

---

ticketscloud

*Выпуск v2*

июн. 22, 2022



<b>1</b>	<b>Содержание</b>	<b>1</b>
1.1	Пошаговая интеграция для распространителей	1
1.1.1	Базовые понятия	1
1.1.1.1	Основные URL	1
1.1.1.2	Авторизация	1
1.1.2	Шаг 1: Информация о мероприятии	2
1.1.3	Шаг 2: Создание заказа	2
1.1.3.1	Создание	2
1.1.3.2	Резервирование билетов	8
1.1.4	Шаг 3: Завершение заказа	12
1.1.4.1	Смена статуса заказа	12
1.1.5	Пример проведения заказа	15
1.1.5.1	0. Получаем информацию о событиях	16
1.1.5.2	1. Получаем информацию о билетах	16
1.1.5.3	2a. Создаем заказ (добавляем билет без места)	16
1.1.5.4	3a. Заполняем заказ (добавляем билет с местом)	18
1.1.5.5	2b. Создаем заказ (добавляем билет с местом)	20
1.1.5.6	3b. Заполняем заказ (добавляем билет без места)	22
1.1.5.7	4. Завершаем заказ	22
1.1.6	Шаг 4: Возврат заказа	25
1.1.6.1	Создание возврата	25
1.1.6.2	Ошибки при создании возврата	29
1.1.6.3	Подтверждение и отмена возврата	29
1.1.6.4	Ошибки при подтверждении возврата	30
1.1.6.5	Ошибки при отмене возврата	30
1.2	Продвинутое использование	30
1.2.1	Жизненный цикл заказа	30
1.2.1.1	Список состояний (статусов) заказа без платежей	30
1.2.2	Действия с заказом	31
1.2.2.1	Добавить информацию о покупателе	31
1.2.2.2	Добавить произвольную информацию в объект заказа	33
1.2.2.3	Отправка билетов покупателю на email	34
1.2.2.4	Добавить промокоды к заказу	36
1.2.3	Промокоды	39
1.2.3.1	Проверка промокода	39
1.2.4	Работа со списком заказов	42

1.2.4.1	Получение списка заказов . . . . .	42
1.2.4.2	Экспорт списка заказов на почту . . . . .	44
1.2.5	Работа со списком возвратов . . . . .	45
1.2.5.1	Получаем список возвратов . . . . .	45
1.2.6	Повторяющиеся сущности . . . . .	46
1.2.6.1	VEVENT . . . . .	46
1.2.6.2	Media . . . . .	46
1.2.6.3	ObjectId . . . . .	47
1.2.6.4	ISODatetime . . . . .	47
<b>2</b>	<b>Дополнительно</b>	<b>49</b>
	<b>TicketsCloud HTTP API</b>	<b>51</b>
	<b>Алфавитный указатель</b>	<b>53</b>

## 1.1 Пошаговая интеграция для распространителей

### 1.1.1 Базовые понятия

**Примечание:** Для интеграции вам необходимо пройти [регистрацию](#) и [заключить](#) сделку с организатором мероприятия

Ticketcloud API построено с учетом *REST*. С авторизацией через *HTTP Basic Auth*. В теле запросов и ответов используется *JSON*. Для демонстрации запросов к API используется утилита [HTTPIe](#)

#### 1.1.1.1 Основные URL

Все ссылки на запросы к API в данной документации оюращаются к основному URL: <https://ticketcloud.com>

Для облегчения интеграции и отладки существует также стейдж сервер: <https://stage.freetc.net>

#### 1.1.1.2 Авторизация

Для авторизации по ключу, значение заголовка [Authorization](#) имеет префикс *key*

Пример запроса:

```
GET /v2/resources/orders HTTP/1.1
Authorization: key 9bd8359943b545500278875r49c5b96d
```

Пример ответа:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "data": [
    ...
  ],
  "pagination": {
    ...
  },
  "refs": {
    ...
  }
}
```

**Предупреждение:** Все запросы к API осуществляются только через *HTTPS*. Запросы через *HTTP* будут отклонены со статусом *301 Moved Permanently*.

### 1.1.2 Шаг 1: Информация о мероприятии

Для получения списка мероприятий пользуйтесь gRPC-сервисом `tc-simple`.

Ранее использовался устаревший сервис, более не поддерживаемый.

### 1.1.3 Шаг 2: Создание заказа

---

**Примечание:** Примеры ответов и запросов в данном разделе упрощены. Тем не менее этого достаточно для интеграции. Детальная информация по каждому разделу присутствует в блоках *См. также*

---

#### 1.1.3.1 Создание

##### Запрос

POST `/v2/resources/orders`

##### JSON Parameters

- `event` – Обязателен

##### Ответ

Объект заказа

```
{
  "data": {
    "id": objectid
    "status": 'executed' | 'in_progress' | 'done' | 'cancelled' | 'expired'
    "created_at": isodatetime
    "done_at": isodatetime
    "expired_after": isodatetime
  }
}
```

(continues on next page)

(продолжение с предыдущей страницы)

```

"event": objectid
"number": int
"org": objectid
"origin": 'api' | 'salespoint' | 'widget' | 'control_panel'
"promocodes": [
    str
]
"settings": {
    "invitation": bool
    "send_tickets": bool
    "courier": bool
    "customer": {
        "name": str
        "email": email
        "phone": str
        "lang": 'en' | 'ru'
    }
}
"tickets": [
    {
        "id": objectid
        "serial": str
        "number": int
        "seat": {
            "row": str
            "number": str
            "sector": objectid
        }
        "status": str
        "price": money
        "nominal": money
        "discount": money
        "extra": money
        "full": money
    },
]
"values": {
    "discount": money
    "extra": money
    "full": money
    "nominal": money
    "price": money
    "sets_values": {
        id: {
            "discount": money
            "id": objectid
            "nominal": money // price with discount
            "price": money
            "promocode": objectid | None
        },
        ...
    }
    "viral_promocodes": [...]
},
"vendor": objectid
"vendor_data": {

```

(continues on next page)

(продолжение с предыдущей страницы)

```
        "order_id": str
        "raw": {
            ...
        }
    },
    "refs": {
        "events": {
            id: {
                "id": objectid
                "lifetime": {
                    "finish": isodatetime
                    "start": isodatetime
                },
                "org": objectid
                "status": str
                "timezone": str
                "title": {
                    "desc": str
                    "text": str
                }
                "venue": objectid
                "age_rating": int
            }
        },
        "partners": {
            id: {
                "id": objectid
                "name": str
            }
        },
        "promocodes": {
            id: {
                "code": str
                "discount": {
                    "percentage" | "fix": str
                },
                "id": objectid
                "lifetime": {
                    "start": isodatetime
                    "finish": isodatetime
                },
                "viral": bool
            }
        },
        "sets": {
            id: {
                "id": objectid
                "name": str
                "price": money
                "with_seats": bool
            }
        },
        "venues": {
            id: {
                "address": str
            }
        }
    }
}
```

(continues on next page)



(продолжение с предыдущей страницы)

```

        "city": {
            "_id": int,
            "name": {
                "default": str
                "en": str
                "ru": str
            },
            "timezone": str
        },
        "country": {
            "_id": str
            "name": {
                "default": str
                "en": str
                "ru": str
            }
        },
        "desc": str
        "id": objectid
        "name": str
    }
}
}
}

```

### Пример запроса

```

POST /v2/resources/orders HTTP/1.1
Authorization: key 9bd8359943b545500278875r49c5b96d
Content-Type: application/json

{
    "event": "5d31fbdd27649b0dff076117"
}

```

### Пример ответа

```

HTTP/1.1 200 OK
Content-Type: application/json

{
    "data": {
        "created_at": "2019-07-25 14:41:11",
        "event": "5d31fbdd27649b0dff076117",
        "expired_after": "2019-07-25 14:56:11",
        "id": "5d39bf878cd381a4891e07b7",
        "number": 54877,
        "org": "5b04229196c055000d87c2b5",
        "origin": "api",
        "promocodes": [],
        "settings": {
            "invitation": false,
            "send_tickets": false,
            "subscribe_agree": false
        },
        "status": "executed",
    }
}

```

(continues on next page)

(продолжение с предыдущей страницы)

