

# API Documentation

## Table of Contents

|  |   |
|--|---|
| Integration Reference .....                | 5 |
| General information and capabilities ..... | 5 |
| Features .....                             | 5 |
| Orders and operation paradigm .....        | 5 |
| Order status .....                         | 5 |
| Operation status .....                     | 6 |
| Workflows .....                            | 6 |
| Using payment page .....                   | 6 |
| Server-to-server .....                     | 6 |
| Charge-off through back-office .....       | 6 |
| API overview .....                         | 6 |
| General information .....                  | 7 |
| Authentication .....                       | 7 |
| Response codes .....                       | 7 |
| Failure types .....                        | 7 |
| Working flow .....                         | 7 |
| Verifying successful operation .....       | 7 |
| Error handling .....                       | 8 |
| Processing declined transactions .....     | 8 |
| Validation .....                           | 8 |
| Sample code .....                          | 8 |
| Actions .....                              | 9 |
| Test request (GET /ping) .....             | 9 |
| Fetching data .....                        | 9 |

|   |    |
|---|----|
| Order information (GET /orders/:id) . . . . .   | 10 |
| List of orders (GET /orders/) . . . . .   | 12 |
| List of operations (GET /operations/) . . . . .   | 13 |
| Export of currency rates (GET /exchange_rates/) . . . . .                               | 14 |
| Working with orders . . . . .   | 15 |
| Order creation (POST /orders/create) . . . . .  | 15 |
| Authorize (POST /orders/authorize) . . . . .  | 17 |
| Reverse (PUT /orders/:id/reverse) . . . . .   | 22 |
| Charge (PUT /orders/:id/charge) . . . . .   | 22 |
| Refund (PUT /orders/:id/refund) . . . . .   | 23 |
| Cancel (PUT /orders/:id/cancel) . . . . .   | 25 |
| Repeating payment (POST /orders/:id/rebill) . . . . .                                   | 25 |
| Original credit transaction (OCT) (POST /orders/:id/credit) . . . . .                   | 26 |
| Original credit transaction without link to order (OCT) (POST /orders/credit) . . . . . | 26 |
| Completion of 3-D Secure Authentication (POST /orders/:id/complete) . . . . .           | 28 |
| Completion of 3-D Secure 2.0 Authentication (POST . . . . .                             | 29 |
| /orders/:id/complete3d20) . . . . .   | 30 |
| Continue 3DS Method URL of authentication (POST /orders/:id/resume) . . . . .           | 31 |
| Additional capabilities . . . . .   | 32 |
| Dynamic descriptor . . . . .  | 32 |
| Unique conversion of payments . . . . .   | 33 |
| Payments with 3-D Secure 2.0 . . . . .  | 33 |
| DSRP . . . . .  | 34 |
| Binding a payment card (Oneclick) . . . . .   | 34 |
| Paged downloads of lists of orders and operations . . . . .                             | 35 |
| Testing . . . . .   | 35 |

Overview ..... 35

# Integration Reference

## General information and capabilities

Gateway's API is built in a REST style, based on the JSON-API specification. API provides a universal programmatic access to operate through payment page or directly in a host-to-host way.

### Features

- JSON-API based on RESTful JSON API standard.
- Advanced authentication using client certificates.
- 1-stage and 2-stage payments.
- Payment flows: payment page or server-to-server using universal API.
- 3-D Secure support.
- Repeating payment without providing card number, expiration date and cardholder name.
- Optional fields for extra parameters.
- Segment selection to divide payments .
- Selection of the payment page language and layout.
- Setting payment session expiration timeout.
- Payments to the card (OCT).
- Passing of dynamic descriptor.
- Recurrent payments.
- Multicurrency.
- Perform authorizations with zero-amount.
- DSRP.
- 3-D Secure 2.0 support.

### Orders and operation paradigm

Payments are represented as orders. Orders keeps general information about payment as well as a cardholder details. Transactions are represented as a sequence of operations associated with the order, for instance: authorize, charge, refund.

Order is associated with a unique ID upon creation, which is used in following requests regarding this order.

### Order status

Status represents the current order state.

| Status                  | Description   | Possible operations                  |
|-------------------------|---|--------------------------------------|
| <code>new</code>        | Order has been created, but has yet to be processed | —                                    |
| <code>prepared</code>   | Waiting for outside action                          | —                                    |
| <code>authorized</code> | Authorization completed                             | <code>reverse, charge, rebill</code> |
| <code>charged</code>    | Order has been charged                              | <code>refund, credit, rebill</code>  |
| <code>reversed</code>   | Authorization has been cancelled                    | <code>rebill</code>                  |
| <code>refunded</code>   | Order was fully or partially refunded               | <code>refund, credit, rebill</code>  |
| <code>rejected</code>   | Order has been rejected                             | <code>rebill</code>                  |
|                         | Order has been determined as a fraud and rejected   |                                      |

|             |   |        |
|-------------|---|--------|
| fraud       | because of that   | rebill |
| declined    | Order has been declined by the acquirer   | rebill |
| chargedback | Order is marked as chargeback   | rebill |
| credited    | Order has been credited (OCT)   | rebill |
| error       | An error occurred during order processing. You should probably contact the support team | rebill |

## Operation status

Status represents the final result of processing the operation.

| Status  | Description  |
|---------|--|
| success | Operation processed successfully   |
| failure | Operation has been declined (for instance, an authorization decline occurred)                  |
| error   | An error occurred during processing of operation. You should probably contact the support team |

## Workflows

### Using payment page

1. Customer presses the "Pay" button on a seller's website.
2. The seller handles customer's payment request and sends a `create` request to the API, specifying description of the purchase, payment amount and other details if needed (see [Order creation](#)).
3. Gateway responds with a `URI` of the payment page and order ID.
4. The seller saves order ID and redirects customer to the `URI`.
5. Customer fills up a payment form on the payment page.
6. Gateway processes authorization using cardholder's credentials and redirects cardholder back to the seller's website.
7. The seller requests order information using saved order ID (see [Order information](#)) and depending on the order status renders result page for the customer.

### Server-to-server

1. Customer enters payment info and presses "Pay" button while on a seller's website.
2. Sellers handles the customer's request and sends an `authorize` request to the API (see [Order authorization](#)).
3. Gateway creates order and handles payment using payment info received from seller.
4. Gateway responds with the authorization results.
5. Seller handles the authorization results and renders the result page for the customer.

### Charge-off through back-office

After successful payment via payment page is performed charge-off of funds from the customer's card such methods:

1. Enabling autocharge in the project (the delay of autocharge is set in minutes).
2. Manually charge-off of funds in the back-office ("Orders => Needed order => Charge").

## API overview

## General information

- Requests are made through HTTP 1.1 using SSL and *Basic-Authentication*.
- Requests and responses are encoded in *UTF-8*.
- Requests and responses are *JSON* structures.
- When getting information use method *GET*, when manipulation with data methods *POST* or *PUT* are required.
- In *GET*-requests parameters are passed as *QUERY STRING*.
- In *POST*-requests parameters are passed as a body of the *POST*-request, in some cases *QUERY STRING* could be used too.
- Success responses are sent with a *HTTP*-status 200 or 201. In case of an error the status may vary (see [Response codes](#)).

## Authentication

API uses *Basic-Authentication*. Login and password should be passed every time you send request to API.

