

Описание API

Оглавление

Общая информация и возможности.....	4
Программный интерфейс	4
Организация заказов и операций	4
Статусы заказов	4
Статусы операций.....	5
Схемы взаимодействия	5
Оплата через платежную страницу.....	5
Сервер-сервер.....	6
Списание средств в бэкофисе.....	6
Работа через API.....	6
Аутентификация.....	6
HTTP-коды	7
Типы отказов	7
Обработка ошибок	7
Валидация	7
Пример кода обработки.....	8
Запросы	9
Тестовый запрос (GET /ping).....	9
Выборка данных.....	9
Информация о заказе (GET /orders/:id).....	10
Список заказов (GET /orders/).....	12
Список операций (GET /operations/).....	14
Экспорт курсов валют (GET /exchange_rates/)	15
Работа с заказами.....	16
Создание заказа (POST /orders/create)	16
Авторизация (POST /orders/authorize)	20
Отмена авторизации (PUT /orders/:id/reverse)	27
Списание средств (PUT /orders/:id/charge).....	29
Возврат средств (PUT /orders/:id/refund)	33
Отмена (PUT /orders/:id/cancel).....	38
Повторный платеж (POST /orders/:id/rebill)	40
Выплата на карту (OCT, Original credit transaction) (POST /orders/:id/credit).....	42
Завершение 3-D Secure аутентификации (POST /orders/:id/complete).....	44
Дополнительные возможности	45
Динамический дескриптор	45

Поддержка длинной записи (Airline Addendum).....	46
Уникальная конверсия платежей.....	46
DSRP-платежи.....	46
Привязка платежной карты (Oneclick).....	47
Фискализация по 54-ФЗ.....	47
Общая информация.....	47
Регистрация чека.....	48
Запрос реквизитов чека.....	51
Тестирование.....	53
Общая информация.....	53
Подключение к тестовой среде.....	53
Параметры подключения.....	53
Примеры запросов на РНР.....	54
Create.....	54
GetOrder.....	56
Refund.....	57
Rebill.....	59

Общая информация и возможности

Программный интерфейс построен по принципам REST, на основе спецификации JSON- API. API предоставляет единый интерфейс как для интеграции через платежную страницу, так и для работы по схеме сервер-сервер.

Возможности:

- JSON-API на основе стандарта RESTful JSON API.
- Дополнительная аутентификация на основе клиентских сертификатов.
- Одностадийная и двухстадийная схемы оплаты.
- Схемы интеграции: через платежную страницу или сервер-сервер с использованием единого API.
- Поддержка 3-D Secure.
- Повторная оплата без ввода номера, даты истечения и держателя карты. Поля для передачи дополнительных данных.
- Выбор сегмента для разделения потока транзакций.
- Выбор языка и шаблона платежной страницы.
- Установка времени жизни сессии.
- Выплаты на карту (кредитные транзакции).
- Передача динамического дескриптора.
- Рекуррентные платежи.
- Проведение авторизаций с нулевой суммой.
- Проведение DSRP-платежей.

Программный интерфейс

Организация заказов и операций

Платежи представляются в виде заказов. Заказ хранит общую информацию о платеже и реквизиты держателя. Транзакции представлены в виде последовательности операций, которые ассоциируются с заказом, например: авторизация, финансовое подтверждение, возврат средств.

Созданному заказу присваивается идентификатор, позже этот идентификатор используется в запросах, связанных этим заказом.

Статусы заказов

По статусу заказа можно судить о текущем состоянии.

Статус	Расшифровка	Возможные операции
new	Заказ создан, но обработка еще не началась	—
processing	Выполняется обработка заказа	—
prepared	Ожидаются действия извне	—
authorized	Проведена успешная авторизация	reverse, charge, rebill
charged	Проведено успешное списание средств	refund, rebill
reversed	Проведено успешное аннулирование	rebill
refunded	Проведен успешный возврат средств	refund, rebill
rejected	Заказ был отклонен системой	rebill
fraud	Заказ был определен как фродный и отклонен системой	rebill
declined	Заказ был отклонен банком-эквайером	rebill
chargedback	Заказ был отмечен как чарджбек	rebill
error	В процессе выполнения произошла ошибка, возможно, требуется вмешательство службы поддержки	rebill

Статусы операций

По статусу операции можно судить об успешности ее выполнения.

Статус	Расшифровка
success	Операция проведена успешно
failure	Операция отклонена (например, произошел отказ авторизации)
error	В процессе выполнения произошла ошибка, возможно, требуется вмешательство службы поддержки

Схемы взаимодействия

Оплата через платежную страницу

1. Пользователь нажимает кнопку "Оплатить" на сайте продавца.
2. Продавец обрабатывает запрос пользователя на совершение покупки и отправляет в систему запрос `create`, указывая описание товара, сумму платежа и при необходимости другие данные (см. [Создание заказа](#)). (для одностадийной оплаты нужно передать `auto_charge:1`)
3. В ответ в Header в параметре Location система передает продавцу URI платежной страницы.
4. Продавец перенаправляет пользователя на полученный URI.
5. Пользователь вводит платежную информацию в форму.
6. Система обрабатывает форму и перенаправляет пользователя в магазин продавца на `return_url`, указанный в запросе `create`.
7. Продавец по номеру заказа из URI запрашивает состояние платежа у системы (см. [Информация о заказе](#)) и на основе полученного статуса заказа формирует страницу-

результат.

Сервер-сервер

1. Пользователь вводит платежную информацию и нажимает кнопку "Оплатить" на сайте продавца.
2. Продавец обрабатывает запрос пользователя на совершение покупки и отправляет в систему запрос `authorize` (см. [Авторизация заказа](#)).
3. Система создает заказ и обрабатывает платеж, используя полученные реквизиты.
4. Система возвращает информацию о результате проведения операции.
5. Продавец обрабатывает результат проведения операции и формирует страницу- результат.

Списание средств в бэкофисе

После успешного проведения оплаты через платежную страницу производится списание средств с карты заказчика. Способы списания:

1. Включение автоклиринга в проекте (задержка автоклиринга задается в минутах).
2. Ручное списание средств в бэкофисе ("Заказы => Искомый заказ => Списать").

Работа через API

- Запросы осуществляются по HTTP 1.1 с использованием SSL и *Basic*-авторизации.
- Кодировка запросов и ответов: *UTF-8*.
- Запросы и ответы в API передаются в формате *JSON*.
- Для выгрузки информации используется метод *GET*, для проведения манипуляций над данными — методы *POST* и *PUT*.
- Для *GET*-запросов параметры запроса передаются как *QUERY STRING*.
- Для *POST*-запросов параметры передаются в теле *POST*-запроса, иногда передается и *QUERY STRING*.
- Успешные ответы отдаются с *HTTP*-статусом 200 (или 201). В случае ошибки статус может отличаться (см. [HTTP-коды](#)).

Аутентификация

Для доступа к API используется *Basic*-авторизация. Для каждого проекта выдается логин и пароль.

```
GET /orders/ HTTP/1.1
Authorization: Basic Zm9vOmJhcg==
Host: example.com
```

HTTP-коды

Дополнительную информацию несет HTTP-код ответа:

Код	Описание
200	Этот код означает, что запрошенная операция была проведена успешно. В зависимости от запрошенной операции это может означать успешно проведенную авторизацию, отмену, списание или возврат.
201	Заказ создан и ожидает ввода данных от пользователя.
402	Операция отклонена эквайером или шлюзом.
422	Ошибка валидации данных запроса.
500	Внутренняя ошибка в эквайере или шлюзе.

Типы отказов

Типы отказов описывают причину отказа:

Код	Описание
declined	Эквайер отклонил транзакцию
fraud	Транзакция отклонена системой защиты от мошеннических транзакций в шлюзе.
rejected	Транзакция отклонена шлюзом.
error	Отказ из-за ошибки на стороне эквайера или шлюза.
validation	Запрос не обработан, т.к. не прошел валидацию.

Обработка ошибок

Все отказы и ошибки возвращаются в едином формате:

```
{
  "failure type" : "error",
  "failure message" : "This is an error",
  "order id" : null,
}
```

Причина отказа передается в поле `failure_message`. Если в процессе обработки заказа был создан заказ, его идентификатор передается в поле `order_id`.

Валидация

В случае ошибки валидации передается дополнительный параметр `errors`, где содержится информация об ошибочных полях. Для удобства передается дополнительная информация о поле, которое вызвало ошибку.

Пример:

```

curl -v -X POST -H 'Content-Type: application/json' -d '{"foo" : "bar"}' 'http://project:***@api.box:5001/orders/create'
> POST /orders/create HTTP/1.1
> Authorization: Basic CHJvamVjdDpwYXNzd29yZA==
> Host: api.box:5001
> Content-Type: application/json
> Content-Length: 13
>
< HTTP/1.1 422 Unprocessable Entity
< Content-Type: application/json
< Content-Length: 223
< {
  "errors" : [
    {
      "attribute" : "required",
      "details" : [
        "(true)"
      ],
      "message" : "Required",
      "uri" : "#/amount"
    },
    {
      "message" : "Unknown property",
      "uri" : "#/foo"
    }
  ],
  "failure_message" : "Validation failed",
  "failure_type" : "validation",
  "order_id" : null
}

```

Пример кода обработки

```

response = http.post('https://api/authorize', pan => 4111111111111111, ... , cvv =>
123);

if response.code == 200 {

    order = decode_json(response.body);

    print "Authorization success";

    print "Order ID: " . order.id

} else if response.code == 402 {

    print "Authorization declined";

    print "Decline reason: " . response.failure_message;

} else if response.code == 422

    { print "Input data is incorrect";

    print "Reason: " . response.failure_message;

```



```
} else {  
  
    print "An error occurred: " . response.failure_message;  
  
}
```

Запросы

Тестовый запрос (GET /ping)

Параметры

Не принимаются.

Ответ

Название	Описание	Пример
message	Сообщение	PONG!
date	Текущие дата и время на сервере	2014-06-30 14:09:20

Пример:

```
curl -v -X GET -H 'Content-Type: application/json' 'http://project:***@api.box:5001/ping'  
> GET /ping HTTP/1.1  
> Authorization: Basic cHJvamVjdDpwYXNzd29yZA==  
> Host: api.box:5001  
> Content-Type: application/json  
>  
< HTTP/1.1 200 OK  
< Content-Type: application/json  
< Content-Length: 48  
< {  
    "date" : "2016-08-22 15:39:42",  
    "message" : "PONG!"  
}
```

Выборка данных

При выгрузке данных, таких как, например, список заказов, максимальное количество записей, которые можно получить за один запрос, составляет 2000 записей. Рекомендуется использовать параметры фильтрации, чтобы ограничить количество выгружаемых данных.

