



Version 1.0

Base banking platform API: Transactions

- **1. Introduction**
- 1.1 Requirements
 - **2. Quick start**
 - 2.1 How to find terminal ID and terminal secret key
 - 2.2 Testing
 - **3. Transaction types**
 - 3.1 Payment
 - 3.2 Authorization
 - 3.3 Capture
 - 3.4 Cancel
 - 3.5 Refund
 - 3.6 Reverse
 - **4. API for PaySeven customers**
 - 4.1 Payment
 - 4.2 Authorization
 - 4.3 Capture
 - 4.4 Cancel
 - 4.5 Refund
 - 4.6 Reverse
 - **5. Check transaction status**
 - **6. Handling 3-D Secure transactions**
 - **7. Error response**
 - **8. Code example**

Introduction

This document is designed to provide you with details on how to integrate your business to the base banking platform processing system. This guide is a step-by-step instruction as well as a detailed reference material on how to simply and quickly get your website up and running with our services. To get a technical assistance please email to: ***** Supported gateways: RBK Money, MilliKart, RoyalPay, Payally, Connectum, Decta. Currently url of base banking platform is <https://gate.rusimex.com:15002>

Supported transaction for RBK Money:

- Authorization
- Payment
- Capture
- Cancel
- Refund

Supported transaction for MilliKart:

- Payment
- Reverse
- Cancel
- Refund

Supported transaction for RoyalPay:

- Payment
- Refund

Supported transaction for Payally:

- Authorization
- Payment
- Capture
- Cancel
- Refund
- Reverse

Supported transaction for Connectum:

- Authorization
- Payment
- Capture
- Cancel
- Refund

Supported transaction for Decta:

- Authorization
- Payment
- Capture
- Reverse
- Refund

Requirements

The transaction request takes the form of a HTTP POST or GET request containing JSON message with a transaction details. The request must:

- use HTTP Basic authentication with terminal hash key and terminal secret;

- all the requests should contain the Authorization header and signature in data:

```
Authorization: (Basic authentication with terminal secret and terminal  
secret hash key)
```

```
signature: (sha256 hash from a request body with a terminal hash key  
concatenating)
```

- have the headers if it is a POST request;

```
Content-Type: application/json  
Accept: application/json  
Authorization: (Basic authentication with terminal secret and terminal  
secret hash key)
```

- use SSL connection with 128-bit (or stronger) encryption to meet PCI DSS; requirements if you are going to use the gateway API integration;
- only ASCII characters are allowed with codes from 32 to 126.

All the requests regarding Base banking platform API should be sent to: *****

Quick start

Start using our system by requesting an account; please write a message to: *****

After registration you'll get an email with username and password, you can log into the system.

How to find terminal ID and terminal secret key

Log into your merchant account using username and password. On the left sidebar click "SHOPS". Click on the id of the required shop for see detailed information this shop. In detailed information click on the required terminal id, then you can get terminal secret and terminal hash key. If you don't see terminal's secret then click edit icon on specific terminal and then check box to "GENERATE NEW SECRET". After that go to the detail information of your terminal and you will be see the generated "SECRET" and "HASH-KEY".

Testing

In "TEST FORM" in sidebar, you can make a transaction with test data using terminal secret and terminal hash key.

For successful payment you need to use card:

```
testCard: {
  number: "4242424242424242",
  verification_value: "123",
  holder: "DAN BALAN",
  exp_month: "12",
  exp_year: "2020"
}
```

Any other card numbers will initiate an error. Use "gateway": "TestGateway" for testing transactions.

Transaction types

Payment

Payment transaction is a combination of authorization and capture processed at a time. This transaction type is generally used when the goods or services can be immediately provided to the customer. Request for payment transactions should be sent via HTTP POST to the following URL:

<http://{{host}}:{{port}}/api/v1/transactions/create>

Request parameters

Name	Mandatory	Description
type	Yes	"Payment"
terminalGuid	Yes	guid of terminal
amount	Yes	Cost in minimal currency units, e.g. \$32.45 must be sent as 3245
currency	Yes	Currency in ISO-4217 format, e.g. USD
description	Yes	Order short description. Max length: 255 chars.
first_name	Yes	Customer's first name. Max length: 255 chars
last_name	Yes	Customer's last name. Max length: 255 chars.
country	Yes	Customer's billing country in ISO 3166-1 Alpha-2 format.
city	Yes	Customer's billing city. Max length: 255 chars.
address	Yes	Customer's billing street. Max length: 255 chars.
zip	Yes	Customer's billing ZIP or postal code.
number	No	Card number. Length: 12 - 19 digits.
verification_value	No	3- or 4-digit security code (called CVC2, CVV2 or CID depending on the credit card brand).
holder	No	Card holder name as it appears in the card. Max length: 255 chars.

