

# API для интеграции с сервисом Nethouse.События

---

API предоставляет доступ к событиям, полям регистрации, билетам и участникам. Также, api позволяет передавать информацию об участниках, посетивших мероприятие.

## Общие понятия

---

Api представляет собой rest-api в формате JSON. Api принимает и отправляет данные в формате JSON. Для корректных ответов об ошибках в формате JSON, желательно в запросах передавать заголовок `Accept: application/json`

Даты и время передаются в формате `TIMESTAMP` в часовом поясе `UTC+3`.

Путь API на основном сервере: `https://events.nethouse.ru/api/public/v1/`

## Ограничения на использование

Допускается делать до 60 запросов к API в течение 1 минуты для одного токена. При превышении лимита будет выдан ответ с HTTP-кодом `429 Too Many Requests`.

## Коды ошибок HTTP

- 200 - входные данные верные, ответ сформирован и отправлен верно.
- 201 - входные данные верные, в результате запроса был создан ресурс.
- 204 - входные данные верные, в результате запроса был удалён ресурс.
- 400 - входные данные неверные, в ответе также будет содержаться информация о конкретных ошибках.
- 403 - входные данные неверные, по переданному токenu отсутствует доступ к указанному ресурсу.
- 404 - входные данные неверные, указанный ресурс не найден.
- 429 - превышено количество запросов для текущего токена.
- 500 - внутренняя ошибка сервиса. Мы пишем логи о таких ошибках.

## Авторизация

Авторизация происходит по токену, получаемому организатором события при подключении api. Токен состоит из:

- token - токен доступа к api
- expires - дата и время истечения токена доступа, период жизни токена - 7 дней
- refresh\_token - токен для обновления основного токена доступа
- refresh\_token\_expires - дата и время истечения токена обновления, период жизни токена - 6 месяцев

Токен принадлежит определённому профилю организатора. По этому токену клиент может обращаться ко всем событиям, принадлежащим профилю организатора.

Одному организатору доступен только один токен одновременно. При обновлении токена, устаревший токен использовать не получится.

Если соорганизатору выдан доступ к чужим событиям (через раздел Доступы) - к таким событиям не удастся получить доступ по токену соорганизатора. Для этого нужен токен, созданный от профиля основного организатора.

Токен передаётся при каждом запросе к api в виде заголовка

```
Authorization: Bearer [...token...]
```

Пример запроса

```
curl --location 'https://events.nethouse.ru/api/public/v1/ping' \  
--header 'Authorization: Bearer ACCESS_TOKEN_STRING'
```

## Доступные методы

- GET api/v1/ping - проверка связи с сервером по токену. Если токен ошибочный или устарел, будет выдана ошибка.
- POST api/v1/refreshToken - метод обновления токена доступа
- GET api/v1/constants - список констант и справочников для работы с записями. Эта информация статичная и меняется крайне редко.
- GET api/v1/events - список событий и курсы.
- GET api/v1/events/{id} - запись одного события или курса
- GET api/v1/tickets - список типов билетов
- GET api/v1/tickets/{id} - запись типа билета
- GET api/v1/options - список опций для типов билетов
- GET api/v1/options/{id} - запись одной опции для типа билета
- GET api/v1/forms - список полей формы регистрации и покупки билетов
- GET api/v1/forms/{id} - запись поля формы регистрации
- GET api/v1/orders - список заказов по событиям и курсам
- GET api/v1/orders/{id} - запись заказа
- GET api/v1/orderTickets - список купленных билетов (участников)
- GET api/v1/orderTickets/{id} - запись купленного билета участника
- PATCH api/v1/orderTickets/{id} - передача информации о посещении мероприятия участником

## Проверка и обновление токена

Проверить работу токена можно, обратившись на метод `ping`. В ответ вернётся json с параметрами текущего токена доступа. Если текущий токен истёк или передан неверно - вернётся соответствующая ошибка.

- Если токен доступа устарел - его можно обновить, передав в качестве токена `refresh_token` в запросе к методу `refreshToken`. В ответ будет выдан обновлённый токен доступа и токен обновления.
- Если для действующего токена доступа запросить обновление токена - действующий токен будет считаться устаревшим, обращаться по нему к api будет нельзя.
- Если истечёт токен обновления, новый токен можно запросить только через панель управления либо через поддержку.