Информация о заказе (GET /orders/:id)

Параметры

Name	Description	Example
expand	List of extra objects to display. Possible expands are: <code>card</code> , <code>client</code> , <code>location</code> , <code>custom_fields</code> , <code>issuer</code> , <code>secure3d</code> , <code>operations.cashflow</code>	<code>card</code> , <code>location</code> , <code>client</code>

Ответ

Название	Описание	Пример
<code>id</code>	Уникальный идентификатор заказа	5673044075769513
<code>merchant_order_id</code>	Идентификатор заказа в системе мерчанта	5678
<code>auth_code</code>	Код авторизации	AUTH12
<code>amount</code>	Сумма заказа	9.99
<code>amount_charged</code>	Сумма списанных средств	9.99
<code>amount_refunded</code>	Сумма возвращенных средств	5.99
<code>pan</code>	Номер карты	411111****1111
<code>segment</code>	Сегмент	987654321
<code>status</code>	Статус заказа	authorized
<code>created</code>	Дата создания заказа	2014-06-30 14:38:48
<code>updated</code>	Дата последнего действия над заказом	2014-06-30 14:38:48
<code>card</code>	Информация о карте (*)	—
<code>*.subtype</code>	Подтип карты	classic
<code>*.type</code>	Тип карты	visa
<code>client</code>	Информация о клиенте (*)	—
<code>*.address</code>	Адрес	Main ave. 1
<code>*.city</code>	Город	Omsk
<code>*.country</code>	Страна	RUSSIA
<code>*.email</code>	Е-mail	foo@bar.com
<code>*.name</code>	Имя	Ivanov
<code>*.phone</code>	Телефон	+7 456 890
<code>*.state</code>	Штат, область	Leningradskaya obl.
<code>*.zip</code>	Почтовый индекс	123456
<code>custom_fields</code>	Пользовательские поля	—
<code>*.description</code>	Описание заказа	Book sale #453
<code>issuer</code>	Информация о банке-эмитенте	—
<code>*.bin</code>	VIN-код	411111
<code>*.country</code>	Страна	RUS
<code>*.title</code>	Название банка	Bank
<code>location</code>	Местоположение клиента (*)	—

*.ip	IP-адрес	8.8.8.8
operations	Массив выполненных операций над заказом (*)	—

Описание полей см. в разделе [Список операций](#).

Пример:

```
curl -v -X GET -H 'Content-Type: application/json'
'http://project:***@api.box:5001/orders/23014806815093624?
expand=card,client,location,custom_fields,issuer,secure3d,operations.cashflow'

> GET /orders/23014806815093624?
expand=card,client,location,custom_fields,issuer,secure3d,operations.cashflow HTTP/1.1
> Authorization: Basic cHJvamVjdDpwYXNzd29yZA==
> Host: api.box:5001
> Content-Type: application/json
>
< HTTP/1.1 200 OK
< Content-Type: application/json
< Content-Length: 1018
< {
  "orders" : [
    {
      "amount" : "9.99",
      "amount_charged" : "0.00",
      "amount_refunded" : "0.00",
      "auth_code" : "AUTH12",
      "card" : {
        "holder" : "Ivan Ivanov",
        "subtype" : "classic",
        "type" : "visa"
      },
      "client" : {
        "address" : "Main ave. 1",
        "city" : "Omsk",
        "country" : "RUSSIA",
        "email" : "foo@bar.com",
        "name" : "Ivan Ivanov",
        "phone" : "+7 456 890",
        "state" : "Omskiy kr.",
        "zip" : "123456"
      },
      "created" : "2016-08-22 15:39:49",
      "currency" : "RUB",
      "custom_fields" : {},
      "description" : "Book sale #453",
      "descriptor" : "TESTMERCH/TERM",
      "id" : "23014806815093624",
      "issuer" : {
        "bin" : "411111",
        "country" : "RUS",
        "title" : "TestBank"
```

```

    },
    "location" : {
        "ip" : "6.6.6.6"
    },
    "merchant_order_id" : "5678",
    "operations" : [
        {
            "amount" : "9.99",
            "auth_code" : "AUTH12",
            "id" : "47575",
            "cashflow" : {
                "amount" : "0.00",
                "currency" : "RUB",
                "fee" : "0.00",
                "incoming" : "0.00",
                "receivable" : "0.00",
                "reserve" : "0.00"
            },
            "created" : "2016-08-22 15:39:49",
            "currency" : "RUB",
            "iso_message" : "Approved",
            "iso_response_code" : "00",
            "status" : "success",
            "type" : "authorize"
        }
    ],
    "pan" : "411111****1111",
    "secure3d" : {},
    "segment" : "987654321",
    "status" : "authorized",
    "updated" : "2016-08-22 15:39:49"
}
]
}

```

Список заказов (GET /orders/)

Параметры

Name	Description	Example
<code>expand</code>	Список расширений, которые требуется показать. Возможные расширения: <code>card</code> , <code>client</code> , <code>location</code> , <code>custom_fields</code> , <code>issuer</code> , <code>secure3d</code> , <code>operations.cashflow</code>	<code>card, location, client</code>
<code>status</code>	Фильтр по статусу	<code>charged, refunded</code>
<code>created_from</code>	Фильтр по дате создания	<code>2016-01-01 00:00:00</code>
<code>created_to</code>	Фильтр по дате создания	<code>2016-01-31 23:59:59</code>
<code>merchant_order_id</code>	Фильтр по идентификатору	<code>12345, 123456</code>
<code>card.type</code>	Фильтр по типу карты	<code>visa, mastercard</code>
<code>card.subtype</code>	Фильтр по субтипу карты	<code>electron, maestro</code>

location.ip	Фильтр по IP-адресу	8.8.8.8, 8.8.8.9
client.name	Поиск по имени клиента или по части имени	Ivanov
client.email	Поиск по адресу электронной почты или по части адреса	@yahoo.com
issuer.country	Фильтр по стране эмитента (ISO 3166-1 alpha-3)	RUS
issuer.bin	Фильтр по БИН-коду	555555, 411111
issuer.title	Фильтр по названию эмитента	Credit Union

Ответ

Поля ответа повторяют набор полей из раздела [Информация о заказе](#).

Пример:

```
curl -v -X GET -H 'Content-Type: application/json'
'http://project:***@api.box:5001/orders/'
> GET /orders/ HTTP/1.1
> Authorization: Basic cHJvamVjdDpwYXNzd29yZA==
> Host: api.box:5001
> Content-Type: application/json
>
< HTTP/1.1 200 OK
< Content-Type: application/json
< Content-Length: 5550
< {
  "orders" : [
    {
      "amount" : "9.99",
      "amount_charged" : "0.00",
      "amount_refunded" : "0.00",
      "auth_code" : "AUTH12",
      "created" : "2016-08-22 15:39:49",
      "currency" : "RUB",
      "description" : "Book sale #453",
      "descriptor" : "TESTMERCH/TERM",
      "id" : "23014806815093624",
      "merchant_order_id" : "5678",
      "pan" : "411111****1111",
      "segment" : "987654321",
      "status" : "authorized",
      "updated" : "2016-08-22 15:39:49"
    },
    {
      "amount" : "9.99",
      "amount_charged" : "0.00",
      "amount_refunded" : "0.00",
      "auth_code" : "AUTH12",
      "created" : "2016-08-22 15:39:48",
      "currency" : "RUB",
      "description" : null,
      "descriptor" : "TESTMERCH/TERM",
      "id" : "23014806643308174",
```

```

    "merchant_order_id" : null,
    "pan" : "411111****1111",
    "status" : "authorized",
    "updated" : "2016-08-22 15:39:48"
  }
]
}

```

Список операций (GET /operations/)

Параметры

Name	Description	Example
<code>expand</code>	Список расширений которые требуется показать. Возможные расширения: <code>cashflow</code>	<code>cashflow</code>
<code>status</code>	Фильтр по статус	<code>error</code>
<code>type</code>	Фильтр по типу	<code>charge, refund</code>
<code>created_from</code>	Фильтр по дате создания	<code>2016-01-01 00:00:00</code>
<code>created_to</code>	Фильтр по дате создания	<code>2016-01-31 23:59:59</code>

Ответ

Название	Описание	Пример
<code>amount</code>	Сумма операции	<code>9.99</code>
<code>auth_code</code>	Код авторизации	<code>AUTH12</code>
<code>created</code>	Дата проведения операции	<code>2014-06-30 14:38:48</code>
<code>iso_message</code>	Сообщение	<code>Approved</code>
<code>iso_response_code</code>	Код ответа	<code>00</code>
<code>status</code>	Статус операции	<code>success</code>
<code>type</code>	Тип операции	<code>authorize</code>
<code>cashflow</code>	Финансовая информация	<code>—</code>
<code>*.amount</code>	Сумма	<code>9.99</code>
<code>*.fee</code>	Комиссия	<code>0.00</code>
<code>*.incoming</code>	Доход	<code>6.00</code>
<code>*.receivable</code>	К получению	<code>5.00</code>

Пример:

```

curl -v -X GET -H 'Content-Type: application/json'
'http://project:***@api.box:5001/operations/'
> GET /operations/ HTTP/1.1
> Authorization: Basic cHJvamVjdDpwYXNzd29yZA==
> Host: api.box:5001
> Content-Type: application/json
>
< HTTP/1.1 200 OK
< Content-Type: application/json

```

```

< Content-Length: 2931
< {
  "operations" : [
    {
      "amount" : "11991.00",
      "auth_code" : "AUTH12",
      "id" : "47575",
      "created" : "2016-08-24 12:07:00",
      "currency" : "RUB",
      "iso_message" : "Approved",
      "iso_response_code" : "00",
      "order_id" : "23011537206894183",
      "status" : "success",
      "type" : "charge"
    },
    {
      "amount" : "8997.00",
      "auth_code" : "AUTH12",
      "id" : "47574",
      "created" : "2016-08-23 12:06:55",
      "currency" : "RUB",
      "iso_message" : "Approved",
      "iso_response_code" : "00",
      "order_id" : "23011536237676397",
      "status" : "success",
      "type" : "authorize"
    }
  ]
}

```

Экспорт курсов валют (GET /exchange_rates/)

Обязательные_параметры

Название	Описание	Пример
<code>date</code>	Дата курсов валют	2014-06-30

Ответ

Название	Описание	Пример
<code>exchange_rates</code>	Обменные данные	-
<code>*.from</code>	Котируемая валюта	USD
<code>*.to</code>	Базовая валюта	EUR
<code>*.rate</code>	Котировка	0.800000

```
curl -v -X GET -H 'Content-Type: application/json'
'http://project:***@api.box:5001/exchange_rates/?date=2014-06-30'
> GET /exchange_rates/ HTTP/1.1
> Authorization: Basic cHJvamVjdDpwYXNzd29yZA==
> Host: api.box:5001
> Content-Type: application/json
>
< HTTP/1.1 200 OK
< Content-Type: application/json
< Content-Length: 1805
< {
  "exchange_rates" : [
    {
      "from" : "EUR",
      "rate" : "75.132109",
      "to" : "RUB"
    },
    {
      "to" : "USD",
      "from" : "EUR",
      "rate" : "1.145036"
    }
  ]
}
```

Работа с заказами

Создание заказа (POST /orders/create)

Успешный ответ на этот запрос возвращается с кодом 201. Это означает, что заказ создан и он ожидает ввода данных от держателя карты.