Name	Mandatory	Description
exp_month	No	Card expiration month expressed with 2 digits (e.g. 01).
exp_year	No	Card expiration year expressed with 4 digits (e.g. 2022).
email	Yes	Email address of a customer making a purchase at your shop.
ip	Yes	IP address of a customer making a purchase at your shop.
succeed_url	Yes	Url for notification succeed transaction.
failed_url	Yes	Url for notification failed transaction.

Request example

```
{
  "terminalGuid" : "1d1a1ea2664c11e98eab0242ac110002",
  "type": "Payment",
  "amount": 100,
  "currency": "RUB",
  "description": "Test transaction",
  "billing_address": {
    "first_name": "John",
    "last_name": "Doe",
    "country": "US",
    "city": "New York",
    "address": "Groove st.",
    "zip": "646464"
  },
  "credit_card": {
    "number": "4242424242424242",
    "verification_value": "122",
    "holder": "DAN BALAN",
    "exp_month": "12",
    "exp_year": "2020"
  },
  "customer": {
    "email": "mail@mail.com",
    "ip": "200.200.200.200"
  },
  "signature": "ONTgHxxXqk03aS8y3RK/yUD7UIm+dsnmd0fmKMRvIwk=",
  "succeed_url": "https://www.yourshop.com/search?q=succeed",
  "failed_url": "https://www.yourshop.com/search?q=failed"
}
```

Response parameters

Name	Description
guid	Transaction's guid

Name	Description
status	Status of processed transaction.
succeed_url	Url for notification of succeed transaction.

Response example

```
{
  "guid": "ed0b0fddb35111e9946d0242ac110002",
  "status": "Success",
  "succeed_url": "https://www.yourshop.com/search?q=succeed"
}
```

Note: You can make transactions without card data, platform will send you form with necessary credit card fields.

```
{
  *request body data*
  ...,
  "redirect_to": {
    "url": string,
    "params": {
      "auth_key": string
    }
  }
}
```

You need to redirect the user by url + params link (where params query parameters), so that the user can enter their card data.

Authorization

With this transaction type the payment is sent for authorization only. The amount will be reserved on the customer's credit card but will not be charged yet. In order to charge the money the transaction needs to be captured in a second step (e.g. after the order is shipped). This method is mostly used by merchants who have a delayed order fulfillment process. Request for authorization transactions should be sent via HTTP POST to the following URL:

<http://{{host}}:{{port}}/api/v1/transactions/create>

Request parameters

Name	Mandatory	Description
type	Yes	"authorization"
terminalGuid	Yes	guid of terminal

Name	Mandatory	Description
amount	Yes	Cost in minimal currency units, e.g. \$32.45 must be sent as 3245
currency	Yes	Currency in ISO-4217 format, e.g. USD
description	Yes	Order short description. Max length: 255 chars.
first_name	Yes	Customer's first name. Max length: 255 chars
last_name	Yes	Customer's last name. Max length: 255 chars.
country	Yes	Customer's billing country in ISO 3166-1 Alpha-2 format.
city	Yes	Customer's billing city. Max length: 255 chars.
address	Yes	Customer's billing street. Max length: 255 chars.
zip	Yes	Customer's billing ZIP or postal code.
number	No	Card number. Length: 12 - 19 digits.
verification_value	No	3- or 4-digit security code (called CVC2, CVV2 or CID depending on the credit card brand).
holder	No	Card holder name as it appears in the card. Max length: 255 chars.
exp_month	No	Card expiration month expressed with 2 digits (e.g. 01).
exp_year	No	Card expiration year expressed with 4 digits (e.g. 2022).
email	Yes	Email address of a customer making a purchase at your shop.
ip	Yes	IP address of a customer making a purchase at your shop.
succeed_url	Yes	url for notification succeed transaction.
failed_url	Yes	url for notification failed transaction.