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

## Response codes

HTTP response code describes the result of requested operation, as follows:

| Code | Description   |
|------|---|
| 200  | This code means the operation you requested has been processed successfully. Depending on which operation you have requested, it may tell that authorization, reversal, charge or refund has been made. |
| 201  | An order has been created and it is waiting for additional input.   |
| 402  | Operation has been declined by acquirer or rejected by gateway  |
| 422  | Request validation failed.  |
| 500  | Error on acquirer or gateway side.  |

## Failure types

Failure type describes the source of failure:

| Code       | Description  |
|------------|--|
| declined   | The acquirer has declined the transaction.                               |
| fraud      | Transaction has been declined by fraud prevention rules on gateway side. |
| rejected   | Transaction has been declined by gateway.                                |
| error      | Failure due to an error on acquirer or gateway side.                     |
| validation | Request not processed because of incorrect input.                        |

## Working flow

### Verifying successful operation

The successful response always returned with the code 200, details of an order are in the response body. To ensure that you have completed the operation, it is necessary and sufficient to check response code and order status in the response body.

## Error handling

All errors are returned as a standard failure response, error message can be found in the response body:

```
{
  "failure_type" : "error",
  "failure_message" : "This is an error",
  "order_id" : null,
}
```

Description of an error is located in a *failure\_message* field. If there is an order created during request, its identifier returned in an *order\_id* field.

## Processing declined transactions

All responses for transactions declined by the issuer are returned in a uniform format:

```
{
  "failure_message": "Soft-Denied",
  "order_id": "123456789",
  "failure_type": "declined"
}
```

The principle for forming a response is the same as for [handling errors](#).

If authorization is rejected according to the Soft-Denied scenario, it means that the request parameters are not meets normativs of SCA (Strong Customer Authentication). Such authorization must be done with the *force3d* flag.

In the order information for Soft-Denied, *iso\_response\_code* will be *A1*.

## Validation

In case of validation error there is an extra field *errors* returned which contains the list of incorrect parameters and some extra information.

### Example:

```
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 }
```

## Sample code

```

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 occured: " . response.failure_message;
}

```

## Actions

### Test request (GET /ping)

#### Parameters

Not supported.

#### Response

| Name    | Description                     | Example             |
|---------|---------------------------------|---------------------|
| message | Message                         | PONG!               |
| date    | Current date and time on server | 2016-06-30 14:09:20 |

#### Example:

```

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!" }
```

## Fetching data

When fetching data such as a list of orders, the number of rows in response is limited to 2000 per page. Please use some filters to narrow search. Follow links in HTTP-headers to scroll through pages.

## Order information (GET /orders/:id)

### Parameters

| Name                | Description   | Example                           |
|---------------------|---|-----------------------------------|
| <code>expand</code> | 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,location,client</code> |

### Response

| Name                           | Description  | Example             |
|--------------------------------|--|---------------------|
| <code>amount</code>            | Order amount   | 9.99                |
| <code>amount_charged</code>    | Charged amount   | 9.99                |
| <code>amount_refunded</code>   | Refunded amount  | 5.99                |
| <code>auth_code</code>         | Authorization code   | AUTH12              |
| <code>card</code>              | Card details (*)   | —                   |
| <code>*.holder</code>          | Cardholder name  | John Smith          |
| <code>*.subtype</code>         | Card subtype   | classic             |
| <code>*.type</code>            | Card type  | visa                |
| <code>client</code>            | Client details (*)   | —                   |
| <code>*.address</code>         | Street address   | Main ave. 1         |
| <code>*.city</code>            | City   | San Francisco       |
| <code>*.country</code>         | Country  | USA                 |
| <code>*.email</code>           | E-mail   | foo@bar.com         |
| <code>*.name</code>            | Full name  | John Smith          |
| <code>*.phone</code>           | Phone number   | +14568901231        |
| <code>*.state</code>           | State  | CA                  |
| <code>*.zip</code>             | Postal code  | 12345               |
| <code>created</code>           | Creation date and time   | 2014-06-30 14:38:48 |
| <code>currency</code>          | Requested payment currency (ISO-4217 3-alpha code)   | USD                 |
| <code>description</code>       | Order description  | Book sale #453      |
| <code>descriptor</code>        | Terminal Descriptor  | EXAMPLE.COM/12345   |
| <code>id</code>                | Unique order ID  | 5673044075769513    |
| <code>issuer</code>            | Card issuer details  | —                   |
| <code>*.bin</code>             | BIN  | 411111              |
| <code>*.country</code>         | Country (ISO 3166-1 alpha-3)   | RUS                 |
| <code>*.title</code>           | Bank name  | Bank                |
| <code>location</code>          | Client location details (*)  | —                   |
| <code>*.country</code>         | Country (ISO 3166-1 alpha-3)   | USA                 |
| <code>*.city</code>            | City   | San Francisco       |
| <code>*.region</code>          | Region   | California          |
| <code>*.ip</code>              | IP-address   | 8.8.8.8             |
| <code>merchant_order_id</code> | Merchant's order ID  | 5678                |
| <code>operations</code>        | Array of the order operations (*). See description of fields in <a href="#">List of operations</a> | —                   |
| <code>pan</code>               | Card number (PAN)  | 411111****1111      |
| <code>secure3d</code>          | 3-D Secure details (*)   | —                   |
| <code>*.reason</code>          | The reason why 3-D Secure was enabled  | force               |

|                        |                                     |                     |
|------------------------|-------------------------------------|---------------------|
| *.scenario             | 3-D Secure scenario used            | full                |
| *.authorization_status | Status of authorization             | Y                   |
| segment                | Segment                             | 987654321           |
| status                 | Order status                        | authorized          |
| updated                | Date and time of the last operation | 2014-06-30 14:38:48 |

### Example:

```
curl -v -X GET -H 'Content-Type: application/json'
'http://project:***@api.box:5001/orders/65213580296542862?
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" : "John Smith",
        "subtype" : "classic",
        "type" : "visa"
      },
      "client" : {
        "address" : "Main ave. 1",
        "city" : "San Francisco",
        "country" : "USA",
        "email" : "foo@bar.com",
        "name" : "John Smith",
        "phone" : "+1 456 890",
        "state" : "CA",
        "zip" : "123456"
      },
      "created" : "2021-11-12 12:16:13",
      "currency" : "USD",
      "custom_fields" : {},
      "description" : "Book sale #453",
      "descriptor" : "TESTMERCH/TERM",
      "id" : "65213580296542862",
      "issuer" : {
        "bin" : "411111",
        "country" : "UKR",
        "title" : "TestBank"
      },
      "location" : {
        "ip" : "6.6.6.6"
      },
      "merchant_order_id" : "5678",
      "operations" : [
        {
          "type": "authorize",
```

```

    "iso_message": "Approved",
    "created": "2021-11-12 12:16:13",
    "amount": "1213.00",
    "auth_code": "212636",
    "iso_response_code": "00",
    "cashflow": {
      "incoming": "0.00",
      "fee": "0.00",
      "receivable": "-36.39",
      "amount": "0.00",
      "currency": "USD",
      "reserve": "36.39"
    },
    "currency": "USD",
    "status": "success"
  },
  {
    "currency": "USD",
    "status": "success",
    "cashflow": {
      "receivable": "1164.48",
      "amount": "1213.00",
      "reserve": "36.39",
      "currency": "USD",
      "incoming": "1200.87",
      "fee": "12.13"
    },
    "arn": "73922059535879175995901",
    "iso_response_code": "00",
    "iso_message": "Approved",
    "created": "2021-11-12 12:21:13",
    "amount": "1213.00",
    "type": "charge"
  }
],
"pan" : "411111****1111",
"secure3d" : {},
"segment" : "987654321",
"status" : "authorized",
"updated" : "2016-08-22 15:39:49"
}
]
}

```

## List of orders (GET /orders/)

### Optional parameters

| Name                           | Description   | Example                           |
|--------------------------------|---|-----------------------------------|
| <code>expand</code>            | 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,location,client</code> |
| <code>status</code>            | Filter by status  | <code>charged,refunded</code>     |
| <code>created_from</code>      | Filter by date of creation  | <code>2016-01-01 00:00:00</code>  |
| <code>created_to</code>        | Filter by date of creation  | <code>2016-01-31 23:59:59</code>  |
| <code>merchant_order_id</code> | Filter by merchant order ID   | <code>12345,123456</code>         |
| <code>card.type</code>         | Filter by card type   | <code>visa,mastercard</code>      |
| <code>card.subtype</code>      | Filter by card subtype  | <code>electron,maestro</code>     |
|                                | Filter by IP address  |                                   |

|                |   |                 |
|----------------|---|-----------------|
| location.ip    |   | 8.8.8.8,8.8.8.9 |
| client.name    | Find by client name or part of it             | Smith           |
| client.email   | Find by email or part of it                   | @yahoo.com      |
| issuer.country | Filter by issuer country (ISO 3166-1 alpha-3) | USA             |
| issuer.bin     | Filter by issuer BIN                          | 555555,411111   |
| issuer.title   | Find by issuer name or part of it             | Credit Union    |

## Response

All the fields are the same as described in [Order information](#).