```

"tickets": [],
"values": {
  "discount": "0.00",
  "extra": "0.00",
  "full": "0.00",
  "nominal": "0.00",
  "price": "0.00",
  "sets_values": {
    "5d31fbfa27649b0dff07611b": {
      "discount": "0.00",
      "id": "5d31fbfa27649b0dff07611b",
      "nominal": "150.00",
      "price": "150.00",
      "promocode": null
    },
    "5d31fbfa8a75c12c9d64de13": {
      "discount": "0.00",
      "id": "5d31fbfa8a75c12c9d64de13",
      "nominal": "1050.00",
      "price": "1050.00",
      "promocode": null
    }
  },
  "viral_promocodes": []
},
"vendor": "5b02d6e9517565000d9cb1ce",
"vendor_data": {}
},
"refs": {
  "events": {
    "5d31fbdd27649b0dff076117": {
      "id": "5d31fbdd27649b0dff076117",
      "lifetime": {
        "finish": "2019-08-28 21:50:00",
        "start": "2019-08-15 21:20:00"
      },
      "org": "5b04229196c055000d87c2b5",
      "status": "public",
      "timezone": "Europe/Moscow",
      "title": {
        "desc": "1",
        "text": "1234567890"
      },
      "age_rating": 16,
      "venue": "552322649cb5384154e028b2"
    },
  },
  "partners": {
    "5b02d6e9517565000d9cb1ce": {
      "id": "5b02d6e9517565000d9cb1ce",
      "name": "Rasp new"
    },
    "5b04229196c055000d87c2b5": {
      "id": "5b04229196c055000d87c2b5",
      "name": "My best org"
    }
  }
}

```

(continues on next page)

(продолжение с предыдущей страницы)

```

    },
    "promocodes": {},
    "sets": {},
    "venues": {
      "552322649cb5384154e028b2": {
        "address": "ул. Родионова, 4",
        "city": {
          "_id": 520555,
          "name": {
            "default": "Nizhniy Novgorod",
            "en": "Nizhny Novgorod",
            "ru": "Нижний Новгород"
          },
          "timezone": "Europe/Moscow"
        },
        "country": {
          "_id": "RU",
          "name": {
            "default": "Russia",
            "en": "Russia",
            "ru": "Россия"
          }
        },
        "desc": "MILO Concert Hall - новая большая профессиональная площадка в ↵
        ↵Нижнем Новгороде",
        "id": "552322649cb5384154e028b2",
        "name": "MILO Concert Hall"
      }
    }
  }
}

```

**Примечание:** ШК появляется только у заказа в статусе done

**См.также:**

*Заказ в статусе executed, Жизненный цикл заказа*

**Возможные ошибки при создании заказа** При ошибках возвращается список текстовых сообщений:

```

HTTP/1.1 400
Content-Type: application/json

{
  "errors": [
    "Event <event_id> not found"
  ]
}

```

Сообщение	Причина
Event <event_id> not found	не найдено мероприятие для заказа
Deal not found	не найдена сделка
Promokey needed	промоключ должен быть
Promokey not founded	промоключ не найден
Promokey expired	промоключ истек
Promokey fully reserved	промоключ полностью зарезервирован
Promokey already reserved	промоключ уже зарезервирован
Only one of tickets or random can be set	только один: tickets или random может быть установлено
„tickets“ or „random“ must be	обязан быть: tickets или random
Cant use promocode when event have promotion	нельзя использовать промокод, пока на мероприятии действует промоакция
expired_after is not changable	время жизни не может быть изменено
max expired_after is {dt}	время жизни истекло <dt :%Y-%m-%d %H:%M:%S>
Promokey needed to add these tickets	необходим промоключ, чтобы добавить эти билеты
ticket (id = {}) does not belong to current event	билет не относится к текущему мероприятию
value should be formatted „YYYY-MM-DD“	для дополнительного поля «Дата», значение имеет неверный формат
value should be True or False	для дополнительного поля «Галочка», значение должно быть булевым
value is not a list	для дополнительного поля «Выбор нескольких вариантов», значение должно быть списком
list length is less than <len>	для дополнительного поля «Выбор нескольких вариантов», значение должно быть длиннее чем <len>
list length is greater than <len>	для дополнительного поля «Выбор нескольких вариантов», значение должно быть короче чем <len>
value is less than <value>	для дополнительного поля «Выбор одного варианта», значение должно быть больше чем <value>
value is greater than <value>	для дополнительного поля «Выбор одного варианта», значение должно быть меньше чем <value>
value should be less than <value>	для дополнительного поля «Выбор одного варианта», значение должно быть меньше либо равно <value>
value should be greater than <value>	для дополнительного поля «Выбор одного варианта», значение должно быть больше либо равно <value>
value is not a string	для дополнительного поля «Текст/Длинный текст», значение должно быть представлено строкой
blank value is not allowed	для дополнительного поля «Текст/Длинный текст», значение не должно быть пустым
String is shorter than <len> characters	для дополнительного поля «Текст/Длинный текст», значение должно быть длиннее <len> символов
String is longer than <len> characters	для дополнительного поля «Текст/Длинный текст», значение должно быть короче <len> символов
does not match pattern <pattern>	для дополнительного поля «Текст/Длинный текст», значение должно соответствовать формату <pattern>

### 1.1.3.2 Резервирование билетов

**Примечание:** Все действия с заказом, кроме его создания, делаются по запросу `PATCH /v2/resources/orders/:id`. В один запрос одновременно может быть добавлено несколько действий. Все запросы на изменение конкретного заказа, должны делаться синхронно. В случае получения запроса до конца обработки предыдущего, будет возвращена ошибка `409 Conflict`.

Это ограничение касается только работы с одним заказом, а работать одновременно с несколькими заказами можно.

За резервирование билетов отвечают три поля:

- `tickets`
- `random`
- `all_or_nothing`

#### Поле `tickets`

В поле `tickets` передаются все `id` билетов, которые должны быть зарезервированы текущим заказом. Если заказ изменяется (покупатель решил добавить ещё один билет), то в обязательном порядке передаются все билеты, которые должны быть в заказе (в т.ч. те, что уже зарезервированы). Для удаления конкретного билета из заказа, нужно передать все билеты, кроме удаляемого.

**Предупреждение:** Нельзя использовать в одном запросе с `random`.

#### Запрос

`PATCH /v2/resources/orders/:id`

##### JSON Parameters

- `tickets` – список `id` билетов

#### Пример запроса

```
PATCH /v2/resources/orders/5b0eab671b2042000ea83850 HTTP/1.1
Authorization: key 9bd8359943b545500278875r49c5b96d
Content-Type: application/json

{
  "tickets": [
    "5b0e8af09adc660001b0ab25",
    "5b0e8afa9adc660001b0ab6c"
  ]
}
```

#### Поле `random`

Поле `random` предназначено для резервирования случайных билетов из указанных категорий. Оно нужно для того, чтобы добавлять в заказ билеты без мест. Имеет вид объекта, где ключ является `id` категории, а значение — количество билетов. Так же, как и в `tickets`, всегда нужно передавать желаемое состояние. Т.е. если пользователь удалил один билет из категории, то передать надо `random` со всеми категориями и количествами, только в одной из категорий будет на один билет меньше. В ответе от сервера всегда будет список забронированных билетов в поле `tickets`.

**Предупреждение:** Нельзя использовать в одном запросе с *tickets*.

### Запрос

PATCH /v2/resources/orders/:id

#### JSON Parameters

- random – массив: ключ – id категории, значение – кол-во мест

### Пример запроса

```
PATCH /v2/resources/orders HTTP/1.1
Authorization: key 9bd8359943b545500278875r49c5b96d
Content-Type: application/json

{
  "random": {
    "5d31fbfa27649b0dff07611b": 1
  }
}
```

### Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "data": {
    "created_at": "2019-07-25 14:41:11",
    "event": "5d31fbdd27649b0dff076117",
    "expired_after": "2019-07-25 14:56:11",
    "id": "5d39bf878cd381a4891e07b7",
    "number": 54877,
    "org": "5b04229196c055000d87c2b5",
    "origin": "api",
    "settings": {
      "invitation": false,
      "send_tickets": false,
      "subscribe_agree": false
    },
    "status": "executed",
    "tickets": [
      {
        "barcode": null,
        "discount": "0.00",
        "extra": "6.00",
        "full": "156.00",
        "id": "5d31fbfb306fdcc187b91179",
        "nominal": "150.00",
        "number": 157495,
        "price": "150.00",
        "serial": "00X",
        "set": "5d31fbfa27649b0dff07611b",
        "status": "reserved"
      }
    ],
    "values": {
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "discount": "0.00",
    "extra": "6.00",
    "full": "156.00",
    "nominal": "150.00",
    "price": "150.00",
    "sets_values": {
      "5d31fbfa27649b0dff07611b": {
        "discount": "0.00",
        "id": "5d31fbfa27649b0dff07611b",
        "nominal": "150.00",
        "price": "150.00",
        "promocode": null
      },
      "5d31fbfa8a75c12c9d64de13": {
        "discount": "0.00",
        "id": "5d31fbfa8a75c12c9d64de13",
        "nominal": "1050.00",
        "price": "1050.00",
        "promocode": null
      }
    },
    "viral_promocodes": [],
  },
  "vendor": "5b02d6e9517565000d9cb1ce",
  "vendor_data": {}
},
"refs": {
  "events": {
    "5d31fbdd27649b0dff076117": {
      "id": "5d31fbdd27649b0dff076117",
      "lifetime": {
        "finish": "2019-08-28 21:50:00",
        "start": "2019-08-15 21:20:00"
      },
      "org": "5b04229196c055000d87c2b5",
      "status": "public",
      "timezone": "Europe/Moscow",
      "title": {
        "desc": "1",
        "text": "1234567890"
      }
    }
  },
  "partners": {
    "5b02d6e9517565000d9cb1ce": {
      "id": "5b02d6e9517565000d9cb1ce",
      "name": "Rasp new"
    },
    "5b04229196c055000d87c2b5": {
      "id": "5b04229196c055000d87c2b5",
      "name": "My best org"
    }
  },
  "sets": {
    "5d31fbfa27649b0dff07611b": {
      "id": "5d31fbfa27649b0dff07611b",

```

(continues on next page)

(продолжение с предыдущей страницы)

```
        "name": "обычные",
        "price": "150.00",
        "with_seats": false
      }
    }
  }
}
```

---

**Примечание:** Хотя `tickets` и `random` нельзя использовать в одном запросе, передать все билеты можно.

Алгоритм такой:

- Запрос с `random`.
  - Получить из ответа `id` билетов в поле `tickets`.
  - Расширить список полученных `id` новым билетом.
  - Отправить запрос с полем `tickets`, заполненным списком `id` билетов.
  - Проверить в ответе, что всё забронировалось корректно.
- 

### Поле `all_or_nothing`

Если значение поля `all_or_nothing` равно `true`, то резервируются либо все билеты, либо ни одного.