Обязательные параметры

Название	Описание	Пример
<code>amount</code>	Сумма заказа	9.99

Необязательные параметры

Название	Описание	Пример
----------	----------	--------

currency	Order currency	RUB
merchant_order_id	Идентификатор заказа в системе мерчанта	5678
segment	Сегмент	987654321
client	Информация о клиенте (*)	—
*.address	Адрес	Main ave. 1
*.city	Город	Omsk
*.country	Страна	RUSSIA
*.login	Логин клиента	john_doe
*.email	E-mail	foo@bar.com
*.name	Имя	Ivanov
*.phone	Телефон	+7 456 890
*.state	Штат	Leningradskaya obl.
*.zip	Почтовый индекс	123456
custom_fields	Пользовательские поля (*)	—
description	Описание заказа	Book sale #453
location	Местоположение клиента (*)	—
*.ip	IP-адрес	8.8.8.8
options	Опции (*)	—
*.expiration_timeout	Время сессии	30m
*.force3d	Принудительное направление в 3-D Secure	1
*.auto_charge	Провести операцию Charge (используется (для одностадийных платежей))	1
*.language	Язык платежной формы	ru
*.return_url	URL для перенаправления клиента после оплаты	http://my.url
*.template	Шаблон платежной формы	12
*.mobile	Мобильный шаблон	1
*.terminal	Терминал	TERM12

Ответ

Поля ответа повторяют набор полей из раздела [Информация о заказе](#).

Дополнительно возвращается заголовок *Location* в котором указывается ссылка на платежную страницу.

Пример (минимальный набор данных):

```
curl -v -X POST -H 'Content-Type: application/json' -d '{ "amount" : 9.99
  }' 'http://project:***@api.box:5001/orders/create'

> POST /orders/create HTTP/1.1
> Authorization: Basic cHJvamVjdDpwYXNzd29yZA==
> Host: api.box:5001
```

```

> Content-Type: application/json
> Content-Length: 15
>
< Location: http://localhost:5002/pay/23014805213622230
< Content-Type: application/json
< Content-Length: 341
< {
  "orders" : [
    {
      "amount" : "9.99",
      "amount_charged" : "0.00",
      "amount_refunded" : "0.00",
      "card" : {},
      "client" : {},
      "created" : "2016-08-22 15:39:42",
      "currency" : "RUB",
      "custom_fields" : {},
      "description" : null,
      "id" : "23014805213622230",
      "issuer" : {},
      "location" : {},
      "merchant_order_id" : null,
      "operations" : [],
      "secure3d" : {},
      "status" : "new",
      "updated" : "2016-08-22 15:39:42"
    }
  ]
}

```

Пример (расширенный набор данных):

```

curl -v -X POST -H 'Content-Type: application/json' -d '{
  "amount" : 9.99,
  "client" : {
    "address" : "Main ave. 1",
    "city" : "Omsk",
    "country" : "RUSSIA",
    "email" : "foo@bar.com",
    "name" : "Ivan Ivanov",
    "phone" : "+7 456 890",
    "state" : "Omskiy kr.",
    "zip" : "123456"
  },
  "description" : "Book sale #453",
  "location" : {
    "ip" : "6.6.6.6"
  },
  "merchant_order_id" : "5678",
}

```

```

    "options" : {
      "force3d" : 0,
      "return_url" : "http://shop.com",
      "terminal" : ""
    },
    "segment" : "987654321"
  }
' 'http://project:***@api.box:5001/orders/create'

> POST /orders/create HTTP/1.1
> Authorization: Basic cHJvamVjdDpwYXNzd29yZA==
> Host: api.box:5001
> Content-Type: application/json
> Content-Length: 357
>
< HTTP/1.1 201 Created
< Location: http://localhost:5002/pay/23014805439541518
< Content-Type: application/json
< Content-Length: 544
< {
  "orders" : [
    {
      "amount" : "9.99",
      "amount_charged" : "0.00",
      "amount_refunded" : "0.00",
      "card" : {},
      "client" : {
        "address" : "Main ave. 1",
        "city" : "Omsk",
        "country" : "RUSSIA",
        "email" : "foo@bar.com",
        "name" : "Ivan Ivanov",
        "phone" : "+7 456 890",
        "state" : "Omskiy kr.",
        "zip" : "123456"
      },
      "created" : "2016-08-22 15:39:43",
      "currency" : "RUB",
      "custom_fields" : {},
      "description" : "Book sale #453",
      "id" : "23014805439541518",
      "issuer" : {},
      "location" : {
        "ip" : "6.6.6.6"
      },
      "merchant_order_id" : "5678",

```

```

        "operations" : [],
        "secure3d" : {},
        "segment" : "987654321",
        "status" : "new",
        "updated" : "2016-08-22 15:39:43"
    }
}
1
}

```

Авторизация (POST /orders/authorize)

Обязательные параметры

Название	Описание	Пример
<code>amount</code>	Сумма заказа	9.99
<code>pan</code>	Номер карты	4111111111111111
<code>card</code>	Информация о карте	—
<code>*.cvv</code>	CVV-код (может быть отключен в настройках проекта)	123
<code>*.holder</code>	Держатель карты (может быть отключен в настройках проекта)	Ivan Ivanov
<code>*.expiration_month</code>	Месяц истечения карты	06
<code>*.expiration_year</code>	Год истечения карты	2022
<code>location</code>	Местоположение клиента (*)	—
<code>*.ip</code>	IP-адрес	8.8.8.8

Необязательные параметры

Название	Описание	Пример
<code>currency</code>	Валюта	RUB
<code>merchant_order_id</code>	Идентификатор заказа в системе мерчанта	5678
<code>segment</code>	Сегмент	987654321
<code>client</code>	Информация о клиенте (*)	—
<code>*.address</code>	Адрес	Main ave. 1
<code>*.city</code>	Город	Omsk
<code>*.country</code>	Страна	RUSSIA
<code>*.email</code>	E-mail	foo@bar.com
<code>*.name</code>	Имя	Ivanov
<code>*.phone</code>	Телефон	+7 456 890
<code>*.state</code>	Штат	Leningradskaya obl.
<code>*.zip</code>	Почтовый индекс	123456
<code>custom_fields</code>	Пользовательские поля (*)	—
<code>*.description</code>	Описание заказа	Book sale #453
<code>location</code>	Местоположение клиента (*)	—

*.ip	IP-адрес	8.8.8.8
options	Опции (*)	—
*.force3d	Принудительное направление в 3-D Secure	1
*.auto_charge	Провести операцию Charge (используется для одностадийных платежей)	1
*.terminal	Терминал	TERM12
*.recurring	Флаг рекуррентного платежа	1

Ответ

Поля ответа повторяют набор полей из раздела [Информация о заказе](#) и дополнительно могут содержать:

Название	Описание	Пример
form3d_html	3-D Secure форма в формате HTML	<form id="form3d" ...
form3d	3-D Secure форма (*)	—
*.MD	Параметр MD	NTY3NDg0NjMzMjA0MDU1OQ==
*.PaReq	Параметр PaReq	Y3NDg0NjMzM0MDU1OQ
*.TermUrl	URL системы для завершения операции	http://box.com/complete
*.action	URL ACS банка-эквайера	http://bank.com
*.method	Метод формы	POST

Пример (минимальный набор данных):

```
curl -v -X POST -H 'Content-Type: application/json' -d '{
  "amount" : 9.99,
  "card" : {
    "cvv" : "333",
    "expiration_month" : 12,
    "expiration_year" : 2020,
    "holder" : "Ivan Ivanov"
  },
  "location" : {
    "ip" : "6.6.6.6"
  },
  "pan" : "4111111111111111"
}' 'http://project:***@api.box:5001/orders/authorize'
```

```
> POST /orders/authorize HTTP/1.1
> Authorization: Basic cHJvamVjdDpwYXNzd29yZA==
> Host: api.box:5001
> Content-Type: application/json
> Content-Length: 156
>
```

```
< Content-Type: application/json
< Content-Length: 829
< {
  "orders" : [
    {
      "amount" : "9.99",
      "amount_charged" : "0.00",
      "amount_refunded" : "0.00",
      "auth_code" : "AUTH12",
      "card" : {
        "holder" : "Ivan Ivanov",
        "subtype" : "classic",
        "type" : "visa"
      },
      "client" : {},
      "created" : "2016-08-22 15:39:48",
      "currency" : "RUB",
      "custom_fields" : {},
      "description" : null,
      "descriptor" : "TESTMERCH/TERM",
      "id" : "23014806643308174",
      "issuer" : {
        "bin" : "411111",
        "country" : "RUS",
        "title" : "TestBank"
      },
      "location" : {
        "ip" : "6.6.6.6"
      },
      "merchant_order_id" : null,
      "operations" : [
        {
          "amount" : "9.99",
          "auth_code" : "AUTH12",
          "id" : "47575",
          "cashflow" : {
            "amount" : "0.00",
            "currency" : "RUB",
            "fee" : "0.00",
            "incoming" : "0.00",
            "receivable" : "0.00",
            "reserve" : "0.00"
          },
          "created" : "2016-08-22 15:39:48",
          "currency" : "RUB",
          "iso_message" : "Approved",
```

```

        "iso_response_code" : "00",
        "status" : "success",
        "type" : "authorize"
    }
},
"pan" : "411111****1111",
"secure3d" : {},
"status" : "authorized",
"updated" : "2016-08-22 15:39:48"
}
]
}

```

Пример (расширенный набор данных):

```

curl -v -X POST -H 'Content-Type: application/json' -d
'{ "amount" : 9.99,
  "card" : {
    "cvv" : "333",
    "expiration_month" : 12,
    "expiration_year" : 2020,
    "holder" : "Ivan Ivanov"
  },
  "client" : {
    "address" : "Main ave. 1",
    "city" : "Omsk",
    "country" : "RUSSIA",
    "email" : "foo@bar.com",
    "name" : "Ivan Ivanov",
    "phone" : "+7 456 890",
    "state" : "Omskiy kr.",
    "zip" : "123456"
  },
  "description" : "Book sale #453",
  "location" : {
    "ip" : "6.6.6.6"
  },
  "merchant_order_id" : "5678",
  "options" : {
    "force3d" : 0,
    "return_url" : "http://shop.com",
    "terminal" : ""
  },
  "pan" : "4111111111111111",
  "segment" : "987654321"
}' 'http://project:***@api.box:5001/orders/authorize'