Пример запроса проверку связи

```
curl --location 'https://events.nethouse.ru/api/public/v1/ping' \
--header 'Authorization: Bearer ACCESS_TOKEN_STRING' \
--header 'Content-Type: application/json' \
--header 'Accept: application/json'
```

Ответ

```
{"ping": "ok"}
```

Для обновления токена отправьте POST-запрос на метод `refreshToken`

```
curl -X POST --location 'https://events.nethouse.ru/api/public/v1/refreshToken' \
--header 'Authorization: Bearer REFRESH_TOKEN_STRING' \
--header 'Content-Type: application/json' \
--header 'Accept: application/json'
```

Ответ

```
{"data": {
  "token": "NEW_TOKEN_STRING",
  "expires_at": 1702904023,
  "refresh_token": "NEW_REFRESH_TOKEN_STRING",
  "refresh_token_expires_at": 1718110423
}}
```

## Постраничный вывод списков

Списки заказов и участников выводятся с разбивкой на страницы. В качестве перехода на следующую страницу используется курсор.

Если в результатах поиска есть следующая страница, будет передан курсор следующей страницы. Для получения результата следующей страницы, его необходимо передать в запросе с этими же параметрами плюс параметр `cursor=[...значение следующего курсора...]`

```
{
  "data": [
  ],
  "meta": {
    "next_cursor": "CURSOR_STRING"
  }
}
```

Принимаемые GET-параметры для списка записей

| Параметр | Тип данных      | Пример значения | Описание                                                                 |
|----------|-----------------|-----------------|--------------------------------------------------------------------------|
| limit    | unsigned int    | 10              | Количество записей на страницу, по умолчанию 30, минимум 1 максимум 100. |
| cursor   | string или null | "..."           | Курсор следующей страницы результатов поиска                             |

Примеры запроса

```
# первая страница
curl --location 'https://events.nethouse.ru/api/public/v1/orders' \
--header 'Authorization: Bearer ACCESS_TOKEN_STRING'

# следующая страница
curl --location 'https://events.nethouse.ru/api/public/v1/orders?cursor=CURSOR' \
--header 'Authorization: Bearer ACCESS_TOKEN_STRING'
```

## 1. Константы и справочники

GET api/public/v1/constants

```
{
  "success": true,
  "registration_form_types": {
    "Фамилия Имя Отчество": 1,
    "Имя": 2,
    "Отчество": 3,
    "Фамилия": 4,
    "Email": 5,
    "Телефон": 6,
    "Город": 7,
    "Текст": 8,
    "Галочка": 9,
    "Несколько вариантов": 10,
    "Один из вариантов": 11
  },
  "states": {
    "Черновик": 0,
    "Опубликовано": 1,
    "Отменено": 2,
    "Заблокировано": 3,
    "Завершено": 4,
    "Курс временно недоступен": 5,
    "В архиве": 6
  },
  "ticket_types": {
    "Офлайн": 1,
    "Онлайн": 2,
    "Офлайн+Онлайн": 3,
    "Видеозапись": 4,
    "Импортирован": 8,
    "Сертификат": 16
  }
}
```

| Параметр                | Тип данных | Пример значения | Описание                     |
|-------------------------|------------|-----------------|------------------------------|
| registration_form_types | object     |                 | Типы полей формы регистрации |
| states                  | object     |                 | Статусы событий              |
| ticket_types            | object     |                 | Типы билетов                 |

## 2. Получение списка доступных событий

Метод принимает необязательные фильтры:

- event\_id (unsigned int) - фильтр по указанному событию
- limit (unsigned int) - количество записей на страницу

GET api/public/v1/events

```
{
  "data": [
    {
      "id": 22,
      "name": "Пример события",
      "dates": "27.12.2022, 10:00 - 20.11.2023, 18:00",
      "state": 1,
      "url_public": "https://events.nethouse.ru/msk/22/",
      "url_admin": "https://events.nethouse.ru/admin/settings/22/"
    }
  ]
}
```

| Параметр   | Тип данных   | Пример значения  | Описание                                                           |
|------------|--------------|------------------|--------------------------------------------------------------------|
| id         | unsigned int | 22               | Идентификатор события                                              |
| name       | string(255)  | "Пример события" | Название события                                                   |
| dates      | string(255)  | "Бессрочно"      | Строка с указанием периода проведения мероприятия.                 |
| state      | unsigned int | 1                | Статус события, значение из справочника.                           |
| url_public | string       | "..."            | Абсолютный путь публичной страницы события.                        |
| url_admin  | string       | "..."            | Абсолютный путь страницы события в панели управления организатора. |

### 3. Получение списка билетов события

Билеты имеют разные типы, все типы представлены в методе передачи констант. Тип билета по сути представляет собой битовую маску, но может быть использован и как отдельные значения из справочника.