Request example

```
{
  "terminalGuid" : "1d1a1ea2664c11e98eab0242ac110002",
  "type": "Authorization",
  "amount": 100,
  "currency": "RUB",
  "description": "Test transaction",
  "billing_address": {
    "first_name": "John",
    "last_name": "Doe",
    "country": "US",
    "city": "New York",
    "address": "Groove st.",
    "zip": "646464"
  },
  "credit_card": {
    "number": "4242424242424242",
```

```

        "verification_value": "122",
        "holder": "DAN BALAN",
        "exp_month": "12",
        "exp_year": "2020"
    },
    "customer": {
        "email": "mail@mail.com",
        "ip": "200.200.200.200"
    },
    "signature": "ONTgHxxXqk03aS8y3RK/yUD7UIm+dsnmd0fmKMRvIwk=",
    "succeed_url": "https://www.yourshop.com/search?q=succeed",
    "failed_url": "https://www.yourshop.com/search?q=failed"
}

```

Response parameters

Name	Description
guid	Transaction's guid
status	Status of processed transaction.
succeed_url	Url for notification of succeed transaction.

Response example

```

{
    "guid": "ed0b0fddb35111e9946d0242ac110002",
    "status": "Success",
    "succeed_url": "https://www.yourshop.com/search?q=succeed"
}

```

Note: You can make transactions without card data, platform will send you form with necessary credit card fields

```

{
    *request body data*
    ...,
    "redirect_to": {
        "url": string,
        "params": {
            "auth_key": string
        }
    }
}

```

Capture

After an order is shipped, a previously authorized amount can be settled (captured). The card issuing bank credits the funds to the merchant's bank account and updates the cardholder's statement. Card regulations require a merchant to ship goods before settling the funds for an order. Request for capture transactions should be sent via HTTP POST to the following URL:

<http://{{host}}:{{port}}/api/v1/transactions/create>

Request parameters

Name	Mandatory	Description
type	Yes	"Capture"
terminalGuid	Yes	guid of terminal
transactionId	Yes	Id of succeed authorization transaction
succeed_url	No	url for notification succeed transaction.
failed_url	No	url for notification failed transaction.

Request example

```
{
  "terminalGuid" : "1d1a1ea2664c11e98eab0242ac110002",
  "type": "Capture",
  "transactionId": "88da3385b35a11e9946d0242ac110002",
  "succeed_url": "https://www.yourshop.com/search?q=succeed",
  "failed_url": "https://www.yourshop.com/search?q=failed"
}
```

Response parameters

Name	Description
guid	Transaction's guid
status	Status of processed transaction.
succeed_url	Url for notification of succeed transaction.

Response example

```
{
  "guid": "ed0b0fddb35111e9946d0242ac110002",
  "status": "Success",
  "succeed_url": "https://www.yourshop.com/search?q=succeed"
}
```

Cancel

After an order is shipped, a previously authorized amount can be cancel. Request for cancel transactions should be sent via HTTP POST to the following URL:

<http://{{host}}:{{port}}/api/v1/transactions/create>

Request parameters

Name	Mandatory	Description
type	Yes	"Cancel"
terminalGuid	Yes	guid of terminal
transactionId	Yes	Id of succeed authorization transaction
succeed_url	No	url for notification succeed transaction.
failed_url	No	url for notification failed transaction.

Request example

```
{
  "terminalGuid" : "1d1a1ea2664c11e98eab0242ac110002",
  "type": "Cancel",
  "transactionId": "88da3385b35a11e9946d0242ac110002",
  "succeed_url": "https://www.yourshop.com/search?q=succeed",
}
```

Response parameters

Name	Description
guid	Transaction's guid
status	Status of processed transaction.
succeed_url	Url for notification of succeed transaction.

Response example

```
{
  "guid": "ed0b0fddb35111e9946d0242ac110002",
  "status": "Success",
  "succeed_url": "https://www.yourshop.com/search?q=succeed"
}
```

Refund

Refund allows to credit the customer, e.g. in case of returned goods or cancellation. To post a refund request, a valid transaction id from a former capture or payment transaction is required. It is only possible

to credit an amount less than or equal to the initial transaction using the same currency as with the original transaction. This feature also allows you to issue multiple partial refunds against an original transaction. Request for refund transactions should be sent via HTTP POST to the following URL:

<http://{{host}}:{{port}}/api/v1/transactions/create>

Request parameters

Name	Mandatory	Description
type	Yes	"Refund"
terminalGuid	Yes	guid of terminal
transactionId	Yes	Id of succeed authorization transaction
succeed_url	No	url for notification succeed transaction.
failed_url	No	url for notification failed transaction.