При изменении заказа с одновременным разрезервированием и резервированием билетов, в случае неудачи с резервированием хотя бы одного билета, разрезервирования не происходит, т.е. список зарезервированных билетов не изменяется.

Если значение не указано, или `false`, то билеты, которые не удалось забронировать пропускаются и отсутствуют поле `tickets` в ответе.

---

**Примечание:** Можно использовать, как с `tickets`, так и с `random`

---

## 1.1.4 Шаг 3: Завершение заказа

---

**Примечание:** На данном шаге рассматривается завершение заказа без работы с платежными системами.

---

### 1.1.4.1 Смена статуса заказа

Запрос

PATCH /v2/resources/orders/:id

JSON Parameters

- `status` – Обязателен (`done` | `cancelled`)



**Ответ***Объект заказа***Пример запроса**

```
PATCH /v2/resources/orders/5d39c3108cd381a4891e07e0 HTTP/1.1
Authorization: key 9bd8359943b545500278875r49c5b96d
Content-Type: application/json

{
  "status": "done"
}
```

**Пример ответа**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "data": {
    "code": "97elgi8j",
    "created_at": "2019-07-25 14:56:16",
    "done_at": "2019-07-25 14:57:47",
    "event": "5d31fbdd27649b0dff076117",
    "expired_after": "2019-07-25 15:11:16",
    "id": "5d39c3108cd381a4891e07e0",
    "number": 54881,
    "org": "5b04229196c055000d87c2b5",
    "origin": "api",
    "promocodes": [
      "5b0eab2d1b2042000db2438f"
    ],
    "settings": {
      "invitation": false,
      "send_tickets": false,
      "subscribe_agree": false
    },
    "status": "done",
    "tickets": [
      {
        "barcode": null,
        "discount": "0.00",
        "extra": "6.00",
        "full": "156.00",
        "id": "5d31fbfb306fdcc187b91179",
        "nominal": "150.00",
        "number": 157495,
        "price": "150.00",
        "serial": "00X",
        "set": "5d31fbfa27649b0dff07611b",
        "status": "reserved"
      }
    ],
    "values": {
      "discount": "0.00",
      "extra": "6.00",
      "full": "156.00",
      "nominal": "150.00",

```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "price": "150.00",
    "sets_values": {
      "5d31fbfa27649b0dff07611b": {
        "discount": "0.00",
        "id": "5d31fbfa27649b0dff07611b",
        "nominal": "150.00",
        "price": "150.00",
        "promocode": "5b0eab2d1b2042000db2438f"
      },
      "5d31fbfa8a75c12c9d64de13": {
        "discount": "0.00",
        "id": "5d31fbfa8a75c12c9d64de13",
        "nominal": "1050.00",
        "price": "1050.00",
        "promocode": "5b0eab2d1b2042000db2438f"
      }
    },
    "viral_promocodes": [],
  },
  "vendor": "5b02d6e9517565000d9cb1ce",
  "vendor_data": {}
},
"refs": {
  "events": {
    "5d31fbdd27649b0dff076117": {
      "id": "5d31fbdd27649b0dff076117",
      "lifetime": {
        "finish": "2019-08-28 21:50:00",
        "start": "2019-08-15 21:20:00"
      },
      "org": "5b04229196c055000d87c2b5",
      "status": "public",
      "timezone": "Europe/Moscow",
      "title": {
        "desc": "1",
        "text": "1234567890"
      }
    }
  },
  "partners": {
    "5b02d6e9517565000d9cb1ce": {
      "id": "5b02d6e9517565000d9cb1ce",
      "name": "Rasp new"
    },
    "5b04229196c055000d87c2b5": {
      "id": "5b04229196c055000d87c2b5",
      "name": "My best org"
    }
  },
  "promocodes": {
    "5b0eab2d1b2042000db2438f": {
      "code": "100",
      "discount": {
        "percentage": "0%"
      },
      "id": "5b0eab2d1b2042000db2438f",

```

(continues on next page)

(продолжение с предыдущей страницы)

```

        "lifetime": null,
        "viral": false
      },
      "sets": {
        "5d31fbfa27649b0dff07611b": {
          "id": "5d31fbfa27649b0dff07611b",
          "name": "обычные",
          "price": "150.00",
          "with_seats": false
        }
      }
    }
  }
}

```

См.также:

*Жизненный цикл заказа Пример проведения заказа*

**Предупреждение:** Заказ в статусе **done** больше нельзя изменять. Любые PATCH запросы на данный заказ будут отклонены со статусом **400 Bad Request**.

**Возможные ошибки при завершении заказа** При ошибках возвращается список текстовых сообщений:

```

HTTP/1.1 400
Content-Type: application/json

{
  "errors": [
    "Not enough money, contact to TicketsCloud manager"
  ]
}

```

Сообщение	Причина
Not enough money, contact to TicketsCloud manager	недостаточно денег, свяжитесь с менеджером TicketsCloud
there is no tickets in order	в заказе нет ни одного билета
Incorrect status <status> for this operation	нельзя выполнить перевод заказа в статус: <status>
Order <order_id> is not cancellable	заказ с id: <order_id> не может быть отменен

**Поздравляем! Вы провели свой первый заказ через API!**

### 1.1.5 Пример проведения заказа

**Примечание:** С нуля создадим и проведем смешанный заказ: один билет с рассадкой, другой – без. Будет два примера, отличающихся порядком добавления этих билетов.

### 1.1.5.1 0. Получаем информацию о событиях

На этом этапе выбираем мероприятие, на которое будем проводить заказ. Отправляем запрос Events в gRPC-сервис `tc-simple`, в полученном списке находим нужное мероприятие и берём его id. Допустим, это `5d7134962110d30a34e95b96`. Также в ответе будут id сетов – например, `5d7135112110d30a34e97e2d`.

### 1.1.5.2 1. Получаем информацию о билетах

Для последующего заказа билетов с местом – передаём id мероприятия в запросе Seats в gRPC-сервис `tc-simple`, в полученном списке находим места и получаем id билетов (допустим, `5d7134962110d30a34e95cfe`). Если билетов с местами в продаже не будет, этот шаг пропускается.

### 1.1.5.3 2а. Создаем заказ (добавляем билет без места)

Берем id добавляемого сета из п.0

```
POST /v2/resources/orders HTTP/1.1
Authorization: key 9bd8359943b545500278875r49c5b96d
Content-Type: application/json

{
  "event": "5d7134962110d30a34e95b96",
  "random": {
    "5d7135112110d30a34e97e2d": 1
  }
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "data": {
    "created_at": "2019-09-25 16:14:01",
    "event": "5d7134962110d30a34e95b96",
    "expired_after": "2019-09-25 16:29:01",
    "id": "5d8b924971a0bf323bd6a6ed",
    "number": 59743,
    "org": "5ba10ea90c43fc000b0fc786",
    "origin": "api",
    "status": "executed",
    "tickets": [
      {
        "barcode": null,
        "discount": "0.00",
        "extra": "560.00",
        "full": "6160.00",
        "id": "5d7135113f18da51a186ad16",
        "nominal": "5600.00",
        "number": 168475,
        "price": "5600.00",
        "serial": "PYX",
        "set": "5d7135112110d30a34e97e2d",
        "status": "reserved"
      }
    ]
  }
},
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "values": {
      "discount": "0.00",
      "extra": "560.00",
      "full": "6160.00",
      "nominal": "5600.00",
      "price": "5600.00",
      "sets_values": {
        "5d713505255895db3c30b0c5": {
          "discount": "0.00",
          "id": "5d713505255895db3c30b0c5",
          "nominal": "6666.00",
          "price": "6666.00",
          "promocode": null
        },
        ...
      },
      "viral_promocodes": []
    },
    "vendor": "5ba10ea90c43fc000b0fc786",
  },
  "refs": {
    "events": {
      "5d7134962110d30a34e95b96": {
        "id": "5d7134962110d30a34e95b96",
        "lifetime": {
          "finish": "2020-06-12 18:00:00",
          "start": "2020-06-12 15:00:00"
        },
        "org": "5ba10ea90c43fc000b0fc786",
        "status": "public",
        "timezone": "Europe/Moscow",
        "title": {
          "desc": "ref",
          "text": "Slipknot"
        }
      }
    },
    "partners": {
      "5ba10ea90c43fc000b0fc786": {
        "id": "5ba10ea90c43fc000b0fc786",
        "name": "Тест VK Pay"
      }
    },
    "promocodes": {},
    "sets": {
      "5d7135112110d30a34e97e2d": {
        "id": "5d7135112110d30a34e97e2d",
        "name": "Фан зона",
        "price": "5600.00",
        "with_seats": false
      }
    }
  }
}