ВАЖНО: Тип билета "16": "Сертификат" не подразумевает посещение мероприятия. Этот билет - это сертификат на покупку настоящего билета.

Метод принимает необязательные фильтры:

- event\_id (unsigned int) - фильтр по указанному событию
- limit (unsigned int) - количество записей на страницу

GET api/public/v1/tickets

```
{
  "data": [
    {
      "id": 486,
      "event_id": 22,
      "name": "Билет",
      "price": 0,
      "type": 3,
      "deleted_at": null
    }
  ]
}
```

| Параметр   | Тип данных         | Пример значения | Описание                                      |
|------------|--------------------|-----------------|-----------------------------------------------|
| id         | unsigned int       | 486             | Идентификатор билета                          |
| event_id   | unsigned int       | 22              | Идентификатор события                         |
| name       | string(255)        | "Билет"         | Название билета                               |
| price      | unsigned int       | 0               | Цена билета                                   |
| type       | unsigned int       | 3               | Тип билета, согласно справочнику в константах |
| deleted_at | timestamp или null | null            | Дата и время удаления билета                  |

## 4. Получение списка опций билетов события

Метод принимает необязательные фильтры:

- event\_id (unsigned int) - фильтр по указанному событию
- limit (unsigned int) - количество записей на страницу

GET api/public/v1/options

```
{
  "data": [
    {
      "id": 929,
      "event_id": 22,
      "ticket_id": 491,
      "name": "Бесплатная неограниченная опция",
      "price": 0,
      "deleted": false
    }
  ]
}
```

| Параметр  | Тип данных   | Пример значения | Описание              |
|-----------|--------------|-----------------|-----------------------|
| id        | unsigned int | 486             | Идентификатор опции   |
| event_id  | unsigned int | 22              | Идентификатор события |
| ticket_id | unsigned int | 491             | Идентификатор билета  |
| name      | string(255)  | "Билет"         | Название опции        |
| price     | unsigned int | 0               | Цена опции            |
| deleted   | bool         | false           | Флаг удаления опции   |

## 5. Получение списка полей регистрационной формы

Метод принимает необязательные фильтры:

- `event_id` (unsigned int) - фильтр по указанному событию
- `limit` (unsigned int) - количество записей на страницу

Важно: если поле отмечено как `is_custom` у полей 10 (Несколько вариантов) или 11 (Один из вариантов), это означает что организатор разрешил участникам для этого поля добавлять значения не из справочника `variants`. Таким образом, для регистрационного поля значения `variants` не являются полным списком возможных значений.

GET `api/public/v1/forms`

```
{
  "data": [
    {
      "id": 73,
      "type": "email",
      "name": "Email",
      "is_custom": false,
      "is_required": true,
      "variants": null
    },
    {
      "id": 84,
      "type": "checkboxList",
      "name": "Необязательный выбор радиобатонов",
      "is_custom": true,
      "is_required": false,
      "variants": [
        "Вариант №10",
        "Вариант №11",
        "Вариант №12"
      ]
    }
  ]
}
```

| Параметр    | Тип данных            | Пример значения                                 | Описание                                                   |
|-------------|-----------------------|-------------------------------------------------|------------------------------------------------------------|
| id          | unsigned int          | 1407                                            | Идентификатор заказа                                       |
| name        | string(255)           | "Обязательный<br>мультивыбор чекбоксов"         | Название поля                                              |
| type        | string(255)           | "checkboxList"                                  | Тип поля, согласно справочнику<br>в константах             |
| is_custom   | bool                  | true                                            | Допускаются пользовательские<br>значение не из справочника |
| is_required | bool                  | false                                           | Поле обязательно для<br>заполнения                         |
| variants    | array                 | [ "Вариант №1", "Вариант<br>№2", "Вариант №3" ] | Варианты значений для полей с<br>несколькими вариантами    |
| deleted_at  | timestamp<br>или null | null                                            | Дата и время удаления поля                                 |