```

```
> POST /orders/authorize HTTP/1.1
> Authorization: Basic cHJvamVjdDpwYXNzd29yZA==
> Host: api.box:5001
> Content-Type: application/json
> Content-Length: 470
>
< HTTP/1.1 200 OK
< Content-Type: application/json
< Content-Length: 1018
< {
  "orders" : [
    {
      "amount" : "9.99",
      "amount_charged" : "0.00",
      "amount_refunded" : "0.00",
      "auth_code" : "AUTH12",
      "card" : {
        "holder" : "Ivan Ivanov",
        "subtype" : "classic",
        "type" : "visa"
      },
      "client" : {
        "address" : "Main ave. 1",
        "city" : "Omsk",
        "country" : "RUSSIA",
        "email" : "foo@bar.com",
        "name" : "Ivan Ivanov",
        "phone" : "+7 456 890",
        "state" : "Omskiy kr.",
        "zip" : "123456"
      },
      "created" : "2016-08-22 15:39:49",
      "currency" : "RUB",
      "custom_fields" : {},
      "description" : "Book sale #453",
      "descriptor" : "TESTMERCH/TERM",
      "id" : "23014806815093624",
      "issuer" : {
        "bin" : "411111",
        "country" : "RUS",
        "title" : "TestBank"
      },
      "location" : {
        "ip" : "6.6.6.6"
      },
      "merchant_order_id" : "5678",
```



```

"operations" : [
  {
    "amount" : "9.99",
    "auth_code" : "AUTH12",
    "id" : "47575",
    "cashflow" : {
      "amount" : "0.00",
      "currency" : "RUB",
      "fee" : "0.00",
      "incoming" : "0.00",
      "receivable" : "0.00",
      "reserve" : "0.00"
    },
    "created" : "2016-08-22 15:39:49",
    "currency" : "RUB",
    "iso_message" : "Approved",
    "iso_response_code" : "00",
    "status" : "success",
    "type" : "authorize"
  }
],
"pan" : "411111****1111",
"secure3d" : {},
"segment" : "987654321",
"status" : "authorized",
"updated" : "2016-08-22 15:39:49"
}
]
}

```

Пример (3-D Secure):

```

curl -v -X POST -H 'Content-Type: application/json' -d '{
  "amount" : 9.99,
  "card" : {
    "cvv" : "333",
    "expiration_month" : 12,
    "expiration_year" : 2020,
    "holder" : "Ivan Ivanov"
  },
  "location" : {
    "ip" : "6.6.6.6"
  },
  "options" : {
    "force3d" : 1
  },
  "pan" : "4111111111111111"
}

```

```

}' 'http://project:***@api.box:5001/orders/authorize'

> POST /orders/authorize HTTP/1.1
> Authorization: Basic cHJvamVjdDpwYXNzd29yZA==
> Host: api.box:5001
> Content-Type: application/json
> Content-Length: 180
>
< HTTP/1.1 201 Created
< Content-Type: application/json
< Content-Length: 1083
<
{
  "orders" : [
    {
      "amount" : "9.99",
      "amount_charged" : "0.00",
      "amount_refunded" : "0.00",
      "card" : {
        "holder" : "Ivan Ivanov",
        "subtype" : "classic",
        "type" : "visa"
      },
      "client" : {},
      "created" : "2016-08-22 15:39:46",
      "currency" : "RUB",
      "custom_fields" : {},
      "description" : null,
      "form3d" : {
        "MD" : "MjMwMTQ4MDYwNzgwNjEzMDM=",
        "PaReq" : "PaReq",
        "TermUrl" : "http://localhost:5002/complete/secure3d",
        "action" : "http://localhost:5002/test/bank",
        "method" : "GET"
      },
      "form3d_html" : "<form id=\"form3d\" method=\"GET\"
action=\"http://localhost:5002/test/bank\"><input type=\"hidden\" name=\"MD\"
value=\"MjMwMTQ4MDYwNzgwNjEzMDM=\"><input type=\"hidden\" name=\"PaReq\"
value=\"PaReq\"><input type=\"hidden\" name=\"TermUrl\"
value=\"http://localhost:5002/complete/secure3d\"></form><script>window.onload =
function() {document.forms['form3d'].submit();}</script>",
      "id" : "23014806078061303",
      "issuer" : {
        "bin" : "411111",
        "country" : "RUS",
        "title" : "TestBank"
      },
      "location" : {

```

```

        "ip" : "6.6.6.6"
    },
    "merchant_order_id" : null,
    "operations" : [],
    "pan" : "411111****1111",
    "secure3d" : {
        "reason" : "force3d",
        "scenario" : "unknown"
    },
    "status" : "prepared",
    "updated" : "2016-08-22 15:39:46"
}
]
}

```

Примечание

После успешной авторизации необходимо провести операцию [Списание средств](#), которая проводит списание средств с карты. Если в настройках аккаунта не установлено автоматическое списание, и в запросе не была передана опция *auto_charge*, и операция *charge* не была проведена явно - списание проведено не будет, а сумма авторизации через некоторое время разблокируются эмитентом и средства вернутся держателю карты.

Отмена авторизации (PUT /orders/:id/reverse)

Параметры

Не принимаются.

Ответ

Поля ответа повторяют набор полей из раздела [Информация о заказе](#).

Пример:

```

curl -v -X PUT -H 'Content-Type: application/json'
'http://project:***@api.box:5001/orders/23014805692462630/reverse'

> PUT /orders/23014805692462630/reverse HTTP/1.1
> Authorization: Basic cHJvamVjdDpwYXNzd29yZA==
> Host: api.box:5001
> Content-Type: application/json
>
< Content-Type: application/json
< Content-Length: 1115
< {

```

```
"orders" : [
  {
    "amount" : "9.99",
    "amount_charged" : "0.00",
    "amount_refunded" : "0.00",
    "auth_code" : "AUTH12",
    "card" : {
      "holder" : "Ivan Ivanov",
      "subtype" : "classic",
      "type" : "visa"
    },
    "client" : {},
    "created" : "2016-08-22 15:39:44",
    "currency" : "RUB",
    "custom_fields" : {},
    "description" : null,
    "descriptor" : "TESTMERCH/TERM",
    "id" : "23014805692462630",
    "issuer" : {
      "bin" : "411111",
      "country" : "RUS",
      "title" : "TestBank"
    },
    "location" : {
      "ip" : "6.6.6.6"
    },
    "merchant_order_id" : null,
    "operations" : [
      {
        "amount" : "9.99",
        "auth_code" : "AUTH12",
        "id" : "47575",
        "cashflow" : {
          "amount" : "0.00",
          "currency" : "RUB",
          "fee" : "0.00",
          "incoming" : "0.00",
          "receivable" : "0.00",
          "reserve" : "0.00"
        },
        "created" : "2016-08-22 15:39:45",
        "currency" : "RUB",
        "iso_message" : "Approved",
        "iso_response_code" : "00",
        "status" : "success",
        "type" : "authorize"
      }
    ]
  }
]
```

```

    },
    {
      "amount" : "9.99",
      "auth_code" : "AUTH12",
      "id" : "47574",
      "cashflow" : {
        "amount" : "0.00",
        "currency" : "RUB",
        "fee" : "0.00",
        "incoming" : "0.00",
        "receivable" : "0.00",
        "reserve" : "0.00"
      },
      "created" : "2016-08-22 15:39:50",
      "currency" : "RUB",
      "iso_message" : "Approved",
      "iso_response_code" : "00",
      "status" : "success",
      "type" : "reverse"
    }
  ],
  "pan" : "411111****1111",
  "secure3d" : {},
  "status" : "reversed",
  "updated" : "2016-08-22 15:39:50"
}
]
}

```

Списание средств (PUT /orders/:id/charge)

Параметры

Название	Описание	Пример
<code>amount</code>	Сумма списываемых средств	9.99

Ответ

Поля ответа повторяют набор полей из раздела [Информация о заказе](#).

Пример (полное списание):

```

curl -v -X PUT -H 'Content-Type: application/json'
'http://project:***@api.box:5001/orders/23014806643308174/charge'

```

```
> PUT /orders/23014806643308174/charge HTTP/1.1
```

```
> Authorization: Basic cHJvamVjdDpwYXNzd29yZA==
```

```
> Host: api.box:5001
> Content-Type: application/json
>
< HTTP/1.1 200 OK
< Content-Type: application/json
< Content-Length: 1113
< {
  "orders" : [
    {
      "amount" : "9.99",
      "amount_charged" : "9.99",
      "amount_refunded" : "0.00",
      "auth_code" : "AUTH12",
      "card" : {
        "holder" : "Ivan Ivanov",
        "subtype" : "classic",
        "type" : "visa"
      },
      "client" : {},
      "created" : "2016-08-22 15:39:48",
      "currency" : "RUB",
      "custom_fields" : {},
      "description" : null,
      "descriptor" : "TESTMERCH/TERM",
      "id" : "23014806643308174",
      "issuer" : {
        "bin" : "411111",
        "country" : "RUS",
        "title" : "TestBank"
      },
      "location" : {
        "ip" : "6.6.6.6"
      },
      "merchant_order_id" : null,
      "operations" : [
        {
          "amount" : "9.99",
          "auth_code" : "AUTH12",
          "id" : "47575",
          "cashflow" : {
            "amount" : "0.00",
            "currency" : "RUB",
            "fee" : "0.00",
            "incoming" : "0.00",
            "receivable" : "0.00",
            "reserve" : "0.00"
```

```

    },
    "created" : "2016-08-22 15:39:48",
    "currency" : "RUB",
    "iso_message" : "Approved",
    "iso_response_code" : "00",
    "status" : "success",
    "type" : "authorize"
  },
  {
    "amount" : "9.99",
    "auth_code" : "AUTH12",
    "id" : "47576",
    "cashflow" : {
      "amount" : "9.99",
      "currency" : "RUB",
      "fee" : "0.30",
      "incoming" : "9.69",
      "receivable" : "9.69",
      "reserve" : "0.00"
    },
    "created" : "2016-08-22 15:39:53",
    "currency" : "RUB",
    "iso_message" : "Approved",
    "iso_response_code" : "00",
    "status" : "success",
    "type" : "charge"
  }
],
"pan" : "411111****1111",
"secure3d" : {},
"status" : "charged",
"updated" : "2016-08-22 15:39:53"
}
]
}

```

Пример (частичное списание):

```

curl -v -X PUT -H 'Content-Type: application/json' -d '{
  "amount" : 1.99
}' 'http://project:***@api.box:5001/orders/23014806312026412/charge'

> PUT /orders/23014806312026412/charge HTTP/1.1
> Authorization: Basic cHJvamVjdDpwYXNzd29yZA==
> Host: api.box:5001
> Content-Type: application/json
> Content-Length: 15