Request example

```
{
  "terminalGuid" : "1d1a1ea2664c11e98eab0242ac110002",
  "type": "Refund",
  "transactionId": "88da3385b35a11e9946d0242ac110002",
  "succeed_url": "https://www.yourshop.com/search?q=succeed"
}
```

Response parameters

Name	Description
guid	Transaction's guid
status	Status of processed transaction.
succeed_url	Url for notification of succeed transaction.

Response example

```
{
  "guid": "ed0b0fddb35111e9946d0242ac110002",
  "status": "Success",
  "succeed_url": "https://www.yourshop.com/search?q=succeed"
}
```

Reverse

This request performs a reversal of a previous original payment referenced by parameter. The result is negative if called after the business day is closed. Similar to calling a refund request or cancel request.

API for PaySeven customers

API uses the same methods as PaySeven for quick client migration.

Payment-old

Request for payment transactions should be sent via HTTP POST to the following URL:

<http://{{host}}:{{port}}/api/v1/transactions/payments>

Request parameters

Name	Mandatory	Description
amount	Yes	Cost in minimal currency units, e.g. \$32.45 must be sent as 3245
currency	Yes	Currency in ISO-4217 format, e.g. USD
description	Yes	Order short description. Max length: 255 chars.
tracking_id	Yes	ID of your transaction or order. Max length: 255 chars.Note: though Payseven gateway allows to send non-unique values, you must send unique values to avoid possible technical issues in the future.
notification_url	Yes	URL where notification about transaction will be posted to. Notification request format equals to transaction response format.
return_url	Yes	Parameter is mandatory if your merchant account is 3- D Secure enabled. Verify it with your account manager. return_url is URL on merchant side where Payseven will send customer's browser when customer returns from 3-D Secure verification. See Handling 3-D Secure transactions for details.
first_name	Yes	Customer's first name. Max length: 255 chars
last_name	Yes	Customer's last name. Max length: 255 chars.
country	Yes	Customer's billing country in ISO 3166-1 Alpha-2 format.
city	Yes	Customer's billing city. Max length: 255 chars.
address	Yes	Customer's billing street. Max length: 255 chars.
zip	Yes	Customer's billing ZIP or postal code.
phone	Yes	Customer's phone number.
number	No	Card number. Length: 12 - 19 digits.
verification_value	No	3- or 4-digit security code (called CVC2, CVV2 or CID depending on the credit card brand).

Name	Mandatory	Description
holder	No	Card holder name as it appears in the card. Max length: 255 chars.
exp_month	No	Card expiration month expressed with 2 digits (e.g. 01).
exp_year	No	Card expiration year expressed with 4 digits (e.g. 2022).
email	Yes	Email address of a customer making a purchase at your shop.
ip	Yes	IP address of a customer making a purchase at your shop.

Request example

```
{
  "amount": 100,
  "currency": "RUB",
  "description": "Test transaction",
  "tracking_id": "123",
  "notification_url": "http://www.url.ru",
  "return_url": "http://www.url.ru",
  "billing_address": {
    "first_name": "John",
    "last_name": "Doe",
    "country": "US",
    "city": "New York",
    "address": "Groove st.",
    "zip": "646464",
    "phone": "+79562323256"
  },
  "credit_card": {
    "number": "4242424242424242",
    "verification_value": "122",
    "holder": "DAN BALAN",
    "exp_month": "12",
    "exp_year": "2020"
  },
  "customer": {
    "email": "mail@mail.com",
    "ip": "200.200.200.200"
  }
}
```

Response parameters

Name	Description
uid	Processed transaction UID.
status	Status of processed transaction.
message	Processing result message.

Name	Description
type	Transaction type.
brand	Detected card brand: visa, master.
last_4	Last 4 digits of card
first_1	The first digit of card.
auth_code	Acquirer authorization code.
bank_code	Transaction bank code
rrn	Retrieval reference number. Transaction ID issued by cards processing network
ref_id	Acquirer transaction reference ID
billing_descriptor	Billing descriptor assigned to transaction.
gateway_id	Gateway ID.
redirect_url	URL at Payseven where customer must be redirected to by merchant for 3-D Secure verification. See Handling 3-D Secure transactions for details.