## 6. Получение списка заказов

Метод принимает необязательные фильтры:

- `event_id` (unsigned int) - фильтр по указанному событию
- `orderId` (unsigned int) - фильтр по указанному заказу
- `updated_from` (timestamp) - показать заказы, обновлённые начиная с указанной даты (включая)
- `limit` (unsigned int) - количество записей на страницу

GET `api/public/v1/orders`

```
{
  "data": [
    {
      "id": 1407,
      "event_id": 22,
      "external_id": null,
      "buyer_email": "user.961836@nethouse.ru",
      "buyer_name": "Трофим Иванович Мельников",
      "discount": {
        "amount": 0,
        "currency": "RUB"
      },
      "total": {
        "amount": 0,
        "currency": "RUB"
      },
      "pay_until": 1682931600,
      "is_completed": false,
      "is_canceled": false,
      "created_at": 1682931600,
      "updated_at": 1682931600,
      "deleted_at": null
    }
  ],
  "meta": {
    "next_cursor": "CURSOR_STRING"
  }
}
```

| Параметр     | Тип данных              | Пример значения                    | Описание                                                                         |
|--------------|-------------------------|------------------------------------|----------------------------------------------------------------------------------|
| id           | unsigned int            | 1407                               | Идентификатор заказа                                                             |
| event_id     | unsigned int            | 22                                 | Идентификатор события                                                            |
| external_id  | string(255)<br>или null | "322"                              | Идентификатор заказа во внешней системе, используется при импортировании заказов |
| buyer_email  | string                  | "user.961836@nethouse.ru"          | Email плательщика                                                                |
| buyer_name   | string или null         | "Трофим Иванович Мельников"        | ФИО плательщика                                                                  |
| discount     | object                  | "{"amount": 0, "currency": "RUB"}" | Скидка по заказу                                                                 |
| total        | object                  | "{"amount": 0, "currency": "RUB"}" | Стоимость заказа с учётом скидки                                                 |
| pay_until    | timestamp<br>или null   | 1682931600                         | Дата и время обновления заказа                                                   |
| is_completed | bool                    | false                              | Заказ завершён. Для платных заказов подразумевается что заказ оплачен            |
| is_canceled  | bool                    | false                              | Заказ отменён автоматически, участником или организатором                        |
| created_at   | timestamp               | 1682931600                         | Дата и время создания заказа                                                     |
| updated_at   | timestamp<br>или null   | 1682931600                         | Дата и время обновления заказа                                                   |
| deleted_at   | timestamp<br>или null   | null                               | Дата и время удаления заказа                                                     |

## 7. Получение списка участников события

GET `api/public/v1/orderTickets` - СПИСОК участников события.

```
{
  "data": [
    {
      "id": 1380,
      "event_id": 22,
      "order_id": 1407,
      "ticket_id": 491,
      "external_id": null,
      "price": {
        "amount": 0,
        "currency": "RUB"
      },
      "discount": {
        "amount": 0,
        "currency": "RUB"
      },
      "sum": {
        "amount": 0,
        "currency": "RUB"
      },
      "fields": {
        "73": "user.92440@nethouse.ru",
        "74": "Некрасоваа Дина Ивановна",
        "75": "Text 92440",
        "76": "+73456348545",
        "77": "Да",
        "78": "Вариант №1, Вариант №2, Вариант №3",
        "79": "Вариант №4",
        "80": "",
        "81": "",
        "82": "Нет",
        "83": "",
        "84": "Вариант №10"
      },
      "hall": {
        "hall": "Большой зал",
        "group": "Партер",
        "place": "Ряд 4 место 21"
      },
      "key": "qwe-123-asd",
      "options": {
        "929": true
      },
      "person": {
        "full_name": "Людмила Фёдоровна Калининаа",
        "first_name": "Фёдоровна",
        "middle_name": "Калининаа",