```

```
>
< HTTP/1.1 200 OK
< Content-Type: application/json
< Content-Length: 1113
< {
  "orders" : [
    {
      "amount" : "9.99",
      "amount_charged" : "1.99",
      "amount_refunded" : "0.00",
      "auth_code" : "AUTH12",
      "card" : {
        "holder" : "Ivan Ivanov",
        "subtype" : "classic",
        "type" : "visa"
      },
      "client" : {},
      "created" : "2016-08-22 15:39:47",
      "currency" : "RUB",
      "custom_fields" : {},
      "description" : null,
      "descriptor" : "TESTMERCH/TERM",
      "id" : "23014806312026412",
      "issuer" : {
        "bin" : "411111",
        "country" : "RUS",
        "title" : "TestBank"
      },
      "location" : {
        "ip" : "6.6.6.6"
      },
      "merchant_order_id" : null,
      "operations" : [
        {
          "amount" : "9.99",
          "auth_code" : "AUTH12",
          "id" : "47575",
          "cashflow" : {
            "amount" : "0.00",
            "currency" : "RUB",
            "fee" : "0.00",
            "incoming" : "0.00",
            "receivable" : "0.00",
            "reserve" : "0.00"
          },
          "created" : "2016-08-22 15:39:47",
          "currency" : "RUB",
          "iso_message" : "Approved",
          "iso_response_code" : "00",
          "status" : "success",
          "type" : "authorize"
        },
        {
```



```

        "amount" : "1.99",
        "auth_code" : "AUTH12",
        "id" : "47576",
        "cashflow" : {
            "amount" : "1.99",
            "currency" : "RUB",
            "fee" : "0.06",
            "incoming" : "1.93",
            "receivable" : "1.93",
            "reserve" : "0.00"
        },
        "created" : "2016-08-22 15:39:51",
        "currency" : "RUB",
        "iso_message" : "Approved",
        "iso_response_code" : "00",
        "status" : "success",
        "type" : "charge"
    }
},
    "pan" : "411111****1111", "secure3d" : {},
    "status" : "charged",
    "updated" : "2016-08-22 15:39:51"
}
]
}

```

Возврат средств (PUT /orders/:id/refund)

Параметры

Название	Описание	Пример
<code>amount</code>	Сумма возвращаемых средств	9.99

Ответ

Поля ответа повторяют набор полей из раздела [Информация о заказе](#).

Пример:

Пример (полный возврат):

```

curl -v -X PUT -H 'Content-Type: application/json'
'http://project:***@api.box:5001/orders/23014806643308174/refund'
> PUT /orders/23014806643308174/refund HTTP/1.1
> Authorization: Basic cHJvamVjdDpwYXNzd29yZA==
> Host: api.box:5001
> Content-Type: application/json
>
< Content-Type: application/json
< Content-Length: 1404
< {
  "orders" : [
    {

```

```
"amount" : "9.99",
"amount_charged" : "9.99",
"amount_refunded" : "9.99",
"auth_code" : "AUTH12",
"card" : {
  "holder" : "Ivan Ivanov",
  "subtype" : "classic",
  "type" : "visa"
},
"client" : {},
"created" : "2016-08-22 15:39:48",
"currency" : "RUB",
"custom_fields" : {},
"description" : null,
"descriptor" : "TESTMERCH/TERM",
"id" : "23014806643308174",
"issuer" : {
  "bin" : "411111",
  "country" : "RUS",
  "title" : "TestBank"
},
"location" : {
  "ip" : "6.6.6.6"
},
"merchant_order_id" : null,
"operations" : [
  {
    "amount" : "9.99",
    "auth_code" : "AUTH12",
    "id" : "47575",
    "cashflow" : {
      "amount" : "0.00",
      "currency" : "RUB",
      "fee" : "0.00",
      "incoming" : "0.00",
      "receivable" : "0.00",
      "reserve" : "0.00"
    },
    "created" : "2016-08-22 15:39:48",
    "currency" : "RUB",
    "iso_message" : "Approved",
    "iso_response_code" : "00",
    "status" : "success",
    "type" : "authorize"
  },
  {
    "amount" : "9.99",
    "auth_code" : "AUTH12",
    "id" : "47576",
    "cashflow" : {
      "amount" : "9.99",
      "currency" : "RUB",
```

```

        "fee" : "0.30",
        "incoming" : "9.69",
        "receivable" : "9.69",
        "reserve" : "0.00"
    },
    "created" : "2016-08-22 15:39:53",
    "currency" : "RUB",
    "iso_message" : "Approved",
    "iso_response_code" : "00",
    "status" : "success",
    "type" : "charge"
  },
  {
    "amount" : "9.99",
    "auth_code" : "AUTH12",
    "id" : "47577",
    "cashflow" : {
      "amount" : "-9.99",
      "currency" : "RUB",
      "fee" : "0.00",
      "incoming" : "-9.99",
      "receivable" : "-9.99",
      "reserve" : "0.00"
    },
    "created" : "2016-08-22 15:39:55",
    "currency" : "RUB",
    "iso_message" : "Approved",
    "iso_response_code" : "00",
    "status" : "success",
    "type" : "refund"
  }
],
"pan" : "411111****1111",
"secure3d" : {},
"status" : "refunded",
"updated" : "2016-08-22 15:39:55"
}
]
}

```

Пример (частичный возврат):

```

curl -v -X PUT -H 'Content-Type: application/json' -d '{
"amount" : 1.99
}' 'http://project:***@api.box:5001/orders/23014806312026412/refund'

> PUT /orders/23014806312026412/refund HTTP/1.1
> Authorization: Basic cHJvamVjdDpwYXNzd29yZA==
> Host: api.box:5001
> Content-Type: application/json

```

```
> Content-Length: 15
>
< Content-Type: application/json
< Content-Length: 1404
< {
  "orders" : [
    {
      "amount" : "9.99",
      "amount_charged" : "1.99",
      "amount_refunded" : "1.99",
      "auth_code" : "AUTH12",
      "card" : {
        "holder" : "Ivan Ivanov",
        "subtype" : "classic",
        "type" : "visa"
      },
      "client" : {},
      "created" : "2016-08-22 15:39:47",
      "currency" : "RUB",
      "custom_fields" : {},
      "description" : null,
      "descriptor" : "TESTMERCH/TERM",
      "id" : "23014806312026412",
      "issuer" : {
        "bin" : "411111",
        "country" : "RUS",
        "title" : "TestBank"
      },
      "location" : {
        "ip" : "6.6.6.6"
      },
      "merchant_order_id" : null,
      "operations" : [
        {
          "amount" : "9.99",
          "auth_code" : "AUTH12",
          "id" : "47575",
          "cashflow" : {
            "amount" : "0.00",
            "currency" : "RUB",
            "fee" : "0.00",
            "incoming" : "0.00",
            "receivable" : "0.00",
            "reserve" : "0.00"
          },
          "created" : "2016-08-22 15:39:47",
```

```
    "currency" : "RUB",
    "iso_message" : "Approved",
    "iso_response_code" : "00",
    "status" : "success",
    "type" : "authorize"
  },
  {
    "amount" : "1.99",
    "auth_code" : "AUTH12",
    "id" : "47576",
    "cashflow" : {
      "amount" : "1.99",
      "currency" : "RUB",
      "fee" : "0.06",
      "incoming" : "1.93",
      "receivable" : "1.93",
      "reserve" : "0.00"
    },
    "created" : "2016-08-22 15:39:51",
    "currency" : "RUB",
    "iso_message" : "Approved",
    "iso_response_code" : "00",
    "status" : "success",
    "type" : "charge"
  },
  {
    "amount" : "1.99",
    "auth_code" : "AUTH12",
    "id" : "47577",
    "cashflow" : {
      "amount" : "-1.99",
      "currency" : "RUB",
      "fee" : "0.00",
      "incoming" : "-1.99",
      "receivable" : "-1.99",
      "reserve" : "0.00"
    },
    "created" : "2016-08-22 15:39:52",
    "currency" : "RUB",
    "iso_message" : "Approved",
    "iso_response_code" : "00",
    "status" : "success",
    "type" : "refund"
  }
],
"pan" : "411111****1111",
```

```
    "secure3d" : {},
    "status" : "refunded",
    "updated" : "2016-08-22 15:39:52"
  }
]
}
```

Отмена (PUT /orders/:id/cancel)

Аналогично проведению [Отмены авторизации](#) или [Возврата средств](#).

Параметры

Название	Описание	Пример
<code>amount</code>	Сумма возвращаемых средств (кроме аннулирования)	9.99

Ответ

Поля ответа повторяют набор полей из раздела [Информация о заказе](#).

Пример:

```
curl -v -X PUT -H 'Content-Type: application/json'
'http://project:***@api.box:5001/orders/23014808614292147/cancel'
```

```
> PUT /orders/23014808614292147/cancel HTTP/1.1
> Authorization: Basic cHJvamVjdDpwYXNzd29yZA==
> Host: api.box:5001
> Content-Type: application/json
>
< HTTP/1.1 200 OK
< Content-Type: application/json
< Content-Length: 1404
< {
  "orders" : [
    {
      "amount" : "9.99",
      "amount_charged" : "9.99",
      "amount_refunded" : "9.99",
      "auth_code" : "AUTH12",
      "card" : {
        "holder" : "Ivan Ivanov",
        "subtype" : "classic",
        "type" : "visa"
      },
      "client" : {},
      "created" : "2016-08-22 15:39:56",
      "currency" : "RUB",
```

```
"custom_fields" : {},
"description" : null,
"descriptor" : "TESTMERCH/TERM",
"id" : "23014808614292147",
"issuer" : {
    "bin" : "411111",
    "country" : "RUS",
    "title" : "TestBank"
},
"location" : {
    "ip" : "6.6.6.6"
},
"merchant_order_id" : null,
"operations" : [
    {
        "amount" : "9.99",
        "auth_code" : "AUTH12",
        "id" : "47575",
        "cashflow" : {
            "amount" : "0.00",
            "currency" : "RUB",
            "fee" : "0.00",
            "incoming" : "0.00",
            "receivable" : "0.00",
            "reserve" : "0.00"
        },
        "created" : "2016-08-22 15:39:56",
        "currency" : "RUB",
        "iso_message" : "Approved",
        "iso_response_code" : "00",
        "status" : "success",
        "type" : "authorize"
    },
    {
        "amount" : "9.99",
        "auth_code" : "AUTH12",
        "id" : "47576",
        "cashflow" : {
            "amount" : "9.99",
            "currency" : "RUB",
            "fee" : "0.30",
            "incoming" : "9.69",
            "receivable" : "9.69",
            "reserve" : "0.00"
        },
        "created" : "2016-08-22 15:39:56",
```

```

        "currency" : "RUB",
        "iso_message" : "Approved",
        "iso_response_code" : "00",
        "status" : "success",
        "type" : "charge"
    },
    {
        "amount" : "9.99",
        "auth_code" : "AUTH12",
        "id" : "47577",
        "cashflow" : {
            "amount" : "-9.99",
            "currency" : "RUB",
            "fee" : "0.00",
            "incoming" : "-9.99",
            "receivable" : "-9.99",
            "reserve" : "0.00"
        },
        "created" : "2016-08-22 15:39:57",
        "currency" : "RUB",
        "iso_message" : "Approved",
        "iso_response_code" : "00",
        "status" : "success",
        "type" : "refund"
    }
],
"pan" : "411111****1111",
"secure3d" : {},
"status" : "refunded",
"updated" : "2016-08-22 15:39:57"
}
]
}

```

Повторный платеж (POST /orders/:id/rebill)

Обязательные параметры

Название	Описание	Пример
<code>amount</code>	Сумма	9.99
<code>cvv</code>	CVV-код (может быть отключен в настройках проекта)	123
<code>location</code>	Местоположение клиента (*)	—
<code>*.ip</code>	IP-адрес	8.8.8.8

Необязательные параметры

Название	Описание	Пример
----------	----------	--------

options	Опции (*)	—
*.recurring	Флаг рекуррентного платежа	1

Остальные необязательные параметры те же, что и в [Авторизации](#).