```

#### 1.1.5.4 За. Заполняем заказ (добавляем билет с местом)

Берем id добавляемого места из п.1, но при этом еще добавляем id билета, полученного в ответе из п.2а

```
PATCH /v2/resources/orders/5d8b924971a0bf323bd6a6ed HTTP/1.1
Authorization: key 9bd8359943b545500278875r49c5b96d
Content-Type: application/json

{
  "tickets": [
    "5d7135113f18da51a186ad16",
    "5d7134962110d30a34e95cfe"
  ]
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "data": {
    "created_at": "2019-09-25 16:14:01",
    "event": "5d7134962110d30a34e95b96",
    "expired_after": "2019-09-25 16:29:01",
    "id": "5d8b924971a0bf323bd6a6ed",
    "number": 59743,
    "org": "5ba10ea90c43fc000b0fc786",
    "origin": "api",
    "status": "executed",
    "tickets": [
      {
        "barcode": null,
        "discount": "0.00",
        "extra": "560.00",
        "full": "6160.00",
        "id": "5d7135113f18da51a186ad16",
        "nominal": "5600.00",
        "number": 168475,
        "price": "5600.00",
        "serial": "PYX",
        "set": "5d7135112110d30a34e97e2d",
        "status": "reserved"
      },
      {
        "barcode": null,
        "discount": "0.00",
        "extra": "99.00",
        "full": "1089.00",
        "id": "5d7134962110d30a34e95cfe",
        "nominal": "990.00",
        "number": 135830,
        "price": "990.00",
        "seat": {
          "number": "14",
          "row": "2",
          "sector": "5a8dd58e6e55b2064c67c144"
        }
      }
    ]
  }
}
```

(continues on next page)

(продолжение с предыдущей страницы)

```

        "serial": "EOY",
        "set": "5d71353b2110d30a34e97e31",
        "status": "reserved"
    }
],
"values": {
    "discount": "0.00",
    "extra": "659.00",
    "full": "7249.00",
    "nominal": "6590.00",
    "price": "6590.00",
    "sets_values": {
        "5d713505255895db3c30b0c5": {
            "discount": "0.00",
            "id": "5d713505255895db3c30b0c5",
            "nominal": "6666.00",
            "price": "6666.00",
            "promocode": null
        },
        ...
    },
    "viral_promocodes": []
},
"vendor": "5ba10ea90c43fc000b0fc786",
},
"refs": {
    "events": {
        "5d7134962110d30a34e95b96": {
            "id": "5d7134962110d30a34e95b96",
            "lifetime": {
                "finish": "2020-06-12 18:00:00",
                "start": "2020-06-12 15:00:00"
            },
            "org": "5ba10ea90c43fc000b0fc786",
            "status": "public",
            "timezone": "Europe/Moscow",
            "title": {
                "desc": "ref",
                "text": "Slipknot"
            }
        }
    },
    "partners": {
        "5ba10ea90c43fc000b0fc786": {
            "id": "5ba10ea90c43fc000b0fc786",
            "name": "Тест VK Pay"
        }
    },
    "promocodes": {},
    "sets": {
        "5d7135112110d30a34e97e2d": {
            "id": "5d7135112110d30a34e97e2d",
            "name": "Фан зона",
            "price": "5600.00",
            "with_seats": false
        },
    },

```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "5d71353b2110d30a34e97e31": {
      "id": "5d71353b2110d30a34e97e31",
      "name": "A2",
      "price": "990.00",
      "with_seats": true
    }
  }
}

```

### 1.1.5.5 2б. Создаем заказ (добавляем билет с местом)

Берем id добавляемого места из п.1

```

POST /v2/resources/orders/5d8b924971a0bf323bd6a6ed HTTP/1.1
Authorization: key 9bd8359943b545500278875r49c5b96d
Content-Type: application/json

{
  "tickets": [
    "5d7134962110d30a34e95cfe"
  ]
}

```

```

HTTP/1.1 200 OK
Content-Type: application/json

{
  "data": {
    "created_at": "2019-09-25 16:14:01",
    "event": "5d7134962110d30a34e95b96",
    "expired_after": "2019-09-25 16:29:01",
    "id": "5d8b924971a0bf323bd6a6ed",
    "number": 59743,
    "org": "5ba10ea90c43fc000b0fc786",
    "origin": "api",
    "status": "executed",
    "tickets": [
      {
        "barcode": null,
        "discount": "0.00",
        "extra": "99.00",
        "full": "1089.00",
        "id": "5d7134962110d30a34e95cfe",
        "nominal": "990.00",
        "number": 135830,
        "price": "990.00",
        "seat": {
          "number": "14",
          "row": "2",
          "sector": "5a8dd58e6e55b2064c67c144"
        },
        "serial": "EOY",
        "set": "5d71353b2110d30a34e97e31",
        "status": "reserved"
      }
    ]
  }
}

```

(continues on next page)



(продолжение с предыдущей страницы)

```

    }
  ],
  "values": {
    "discount": "0.00",
    "extra": "659.00",
    "full": "7249.00",
    "nominal": "6590.00",
    "price": "6590.00",
    "sets_values": {
      "5d713505255895db3c30b0c5": {
        "discount": "0.00",
        "id": "5d713505255895db3c30b0c5",
        "nominal": "6666.00",
        "price": "6666.00",
        "promocode": null
      },
      ...
    },
    "viral_promocodes": []
  },
  "vendor": "5ba10ea90c43fc000b0fc786",
},
"refs": {
  "events": {
    "5d7134962110d30a34e95b96": {
      "id": "5d7134962110d30a34e95b96",
      "lifetime": {
        "finish": "2020-06-12 18:00:00",
        "start": "2020-06-12 15:00:00"
      },
      "org": "5ba10ea90c43fc000b0fc786",
      "status": "public",
      "timezone": "Europe/Moscow",
      "title": {
        "desc": "ref",
        "text": "Slipknot"
      }
    }
  },
},
"partners": {
  "5ba10ea90c43fc000b0fc786": {
    "id": "5ba10ea90c43fc000b0fc786",
    "name": "Тект VK Pay"
  }
},
"promocodes": {},
"sets": {
  "5d71353b2110d30a34e97e31": {
    "id": "5d71353b2110d30a34e97e31",
    "name": "A2",
    "price": "990.00",
    "with_seats": true
  }
}
}
}

```

### 1.1.5.6 3b. Заполняем заказ (добавляем билет без места)

Берем id добавляемого сета из п.2b (или из п.0), но при этом еще дописываем id сета от добавленного билета из п.2

```
POST /v2/resources/orders HTTP/1.1
Authorization: key 9bd8359943b545500278875r49c5b96d
Content-Type: application/json

{
  "random": {
    "5d71353b2110d30a34e97e31": 1,
    "5d7135112110d30a34e97e2d": 1
  }
}
```

### 1.1.5.7 4. Завершаем заказ

```
PATCH /v2/resources/orders/5d8b924971a0bf323bd6a6ed HTTP/1.1
Authorization: key 9bd8359943b545500278875r49c5b96d
Content-Type: application/json