Response example

```
{
  "transaction": {
    "customer": {
      "ip": "192.4.84.135/32",
      "email": "nogotochki@mail.com"
    },
    "credit_card": {
      "holder": "DAN BALAN",
      "stamp": "",
      "token": "",
      "brand": "Visa",
      "last_4": "4242",
      "first_1": "4"
    },
    "billing_address": {
      "first_name": "John",
      "last_name": "Doe",
      "address": "Groove st.",
      "country": "US",
      "city": "New York",
      "zip": "646464",
      "phone": "+79562323256"
    },
    "payment": {
      "auth_code": "000",
      "bank_code": "779819",
      "rrn": "005809838535",
      "ref_id": "2b6aea7c",
    }
  }
}
```

```

        "message": "OK",
        "gateway_id": "47532d37-b330-4011-b858-4244444b2a00",
        "billing_descriptor": "TEST GATEWAY BILLING DESCRIPTOR",
        "status": "0"
    },
    "uid": "9115d09c-4446-4dd9-a817-efb11be42fae",
    "status": "success",
    "message": "Successfully processed",
    "amount": "10",
    "currency": "RUB",
    "description": "Test transaction",
    "type": "payment"
}

```

Note: You can make transactions without card data, platform will send you form with necessary credit card fields.

```

{
  *request body data*
  ...,
  "redirect_to": {
    "url": string,
    "params": {
      "auth_key": string
    }
  }
}

```

You need to redirect the user by url + params link (where params query parameters), so that the user can enter their card data.

Authorization-old

Request for authorization transactions should be sent via HTTP POST to the following URL:

<http://{{host}}:{{port}}/api/v1/transactions/authorizations>

Request parameters

Name	Mandatory	Description
amount	Yes	Cost in minimal currency units, e.g. \$32.45 must be sent as 3245
currency	Yes	Currency in ISO-4217 format, e.g. USD
description	Yes	Order short description. Max length: 255 chars.

Name	Mandatory	Description
tracking_id	Yes	ID of your transaction or order. Max length: 255 chars.Note: though Payseven gateway allows to send non-unique values, you must send unique values to avoid possible technical issues in the future.
notification_url	Yes	URL where notification about transaction will be posted to. Notification request format equals to transaction response format.
return_url	Yes	Parameter is mandatory if your merchant account is 3- D Secure enabled. Verify it with your account manager. return_url is URL on merchant side where Payseven will send customer's browser when customer returns from 3-D Secure verification. See Handling 3-D Secure transactions for details.
first_name	Yes	Customer's first name. Max length: 255 chars
last_name	Yes	Customer's last name. Max length: 255 chars.
country	Yes	Customer's billing country in ISO 3166-1 Alpha-2 format.
city	Yes	Customer's billing city. Max length: 255 chars.
address	Yes	Customer's billing street. Max length: 255 chars.
zip	Yes	Customer's billing ZIP or postal code.
phone	Yes	Customer's phone number.
number	No	Card number. Length: 12 - 19 digits.
verification_value	No	3- or 4-digit security code (called CVC2, CVV2 or CID depending on the credit card brand).
holder	No	Card holder name as it appears in the card. Max length: 255 chars.
exp_month	No	Card expiration month expressed with 2 digits (e.g. 01).
exp_year	No	Card expiration year expressed with 4 digits (e.g. 2022).
email	Yes	Email address of a customer making a purchase at your shop.
ip	Yes	IP address of a customer making a purchase at your shop.