Ответ

Поля ответа повторяют набор полей из раздела [Информация о заказе](#).

Пример:

```
curl -v -X POST -H 'Content-Type: application/json' -d '{
  "amount" : 9.99,
  "cvv" : "123"
}' 'http://project:***@api.box:5001/orders/23014808614292147/rebill'

> POST /orders/23014808614292147/rebill HTTP/1.1
> Authorization: Basic cHJvamVjdDpwYXNzd29yZA==
> Host: api.box:5001
> Content-Type: application/json
> Content-Length: 27
>
< Content-Type: application/json
< Content-Length: 815
< {
  "orders" : [
    {
      "amount" : "9.99",
      "amount_charged" : "0.00",
      "amount_refunded" : "0.00",
      "auth_code" : "AUTH12",
      "card" : {
        "holder" : "Ivan Ivanov",
        "subtype" : "classic",
        "type" : "visa"
      },
      "client" : {},
      "created" : "2016-08-22 15:39:58",
      "currency" : "RUB",
      "custom_fields" : {},
      "description" : null,
      "descriptor" : "TESTMERCH/TERM",
      "id" : "23014809139351770",
      "issuer" : {
        "bin" : "411111",
        "country" : "RUS",
        "title" : "TestBank"
```

```

    },
    "location" : {},
    "merchant_order_id" : null,
    "operations" : [
        {
            "amount" : "9.99",
            "auth_code" : "AUTH12",
            "id" : "47575",
            "cashflow" : {
                "amount" : "0.00",
                "currency" : "RUB",
                "fee" : "0.00",
                "incoming" : "0.00",
                "receivable" : "0.00",
                "reserve" : "0.00"
            },
            "created" : "2016-08-22 15:39:58",
            "currency" : "RUB",
            "iso_message" : "Approved",
            "iso_response_code" : "00",
            "status" : "success",
            "type" : "authorize"
        }
    ],
    "pan" : "411111****1111",
    "secure3d" : {},
    "status" : "authorized",
    "updated" : "2016-08-22 15:39:58"
}
1
}

```

Выплата на карту (OCT, Original credit transaction) (POST /orders/:id/credit)

Обязательные параметры

Название	Описание	Пример
amount	Сумма	9.99
currency	Валюта	RUB
location	Местоположение клиента (*)	—
*.ip	IP-адрес	8.8.8.8

Необязательные параметры

Необязательные параметры те же, что и в [Авторизации](#).

Ответ

Поля ответа повторяют набор полей из раздела [Информация о заказе](#).

Пример:

```
curl -v -X POST -H 'Content-Type: application/json' -d '{
  "amount" : 9.99,
  "currency" : "RUB"
  "location" : {
    "ip" : "8.8.8.8"
  }
}' 'http://project:***@api.box:5001/orders/23014808614292147/credit'

> POST /orders/23014808614292147/credit HTTP/1.1
> Authorization: Basic cHJvamVjdDpwYXNzd29yZA==
> Host: api.box:5001
> Content-Type: application/json
> Content-Length: 27
>
< Content-Type: application/json
< Content-Length: 815
<
{
  "orders" : [
    {
      "amount" : "9.99",
      "amount_charged" : "0.00",
      "amount_refunded" : "0.00",
      "auth_code" : "AUTH12",
      "card" : {
        "holder" : "Ivan Ivanov",
        "subtype" : "classic",
        "type" : "visa"
      },
      "client" : {},
      "created" : "2016-08-22 15:39:58",
      "currency" : "RUB",
      "custom_fields" : {},
      "description" : null,
      "descriptor" : "TESTMERCH/TERM",
      "id" : "23014809139351770",
      "issuer" : {
        "bin" : "411111",
        "country" : "RUS",
        "title" : "TestBank"
      },
      "location" : {},
```

```

    "merchant_order_id" : null,
    "operations" : [
      {
        "amount" : "9.99",
        "auth_code" : "AUTH12",
        "id" : "47575",
        "cashflow" : {
          "amount" : "0.00",
          "currency" : "RUB",
          "fee" : "0.00",
          "incoming" : "0.00",
          "receivable" : "0.00",
          "reserve" : "0.00"
        },
        "created" : "2016-08-22 15:39:58",
        "currency" : "RUB",
        "iso_message" : "Approved",
        "iso_response_code" : "00",
        "status" : "success",
        "type" : "credit"
      }
    ],
    "pan" : "411111****1111",
    "secure3d" : {},
    "status" : "credited",
    "updated" : "2016-08-22 15:39:58"
  }
]
}

```

Завершение 3-D Secure аутентификации (POST /orders/:id/complete)

Обязательные параметры

Название	Описание	Пример
PaRes	Ответ системы аутентификации (ACS)	424EF64895CAE9797497F

Пример:

```

curl -X POST "http://project:***@api.box:5001/orders/43809080318125930/complete" -H
"accept: application/json" -H "authorization: Basic
bWVYy2hhbnRfdGVzdDpQQVFpemtWeCtDULJVbTEr" -H "Content-Type: application/json" -d "
{ \"PaRes\": \"YAED53A345D\"}"
{
  "orders": [
    {

```

```

        "description": "Book sale 453",
        "auth_code": "AUTH12",
        "amount": "100.99",
        "merchant_order_id": null,
        "id": "43809080318125930",
        "updated": "2019-03-20 18:47:03",
        "pan": "411111****1111",
        "currency": "RUB",
        "descriptor": "321/123",
        "amount_charged": "100.99",
        "created": "2019-03-20 18:46:15",
        "status": "charged",
        "amount_refunded": "0.00"
    }
}

```

Примечание

Эндпойнт передает ответ на запрос авторизации. При повторной отправке запроса на эндпойнт будет получен ответ: "Order already completed".

Дополнительные возможности

Динамический дескриптор

Эта опция доступна, если ее поддерживает банк-эквайер.

Динамический дескриптор позволяет передавать имя и местоположение продавца, которые попадут в банковскую выписку держателя карты и будут указаны в SMS, посылаемом банком эмитентом.

Название и местоположение продавца, установленные при регистрации торговой точки (MID) в банке, можно переопределять актуальным значением для каждого платежа. Динамический дескриптор поддерживается в операциях [Authorize](#), [Create](#) и [Rebill](#).

Данные динамического дескриптора должны быть указаны в блоке `extra_fields`:

Обязательные параметры

Название	Описание	Пример
<code>dynamic_descriptor</code>	Динамический дескриптор (*)	—
<code>*.merchant</code>	Имя продавца	BOOKSTORE LTD
<code>*.city</code>	Город продавца	Moscow
<code>*.country</code>	Страна продавца (ISO 3166-1 alpha-2)	RU

Согласно значениям, приведенным в примере, дескриптор, полученный владельцем карты, может выглядеть следующим образом:

Примечание

Суммарная длина значений полей *merchant* и *city* не должна превышать 37 символов, при этом длина значения *merchant* не должна превышать 22 символов. Значения обоих параметров могут содержать только цифро-буквенные символы и знаки препинания.

Поддержка длинной записи (Airline Addendum)

Для идентификации платежей, связанных с заказом авиабилетов, есть возможность передавать длинную запись в запросах [Create](#), [Authorize](#), [Rebill](#), [Charge](#).

Длинная запись передается эквайеру при списании средств. Если в запросе [Charge](#) указана длинная запись, отличающаяся от той, что была указана при авторизации или создании заказа, то она будет переопределена.

Длинная запись передается в блоке *extra_fields*:

Обязательные параметры

Название	Описание	Пример
<i>airline_addendum</i>	Длинная запись (*)	—
<i>*.raw_data</i>	Данные длинной записи	01QP1INU1237...3JOHNDOO

Уникальная конверсия платежей

Если при платеже передается уникальный идентификатор заказа (инвойса), все заказы, имеющие один и тот же идентификатор, при анализе конверсии на платежной странице могут считаться одной попыткой. Это позволяет более точно учитывать конверсию в тех случаях, когда покупатель делает несколько попыток платежей разными картами, чтобы оплатить один заказ.

Уникальный идентификатор заказа можно передавать в запросах [Create](#), [Authorize](#), [Rebill](#).

Идентификатор передается в блоке *extra_fields*:

Название	Описание	Пример
<i>attributes</i>	Параметры (*)	—
<i>*.invoice_id</i>	Уникальный идентификатор заказа	741852963

DSRP-платежи

Данные DSRP-платежа передаются в параметре *secure3d*

Название	Описание	Пример
<i>secure3d</i>	Данные аутентификации 3-D Secure (*)	—

*.dsrp	Признак DSRP-платежа	1
*.xid	Идентификатор транзакции из PaRes (XID)	8Rg2xYbxLUGsfZlkoh1JxAQFBM=
*.cavv	Значение CAVV из PaRes	AAABCBVRA2EYZXQBV1E DAAAAAAA=
*.eci	Признак ECI	05

При работе с DSRP-платежами мерчант получает токен, в котором содержатся реквизиты для проведения платежа. Заполнение параметров на примере ApplePay:

Название в токене	Название в API
applicationPrimaryAccountNumber	pan
applicationExpirationDate	expiration_month expiration_year
transactionAmount	amount
transactionId	xid
onlinePaymentCryptogram	cavv
eciIndicator	eci

Поля *holder* и *cvv* не передаются, т.к. они не содержатся в токене, *dsrp* всегда имеет значение 1, *eci* передается только, если содержится в токене.

Привязка платежной карты (Oneclick)

Привязка платежной карты на платежной странице дает возможность держателю сохранить данные своей карты при первом платеже, для того чтобы при последующих платежах он смог выбрать эту карту из списка своих привязанных карт и провести платеж с вводом только CVV-кода. Для того, чтобы карта привязалась, пользователь должен отметить соответствующий чекбокс на платежной странице.