{
  "status": "done"
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "data": {
    "code": "lw4bb10o",
    "created_at": "2019-09-25 16:14:01",
    "done_at": "2019-09-25 16:20:40",
    "event": "5d7134962110d30a34e95b96",
    "expired_after": "2019-09-25 16:29:01",
    "id": "5d8b924971a0bf323bd6a6ed",
    "number": 59743,
    "org": "5ba10ea90c43fc000b0fc786",
    "origin": "api",
    "status": "done",
    "tickets": [
      {
        "barcode": "67454655075047921",
        "discount": "0.00",
        "extra": "560.00",
        "full": "6160.00",
        "id": "5d7135113f18da51a186ad16",
        "nominal": "5600.00",
        "number": 168475,
        "price": "5600.00",
        "serial": "PYX",
        "set": "5d7135112110d30a34e97e2d",
        "status": "reserved"
      },
    ]
  }
}
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    {
      "barcode": "35348364979141729",
      "discount": "0.00",
      "extra": "99.00",
      "full": "1089.00",
      "id": "5d7134962110d30a34e95cfe",
      "nominal": "990.00",
      "number": 135830,
      "price": "990.00",
      "seat": {
        "number": "14",
        "row": "2",
        "sector": "5a8dd58e6e55b2064c67c144"
      },
      "serial": "EOY",
      "set": "5d71353b2110d30a34e97e31",
      "status": "reserved"
    },
    ],
    "values": {
      "discount": "0.00",
      "extra": "659.00",
      "full": "7249.00",
      "nominal": "6590.00",
      "price": "6590.00",
      "sets_values": {
        "5d713505255895db3c30b0c5": {
          "discount": "0.00",
          "id": "5d713505255895db3c30b0c5",
          "nominal": "6666.00",
          "price": "6666.00",
          "promocode": null
        },
        ...
      },
      "viral_promocodes": [],
    },
    "vendor": "5ba10ea90c43fc000b0fc786",
  },
  "refs": {
    "events": {
      "5d7134962110d30a34e95b96": {
        "id": "5d7134962110d30a34e95b96",
        "lifetime": {
          "finish": "2020-06-12 18:00:00",
          "start": "2020-06-12 15:00:00"
        },
        "org": "5ba10ea90c43fc000b0fc786",
        "status": "public",
        "timezone": "Europe/Moscow",
        "title": {
          "desc": "ref",
          "text": "Slipknot"
        }
      },
    }
  },
},

```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "partners": {
      "5ba10ea90c43fc000b0fc786": {
        "id": "5ba10ea90c43fc000b0fc786",
        "name": "Тест VK Pay"
      }
    },
    "promocodes": {},
    "sets": {
      "5d7135112110d30a34e97e2d": {
        "id": "5d7135112110d30a34e97e2d",
        "name": "Фан зона",
        "price": "5600.00",
        "with_seats": false
      },
      "5d71353b2110d30a34e97e31": {
        "id": "5d71353b2110d30a34e97e31",
        "name": "A2",
        "price": "990.00",
        "with_seats": true
      }
    }
  }
}
```

**Возможные ошибки при работе с заказом** При ошибках возвращается список текстовых сообщений:

```

HTTP/1.1 400
Content-Type: application/json

{
  "errors": [
    "Event <event_id> not found"
  ]
}
```

Сообщение	Причина
Event <event_id> not found	не найдено мероприятие для заказа
Deal not found	не найдена сделка
Promokey needed	промоключ должен быть
Promokey not founded	промоключ не найден
Promokey expired	промоключ истек
Promokey fully reserved	промоключ полностью зарезервирован
Promokey already reserved	промоключ уже зарезервирован
Only one of tickets or random can be set	только один: tickets или random
„tickets“ or „random“ must be	обязан быть: tickets или random
Not allow to update order in status <status>	нельзя менять заказ в статусе <status>
Allow update the only status to cancelled	разрешено только отменить заказ
expired_after is not changable	время жизни не может быть изменено
max expired_after is {dt}	время жизни истекло <dt :%Y-%m-%d %H:%M:%S>
only org can send invitations	только организатор может отправить приглашения
invitation can send from control panel or salespoint only	приглашения можно отправить только из панели управления или точки продаж
Courier must set only with salespoint	запрос должен быть только из точки продаж

Таблица 1 – продолжение с предыдущей строки

Сообщение	Причина
Kryptonite integrations is disabled	интеграция с криптонитом выключена
Cant set smart_tickets to false while kryptonite integration option is only smart_tickets	нельзя выключить «криптобилеты»
Cant use promocode when event have promotion	нельзя использовать промокод
expired_after is not changable	время жизни не может быть изменено
max expired_after is {dt}	время жизни истекло <dt :%Y-%m-%d>
Promokey needed to add these tickets	необходим промоключ, чтобы добавить билеты
ticket (id = {id}) does not belong to current event	билет не относится к текущему событию
value should be formatted „YYYY-MM-DD“	для дополнительного поля «Дата»
value should be True or False	для дополнительного поля «Готовность»
value is not a list	для дополнительного поля «Билеты»
list length is less than <len>	для дополнительного поля «Билеты»
list length is greater than <len>	для дополнительного поля «Билеты»
value is less than <value>	для дополнительного поля «Билеты»
value is greater than <value>	для дополнительного поля «Билеты»
value should be less than <value>	для дополнительного поля «Билеты»
value should be greater than <value>	для дополнительного поля «Билеты»
value is not a string	для дополнительного поля «Текст»
blank value is not allowed	для дополнительного поля «Текст»
String is shorter than <len> characters	для дополнительного поля «Текст»
String is longer than <len> characters	для дополнительного поля «Текст»
does not match pattern <pattern>	для дополнительного поля «Текст»
Not enough money, contact to TicketsCloud manager	недостаточно денег, свяжитесь с менеджером
there is no tickets in order	в заказе нет ни одного билета
Incorrect status <status> for this operation	нельзя выполнить перевод заказа в статус <status>
Order <order_id> is not cancellable	заказ с id: <order_id> не может быть отменен

## 1.1.6 Шаг 4: Возврат заказа

### 1.1.6.1 Создание возврата

#### Запрос

POST /v2/resources/refund\_requests/

#### JSON Parameters

- **order** – Id заказа
- **culprit** – Виновник возврата (**user** | **org**)
- **tickets** – Список id билетов для возврата
- **requested\_at** – Дата и время заявки на возврат. Необязательный параметр.

#### Ответ

Объект возврата

```
{
  "data": {
    "id": objectid,
    "status": "new" | "in_progress" | "approved" | "rejected",
    "culprit": "user" | "org",
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "created_at": ISODatetime,
    "finished_at": ISODatetime,
    "requested_at": ISODatetime,
    "refund_nominal": money // сумма возврата
    "delta": money // Устарело. Необходимо использовать `refund_nominal`,
    "event": objectid,
    "org": objectid,
    "order": objectid,
    "policy": "general" | "law_ru_193", // политика возврата.
                                         // general - стандартные условия.
                                         // law_ru_193 - применяются условия по ФЗ-
↪193
    "tickets": Array[objectid],
    "vendor": objectid
  }
}

```

### Пример запроса

```

POST /v2/resources/refund_requests HTTP/1.1
Authorization: key 9bd8359943b545500278875r49c5b88d
Content-Type: application/json

{
  "culprit": "user",
  "order": "5def7c43e3ed5ae2a8d6908b",
  "requested_at": "2019-12-15T13:01:02",
  "tickets": [
    "5dd5469b05fe4df8940eb613"
  ]
}

```

### Пример ответа

```

HTTP/1.1 200 OK
Content-Type: application/json

{
  "data": {
    "created_at": "2019-12-18 12:46:28",
    "culprit": "user",
    "customer_money": "1000.00",
    "delta": "1000.00",
    "event": "5dd5468f05fe4df8940eb602",
    "id": "5dfa1fa437077b6b6ed0f6b3",
    "order": "5def7c43e3ed5ae2a8d6908b",
    "org": "5b2d06ac5c3eb3000c475e36",
    "policy": "law_ru_193",
    "refund_nominal": "1000.00",
    "requested_at": "2019-12-15 13:01:02",
    "status": "new",
    "tickets": [
      "5dd5469b05fe4df8940eb613"
    ],
    "vendor": "5b2d06ac5c3eb3000c475e36"
  },
}

```

(continues on next page)

(продолжение с предыдущей страницы)

```

"refs": {
  "events": {
    "5dd5468f05fe4df8940eb602": {
      "age_rating": 0,
      "id": "5dd5468f05fe4df8940eb602",
      "lifetime": {
        "finish": "2020-01-01 20:00:00",
        "start": "2020-01-01 17:00:00"
      },
      "org": "5b2d06ac5c3eb3000c475e36",
      "status": "public",
      "timezone": "Europe/Moscow",
      "title": {
        "desc": "Metal music for all",
        "text": "MetalGrid"
      },
      "venue": "583d9307515e350019da3ef6"
    }
  },
  "orders": {
    "5def7c43e3ed5ae2a8d6908b": {
      "code": "k0xvxvel",
      "created_at": "2019-12-10 11:06:43",
      "custom_fields": {
        "order": [],
        "tickets": []
      },
      "done_at": "2019-12-10 11:11:54",
      "event": "5dd5468f05fe4df8940eb602",
      "expired_after": "2019-12-10 11:21:43",
      "id": "5def7c43e3ed5ae2a8d6908b",
      "number": 121,
      "org": "5b2d06ac5c3eb3000c475e36",
      "origin": "api",
      "payments": [],
      "promocodes": [],
      "settings": {
        "invitation": false,
        "send_tickets": false,
        "subscribe_agree": false
      },
      "status": "done",
      "tickets": [
        {
          "barcode": "61534165115017991",
          "discount": "0.00",
          "extra": "0.00",
          "full": "1000.00",
          "id": "5dd5469b05fe4df8940eb613",
          "nominal": "1000.00",
          "number": 324762,
          "price": "1000.00",
          "serial": "APP",
          "set": "5dd5469b05fe4df8940eb606",
          "status": "reserved"
        }
      ]
    }
  }
}

```

(continues on next page)