## Example:

```
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" : "USD", "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" : "USD", "description" : null, "descriptor" : "TESTMERCH/TERM", "id" : "23014806643308174",
"merchant_order_id" : null, "pan" : "411111****1111", "status" : "authorized", "updated" : "2016-08-22 15:39:48" } ] }
```

## List of operations (GET /operations/)

### Optional parameters

| Name         | Description   | Example             |
|--------------|---|---------------------|
| expand       | List of extra objects to display. Possible expand is <code>cashflow</code> , <code>extra</code> | cashflow            |
| status       | Filter by status  | error               |
| type         | Filter by type  | charge, refund      |
| created_from | Filter by date of creation  | 2016-01-01 00:00:00 |
| created_to   | Filter by date of creation  | 2016-01-31 23:59:59 |

## Response

| Name              | Description  | Example             |
|-------------------|--|---------------------|
| amount            | Operation amount                                   | 9.99                |
| auth_code         | Authorization code                                 | AUTH12              |
| created           | Operation date and time                            | 2014-06-30 14:38:48 |
| currency          | Requested payment currency (ISO-4217 3-alpha code) | USD                 |
| iso_message       | Response message                                   | Approved            |
| iso_response_code | Response ISO-code                                  | 00                  |
|                   |  |                     |

| status<br>type | Operation status<br>Operation type          | success<br>authorize |
|----------------|---|----------------------|
| cashflow       | Financial details (*)                       | —                    |
| *.currency     | Accounting currency (ISO-4217 3-alpha code) | EUR                  |
| *.amount       | Accounting amount                           | 9.00                 |
| *.fee          | Fee amount                                  | 0.00                 |
| *.incoming     | Income amount                               | 6.00                 |
| *.receivable   | Receivable amount                           | 5.00                 |
| *.reserve      | Reserved amount                             | 0.00                 |

### Example:

```
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",
      "created" : "2016-08-24 12:07:00",
      "currency" : "EUR",
      "iso_message" : "Approved",
      "iso_response_code" : "00",
      "order_id" : "23011537206894183",
      "arn": "28281615742473953725862",
      "status" : "success",
      "type" : "charge"
    },
    {
      "amount" : "8997.00",
      "auth_code" : "AUTH12",
      "created" : "2016-08-23 12:06:55",
      "currency" : "USD",
      "iso_message" : "Approved",
      "iso_response_code" : "00",
      "order_id" : "23011536237676397",
      "status" : "success",
      "type" : "authorize"
    }
  ]
}
```

## Export of currency rates (GET /exchange\_rates/)

### Parameters

| Name      | Description                             | Example    |
|-----------|---|------------|
| date      | Date of exchange rates                  | 2014-06-30 |
| date_from | Begin date of period for exchange rates | 2021-06-29 |
| date_to   | Final date of period for exchange rates | 2021-06-30 |

## Response

| Name                        | Description      | Example  |
|-----------------------------|------------------|----------|
| <code>exchange_rates</code> | Data of exchange | —        |
| <code>*.from</code>         | Quoted currency  | USD      |
| <code>*.to</code>           | Base currency    | EUR      |
| <code>*.rate</code>         | Quote            | 0.800000 |

### Example:

```
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" : "UAH", "from" : "EUR", "rate" :
"31.562887" }, { "from" : "EUR", "rate" : "334.000000", "to" : "UBT" }, { "to" : "USD", "from" : "EUR", "rate" :
"1.145036" } ] }
```

### Notes

Filters by date and date period cannot be used at the same time.

## Working with orders

### Order creation (POST /orders/create)

The success response on this request is returned with response code 201. It means that the new order is created and it is waiting for input from cardholder.

### Required parameters

| Name                | Description  | Example |
|---------------------|--------------|---------|
| <code>amount</code> | Order amount | 9.99    |

### Optional parameters

| Name                           | Description                  | Example        |
|--------------------------------|------------------------------|----------------|
| <code>currency</code>          | Order currency               | USD            |
| <code>merchant_order_id</code> | Merchant's order ID          | 5678           |
| <code>segment</code>           | Segment                      | 987654321      |
| <code>description</code>       | Order description            | Book sale #453 |
| <code>client</code>            | Client details (*)           | —              |
| <code>*.address</code>         | Street address               | Main ave. 1    |
| <code>*.city</code>            | City                         | San Francisco  |
| <code>*.country</code>         | Country (ISO 3166-1 alpha-3) | USA            |
| <code>*.email</code>           | E-mail                       | foo@bar.com    |
| <code>*.name</code>            | Full name                    | John Smith     |

|                         |  |                   |
|-------------------------|--|-------------------|
| *.phone                 | Phone number   | +1 456 890        |
| *.state                 | State  | CA                |
| *.zip                   | Postal code  | 123456            |
| custom_fields           | Additional custom fields (*), up to 10 additional values                 | —                 |
| location                | Client location details (*)  | —                 |
| *.ip                    | IP-address   | 8.8.8.8           |
| options                 | Options (*)  | —                 |
| *.expiration_timeout    | Session expiration timeout   | 30m               |
| *.force3d               | Forced 3-D Secure verification   | 1                 |
| *.auto_charge           | Executes <a href="#">Charge</a> operation (is used for 1-stage payments) | 1                 |
| *.language              | Payment page language  | ru                |
| *.return_url            | URL to redirect customer when the payment is completed                   | http://my.url     |
| *.template              | Payment page layout  | 12                |
| *.mobile                | Mobile layout  | 1                 |
| *.terminal              | Terminal   | TERM12            |
| *.apple_pay_enabled     | Apple Pay support on checkout page                                       | 1                 |
| *.google_pay_enabled    | Google Pay™ support on checkout page                                     | 1                 |
| *.secure3d20_return_url | Notification URL 3-D Secure 2.0  | http://return.url |
| *.exemption_mit         | Exemption MIT flag   | 1                 |

## Response

Fields in response are the same as described in [Order information](#).

HTTP-header *Location* contents a URL of a payment page. Please redirect the cardholder to this URL.

## Example (only required fields provided):

```
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" : "USD", "custom_fields" : {}, "description" : null, "id" :
"23014805213622230", "issuer" : {}, "location" : {}, "merchant_order_id" : null, "operations" : [], "secure3d" : {},
"status" : "new", "updated" : "2016-08-22 15:39:42" } ] }
```

## Example (all the fields provided):

```
curl -v -X POST -H 'Content-Type: application/json' -d '{
  "amount" : 9.99,
  "client" : {
    "address" : "Main ave. 1",
```

```

    "city" : "San Francisco",
    "country" : "USA",
    "email" : "foo@bar.com",
    "name" : "John Smith",
    "phone" : "+1 456 890",
    "state" : "CA",
    "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" : "San Francisco", "country" : "USA", "email" : "foo@bar.com", "name" : "John
Smith", "phone" : "+1 456 890", "state" : "CA", "zip" : "123456" }, "created" : "2016-08-22 15:39:43", "currency" :
"USD", "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" } ] }

```

## Notes

To manage the inclusion of *Google Pay*<sup>™</sup> and *Apple Pay* on the checkout page using flags `*_pay_enabled` must be disabled global support in the settings.