Request example

```
{  
  "amount": 100,  
  "currency": "RUB",  
  "description": "Test transaction",  
  "tracking_id": "123",  
  "notification_url": "http://www.url.ru",  
  "return_url": "http://www.url.ru",  
  "billing_address": {  
    "first_name": "John",
```



```
    "last_name": "Doe",
    "country": "US",
    "city": "New York",
    "address": "Groove st.",
    "zip": "646464",
    "phone": "+79562323256"
  },
  "credit_card": {
    "number": "4242424242424242",
    "verification_value": "122",
    "holder": "DAN BALAN",
    "exp_month": "12",
    "exp_year": "2020"
  },
  "customer": {
    "email": "mail@mail.com",
    "ip": "200.200.200.200"
  }
}
```

Response parameters

Name	Description
uid	Processed transaction UID.
status	Status of processed transaction.
message	Processing result message.
type	Transaction type.
brand	Detected card brand: visa, master.
last_4	Last 4 digits of card
first_1	The first digit of card.
auth_code	Acquirer authorization code.
bank_code	Transaction bank code
rrn	Retrieval reference number. Transaction ID issued by cards processing network
ref_id	Acquirer transaction reference ID
billing_descriptor	Billing descriptor assigned to transaction.
gateway_id	Gateway ID.
redirect_url	URL at Payseven where customer must be redirected to by merchant for 3-D Secure verification. See Handling 3-D Secure transactions for details.

Response example

```

{
  "transaction": {
    "customer": {
      "ip": "192.4.84.135/32",
      "email": "nogotochki@mail.com"
    },
    "credit_card": {
      "holder": "DAN BALAN",
      "stamp": "",
      "token": "",
      "brand": "Visa",
      "last_4": "4242",
      "first_1": "4"
    },
    "billing_address": {
      "first_name": "John",
      "last_name": "Doe",
      "address": "Groove st.",
      "country": "US",
      "city": "New York",
      "zip": "646464",
      "phone": "+79562323256"
    },
    "payment": {
      "auth_code": "000",
      "bank_code": "779819",
      "rrn": "005809838535",
      "ref_id": "2b6aea7c",
      "message": "OK",
      "gateway_id": "47532d37-b330-4011-b858-4244444b2a00",
      "billing_descriptor": "TEST GATEWAY BILLING DESCRIPTOR",
      "status": "0"
    },
    "uid": "9115d09c-4446-4dd9-a817-efb11be42fae",
    "status": "success",
    "message": "Successfully processed",
    "amount": "10",
    "currency": "RUB",
    "description": "Test transaction",
    "type": "payment"
  }
}

```

Note: You can make transactions without card data, platform will send you form with necessary credit card fields

```

{
  *request body data*
  ...,
  "redirect_to": {
    "url": string,

```

```
    "params": {
      "auth_key": string
    }
  }
}
```

Capture-old

Request for capture transactions should be sent via HTTP POST to the following URL:

<http://{{host}}:{{port}}/api/v1/transactions/captures>

Request parameters

Name	Mandatory	Description
parent_uid	Yes	UID of an authorization transaction.
amount	Yes	Amount to capture in minimal currency units, e.g. 1000 for \$10.00.

Request example

```
{
  "parent_uid": "88da3385b35a11e9946d0242ac110002",
  "amount": "https://www.yourshop.com/search?q=succeed",
}
```

Response parameters

Name	Description
uid	Transaction's guid
status	Status of processed transaction.
type	Transaction type.
amount	Captured amount.
parent_uid	Captured amount.
currency	Currency in ISO-4217 format, e.g. USD.
ref_id	Acquirer transaction reference ID.
gateway_id	Gateway ID.
message	Processing result message.

Response example

```
{
  "transaction": {
    "capture": {
      "ref_id": "51411922672514728",
      "gateway_id": "2b1035a0-5498-4c8a-be77-7d0475ce7496",
      "status": "charged"
    },
    "uid": "bad39a58-0037-4f9e-9596-9514d7b9340c",
    "status": "success",
    "message": "Successfully processed",
    "amount": "250",
    "parent_uid": "dacbdef0-5f12-4d74-95bc-1051572f7e38",
    "currency": "USD",
    "description": "Test transaction",
    "type": "capture"
  }
}
```

Cancel-old

<http://{{host}}:{{port}}/api/v1/transactions/cancels>

Request parameters

Name	Mandatory	Description
parent_uid	Yes	UID of an authorization transaction.
amount	Yes	Amount to cancel in minimal currency units, e.g. 1000 for \$10.00.
reason	Yes	A note why a refund was made. Max length: 255 chars

Request example

```
{
  "parent_uid": "52729a32-107a-417c-8a87-9c9731c5a708",
  "amount": 10,
  "reason": "some reason"
}
```

Response parameters

Name	Description
uid	Processed transaction UID.
status	Status of processed transaction.
message	Processing result message.