(продолжение с предыдущей страницы)

```
    ],
    "values": {
      "discount": "0.00",
      "extra": "0.00",
      "full": "1000.00",
      "nominal": "1000.00",
      "price": "1000.00",
      "sets_values": {
        "5dd5469b05fe4df8940eb606": {
          "discount": "0.00",
          "id": "5dd5469b05fe4df8940eb606",
          "nominal": "1000.00",
          "price": "1000.00",
          "promocode": null
        }
      },
      "viral_promocodes": []
    },
    "vendor": "5b2d06ac5c3eb3000c475e36",
    "vendor_data": {}
  },
  "partners": {
    "5b2d06ac5c3eb3000c475e36": {
      "id": "5b2d06ac5c3eb3000c475e36",
      "name": "Funky Box"
    }
  },
  "tickets": {
    "5dd5469b05fe4df8940eb613": {
      "discount": "0.00",
      "extra": "0.00",
      "full": "1000.00",
      "id": "5dd5469b05fe4df8940eb613",
      "nominal": "1000.00",
      "number": 324762,
      "price": "1000.00",
      "serial": "APP",
      "set": "5dd5469b05fe4df8940eb606",
      "status": "reserved"
    }
  }
}
```



### 1.1.6.2 Ошибки при создании возврата

Сообщение	Причина
<field> is required	не указано поле <field>
culprit Value „<value>“ not in list of available values [„user“, „org“]	неверно указан виновник возврата
Order <order_id> not found	не найден заказ
Operation failed: less than three days before event start	меньше 3 дней до начала, по ФЗ-193 возврат невозможен
Operation failed: event already start	мероприятие началось, по ФЗ-193 возврат невозможен
Operation failed: there is discounted ticket	билет со скидкой, возврат невозможен
Operation failed: Mixed percentage values of tickets in one refund	билеты с разными скидками, возврат невозможен
bad culprit: <value>	неверно указан виновник возврата
wrong order (id: <order_id>) status: <order_status>	заказ не в статусе done
Refund with culprit core must take all tickets from order	в возврате не все билеты
Value Error	возврат не создан, точная причина неизвестна
bad tickets	указаны билеты не из заказа
Unexpected situation	ошибка при создании платежей
RefundRequest in wrong status <refund_status> for create refund payment	неверный статус возврата при создании платежей

#### Пример ошибки

```
{
  "errors": [
    "culprit Value 'idunno' not in list of available values ['user', 'org']",
    "tickets list length is less than 1"
  ]
}
```

### 1.1.6.3 Подтверждение и отмена возврата

#### Запрос

PATCH /v2/resources/refund\_requests/:refund\_id

#### JSON Parameters

- status – approved | rejected

#### Ответ

*Объект возврата*

#### Пример запроса

```
PATCH /v2/resources/refund_requests/5dfa1fa437077b6b6ed0f6b3 HTTP/1.1
Authorization: key 9bd8359943b545500278875r49c5b88d
Content-Type: application/json

{
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "status": "approved"
  }

```

При ошибках возвращается список текстовых сообщений:

```

{
  "errors": [
    "Incorrect status <status> for this operation"
  ]
}

```

#### 1.1.6.4 Ошибки при подтверждении возврата

Сообщение	Причина
Not allowed to approve refund for finished event	запрещено подтверждать возврат после завершения мероприятия
Only org culprit allowed for cancelled events	только организатор может подтвердить возврат для отмененного мероприятия
Only user culprit allowed for non cancelled events	только пользователь может подтвердить возврат для не отмененного мероприятия
RefundRequest in wrong status <status> for approve	возврат находится в неверном статусе для подтверждения, возможно он уже подтвержден или отменен
Some tickets already use in another refund request	некоторые билеты уже используются в другом возврате и не могут быть в этом
Refund with culprit core must take all tickets from order	билеты должны быть из одного заказа
Sums of payments not equal refund request	сумма возврата билетов не совпадает с суммой, которая была за них уплачена
Not enough money, contact to TicketsCloud manager	недостаточно денег для проведения возврата, для решения нужно связаться с менеджером TicketsCloud

#### 1.1.6.5 Ошибки при отмене возврата

Сообщение	Причина
Cant reject refund request <id refund>, one of payments in status done	нельзя отменить возврат, потому что один из платежей уже в статусе <i>done</i>
Cant reject refund request <id refund>, one of payments in status in_progress	нельзя отменить возврат, потому что один из платежей уже в статусе <i>in_progress</i>
Failed to reject refund request <id refund>	Нельзя отменить возврат, возврат сменил статус с new/in_progress на другой

## 1.2 Продвинутое использование

### 1.2.1 Жизненный цикл заказа

#### 1.2.1.1 Список состояний (статусов) заказа без платежей

## executed

Заказ всегда создаётся в статусе `executed`. Через 10 минут, заказ в этом состоянии переходит в *expired*

С заказом в состоянии `executed` возможны следующие *действия*:

- *забронировать билеты*
- *добавить информацию о покупателе*
- *изменить параметр `send_tickets`*
- *добавить произвольные поля в заказ*
- *отменить*
- *перевести в статус `done`*

## done

Заказ в этом статусе уже оплачен.

## cancelled

Заказ отменён.

## expired

Заказ просрочен, забронированные билеты снова вернулись в продажу.

### 1.2.2 Действия с заказом

---

**Примечание:** Все действия с заказом, кроме его создания, делаются по запросу *PATCH /v2/resources/orders/:id*. В один запрос одновременно может быть добавлено несколько действий. Все запросы на изменение конкретного заказа, должны делаться синхронно. В случае получения запроса до конца обработки предыдущего, будет возвращена ошибка *409 Conflict*.

Это ограничение касается только работы с одним заказом, а работать одновременно с несколькими заказами можно.

В ответ на все *PATCH* запросы приходит обновленный *объект заказа*

---

#### 1.2.2.1 Добавить информацию о покупателе

##### Запрос

POST /v2/resources/orders/:id

##### JSON Parameters

- **settings** – объект с одним ключом `customer` и соответствующим ему значением, содержащим:
  - **name** (str)

- **email** (str)
- **phone** (str)
- **lang** (en|ru)

**Ответ**

```
HTTP/1.1 200 OK
Content-Type: application/json
```

*Объект заказа*

**Пример запроса**

```
PATCH /v2/resources/orders/5bacf64ea0eb2f000c45160a HTTP/1.1
Authorization: key 9bd8359943b545500278875r49c5b96d
Content-Type: application/json

{
  "settings": {
    "customer": {
      "email": "hello@world.ru",
      "lang": "ru",
      "name": "Ivan Ivanov",
      "phone": "+79991234576"
    }
  }
}
```

**Пример ответа**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "data": {
    "created_at": "2018-09-27 15:25:02",
    "event": "5b23b53b9c9b19000c6c4180",
    "expired_after": "2018-11-16 00:00:00",
    "id": "5bacf64ea0eb2f000c45160a",
    "number": 41564,
    "org": "5b0286ce517565000d9cb1ca",
    "origin": "api",
    "promocodes": [],
    "settings": {
      "customer": {
        "email": "hello@world.ru",
        "lang": "ru",
        "name": "Ivan Ivanov",
        "phone": "+79991234576"
      },
      "invitation": false,
      "send_tickets": true
    },
    "status": "executed",
    "tickets": [
      ...
    ],
  },
}
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "values": {
      ...
    },
    "vendor": "5b0286ce517565000d9cb1ca",
    "vendor_data": {}
  },
  "refs": {
    ...
  }
}

```

### 1.2.2.2 Добавить произвольную информацию в объект заказа

#### Запрос

POST /v2/resources/orders/:id

#### JSON Parameters

- **vendor\_data** –
  - **order\_id** (str) (Необязательно) Номер заказа в системе распространителя. Максимальная длина 64 символа
  - **raw** (object) Объект с произвольными полями.

#### Предупреждение:

Максимальное кол-во ключей в поле **raw** - 20

Максимальная длина ключа - 40 символов

Все значения - String с максимальной длиной 128 символов.

#### Ответ

```

HTTP/1.1 200 OK
Content-Type: application/json

```

*Объект заказа*

#### Пример запроса

```

PATCH /v2/resources/orders/5bacf64ea0eb2f000c45160a HTTP/1.1
Authorization: key 9bd8359943b545500278875r49c5b96d
Content-Type: application/json

{
  "vendor_data": {
    "order_id": "5bacf64ea0eb2f000c45160a",
    "raw": {
      "enable_call_to_customer": true,
      "call_counter": 3
    }
  }
}

```

#### Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "data": {
    "created_at": "2018-09-27 15:25:02",
    "event": "5b23b53b9c9b19000c6c4180",
    "expired_after": "2018-11-16 00:00:00",
    "id": "5bacf64ea0eb2f000c45160a",
    "number": 41564,
    "org": "5b0286ce517565000d9cb1ca",
    "origin": "api",
    "promocodes": [],
    "settings": {
      ...
    },
    "status": "executed",
    "tickets": [
      ...
    ],
    "values": {
      ...
    },
    "vendor": "5b0286ce517565000d9cb1ca",
    "vendor_data": {
      "order_id": "5bacf64ea0eb2f000c45160a",
      "raw": {
        "enable_call_to_customer": true,
        "call_counter": 3
      }
    }
  },
  "refs": {
    ...
  }
}
```

### 1.2.2.3 Отправка билетов покупателю на email

При значении `true` билеты отправляет платформа ticketscloud на `email`, указанный в поле `customer`. По умолчанию `false`, тогда билеты генерируете и отправляете вы. Обязательное условие — наличие баркода.