## Authorize (POST /orders/authorize)

The success response on this request may be returned with response code 200 or 201.

Code 201 means that the order is prepared for the 3-D Secure authentication and it is waiting for cardholder to return from ACS. You should make the cardholder submit the HTML-form provided in response.

Code 200 means that the authorization is complete. This happens in cases when 3-D Secure was not requested or authentication was completed without cardholder participation.

### Required parameters

| Name   | Description       | Example          |
|--------|-------------------|------------------|
| amount | Order amount      | 9.99             |
| pan    | Card number (PAN) | 4111111111111111 |

|                    |  |            |
|--------------------|--|------------|
| card               | Card details (*)   | —          |
| *.cvv              | CVV-code (can be configured as optional in project settings)                                 | 123        |
| *.holder           | Cardholder name (can be configured as optional in project settings, from 2 to 40 characters) | John Smith |
| *.expiration_month | Month of the card expiration date  | 06         |
| *.expiration_year  | Year of the card expiration date   | 2022       |
| location           | Client location details (*)  | —          |
| *.ip               | IP-address   | 8.8.8.8    |

### Optional parameters

| Name                    | Description  | Example                          |
|-------------------------|--|----------------------------------|
| currency                | Order currency   | USD                              |
| merchant_order_id       | Merchant's order ID  | 5678                             |
| segment                 | Segment  | 987654321                        |
| description             | Order description  | Book sale #453                   |
| client                  | Client details (*)   | —                                |
| *.address               | Street address   | Main ave. 1                      |
| *.city                  | City   | San Francisco                    |
| *.country               | Country (ISO 3166-1 alpha-3)   | USA                              |
| *.login                 | Client login   | john_doe                         |
| *.email                 | E-mail   | foo@bar.com                      |
| *.name                  | Full name  | John Smith                       |
| *.phone                 | Phone number   | +1 456 890                       |
| *.state                 | State  | CA                               |
| *.zip                   | Postal code  | 123456                           |
| custom_fields           | Additional custom fields (*)   | —                                |
| options                 | Options (*)  | —                                |
| *.force3d               | Forced 3-D Secure verification   | 1                                |
| *.return_url            | URL to redirect customer when the payment is completed                   | http://my.url                    |
| *.auto_charge           | Executes <a href="#">Charge</a> operation (is used for 1-stage payments) | 1                                |
| *.terminal              | Terminal   | TERM12                           |
| *.recurring             | Flag of the recurring payment  | 1                                |
| *.secure3d20_return_url | Notification URL 3-D Secure 2.0  | http://return.url                |
| *.exemption_mit         | Exemption MIT flag   | 1                                |
| secure3d                | 3-D Secure authentication (*)  | —                                |
| *.xid                   | PaRes transaction identifier (XID)                                       | 8Rg2xYbxLUGsfZlko<br>h1JxAQFBM=  |
| *.cavv                  | Cardholder authentication verification value from PaRes                  | AAABCBVRA2EYZXQBV<br>1EDAAAAA=   |
| *.eci                   | E-commerce indicator (ECI)   | 05                               |
| *.ds_trans_id           | Directory server transaction ID  | Rcm70YAaT/7SZnqVG<br>FQkm/NJnbQ= |

### Response

Fields in response are the same as described in [Order information](#) and may contain extra fields described below.

| Name        | Description             | Example               |
|-------------|-------------------------|-----------------------|
| form3d_html | 3-D Secure form in HTML | <form id="form3d" ... |

|           |   |                          |
|-----------|---|--------------------------|
| form3d    | 3-D Secure form (*)                       | —                        |
| *.MD      | MD parameter                              | NTY3NDg0NjMzMjA0MDU1OQ== |
| *.PaReq   | PaReq parameter                           | Y3NDg0NjMzM0MDU1OQ       |
| *.TermUrl | System URL where transaction is finalized | http://box.com/complete  |
| *.action  | Acquirer ACS URL                          | http://bank.com          |
| *.method  | Form method                               | POST                     |

**Example (only required fields provided):**

```
curl -v -X POST -H 'Content-Type: application/json' -d '{
  "amount" : 9.99,
  "card" : {
    "cvv" : "333",
    "expiration_month" : 12,
    "expiration_year" : 2020,
    "holder" : "John Smith"
  },
  "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" : "John Smith", "subtype" : "classic", "type" : "visa" }, "client" : {}, "created" : "2016-08-22 15:39:48",
"currency" : "USD", "custom_fields" : {}, "description" : null, "descriptor" : "TESTMERCH/TERM", "id" :
"23014806643308174", "issuer" : { "bin" : "411111", "country" : "UKR", "title" : "TestBank" }, "location" : { "ip" :
"6.6.6.6" }, "merchant_order_id" : null, "operations" : [ { "amount" : "9.99", "auth_code" : "AUTH12", "cashflow" : {
"amount" : "0.00", "currency" : "USD", "fee" : "0.00", "incoming" : "0.00", "receivable" : "0.00", "reserve" : "0.00" },
"created" : "2016-08-22 15:39:48", "currency" : "USD", "iso_message" : "Approved", "iso_response_code" : "00",
"status" : "success", "type" : "authorize" } ], "pan" : "411111****1111", "secure3d" : {}, "status" : "authorized",
"updated" : "2016-08-22 15:39:48" } ] }
```

**Example (all fields provided):**

```
curl -v -X POST -H 'Content-Type: application/json' -d '{
  "amount" : 9.99,
  "card" : {
    "cvv" : "333",
    "expiration_month" : 12,
    "expiration_year" : 2020,
    "holder" : "John Smith"
  },
  "client" : {
    "address" : "Main ave. 1",
    "city" : "San Francisco",
    "country" : "USA",
    "email" : "foo@bar.com",
    "name" : "John Smith",
    "phone" : "+1 456 890",
```

```

    "state" : "CA",
    "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" : "John Smith", "subtype" : "classic", "type" : "visa" }, "client" : { "address" : "Main ave. 1", "city" :
"San Francisco", "country" : "USA", "email" : "foo@bar.com", "name" : "John Smith", "phone" : "+1 456 890", "state" :
"CA", "zip" : "123456" }, "created" : "2016-08-22 15:39:49", "currency" : "USD", "custom_fields" : {}, "description" :
"Book sale #453", "descriptor" : "TESTMERCH/TERM", "id" : "23014806815093624", "issuer" : { "bin" : "411111",
"country" : "UKR", "title" : "TestBank" }, "location" : { "ip" : "6.6.6.6" }, "merchant_order_id" : "5678", "operations" : [ {
"amount" : "9.99", "auth_code" : "AUTH12", "cashflow" : { "amount" : "0.00", "currency" : "USD", "fee" : "0.00",
"incoming" : "0.00", "receivable" : "0.00", "reserve" : "0.00" }, "created" : "2016-08-22 15:39:49", "currency" : "USD",
"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" } ] }

```

### Example (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" : "John Smith"
  },
  "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" : "John Smith",
        "subtype" : "classic",
        "type" : "visa"
      },
      "client" : {},
      "created" : "2016-08-22 15:39:46",
      "currency" : "USD",
      "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" : "UKR",
        "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"
    }
  ]
}

```

## Notes

Once the authorization request is completed, it has to be followed with [Charge](#) operation. It also can be done with `auto_charge` option passed in request or automatically depending on project settings. If `charge` is never processed, authorization will be released by the card issuer after some time and authorized amount will be available

back to the cardholder.

## Reverse (PUT /orders/:id/reverse)

### Parameters

Not supported.

### Response

Fields in response are the same as described in [Order information](#).

### Example:

```
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": "John Smith", "subtype": "classic", "type": "visa" }, "client": {}, "created": "2016-08-22 15:39:44",
"currency": "USD", "custom_fields": {}, "description": null, "descriptor": "TESTMERCH/TERM", "id":
"23014805692462630", "issuer": { "bin": "411111", "country": "UKR", "title": "TestBank" }, "location": { "ip":
"6.6.6.6" }, "merchant_order_id": null, "operations": [ { "amount": "9.99", "auth_code": "AUTH12", "cashflow": {
"amount": "0.00", "currency": "USD", "fee": "0.00", "incoming": "0.00", "receivable": "0.00", "reserve": "0.00" },
"created": "2016-08-22 15:39:45", "currency": "USD", "iso_message": "Approved", "iso_response_code": "00",
"status": "success", "type": "authorize" }, { "amount": "9.99", "auth_code": "AUTH12", "cashflow": { "amount":
"0.00", "currency": "USD", "fee": "0.00", "incoming": "0.00", "receivable": "0.00", "reserve": "0.00" }, "created":
"2016-08-22 15:39:50", "currency": "USD", "iso_message": "Approved", "iso_response_code": "00", "status":
"success", "type": "reverse" } ], "pan": "411111****1111", "secure3d": {}, "status": "reversed", "updated": "2016-
08-22 15:39:50" } ] }
```

## Charge (PUT /orders/:id/charge)

### Parameters

| Name   | Description      | Example |
|--------|------------------|---------|
| amount | Amount of charge | 9.99    |

### Response

Fields in response are the same as described in [Order information](#).

### Example (full charge):

```
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": "John Smith", "subtype": "classic", "type": "visa" }, "client": {}, "created": "2016-08-22 15:39:48",
"currency": "USD", "custom_fields": {}, "description": null, "descriptor": "TESTMERCH/TERM", "id":
"23014806643308174", "issuer": { "bin": "411111", "country": "UKR", "title": "TestBank" }, "location": { "ip":
"6.6.6.6" }, "merchant_order_id": null, "operations": [ { "amount": "9.99", "auth_code": "AUTH12", "cashflow": {
"amount": "0.00", "currency": "USD", "fee": "0.00", "incoming": "0.00", "receivable": "0.00", "reserve": "0.00" },
"created": "2016-08-22 15:39:48", "currency": "USD", "iso_message": "Approved", "iso_response_code": "00",
"status": "success", "type": "authorize" }, { "amount": "9.99", "auth_code": "AUTH12", "cashflow": { "amount":
"9.99", "currency": "USD", "fee": "0.30", "incoming": "9.69", "receivable": "9.69", "reserve": "0.00" }, "created":
"2016-08-22 15:39:53", "currency": "USD", "iso_message": "Approved", "iso_response_code": "00", "status":
"success", "type": "charge" } ], "pan": "411111****1111", "secure3d": {}, "status": "charged", "updated": "2016-08-
22 15:39:53" } ] }

```

### Example (partial charge):

```

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": "John Smith", "subtype": "classic", "type": "visa" }, "client": {}, "created": "2016-08-22 15:39:47",
"currency": "USD", "custom_fields": {}, "description": null, "descriptor": "TESTMERCH/TERM", "id":
"23014806312026412", "issuer": { "bin": "411111", "country": "UKR", "title": "TestBank" }, "location": { "ip":
"6.6.6.6" }, "merchant_order_id": null, "operations": [ { "amount": "9.99", "auth_code": "AUTH12", "cashflow": {
"amount": "0.00", "currency": "USD", "fee": "0.00", "incoming": "0.00", "receivable": "0.00", "reserve": "0.00" },
"created": "2016-08-22 15:39:47", "currency": "USD", "iso_message": "Approved", "iso_response_code": "00",
"status": "success", "type": "authorize" }, { "amount": "1.99", "auth_code": "AUTH12", "cashflow": { "amount":
"1.99", "currency": "USD", "fee": "0.06", "incoming": "1.93", "receivable": "1.93", "reserve": "0.00" }, "created":
"2016-08-22 15:39:51", "currency": "USD", "iso_message": "Approved", "iso_response_code": "00", "status":
"success", "type": "charge" } ], "pan": "411111****1111", "secure3d": {}, "status": "charged", "updated": "2016-08-
22 15:39:51" } ] }

```

## Refund (PUT /orders/:id/refund)

### Parameters

| Name   | Description      | Example |
|--------|------------------|---------|
| amount | Amount of refund | 9.99    |

## Response

Fields in response are the same as described in [Order information](#).

### Example (full refund):

```
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": "John Smith", "subtype": "classic", "type": "visa" }, "client": {}, "created": "2016-08-22 15:39:48",
"currency": "USD", "custom_fields": {}, "description": null, "descriptor": "TESTMERCH/TERM", "id":
"23014806643308174", "issuer": { "bin": "411111", "country": "UKR", "title": "TestBank" }, "location": { "ip":
"6.6.6.6" }, "merchant_order_id": null, "operations": [ { "amount": "9.99", "auth_code": "AUTH12", "cashflow": {
"amount": "0.00", "currency": "USD", "fee": "0.00", "incoming": "0.00", "receivable": "0.00", "reserve": "0.00" },
"created": "2016-08-22 15:39:48", "currency": "USD", "iso_message": "Approved", "iso_response_code": "00",
"status": "success", "type": "authorize" }, { "amount": "9.99", "auth_code": "AUTH12", "cashflow": { "amount":
"9.99", "currency": "USD", "fee": "0.30", "incoming": "9.69", "receivable": "9.69", "reserve": "0.00" }, "created":
"2016-08-22 15:39:53", "currency": "USD", "iso_message": "Approved", "iso_response_code": "00", "status":
"success", "type": "charge" }, { "amount": "9.99", "auth_code": "AUTH12", "cashflow": { "amount": "-9.99",
"currency": "USD", "fee": "0.00", "incoming": "-9.99", "receivable": "-9.99", "reserve": "0.00" }, "created": "2016-
08-22 15:39:55", "currency": "USD", "iso_message": "Approved", "iso_response_code": "00", "status": "success",
"type": "refund" } ], "pan": "411111****1111", "secure3d": {}, "status": "refunded", "updated": "2016-08-22
15:39:55" } ] }
```

### Example (partial refund):

```
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": "John Smith", "subtype": "classic", "type": "visa" }, "client": {}, "created": "2016-08-22 15:39:47",
"currency": "USD", "custom_fields": {}, "description": null, "descriptor": "TESTMERCH/TERM", "id":
"23014806312026412", "issuer": { "bin": "411111", "country": "UKR", "title": "TestBank" }, "location": { "ip":
"6.6.6.6" }, "merchant_order_id": null, "operations": [ { "amount": "9.99", "auth_code": "AUTH12", "cashflow": {
"amount": "0.00", "currency": "USD", "fee": "0.00", "incoming": "0.00", "receivable": "0.00", "reserve": "0.00" },
"created": "2016-08-22 15:39:47", "currency": "USD", "iso_message": "Approved", "iso_response_code": "00",
```