Чтобы задействовать механизм привязки карты покупателя, в запросе создания заказа необходимо указать уникальный идентификатор пользователя в системе мерчанта. Если не передать этот идентификатор, пользователь увидит обычную платежную страницу без сохраненных карт и без опции сохранения введенной карты.

Идентификатор передается в блоке *extra_fields* запроса [Create](#), допускается передавать цифробуквенное значение длиной до 40 символов.

Название	Описание	Пример
oneclick	Привязка карты (*)	—
*.customer_id	Идентификатор покупателя	0123654789

Фискализация по 54-ФЗ

Общая информация

Система позволяет передать вместе с платежом данные для формирования электронного чека. Чек будет зарегистрирован с помощью одного из провайдеров фискализации и после оформления его реквизиты станут доступны для просмотра в бэкофисе мерчанта и в API.

Регистрация чека происходит только после успешной операции списания [Charge](#), оформление проводится асинхронно через некоторое время после регистрации.

В зависимости от используемой схемы интеграции, данные чека можно передать в запросах [Create](#), [Authorize](#), [Rebill](#), [Charge](#).

Если в интеграции не используется явное проведение списаний через API (одностадийная схема или автоклиринг), данные чека следует передавать в запросах на оплату (операции [Create](#), [Authorize](#), [Rebill](#)). Если используется явный вызов списания, данные чека можно передать непосредственно в операцию [Charge](#), при этом ранее переданные данные по этому платежу будут перезаписаны.

При оформлении возврата данные чека "Возврат прихода" передаются вместе с операцией [Refund](#) при проведении ее через API, независимо от схемы интеграции. Если провести операцию возврата через бэкофис, или не передать данные чека в API, возврат средств будет проведен без оформления электронного чека.

Регистрация чека

Данные для формирования чека следует передавать в блоке `extra_fields`:

Обязательные параметры

Name	Description	Example
<code>electronic_receipt</code>	Электронный чек (*)	—
<code>*.phone</code>	Телефон покупателя	9111231212
<code>*.email</code>	E-mail покупателя	mail@server.example
<code>*.inn</code>	ИНН продавца	12345678
<code>*.payment_address</code>	Адрес места расчетов	shop.ru
<code>*.taxation_type</code>	Система налогообложения продавца	1
<code>*.items</code>	Список товарных позиций	—
<code>*.*.name</code>	Наименование	Книга
<code>*.*.price</code>	Цена	300.25
<code>*.*.quantity</code>	Количество	2
<code>*.*.amount</code>	Стоимость	600.50
<code>*.*.tax_type</code>	Ставка НДС	vat18

Опциональные параметры товарной позиции `*.items`

Name	Description	Example
<code>*.*.unit</code>	Единица измерения товара	шт.
<code>*.*.payment_object</code>	Предмет расчета	goods
<code>*.*.payment_method</code>	Способ расчета	full_payment
<code>*.*.agent</code>	Атрибуты агента	—
<code>*.*.*.type</code>	Тип агента	attorney

<code>*.*.*.operation</code>	Операция платежного агента	Платеж
<code>*.*.*.phone_numbers</code>	Телефоны платежного агента (массив строк)	["79111231212", "74959999999"]

В параметре `*.items` передается массив структур с товарными позициями. В таблице перечислены поля для структуры с одной товарной позицией.

В параметре `*.taxation_type` указывается система налогообложения продавца в соответствии с таблицей ниже:

Значение	Система налогообложения
0	Общая
1	Упрощенная (доходы)
2	Упрощенная (доходы минус расходы)
3	Единый налог на вмененный доход
4	Единый сельскохозяйственный налог
5	Патентная

В параметре товарной позиции `*.*.tax_type` указывается ставка, по которой товар облагается НДС:

Значение	Описание
<code>none</code>	НДС не облагается
<code>vat0</code>	Ставка НДС 0%
<code>Vat20</code>	Ставка НДС 20%
<code>vat10</code>	Ставка НДС 10%
<code>vat118</code>	Ставка НДС из расчета 18/118
<code>vat110</code>	Ставка НДС из расчета 10/110

В параметре товарной позиции `*.*.payment_object` указывается признак предмета расчета:

Значение	Описание
<code>goods</code>	Товары
<code>excisable_goods</code>	Подакцизные товары
<code>work</code>	Работа
<code>service</code>	Услуги
<code>gambling_bet</code>	Ставка азартной игры
<code>gambling_prize</code>	Выигрыш азартной игры
<code>lottery_ticket</code>	Лотерейный билет
<code>lottery_prize</code>	Выигрыш лотереи
<code>intellectual_property</code>	Предоставление результатов интеллектуальной деятельности
<code>payment</code>	Платеж
<code>agent_commission</code>	Агентское вознаграждение
<code>composite</code>	Составной предмет расчета
<code>other</code>	Другой предмет расчета

В параметре товарной позиции `*.*.payment_method` указывается признак способа расчета:

Значение	Описание
<code>full_payment</code>	Полная оплата
<code>partial_payment</code>	Частичная оплата
<code>full_prepayment</code>	Полная предоплата
<code>partial_prepayment</code>	Частичная предоплата
<code>advance_payment</code>	Аванс
<code>credit</code>	Передача в кредит
<code>credit_payment</code>	Оплата кредита

В параметре атрибутов агента для товарной позиции `*.*.agent.type` указывается тип платежного агента:

Значение	Описание
<code>bank_paying_agent</code>	Банковский платежный агент
<code>bank_paying_subagent</code>	Банковский платежный субагент
<code>paying_agent</code>	Платежный агент
<code>paying_subagent</code>	Платежный субагент
<code>attorney</code>	Поверенный
<code>commission_agent</code>	Комиссионер
<code>Other</code>	Другой агент

Пример:

```
{
...
"extra_fields":{
  "electronic_receipt":{
    "email":"test@mail.example",
    "items":[
      {
        "name":"Товар1",
        "amount":200,
        "price":200,
        "quantity":1,
        "tax_type":"vat20"
      },
      {
        "name":"Товар2",
        "amount":300,
        "price":300,
        "quantity":1,
        "tax_type":"vat20"
      }
    ]
  },
  ...
}
```

```

    "payment_address": "testshop.example",
    "taxation_type": "1",
    "inn": "7736642031",
    "phone": "+79111231212"
  }
}

```

Запрос реквизитов чека

Все чеки, связанные с ордером, передаются в ответ на запрос [Информации о заказе](#) в списке *electronic_receipts*. Каждый элемент списка содержит следующие поля:

Имя	Описание	Пример
<code>status</code>	Состояние	success
<code>operation</code>	Операция	refund
<code>created</code>	Дата создания	2017-11-01 18:41:15
<code>operation_id</code>	ID операции	47071
<code>attributes</code>	Атрибуты (*)	—
<code>*.shift_receipt_number</code>	Номер чека в смене	2779
<code>*.shift_number</code>	Номер смены.	7971
<code>*.registered</code>	Дата и время документа из ФН.	2017-11-01 18:45:00
<code>*.fiscal_storage_number</code>	Номер ФН.	9999078900001341
<code>*.device_registration_number</code>	Регистрационный номер ККТ.	0000000400054952
<code>*.receipt_number</code>	Фискальный номер документа.	1290
<code>*.receipt_attribute</code>	Фискальный признак документа.	819694071
<code>*.fns_website</code>	Адрес сайта ФНС.	www.nalog.ru

Пример:

```

{
  "orders" : [
    {
      "amount" : "10.00",
      "amount_charged" : "10.00",
      "amount_refunded" : "0.00",
      "auth_code" : "AUTH12",
      "created" : "2022-07-04 15:40:50",

```

```
"currency" : "RUB",
"description" : "ТЕСТ ",
"descriptor" : "MerchID/TestID",
"electronic_receipts" : [
  {
    "attributes" : {
      "device_registration_number" : "0000000000008144",
      "fiscal_storage_number" : "9999078900006846",
      "fns_website" : "www.nalog.ru",
      "receipt_attribute" : "1956641512",
      "receipt_number" : 1259,
      "registered" : "2022-07-04 15:41:00",
      "shift_number" : 233,
      "shift_receipt_number" : 6
    },
    "created" : "2022-07-04 15:41:41",
    "operation" : "charge",
    "operation_id" : "47014",
    "status" : "success"
  }
],
"id" : "70392435330005735",
"merchant_order_id" : "1112538967120549295",
"operations" : [
  {
    "amount" : "10.00",
    "auth_code" : "AUTH12",
    "created" : "2022-07-04 15:41:41",
    "currency" : "RUB",
    "id" : "47013",
    "iso_message" : "Approved",
    "iso_response_code" : "00",
    "status" : "success",
    "type" : "authorize"
  },
  {
    "amount" : "10.00",
    "auth_code" : "AUTH12",
    "created" : "2022-07-04 15:41:41",
    "currency" : "RUB",
    "id" : "47014",
    "iso_message" : "Approved",
    "iso_response_code" : "00",
    "status" : "success",
    "type" : "charge"
  }
]
```

```
    },
    "pan" : "411111****1111",
    "status" : "charged",
    "updated" : "2022-07-04 15:41:41"
  }
}
}
```

Тестирование

Общая информация

Тестовые заказы проводятся только через тестовый терминал. Доступ к тестовому терминалу можно получить через службу поддержки.

В тестовом режиме допускается использовать любые тестовые номера карт с корректными датой истечения, секретным кодом и именем держателя. Тестовый терминал успешно проводит попытки оплаты с любыми реквизитами, которые прошли базовую валидацию формата.