Изменить `email` можно только при статусе заказа `executed`, для этого надо обновить данные `customer`, точно так же как и при первичном добавлении.

#### Запрос

POST /v2/resources/orders/:id

#### JSON Parameters

- settings –
  - send\_tickets (bool)

#### Ответ

```
HTTP/1.1 200 OK
Content-Type: application/json
```

*Объект заказа*

### Пример запроса

```
PATCH /v2/resources/orders/5bacf64ea0eb2f000c45160a HTTP/1.1
Authorization: key 9bd8359943b545500278875r49c5b96d
Content-Type: application/json

{
  "settings": {
    "send_tickets": true
  }
}
```

### Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "data": {
    "created_at": "2018-09-27 15:25:02",
    "event": "5b23b53b9c9b19000c6c4180",
    "expired_after": "2018-11-16 00:00:00",
    "id": "5bacf64ea0eb2f000c45160a",
    "number": 41564,
    "org": "5b0286ce517565000d9cb1ca",
    "origin": "api",
    "promocodes": [],
    "settings": {
      "send_tickets": true,
      ...
    },
    "status": "executed",
    "tickets": [
      ...
    ],
    "values": {
      ...
    },
    "vendor": "5b0286ce517565000d9cb1ca",
    "vendor_data": {
      ...
    }
  },
  "refs": {
    ...
  }
}
```

**Примечание:** У нас нет требований к дизайну билета, кроме наличия обязательной информации на нём.

**Билет должен содержать:**

- баркод
  - серия
  - номер
  - категория
  - ряд-место (если с рассадкой)
  - цена
  - площадка (название-адрес)
  - дата-время начала в часовом поясе площадки
  - название мероприятия
  - наименование юридического лица, инн, огрн, адрес (если не ИП)
- 

#### 1.2.2.4 Добавить промокоды к заказу

Регистр применяемых к заказу промокодов не важен

##### Запрос

PATCH /v2/resources/orders/:id

##### JSON Parameters

- promocodes – (list of string)

##### Ответ

```
HTTP/1.1 200 OK
Content-Type: application/json
```

*Объект заказа*

##### Пример запроса

```
PATCH /v2/resources/orders/5bacf64ea0eb2f000c45160a HTTP/1.1
Authorization: key 9bd8359943b545500278875r49c5b96d
Content-Type: application/json

{
  "promocodes": [
    "PROMO100"
  ]
}
```

##### Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "data": {
    "created_at": "2018-09-27 15:25:02",
    "event": "5b23b53b9c9b19000c6c4180",
    "expired_after": "2018-11-16 00:00:00",
```

(continues on next page)



(продолжение с предыдущей страницы)

```

    "id": "5bacf64ea0eb2f000c45160a",
    "number": 41564,
    "org": "5b0286ce517565000d9cb1ca",
    "origin": "api",
    "promocodes": [
      "5d664d392a4191909a568b40"
    ],
    "status": "executed",
    "tickets": [
      {
        "barcode": null,
        "discount": "100.00",
        "extra": "46.12",
        "full": "946.12",
        "id": "5d664830b8a2cb5ce0576457",
        "nominal": "900.00",
        "number": 142278,
        "price": "1000.00",
        "serial": "BKC",
        "set": "5d6648306b23e8a5f90ef047",
        "status": "reserved"
      }
    ],
    "values": {
      "discount": "100.00",
      "extra": "46.12",
      "full": "946.12",
      "nominal": "900.00",
      "price": "1000.00",
      "sets_values": {
        "5d6648306b23e8a5f90ef047": {
          "discount": "100.00",
          "id": "5d6648306b23e8a5f90ef047",
          "nominal": "900.00",
          "price": "1000.00",
          "promocode": "5d664d392a4191909a568b40"
        },
        "5d664830e4c685ae48bd2d3d": {
          "discount": "100.00",
          "id": "5d664830e4c685ae48bd2d3d",
          "nominal": "400.00",
          "price": "500.00",
          "promocode": "5d664d392a4191909a568b40"
        }
      },
      "viral_promocodes": []
    },
    "vendor": "5b0286ce517565000d9cb1ca",
    "vendor_data": {
      ...
    }
  },
  "refs": {
    "promocodes": {
      "5d664d392a4191909a568b40": {
        "code": "promo100",

```

(continues on next page)

(продолжение с предыдущей страницы)

```
        "discount": {
            "fix": "100.00"
        },
        "id": "5d664d392a4191909a568b40",
        "viral": false
    }
},
"sets": {
    "5d6648306b23e8a5f90ef047": {
        "id": "5d6648306b23e8a5f90ef047",
        "name": "Папреп",
        "price": "1000.00",
        "with_seats": false
    }
},
...
}
}
```

Промокод может быть как на конкретную сумму, так и на процент:

```
{
    ...
    "promocodes": {
        "5d66b9f9a80b147cadf87583": {
            "id": "5d66b9f9a80b147cadf87583",
            "code": "promo5%",
            "discount": {
                "percentage": "5%"
            },
            "viral": false
        }
    }
}
```

Ошибки при работе с промокодами

code	msg
promocode_not_found	Promocode not found
promocode_already_used	Promocode already used
promocode_limit_min_tickets	Need more tickets in order for promocode activation / Minimum count of tickets in order for promocode activation is {}
promocode_apply_info	Can't apply promocode by some reason

**Предупреждение:** Успешность применения промокода не влияет на код ответа

Пример ошибки

```
{
    "data": {
        ...
    }
}
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    },
    "refs": {
        ...
    },
    "errors": [
        {
            "code": "promocode_not_found",
            "msg": "Promocode not found"
        }
    ],
}

```

## 1.2.3 Промокоды

### 1.2.3.1 Проверка промокода

Существует возможность проверить актуальность промокода для мероприятия. В случае если промокод невозможно применить возвращается ответ BadRequest с кодом ответа 400. В остальных случаях возвращаются параметры промокода и ограничения на его использование.

#### Запрос

POST /v2/services/promocodes/check

#### JSON Parameters

- event – id мероприятия (str)
- code – код (str)

#### Ответ

```

HTTP/1.1 200 OK
Content-Type: application/json

{
  "data": {
    "discount": {
      "percentage": str // Размер скидки промокода, в процентах
      "fix": str // или фиксированной величины. Одновременно
                  // может быть только один из параметров.
    },
    "lifetime": {
      "finish": iso8601 // Ограничение по времени действия промокода.
      "start": iso8601 // Необязательное поле.
    },
    "sets": [oid, ... ], // Ограничение по категориям билетов на которые
                        // действует скидка. Пустой массив означает
                        // отсутствие ограничений.

    "tickets_count": {
      "min": int // Ограничение по количеству билетов в заказе,
                 // на которое действует скидка.
                 // min - минимальное количество билетов в заказе,
                 // после которого начинает начисляться скидка.
                 // Необязательное поле.
      "max": int // max - максимальное количество билетов в заказе,
                 // после которого скидка больше не начисляется.
    }
  }
}

```

(continues on next page)

(продолжение с предыдущей страницы)

```

// Необязательное поле.
    }
  },
  "refs": {
    "sets": {
      oid: {
        "id": oid    // id категории
        "name": str   // название категории
        "price": str  // цена категории (оригинальная, без скидки)
      }
      ...
    }
  }
}

```

## Примеры

### Запрос

```

POST /v2/services/promocodes/check HTTP/1.1
Authorization: key 9bd8359943b545500278875r49c5b96d
Content-Type: application/json

{
  "code": "fix100",
  "event": "5db2d8504d9134a8c7ec2cf5"
}

```

### Ответ

```

HTTP/1.1 200 OK
Content-Type: application/json

{
  "data": {
    "discount": {
      "fix": "100.00"
    },
    "sets": [],
    "tickets_count": {}
  },
  "refs": {
    "sets": {}
  }
}

```

### Запрос

```

POST /v2/services/promocodes/check HTTP/1.1
Authorization: key 9bd8359943b545500278875r49c5b96d
Content-Type: application/json

{
  "code": "all",

```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "event": "5d765a4a221988d7da985875"
  }

```

**Ответ**

```

HTTP/1.1 200 OK
Content-Type: application/json

{
  "data": {
    "discount": {
      "percentage": "25%"
    },
    "lifetime": {
      "finish": "2019-11-30 20:59:00",
      "start": "2019-10-30 21:00:00"
    },
    "sets": [
      "5d765a59221988d7da985879"
    ],
    "tickets_count": {
      "min": 2
    }
  },
  "refs": {
    "sets": {
      "5d765a59221988d7da985879": {
        "id": "5d765a59221988d7da985879",
        "name": "Билетище",
        "price": "1099.00"
      }
    }
  }
}

```

**Запрос**

