве **testSessions** поле **correctlyAnsweredQuestions** содержит количество вопросов, на которые пользователь дал правильный ответ(выбрав все возможные правильные ответы)

Поле **correctAnswersCount** в данных вопроса содержит количество правильных ответов, определенных для данного вопроса.