Для получения определенного результата платежа можно использовать специальные номера карт:

Номер карты	Описание
4111111111111111	Успешная авторизация
2222400060000007	Успешная авторизация
4276990011343663	Без 3-D Secure: отклоненная авторизация;
	3-D Secure: сценарий "Unable to verify enrollment".
5555555555555599	Внутренняя ошибка эквайера или шлюза
4276838748917319	Сценарий 3-D Secure "Not enrolled"
4000000000000002	Отклонено системой защиты от мошеннических операций

Подключение к тестовой среде

Параметры подключения

Адрес сервера: <https://api.ecom-sandbox.2can.ru>

Параметры basic-авторизации:

login: testproject

password: pass123123123

Тестовая карта: 4111 1111 1111 1111 (срок действия, CVV, держатель карты – любые);

Примеры запросов на PHP

Create

```
<?
// Url на который будет отправлен запрос на создание заказа
$apiurl = "https://api.ecom-sandbox.2can.ru/orders/create";

// Данные для авторизации
$apilogin = 'testproject';
$apipassword = 'pass123123123';

// Параметры заказа
$request = [
    /* Обязательные параметры */
    "amount" => '10000.00', // Сумма заказа

    /* Необязательные параметры */
    "currency" => "RUB", // Валюта заказа
    "merchant_order_id" => '123', // Идентификатор заказа в системе мерчанта
    "description" => 'Заказ № 123', // Описание заказа - выводится в заголовке формы
оплаты

    "client" => [ // Данные о клиенте
        "address" => "Новый Арбат, 1",
        "city" => "Москва",
        "country" => "RU",
        "login" => "vaspet123",
        "email" => "vaspet@somedomain.ru",
        "name" => "Василий Петров",
        "phone" => "+7 777 777 77 77",
        "state" => "Москва",
        "zip" => "123456"
    ],

    "location" => [ // Местоположение клиента
        "ip" => $_SERVER['REMOTE_ADDR'], // IP-адрес
    ],

    "custom_fields" => [ // Набор пользовательских полей
        "somefiled" => 'Кастомное поле'
    ],

    "options" => [
        "expiration_timeout" => '25m', // Время сессии
        "force3d" => '1', // Принудительное направление в 3-D Secure
        "auto_charge" => '0', //Провести операцию Charge (используется (для одностадийных
платежей)
```

```

        "language" => 'ru', // Язык платежной формы
        "return_url" => 'http://' . $_SERVER['HTTP_HOST'] . '/success/', // URL для
перенаправления клиента после оплаты
        "template" => '2', // Шаблон платежной формы
        "mobile" => '1', // Мобильный шаблон
        "terminal" => 'TestClient2' // Терминал
    ]
};

// Подготовка заголовка запроса
$headers = array(
    'Content-Type:application/json',
    'Authorization: Basic ' . base64_encode($apilogin . ":" . $apipassword) // Заголовок
для авторизации
);

$curl = curl_init();

curl_setopt_array($curl, array(
    CURLOPT_HTTPHEADER => $headers, // Заголовок запроса (авторизация)
    CURLOPT_URL => $apiurl, // URL на который отправлять запрос
    CURLOPT_RETURNTRANSFER => true, // Возвращать тело ответа
    CURLOPT_POST => true, // POST запрос
    CURLOPT_POSTFIELDS => json_encode($request), // Переводим параметры запроса в JSON и
добавляем в тело запроса
    CURLOPT_HEADER => true, // Вывод заголовков ответа в тело ответа
));

// Отправка запроса
$result = curl_exec($curl); // Тело ответа
$info = curl_getinfo($curl); // Информация по ответу

curl_close($curl);

if($result == false)
{
    echo "Ошибка CURL: " . curl_error($curl);
    return false;
}
else
{
    // Разбор ответа

    $resultheaders = substr($result, 0, $info["header_size"]); // Получаем заголовок
ответа из тела ответа

    $resultbody = json_decode(substr($result, $info["header_size"]), true); // Убираем
заголовок из тела, и декодируем JSON

```

```

if(!isset($resultbody["errors"]) && in_array($info['http_code'], ['200', '201'])){
    // Получаем URL формы оплаты из заголовка ответа полу "Location"
    preg_match("!\\r\\n(?:Location): *(.*)? *\\r\\n!", $resultheaders, $matches);
    $redirect_url = $matches[1];

    // Перенаправляем пользователя на форму оплаты
    header('Location: ' . $redirect_url);
}else{
    /*
        Код обработки ошибок
        $resultbody["failure_type"] - Тип ошибки
        $resultbody["failure_message"] - Сообщение ошибки
        $resultbody["errors"] - Массив с подробностями ошибок

    */
}
}
?>

```

GetOrder

```

<?
// Номер заказа шлюза
$orderid = '69747380516125001';

// Url на который будет отправлен запрос на создание заказа с указанием полей, которые
необходимо получить
$apiurl = sprintf('https://api.ecom-
sandbox.2can.ru/orders/%s?expand=card,client,location,custom_fields,issuer,secure3d,operat
ions.cashflow', $orderid);

// Данные для авторизации
$apilogin = 'testproject';
$apipassword = pass123123123;

// Подготовка заголовка запроса
$headers = array(
    'Content-Type:application/json',
    'Authorization: Basic '. base64_encode($apilogin . ":" . $apipassword) // Заголовок
для авторизации
);

$curl = curl_init();

curl_setopt_array($curl, array(
    CURLOPT_HTTPHEADER => $headers, // Заголовок запроса (авторизация)

```



```

    CURLOPT_URL => $apiurl, // URL на который отправлять запрос
    CURLOPT_RETURNTRANSFER => true, // Возвращать тело ответа
));

// Отправка запроса
$result = curl_exec($curl); // Тело ответа
$info = curl_getinfo($curl); // Информация по ответу

curl_close($curl);

if($result == false)
{
    echo "Ошибка CURL: " . curl_error($curl);
    return false;
}
else
{
    // Разбор ответа
    $resultbody = json_decode($result, true); // Декодируем JSON ответа

    if(!isset($resultbody["errors"]) && in_array($info['http_code'], ['200', '201'])) {
        /*
            Код обработки информации по заказу
            $resultbody["orders"][0] - массив данных по заказу
        */

    }else{
        /*
            Код обработки ошибок
            $resultbody["failure_type"] - Тип ошибки
            $resultbody["failure_message"] - Сообщение ошибки
            $resultbody["errors"] - Массив с подробностями ошибок
        */
    }
}
?>

```

Refund

```

<?
// Номер заказа шлюза
$orderid = '69746955116652401';

// Url на который будет отправлен запрос на возврат средств заказа
$apiurl = sprintf('https://api.ecom-sandbox.2can.ru/orders/%s/refund', $orderid);

// Данные для авторизации

```

```

$apilogin = 'testproject';
$apipassword = 'pass123123123';

// Параметры заказа
$request = [
    /* Обязательные параметры */
    "amount" => '10000.00', // Сумма для возврата
];

// Подготовка заголовка запроса
$headers = array(
    'Content-Type:application/json',
    'Authorization: Basic ' . base64_encode($apilogin . ":" . $apipassword) // Заголовок
    для авторизации
);

$curl = curl_init();

curl_setopt_array($curl, array(
    CURLOPT_HTTPHEADER => $headers, // Заголовок запроса (авторизация)
    CURLOPT_URL => $apiurl, // URL на который отправлять запрос
    CURLOPT_RETURNTRANSFER => true, // Возвращать тело ответа
    CURLOPT_POST => true, // Отправлять данные в теле запроса
    CURLOPT_CUSTOMREQUEST => 'PUT', // Типа запроса PUT
    CURLOPT_POSTFIELDS => json_encode($request), // Переводим параметры запроса в JSON и
    добавляем в тело запроса
));

// Отправка запроса
$result = curl_exec($curl); // Тело ответа
$info = curl_getinfo($curl); // Информация по ответу

curl_close($curl);

if($result == false)
{
    echo "Ошибка CURL: " . curl_error($curl);
    return false;
}
else
{
    // Разбор ответа
    $resultbody = json_decode($result, true); // Декодируем JSON ответа

    if(!isset($resultbody["failure_type"]) && in_array($info['http_code'], ['200', '201']))
    {

```

```

    /*
        Код обработки информации по заказу
        $resultbody["orders"][0] - массив данных по заказу
    */
}else{
    /*
        Код обработки ошибок
        $resultbody["failure_type"] - Тип ошибки
        $resultbody["failure_message"] - Сообщение ошибки
        $resultbody["errors"] - Массив с подробностями ошибок

    */
}
}
?>

```

Rebill

```

<?
/// Номер заказа шлюза
$orderid = '69747525875326582';

// Url на который будет отправлен запрос на повторный платеж по заказу
$apiurl = sprintf('https://api.ecom-sandbox.2can.ru/orders/%s/rebill', $orderid);

// Данные для авторизации
$apillogin = 'testproject';
$apipassword = 'pass123123123';

// Параметры заказа
$request = [
    /* Обязательные параметры */
    "amount" => '10000.00', // Сумма заказа
    "cvv" => "123", // CVV-код (может быть отключен в настройках проекта)
    "location" => [ // Местоположение клиента
        "ip" => $_SERVER['REMOTE_ADDR'], // IP-адрес
    ],

    /* Необязательные параметры */
    "currency" => "RUB", // Валюта заказа
    "merchant_order_id" => '123', // Идентификатор заказа в системе мерчанта
    "description" => 'Заказ № 123', // Описание заказа - выводится в заголовке формы
оплаты

    "client" => [ // Данные о клиенте

```

```

        "address" => "Новый Арбат, 1",
        "city" => "Москва",
        "country" => "RU",
        "login" => "vaspet123",
        "email" => "vaspet@somedomain.ru",
        "name" => "Василий Петров",
        "phone" => "+7 777 777 77 77",
        "state" => "Москва",
        "zip" => "123456"
    ],

    "custom_fields" => [ // Набор пользовательских полей
        "somefield" => 'Кастомное поле'
    ],

    "options" => [
        "expiration_timeout" => '25m', // Время сессии
        "force3d" => '1', // Принудительное направление в 3-D Secure
        "auto_charge" => '0', //Провести операцию Charge (используется (для одностадийных платежей)
        "language" => 'ru', // Язык платежной формы
        "return_url" => 'http://' . $_SERVER['HTTP_HOST'] . '/success/', // URL для перенаправления клиента после оплаты
        "template" => '2', // Шаблон платежной формы
        "mobile" => '1', // Мобильный шаблон
        "terminal" => 'TestClient2', // Терминал,
        "recurring" => '0', // Флаг рекуррентного платежа
    ]
];

// Подготовка заголовка запроса
$headers = array(
    'Content-Type:application/json',
    'Authorization: Basic ' . base64_encode($apilogin . ":" . $apipassword) // Заголовок для авторизации
);

$curl = curl_init();

curl_setopt_array($curl, array(
    CURLOPT_HTTPHEADER => $headers, // Заголовок запроса (авторизация)
    CURLOPT_URL => $apiurl, // URL на который отправлять запрос
    CURLOPT_RETURNTRANSFER => true, // Возвращать тело ответа
    CURLOPT_POST => true, // POST запрос
    CURLOPT_POSTFIELDS => json_encode($request), // Переводим параметры запроса в JSON и добавляем в тело запроса

```

```

));

// Отправка запроса
$result = curl_exec($curl); // Тело ответа
$info = curl_getinfo($curl); // Информация по ответу

curl_close($curl);

if($result == false)
{
    echo "Ошибка CURL: " . curl_error($curl);
    return false;
}
else
{
    // Разбор ответа
    $resultbody = json_decode($result,true); // Декодируем JSON ответа

    if(!isset($resultbody["errors"]) && in_array($info['http_code'], ['200', '201'])){
        echo $resultbody; // Вывод ответ для автоматического перенаправления на шлюз
        /*
            Код обработки информации по заказу
            $resultbody["orders"][0] - массив данных по заказу
        */
    }else{
        /*
            Код обработки ошибок
            $resultbody["failure_type"] - Тип ошибки
            $resultbody["failure_message"] - Сообщение ошибки
            $resultbody["errors"] - Массив с подробностями ошибок

        */
    }
}
?>

```