



API Documentation

Table of Contents

Integration Reference	4
General information and capabilities	4
Features	4
Orders and operation paradigm	4
Order status	4
Operation status	5
Workflows	5
Using payment page	5
Server-to-server	5
API overview	6
General information	6
Authentication	6
Response codes	6
Failure types	6
Working flow	7
Verifying successful operation	7
Error handling	7
Validation	7
Sample code	8
Actions	9
Test request (GET /ping)	9
Fetching data	9
Order information (GET /orders/:id)	9
List of orders (GET /orders/)	12

List of operations (GET /operations/)	14
Working with orders	15
Order creation (POST /orders/create)	15
Authorize (POST /orders/authorize)	18
Reverse (PUT /orders/:id/reverse)	25
Charge (PUT /orders/:id/charge)	27
Refund (PUT /orders/:id/refund)	30
Cancel (POST /orders/:id/cancel)	34
Repeating payment (POST /orders/:id/rebill)	36
Original credit transaction (OCT) (POST /orders/:id/credit)	38
Additional capabilities	40
Dynamic descriptor	40
Testing	40
Overview	40

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 jsonapi.org standard.
- Validation based on a JSON Schema standard (json-schema.org).
- 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.

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
new	Order has been created, but has yet to be processed	—
prepared	Waiting for outside action	—
authorized	Authorization completed	reverse, charge,rebill
charged	Order has been charged	refund, credit, rebill
reversed	Authorization has been cancelled	rebill

refunded	Order was fully or partially refunded	refund, credit, rebill
rejected	Order has been rejected	rebill
fraud	Order has been determined as a fraud and rejected 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.

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
<code>declined</code>	The acquirer has declined the transaction.
<code>fraud</code>	Transaction has been declined by fraud prevention rules on gateway side.
<code>rejected</code>	Transaction has been declined by gateway.
<code>error</code>	Failure due to an error on acquirer or gateway side.
<code>validation</code>	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.

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 occurred: " . 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
expand	List of extra objects to display. Possible expands are: card, client, location, custom_fields, issuer, secure3d, operations.cashflow	card,location,client

Response

Name	Description	Example

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

```

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>
<code>location.ip</code>	Filter by IP address	<code>8.8.8.8,8.8.8.9</code>
<code>client.name</code>	Find by client name or part of it	<code>Smith</code>

<code>client.email</code>	Find by email or part of it	@yahoo.com
<code>issuer.country</code>	Filter by issuer country (ISO 3166-1 alpha-3)	USA
<code>issuer.bin</code>	Filter by issuer BIN	555555,411111
<code>issuer.title</code>	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
<code>expand</code>	List of extra objects to display. Possible expand is <code>cashflow</code>	cashflow
<code>status</code>	Filter by status	error
<code>type</code>	Filter by type	charge, refund
<code>created_from</code>	Filter by date of creation	2016-01-01 00:00:00
<code>created_to</code>	Filter by date of creation	2016-01-31 23:59:59

Response

Name	Description	Example
<code>amount</code>	Operation amount	9.99
<code>auth_code</code>	Authorization code	AUTH12
<code>created</code>	Operation date and time	2014-06-30 14:38:48
<code>currency</code>	Requested payment currency (ISO-4217 3-alpha code)	USD
<code>iso_message</code>	Response message	Approved
<code>iso_response_code</code>	Response ISO-code	00
<code>status</code>	Operation status	success
<code>type</code>	Operation type	authorize
<code>cashflow</code>	Financial details (*)	—
<code>*.currency</code>	Accounting currency (ISO-4217 3-alpha code)	EUR
<code>*.amount</code>	Accounting amount	9.00
<code>*.fee</code>	Fee amount	0.00
<code>*.incoming</code>	Income amount	6.00
<code>*.receivable</code>	Receivable amount	5.00
<code>*.reserve</code>	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",
      "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"
    }
  ]
}

```

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
<code>currency</code>	Order currency	USD

Optional parameters

Name	Description	Example
<code>merchant_order_id</code>	Merchant's order ID	5678

segment	Segment	987654321
client	Client details (*)	—
*.address	Street address	Main ave. 1
*.city	City	San Francisco
*.country	Country (ISO 3166-1 alpha-3)	USA
*.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 (*), 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 Charge 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	mobile
*.terminal	Terminal	TERM12

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"
    }
  ]
}

```

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
<code>amount</code>	Order amount	9.99
<code>currency</code>	Order currency	USD
<code>pan</code>	Card number (PAN)	4111111111111111
<code>card</code>	Card details (*)	—
<code>*.holder</code>	Cardholder name	John Smith
<code>*.cvv</code>	CVV-code (can be configured as optional in project settings)	123
<code>*.expiration_month</code>	Month of the card expiration date	06
<code>*.expiration_year</code>	Year of the card expiration date	2022
<code>location</code>	Client location details (*)	—
<code>*.ip</code>	IP-address	8.8.8.8

Optional parameters

Name	Description	Example
<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
<code>*.phone</code>	Phone number	+1 456 890
<code>*.state</code>	State	CA
<code>*.zip</code>	Postal code	123456
<code>custom_fields</code>	Additional custom fields (*)	—
<code>options</code>	Options (*)	—
<code>*.force3d</code>	Forced 3-D Secure verification	1
<code>*.auto_charge</code>	Executes Charge operation (is used for 1-stage payments)	1
<code>*.terminal</code>	Terminal	TERM12

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"
    }
  ]
}

```

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
<code>amount</code>	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 (POST /orders/:id/cancel)

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

Parameters

Name	Description	Example
<code>amount</code>	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
currency	Order currency	USD
*.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

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
currency	Order currency	USD
location	Client location details (*)	—
*.ip	IP-address	8.8.8.8

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"
  }
]
```

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. 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

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
4111111111111111	Success authorization
4276990011343663	Non 3-D Secure: declined authorization;
	3-D Secure: "Unable to verify enrollment" scenario.
5555555555555599	Internal system error
4276838748917319	3-D Secure scenario "Not enrolled"
4000000000000002	Declined as fraud