```
"status" : "success", "type" : "authorize" }, { "amount" : "1.99", "auth_code" : "AUTH12", "cashflow" : { "amount" : "1.99", "currency" : "USD", "fee" : "0.06", "incoming" : "1.93", "receivable" : "1.93", "reserve" : "0.00" }, "created" : "2016-08-22 15:39:51", "currency" : "USD", "iso_message" : "Approved", "iso_response_code" : "00", "status" : "success", "type" : "charge" }, { "amount" : "1.99", "auth_code" : "AUTH12", "cashflow" : { "amount" : "-1.99", "currency" : "USD", "fee" : "0.00", "incoming" : "-1.99", "receivable" : "-1.99", "reserve" : "0.00" }, "created" : "2016-08-22 15:39:52", "currency" : "USD", "iso_message" : "Approved", "iso_response_code" : "00", "status" : "success", "type" : "refund" } ], "pan" : "411111****1111", "secure3d" : {}, "status" : "refunded", "updated" : "2016-08-22 15:39:52" ] ] }
```

## Cancel (PUT /orders/:id/cancel)

This is the equivalent of making [Reverse](#) or [Refund](#) depending on current order state.

### Parameters

| Name   | Description                                     | Example |
|--------|---|---------|
| amount | Amount of refund (not applicable for reversing) | 9.99    |

### Response

Fields in response are the same as described in [Order information](#).

### Example:

```
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" : "John Smith", "subtype" : "classic", "type" : "visa" }, "client" : {}, "created" : "2016-08-22 15:39:56",
"currency" : "USD", "custom_fields" : {}, "description" : null, "descriptor" : "TESTMERCH/TERM", "id" :
"23014808614292147", "issuer" : { "bin" : "411111", "country" : "UKR", "title" : "TestBank" }, "location" : { "ip" :
"6.6.6.6" }, "merchant_order_id" : null, "operations" : [ { "amount" : "9.99", "auth_code" : "AUTH12", "cashflow" : {
"amount" : "0.00", "currency" : "USD", "fee" : "0.00", "incoming" : "0.00", "receivable" : "0.00", "reserve" : "0.00" },
"created" : "2016-08-22 15:39:56", "currency" : "USD", "iso_message" : "Approved", "iso_response_code" : "00",
"status" : "success", "type" : "authorize" }, { "amount" : "9.99", "auth_code" : "AUTH12", "cashflow" : { "amount" :
"9.99", "currency" : "USD", "fee" : "0.30", "incoming" : "9.69", "receivable" : "9.69", "reserve" : "0.00" }, "created" :
"2016-08-22 15:39:56", "currency" : "USD", "iso_message" : "Approved", "iso_response_code" : "00", "status" :
"success", "type" : "charge" }, { "amount" : "9.99", "auth_code" : "AUTH12", "cashflow" : { "amount" : "-9.99",
"currency" : "USD", "fee" : "0.00", "incoming" : "-9.99", "receivable" : "-9.99", "reserve" : "0.00" }, "created" : "2016-
08-22 15:39:57", "currency" : "USD", "iso_message" : "Approved", "iso_response_code" : "00", "status" : "success",
"type" : "refund" } ], "pan" : "411111****1111", "secure3d" : {}, "status" : "refunded", "updated" : "2016-08-22
15:39:57" } ] ] }
```

## Repeating payment (POST /orders/:id/rebill)

### Required parameters

| Name     | Description  | Example |
|----------|--|---------|
| amount   | Amount   | 9.99    |
| cvv      | CVV-code (can be configured as optional in project settings) | 123     |
| location | Client location details (*)                                  | —       |
| *.ip     | IP-address   | 8.8.8.8 |

### Optional parameters

| Name        | Description                   | Example |
|-------------|-------------------------------|---------|
| options     | Options (*)                   | —       |
| *.recurring | Flag of the recurring payment | 1       |

Other optional parameters are the same as for [Authorize](#) operation.

### Response

Fields in response are the same as described in [Order information](#).

### Example:

```
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" : "John Smith", "subtype" : "classic", "type" : "visa" }, "client" : {}, "created" : "2016-08-22 15:39:58",
"currency" : "USD", "custom_fields" : {}, "description" : null, "descriptor" : "TESTMERCH/TERM", "id" :
"23014809139351770", "issuer" : { "bin" : "411111", "country" : "UKR", "title" : "TestBank" }, "location" : {},
"merchant_order_id" : null, "operations" : [ { "amount" : "9.99", "auth_code" : "AUTH12", "cashflow" : { "amount" :
"0.00", "currency" : "USD", "fee" : "0.00", "incoming" : "0.00", "receivable" : "0.00", "reserve" : "0.00" }, "created" :
"2016-08-22 15:39:58", "currency" : "USD", "iso_message" : "Approved", "iso_response_code" : "00", "status" :
"success", "type" : "authorize" } ], "pan" : "411111****1111", "secure3d" : {}, "status" : "authorized", "updated" :
"2016-08-22 15:39:58" } ] }
```

## Original credit transaction (OCT) (POST /orders/:id/credit)

### Required parameters

| Name   | Description | Example |
|--------|-------------|---------|
| amount | Amount      | 9.99    |

### Optional parameters

Optional parameters are the same as for [Authorize](#) operation.

### Response

Fields in response are the same as described in [Order information](#).

**Example:**

```
curl -v -X POST -H 'Content-Type: application/json' -d '{
  "amount" : 9.99,
  "currency" : "USD"
  "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" : "John Smith",
        "subtype" : "classic",
        "type" : "visa"
      },
      "client" : {},
      "created" : "2016-08-22 15:39:58",
      "currency" : "USD",
      "custom_fields" : {},
      "description" : null,
      "descriptor" : "TESTMERCH/TERM",
      "id" : "23014809139351770",
      "issuer" : {
        "bin" : "411111",
        "country" : "UKR",
        "title" : "TestBank"
      },
      "location" : {},
      "merchant_order_id" : null,
      "operations" : [
        {
          "amount" : "9.99",
          "auth_code" : "AUTH12",
          "cashflow" : {
            "amount" : "0.00",
            "currency" : "USD",
            "fee" : "0.00",
            "incoming" : "0.00",
            "receivable" : "0.00",
            "reserve" : "0.00"
          },
          "created" : "2016-08-22 15:39:58",
          "currency" : "USD",
          "iso_message" : "Approved",
          "iso_response_code" : "00",
          "status" : "success",
```

```

        "type" : "credit"
      }
    ],
    "pan" : "411111****1111",
    "secure3d" : {},
    "status" : "credited",
    "updated" : "2016-08-22 15:39:58"
  }
]
}

```

## Original credit transaction without link to order (OCT) (POST /orders/credit)

### Required parameters

| Name   | Description       | Example          |
|--------|-------------------|------------------|
| amount | Amount            | 9.99             |
| pan    | Card number (PAN) | 4111111111111111 |

### Optional parameters

Optional parameters are the same as for [Authorize](#) operation.

### Response

Fields in response are the same as described in [Order information](#).

### Example:

```

curl -X POST -H 'Authorization: Basic cHJvamVjdDpwYXNzd29yZA==' -i
'http://project:***@api.box:5001/orders/credit' --data '

```

```

{
  "amount"      : "99.99",
  "pan"         : "4111111111111111"
}'

```

```

> HTTP/1.1 200 OK
> Server: nginx/1.10.3 (Ubuntu)
> Date: Tue,
> 18 May 2021 10: 37: 12 GMT
> Content-Type: application/json
> Content-Length: 786
> Connection: keep-alive
>

```

```

{
  "orders": [
    {
      "descriptor": " merchant123",
      "amount": "99.99",
      "status": "credited",
      "client": {},
      "amount_refunded": "0.00",
      "description": null,
      "operations": [
        {
          "auth_code": "301541",
          "cashflow": {
            "currency": "USD",
            "fee": "0.00",
            "incoming": "-99.99",
            "reserve": "3.00",

```