```

POST /v2/services/promocodes/check HTTP/1.1
Authorization: key 9bd8359943b545500278875r49c5b96d
Content-Type: application/json

{
  "code": "not_exists",
  "event": "5d765a4a221988d7da985875"
}

```

**Ответ**

```

HTTP/1.1 400 Bad Request
Content-Type: application/json

{
  "error": "HTTPBadRequest",
  "reason": "Promocode not found"
}

```

## 1.2.4 Работа со списком заказов

### 1.2.4.1 Получение списка заказов

#### Запрос

GET /v2/resources/orders

#### Query Parameters

- **created\_at** – Дата создания заказа в формате ISO от и до, через запятую
- **status** – Список статусов заказа
- **events** – Список мероприятий
- **only\_with\_customer** – bool; Если true – Показываем заказы, только если есть данные покупателя
- **page** – Порядковый номер страницы результатов запроса (по умолчанию – первая)
- **page\_size** – Кол-во записей на страницу результатов запроса (минимум 1, максимум 200, по умолчанию 50)

#### Ответ

```
HTTP/1.1 200 OK
Content-Type: application/json
```

Список *объектов заказа*

#### Пример запроса

```
GET /v2/resources/orders?status=done,cancelled&created_at=2000-07-28T13:00:00,2020-07-
↪28T13:00:00 HTTP/1.1
Authorization: key 9bd8359943b545500278875r49c5b96d
```

#### Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "data": [
    {
      "code": "97elgi8j",
      "created_at": "2019-07-25 14:56:16",
      "done_at": "2019-07-25 14:57:47",
      "event": "5d31fbdd27649b0dff076117",
      "expired_after": "2019-07-25 15:11:16",
      "id": "5d39c3108cd381a4891e07e0",
      "number": 54881,
      "org": "5b04229196c055000d87c2b5",
      "origin": "api",
      "promocodes": [],
      "settings": {
        "invitation": false,
        "send_tickets": false,
        "subscribe_agree": false
      },
    },
  ],
}
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    "status": "done",
    "tickets": [
      {
        "barcode": "17174251182011730",
        "discount": "0.00",
        "extra": "6.00",
        "full": "156.00",
        "id": "5d31fbfb306fdcc187b91179",
        "nominal": "150.00",
        "number": 157495,
        "price": "150.00",
        "serial": "00X",
        "set": "5d31fbfa27649b0dff07611b",
        "status": "reserved"
      }
    ],
    "values": {
      "discount": "0.00",
      "extra": "6.00",
      "full": "156.00",
      "nominal": "150.00",
      "price": "150.00",
      "sets_values": {
        "5d31fbfa27649b0dff07611b": {
          "discount": "0.00",
          "id": "5d31fbfa27649b0dff07611b",
          "nominal": "150.00",
          "price": "150.00",
          "promocode": null
        },
        "5d31fbfa8a75c12c9d64de13": {
          "discount": "0.00",
          "id": "5d31fbfa8a75c12c9d64de13",
          "nominal": "1050.00",
          "price": "1050.00",
          "promocode": null
        }
      }
    },
    "viral_promocodes": [],
    "vendor": "5b02d6e9517565000d9cb1ce",
    "vendor_data": {}
  },
  ...
],
"pagination": {
  "page": 1,
  "page_size": 50,
  "total": 1
},
"refs": {
  "events": {
    "5c13b5b1867efb000be1ebd7": {
      "id": "5c13b5b1867efb000be1ebd7",
      "lifetime": {
        "finish": "2019-08-31 16:00:00",

```

(continues on next page)

(продолжение с предыдущей страницы)

```

        "start": "2019-08-31 15:00:00"
      },
      "org": "5ba10ea90c43fc000b0fc786",
      "status": "public",
      "timezone": "Europe/Moscow",
      "title": {
        "desc": "аун",
        "text": "КоЯн"
      }
    },
    ...
  },
  "partners": {
    "5b02d6e9517565000d9cb1ce": {
      "id": "5b02d6e9517565000d9cb1ce",
      "name": "Rasp new"
    },
    "5b04229196c055000d87c2b5": {
      "id": "5b04229196c055000d87c2b5",
      "name": "My best org"
    },
    "5ba10ea90c43fc000b0fc786": {
      "id": "5ba10ea90c43fc000b0fc786",
      "name": "Тест VK Pay"
    },
    "5bb333389049ea000d2ba9c7": {
      "id": "5bb333389049ea000d2ba9c7",
      "name": "Newbie"
    }
  }
},
}

```

#### 1.2.4.2 Экспорт списка заказов на почту

Описание параметров:

Запрос

GET /v2/resources/orders/export

##### Query Parameters

- **created\_at** – Дата создания заказа в формате ISO от и до, через запятую
- **status** – Список статусов заказа
- **events** – Список мероприятий
- **only\_with\_customer** – bool; Если true – Показываем заказы, только если есть данные покупателя
- **email** – Адрес, куда отправить файл экспорта. Обязательно

Ответ

```

HTTP/1.1 200 OK
Content-Length: 0
Content-Type: text/html; charset=UTF-8

```



## Пример запроса

```
POST /v2/resources/orders/export HTTP/1.1
Authorization: key 9bd8359943b545500278875r49c5b96d
Content-Type: application/json

{
  "created_at": "2000-07-28T13:00:00,2020-07-28T13:00:00",
  "email": "hello@world.hello",
  "status": [
    "done",
    "cancelled"
  ]
}
```

## 1.2.5 Работа со списком возвратов

### 1.2.5.1 Получаем список возвратов

#### Запрос

GET /v2/resources/refund\_requests

#### Query Parameters

- **created\_at** – Дата создания возврата, *isodatetime* промежуток через запятую
- **finished\_at** – Дата завершения возврата, *isodatetime* промежуток через запятую
- **events** – Фильтр по списку мероприятий
- **status** – Фильтр по списку статусов возвратов (new | in\_progress | approved | rejected)
- **page** – Порядковый номер страницы результатов запроса (по умолчанию – первая)
- **page\_size** – Кол-во записей на страницу результатов запроса (минимум 1, максимум 200, по умолчанию 50)

## Пример запроса

```
GET /v2/resources/refund_requests?created_at=2018-01-01T00:00:00,2019-07-01T00:00:00&
events=5b23b53b9c9b19000c6c4180 HTTP/1.1
Authorization: key 9bd8359943b545500278875r49c5b96d
```

## Пример ответа

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "data": [
    {
      "created_at": "2018-07-24 13:26:00",
      "culprit": "user",
      "delta": "13500.00",
      "event": "5b23b53b9c9b19000c6c4180",

```

(continues on next page)

(продолжение с предыдущей страницы)

```

        "refund_nominal" "13500.00",
        "finished_at": "2018-07-24 13:26:14",
        "id": "5b5728e886e7040069eb7f86",
        "order": "5b57281886e704000b424741",
        "org": "5b0286ce517565000d9cb1ca",
        "policy": "general",
        "status": "approved",
        "tickets": [
            ...
        ],
        "vendor": "5b0286ce517565000d9cb1ca"
    },
    ...
],
"pagination": {
    "page": 1,
    "page_size": 50,
    "total": 1
},
"refs": {
    "events": {
        ...
    },
    "orders": {
        ...
    },
    "partners": {
        ...
    },
    "tickets": {
        ...
    }
}
}

```

## 1.2.6 Повторяющиеся сущности

### 1.2.6.1 VEVENT

Поле типа vevent, это строка в формате VEVENT из [RFC 2445](#). В настоящий момент поддерживается только два значения: DTSTART и DTEND. Значения указываются только в UTC

Пример:

```

BEGIN:VEVENT\r\n
DTSTART;VALUE=DATE-TIME:20160124T160000Z\r\n
DTEND;VALUE=DATE-TIME:20160124T173000Z\r\n
END:VEVENT\r\n

```

### 1.2.6.2 Media

Различные медиа-данные имеют общий формат.

id

**author** id создателя

**content\_type** тип файла (например, «image/jpeg»)

**length** размер в байтах

**md5hash** хеш md5 от содержимого

**url** полный урл до файла

#### 1.2.6.3 ObjectId

Уникальный идентификатор в рамках сущности – строка из 24 символов ([a-zA-Z0-9])

#### 1.2.6.4 ISODatetime

Строка, содержащая дату-время в формате ISO8601



- `genindex`
- `search`



/v1

GET /v1/resources/events/:id/tickets, ??

/v2

GET /v2/resources/orders, 42

GET /v2/resources/orders/export, 44

GET /v2/resources/refund\_requests, 45

POST /v2/resources/orders, 2

POST /v2/resources/orders/:id, 34

POST /v2/resources/refund\_requests/, 25

POST /v2/services/promocodes/check, 39

PATCH /v2/resources/orders/:id, 12

PATCH /v2/resources/refund\_requests/:refund\_id,  
29





R

RFC

RFC 2445, [46](#)