```

```
    "last_name": "Людмила",
    "email": "user.661721@nethouse.ru",
    "phone": "+78255737621",
    "city": null
  },
  "promocode": {
    "id": 321,
    "name": "SALE-25"
  },
  "session": {
    "id": 1,
    "name": "Утренний сеанс",
    "datetime": 1682931600
  },
  "is_canceled": false,
  "created_at": 1682931600,
  "updated_at": 1682931600,
  "deleted_at": null,
  "canceled_at": null,
  "expires_at": null,
  "visited_at": [
    1682931600
  ]
}
],
"meta": {
  "next_cursor": null
}
}
```

| Параметр    | Тип данных           | Пример значения                                                               | Описание                                                                                                |
|-------------|----------------------|-------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|
| id          | unsigned int         | 1407                                                                          | Идентификатор билета                                                                                    |
| event_id    | unsigned int         | 22                                                                            | Идентификатор события                                                                                   |
| order_id    | unsigned int         | 1407                                                                          | Идентификатор заказа                                                                                    |
| ticket_id   | unsigned int         | 491                                                                           | Идентификатор билета                                                                                    |
| external_id | string(255) или null | "322"                                                                         | Идентификатор билета во внешней системе, используется при импортировании заказов                        |
| price       | object               | "{"amount": 0,"currency": "RUB"}"                                             | Цена билета                                                                                             |
| discount    | object               | "{"amount": 0,"currency": "RUB"}"                                             | Скидка по применённому промокоду                                                                        |
| sum         | object               | "{"amount": 0,"currency": "RUB"}"                                             | Стоимость билета с учётом скидки и заказанных опций                                                     |
| fields      | object               | см. ниже                                                                      | Регистрационные данные, указанные участником при регистрации, для каждого регистрационного поля события |
| hall        | object               | {"hall": "Зал на 300 мест #1", "group": "Сектор 1", "place": "Ряд 4 место 1"} | Выбранное место в зале                                                                                  |
| key         | string или null      | "Трофим Иванович Мельников"                                                   | Ключ доступа участника, применяется на онлайн-событиях                                                  |

| Параметр    | Тип данных          | Пример значения                                          | Описание                                                               |
|-------------|---------------------|----------------------------------------------------------|------------------------------------------------------------------------|
| options     | object              | {"929":true}                                             | Дополнительные опции                                                   |
| person      | object              | см. ниже                                                 | Регистрационные данные с разобранными именем и фамилией                |
| promocode   | object или null     | {"id":123, "name":"PROMO-50"}                            | Промокод, применённый к билету                                         |
| session     | object или null     | {"id":1, "name":"Утренний сеанс", "datetime":1682931600} | Данные выбранного сеанса                                               |
| is_canceled | bool                | false                                                    | Билет отменён                                                          |
| created_at  | timestamp или null  | 1682931600                                               | Дата и время создания билета                                           |
| updated_at  | timestamp или null  | 1682931600                                               | Дата и время обновления билета                                         |
| deleted_at  | timestamp или null  | null                                                     | Дата и время удаления билета                                           |
| canceled_at | timestamp или null  | null                                                     | Дата и время отмены билета участником или организатором                |
| expires_at  | timestamp или null  | null                                                     | Дата и время истечения билета, билет не может быть использован позднее |
| visited_at  | array of timestamps | null                                                     | Дата и время использования билета либо посещения мероприятия           |

Пример объекта `fields`

```
{  
  "73": "user.92440@nethouse.ru",  
  "74": "Некрасоваа Дина Ивановна",  
  "75": "Text 92440",  
  "76": "+73456348545",  
  "77": "Да",  
  "78": "Вариант №1, Вариант №2, Вариант №3",  
  "79": "Вариант №4",  
  "80": "",  
  "81": "",  
  "82": "Нет",  
  "83": "",  
  "84": "Вариант №10"  
}
```

Пример объекта `person`

```
{  
  "full_name": "Людмила Фёдоровна Калинина",  
  "first_name": "Фёдоровна",  
  "middle_name": "Калинина",  
  "last_name": "Людмила",  
  "email": "user.661721@nethouse.ru",  
  "phone": "+78255737621",  
  "city": null  
}
```

## 8. Передача информации о том, что участник посетил событие

PATCH `api/public/v1/orderTickets/1407?visited_at=1682931600`

Метод принимает один параметр: `visited_at`, в котором передаётся дата и время посещения в формате timestamp в часовом поясе UTC+3.

В результате будет возвращён полный объект участника.