```

        "receivable": "-102.99",
        "amount": "-99.99"
    },
    "status": "success",
    "amount": "99.99",
    "iso_response_code": "00",
    "currency": "USD",
    "created": "2021-05-18 13:37:12",
    "type": "credit",
    "iso_message": "Approved"
}
],
"pan": "411111****1111",
"secure3d": {},
"created": "2021-05-18 13:37:11",
"custom_fields": {},
"currency": "USD",
"card": {
    "type": "mastercard",
    "subtype": "standard",
    "expiration": "0/0"
},
"id": "61277755379729234",
"issuer": {
    "bin": "527644"
},
"amount_charged": "0.00",
"location": {},
"auth_code": "301541",
"merchant_order_id": null,
"updated": "2021-05-18 13:37:12"
}
]
}

```

## Completion of 3-D Secure Authentication (POST /orders/:id/complete)

### Required parameters

| Name  | Description                   | Example               |
|-------|-------------------------------|-----------------------|
| PaRes | Payer authentication response | 424EF64895CAE9797497F |

### Example:

```

curl -X POST 'http://project:***@api.box:5001/orders/43809080318125930/complete' -H 'accept: application/json' -H 'authorization: Basic bWVyY2hhbnRfdGVzdDpQQVFpemtWeCtDU1JVbTEr' -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": "USD",
      "descriptor": "321/123",
      "amount_charged": "100.99",

```

```

    "created": "2019-03-20 18:46:15",
    "status": "charged",
    "amount_refunded": "0.00"
  }
]
}

```

## Notes

The endpoint sends a response to the authorization request. When resubmitting the request to the endpoint the answer will be received: "Order already completed".

## Completion of 3-D Secure 2.0 Authentication (POST /orders/:id/complete3d20)

### Optional parameters

| Name              | Description   | Example                                       |
|-------------------|---|---|
| <code>cres</code> | Base64 encoded string containing the results of 3-D Secure authentication | ewogICJhY3NUcmFuc0...W9uIia6ICiYlJjEuMCIKfQ== |

### Response

Fields in response are the same as described in [Order information](#).

### Example:

```

curl -X POST -H 'Authorization: Basic cHJvamVjdDpwYXNzd29yZDEyMw==' -i
'http://project:***@api.box:5001/orders/57937457169525773/complete3d20'

{
  "orders": [{
    "merchant_order_id": null,
    "client": {},
    "secure3d": {
      "reason": "force3d",
      "xid": "635fb7bc-7acc-451e-a37d-606f1febae78",
      "eci": "5",
      "version": "2",
      "authorization_status": "Y",
      "dsrp": "0",
      "cavv": "AAABBZEEBgAAAAAAAAAQGAAAAAAAA=",
      "scenario": "full"
    },
    "updated": "2020-12-18 13:13:43",
    "amount": "123.00",
    "currency": "USD",
    "status": "authorized",
    "pan": "510033***0101",
    "amount_refunded": "0.00",
    "amount_charged": "0.00",
    "operations": [{
      "amount": "123.00",
      "auth_code": "123",
      "cashflow": {
        "currency": "USD",
        "receivable": "-3.69",
        "amount": "0.00",
        "fee": "0.00",
        "incoming": "0.00",
        "reserve": "3.69"
      }
    }
  ]
}

```

```

    },
    "type": "authorize",
    "iso_message": "Approved",
    "created": "2020-12-18 13:13:43",
    "iso_response_code": "00",
    "status": "success",
    "currency": "USD"
  }],
  "auth_code": "123",
  "description": null,
  "custom_fields": {},
  "location": {
    "ip": "6.6.6.6"
  },
  "card": {
    "type": "mastercard",
    "expiration": "10/22",
    "subtype": "standard"
  },
  "issuer": {
    "bin": "510033"
  },
  "id": "57937457169525773",
  "created": "2020-12-18 13:09:52"
}]
}

```

## Notes

The endpoint completes authorization performed with 3-D Secure 2.0 authentication. If there is no data in the order properties identification from *RReq*, a validation error will be received and order will remain in the *prepare* status.

## Continue 3DS Method URL of authentication (POST /orders/:id/resume)

### Parameters

Not supported.

### Response

Fields in response are the same as described in [Order information](#).

### Example:

```

curl -X POST -H 'Authorization: Basic cHJvamVjdDpwYXNzd29yZDEyMw==' -i
'http://project:***@api.box:5001/orders/60391976929806541/resume'

{
  "orders": [{
    "amount": "1.00",
    "custom_fields": {},
    "id": "60391976929806541",
    "form3d": {
      "threeDSData": "NjAzOTE5NzY5Mjk4MDY1NDE=",
      "method": "POST",
      "action": "http://checkout.***.local:80/complete/secure3d20"
    },
    "issuer": {
      "bin": "220138"
    },
    "amount_refunded": "0.00",
    "amount_charged": "0.00",
  }],
  "auth_code": "123",
  "description": null,
  "custom_fields": {},
  "location": {
    "ip": "6.6.6.6"
  },
  "card": {
    "type": "mastercard",
    "expiration": "10/22",
    "subtype": "standard"
  },
  "issuer": {
    "bin": "510033"
  },
  "id": "57937457169525773",
  "created": "2020-12-18 13:09:52"
}

```

```

    "created": "2021-04-08 12:29:19",
    "form3d_html": "<form id=\"form3d\" method=\"POST\"
action=\"http://checkout.***.local:80/complete/secure3d20\"><input type=\"hidden\"
name=\"threeDSSessionData\" value=\"NjAzOTE5NzY5Mjk4MDY1NDE=\"></form><script>window.onload =
function() {document.forms['form3d'].submit();}</script>",
    "updated": "2021-04-08 12:29:20",
    "status": "prepared",
    "description": null,
    "location": {
      "ip": "6.6.6.6"
    },
    "currency": "EUR",
    "client": {},
    "pan": "220138****0013",
    "merchant_order_id": null,
    "operations": [],
    "card": {
      "holder": "John Smith",
      "type": "mir",
      "subtype": "unknown",
      "expiration": "12/21"
    },
    "secure3d": {
      "dsrp": "0",
      "scenario": "unknown",
      "reason": "force3d",
      "eci": "0",
      "version": "2"
    }
  }
}

```

## Notes

The endpoint continues the authorization described in [Payments with 3-D Secure 2.0](#). Maximum time waiting for a response to a request - 10 seconds.

## Additional capabilities

### Dynamic descriptor

This is applicable when the acquirer has support for dynamic descriptor.

Dynamic descriptor allows to customize merchant name and location as they shown in cardholder's bank statement and SMS from issuer.

The default merchant name and location associated with MID may be overridden with custom values for each payment request. Dynamic descriptor is supported in [Authorize](#), [Create](#) and [Rebill](#) requests.

Dynamic descriptor data should be specified within *extra\_fields* block as follows:

#### Required parameters

| Name                            | Description                             | Example       |
|---------------------------------|---|---------------|
| <code>dynamic_descriptor</code> | Dynamic descriptor (*)                  | —             |
| <code>*.merchant</code>         | Merchant name                           | BOOKSTORE LTD |
| <code>*.city</code>             | Merchant's city                         | Moscow        |
| <code>*.country</code>          | Merchant's country (ISO 3166-1 alpha-2) | RU            |

## Notes

Combined length of `*.merchant` and `*.city` values should not be more than 37 symbols and length of `*.merchant` alone should not be greater than 22 symbols. Both parameters may consist of alphanumeric characters and punctuation marks.

According to provided example values, resulting descriptor visible to cardholder may look like this

```
BOOKSTORE LTD>Moscow          RU
```

## Unique conversion of payments

If a unique identifier is transferred at payment, then all orders having one identifier, are considered one attempt. By identifier calculates a unique conversion of payments. A unique order ID is supported in [Authorize](#), [Create](#) and [Rebill](#) requests. For example, there were five attempts of order with the same identifier, one of which was successful. In this case, the conversion will be 20% (5 attempts, successful - 1), but a unique conversion - 100%, because based on the logic of counting unique attempts, it was one and ended in success.

A unique identifier is passed in the `extra_fields` block:

| Name                      | Description                  | Example   |
|---------------------------|------------------------------|-----------|
| <code>attributes</code>   | Parameters (*)               | —         |
| <code>*.invoice_id</code> | Unique identifier of payment | 741852963 |

## Payments with 3-D Secure 2.0

Data for 3-D Secure 2.0 authentication is transmitted in the request [Authorize](#) in the parameter `secure3d`.

### Required parameters

| Name                                   | Description   | Example      |
|--|---|--------------|
| <code>secure3d</code>                  | 3-D Secure authentication (*)                               | —            |
| <code>*.browser_details</code>         | Browser details   | —            |
| <code>*.*.browser_accept_header</code> | Accepted content types                                      | text/html    |
| <code>*.*.browser_color_depth</code>   | Device color depth  | 24           |
| <code>*.*.browser_ip</code>            | User IP   | 109.86.86.86 |
| <code>*.*.browser_language</code>      | Browser language  | en           |
| <code>*.*.browser_screen_height</code> | Browser area height   | 1080         |
| <code>*.*.browser_screen_width</code>  | Browser area width  | 1920         |
| <code>*.*.browser_timezone</code>      | Difference between browser timezone and UTC time in minutes | -180         |
| <code>*.*.browser_user_agent</code>    | Browser user agent  | Mozilla      |
| <code>*.*.browser_java_enabled</code>  | Java enable flag in the browser                             | false        |
| <code>*.*.window_height</code>         | Browser window height                                       | 1080         |
| <code>*.*.window_width</code>          | Browser window width  | 1920         |

### Optional parameters

| Name                           | Description                        | Example |
|--------------------------------|------------------------------------|---------|
| <code>secure3d</code>          | 3-D Secure authentication (*)      | —       |
| <code>*.force_challenge</code> | Force inclusion cash flow scenario | 1       |

If the authorization response looks like this:

```
{
  "method3d_params": {
```

```

    "method_url": "https://ds.testcert.testconnect.su/ma",
    "method_data":
"axJ0aHJlZURTTWV0aG9kTm90aWZpY2F0aW9uVVJMIjoiaHR0cHM6Ly8zZHN0ZXN0Lm1kbWJhbmsucnUvY2dpLWJpb

19jZ2lfbGlua193IiwidGhyZWVEU1NlcnZlc1RyYW5zSUQiOiJkNDI0ZDFjMS02OTUwLTRkNmMtYTMwNy0zMTcxZDgzYjJiN2IifQ"

    },
    "message": "Method3d required",
    "order_id": "60391976929806541"
}

```

then you need to collect in the user's browser a form that sends data to an invisible *iframe* using for these are the values of the *method\_url* and *method\_data* parameters from the authorization response.

After the user's browser has submitted the form to an invisible *iframe*, you need to do the request which described in the section [Continue 3DS Method URL of authentication](#).

## DSRP

The DSRP data is transmitted in the parameter *secure3d*

### Required parameters

| Name            | Description   | Example                     |
|-----------------|---|-----------------------------|
| <i>secure3d</i> | 3-D Secure authentication (*)                           | —                           |
| <i>*.dsrp</i>   | DSRP  | 1                           |
| <i>*.xid</i>    | PaRes transaction identifier (XID)                      | 8Rg2xYbxLUGsfZ1koh1JxAQFBM= |
| <i>*.cavv</i>   | Cardholder Authentication Verification Value from PaRes | AAABCBVRA2EYZXQBV1EDAAAAAA= |
| <i>*.eci</i>    | E-commerce Indicator (ECI)                              | 05                          |

### Optional parameters

| Name                  | Description  | Example    |
|-----------------------|--|------------|
| <i>dsrp_type</i>      | Type system DSRP. Possible types are: <i>masterpass</i> , <i>apple_pay</i> , <i>android_pay</i> , <i>samsung_pay</i> | masterpass |
| <i>dsrp_wallet_id</i> | ID digital wallet  | 765        |

Parameter *dsrp\_wallet\_id* are applied only for the *dsrp\_type == masterpass*. If the type of payment system is not specified, the default is ApplePay.

When working with DSRP, merchant gets a token, which contains the requisites for payment. Example of filling for the ApplePay:

| Name for ApplePay                      | Name for API                     |
|--|----------------------------------|
| <i>applicationPrimaryAccountNumber</i> | pan                              |
| <i>applicationExpirationDate</i>       | expiration_month expiration_year |
| <i>transactionAmount</i>               | amount                           |
| <i>transactionId</i>                   | xid                              |
| <i>onlinePaymentCryptogram</i>         | cavv                             |
| <i>eciIndicator</i>                    | eci                              |

The fields *holder* and *cvv* are not transmitted, because they are not contained in the token, *dsrp* always has the value 1 (*true*), *eci* is passed only if it is contained in a token.

## Binding a payment card (Oneclick)

When creating a payment can be binding a buyer card. In the future, can be select one of the binded cards on the

payment page. For binding the card, must be specify the alphanumeric identifier of the buyer.

Card binding is performed on the payment page using the card save check box.

The id must not exceed 40 symbols, it is transmitted in the block `extra_fields` of the query [Create](#):

| Name                       | Description                                      | Example    |
|----------------------------|--|------------|
| <code>oneclick</code>      | Binding a card (*)                               | —          |
| <code>*.customer_id</code> | Buyer id   | 0123654789 |
| <code>*.prechecked</code>  | Flag of filling the checkbox for saving the card | 1          |

## Paged downloads of lists of orders and operations

For a list of orders or operations, you can specify the number of entries on the page. It is also possible to specify the page number to download list of orders (operations).

| Name                   | Description                   | Example |
|------------------------|-------------------------------|---------|
| <code>page</code>      | Number loadable page          | 2       |
| <code>page_size</code> | Number of records on the page | 50      |

If the `page` and `page_size` parameters are not specified, the page with the number 1 will be downloaded. She will be contain the number of entries by default.

For example, if `page = 2` and `page_size = 50`, then 50 entries will be downloaded, starting from the 51st.

HTTP headers in the response contain in the header `Pagination` links to subsequent and previous pages of list of orders (operations). If is downloaded first page, there will only be a link to the next page of list. For example, if is downloaded a second page from the list of orders, the title will contain links of that type:

```
'<http://mybox.localhost?foo=bar&page_size=2&page=1>; rel="prev",  
<http://mybox.localhost?foo=bar&page_size=2&page=3>; rel="next"'
```

If an incorrect values of the `page_size` or `page` parameters is set, to will be error of validation with that message:

```
'message' : 'Invalid page size' or 'message' => 'Invalid page'
```

## Testing

### Overview

Test payments are being processed via test terminal. Please contact the support team for access to the test terminal.

While testing you can use any valid test card numbers, any valid date of expiration, CVV-code and cardholder name. Test terminal accepts all the payments with any valid credentials.

If you want to determine the way test terminal responds, you can use some special card numbers:

| Card number                   | Description   |
|-------------------------------|---|
| <code>4111111111111111</code> | Success authorization                               |
| <code>2222400060000007</code> | Success authorization                               |
| <code>4276990011343663</code> | Non 3-D Secure: declined authorization;             |
|                               | 3-D Secure: "Unable to verify enrollment" scenario. |
| <code>5555555555555599</code> | Internal system error                               |
| <code>4276838748917319</code> | 3-D Secure scenario "Not enrolled"                  |
| <code>4000000000000002</code> | Declined as fraud                                   |