Name	Description
type	Transaction type.
amount	Canceled amount
parent_uid	UID of an authorization transaction.
currency	Currency in ISO-4217 format, e.g. USD.
ref_id	Acquirer transaction reference ID.
gateway_id	Gateway ID.
reason	A note why a refund was made.
billing_descriptor	Billing descriptor assigned to transaction.

Response example

```
{
  "transaction": {
    "cancel": {
      "message": "Transaction has been reversed successfully",
      "gateway_id": "47532d37-b330-4011-b858-4244444b2a00",
      "billing_descriptor": "TEST GATEWAY BILLING DESCRIPTOR",
      "status": "0"
    },
    "uid": "7f65b561-0927-4b72-8101-b230937b283b",
    "status": "success",
    "message": "Successfully processed",
    "amount": "10",
    "parent_uid": "52729a32-107a-417c-8a87-9c9731c5a708",
    "currency": "EUR",
    "description": "Test transaction",
    "type": "cancel",
    "reason": "some reason"
  }
}
```

Refund-old

<http://{{host}}:{{port}}/api/v1/transactions/refunds>

Request parameters

Name	Mandatory	Description
parent_uid	Yes	UID of an authorization transaction.
amount	Yes	Amount to refund in minimal currency units, e.g. 1000 for \$10.00.
reason	Yes	A note why a refund was made. Max length: 255 chars

Request example

```
{
  "parent_uid": "ac67a625-8705-48bd-934d-b6b60731361f",
  "amount": 10,
  "reason": "without reason"
}
```

Response parameters

Name	Description
uid	Processed transaction UID.
status	Status of processed transaction.
message	Processing result message.
type	Transaction type.
amount	Refunded amount
parent_uid	UID of an authorization transaction.
currency	Currency in ISO-4217 format, e.g. USD.
ref_id	Acquirer transaction reference ID.
gateway_id	Gateway ID.
reason	A note why a refund was made.
billing_descriptor	Billing descriptor assigned to transaction.

Response example

```
{
  "transaction": {
    "refund": {
      "message": "Transaction has been refunded successfully",
      "gateway_id": "47532d37-b330-4011-b858-4244444b2a00",
      "billing_descriptor": "TEST GATEWAY BILLING DESCRIPTOR",
      "status": "0"
    },
    "uid": "c4b1f309-2e17-4fd6-980b-ea9d9ae64c20",
    "status": "success",
    "message": "Successfully processed",
    "amount": "10",
    "parent_uid": "ac67a625-8705-48bd-934d-b6b60731361f",
    "currency": "EUR",
    "description": "Test transaction",
    "type": "refund",
    "reason": "withaout reason"
  }
}
```

```
}  
}
```

Reverse-old

This request performs a reversal of a previous original payment referenced by parameter. The result is negative if called after the business day is closed. Similar to calling a refund request or cancel request.

Check transaction status

This request returns the status of the transaction by its tracking_id.

<http://{{host}}:{{port}}/api/v1/transactions/check>

Request parameters

Name	Mandatory	Description
tracking_id	Yes	ID of your transaction or order. Max length: 255 chars.

Request example

```
{  
  "tracking_id": "g15fdg-112df"  
}
```

Response parameters

Name	Description
status	Status of processed transaction.
transactions	Array of transactions.
status_transaction	Transaction status.
type	Transaction type.
uid	Transaction UID.
amount	Transaction amount.
description	Transaction description.
currency	Currency in ISO-4217 format, e.g. USD.
card_holder	Card holder.
card_number	Card number.
is_refunded	Whether a refund was made.

Name	Description
payment_status	Order status of this payment.

Response example

```
{
  "status": "success",
  "transactions": [
    {
      "status_transaction": "success",
      "type": "payment",
      "uid": "82b5ab2a-399e-45a4-b582-793792614bd7",
      "description": "tyktyk",
      "amount": "10",
      "currency": "EUR",
      "card_holder": "Inga Viktorovna",
      "card_number": "472492*****8434",
      "is_refunded": false,
      "payment_status": "paid"
    },
    {
      "status_transaction": "3dwaiting",
      "type": "payment",
      "uid": "3afd5083-0aba-4d04-8f4d-47cae2ceeeb3",
      "description": "tyktyk",
      "amount": "10",
      "currency": "EUR",
      "card_holder": "Inga Viktorovna",
      "card_number": "472492*****8434",
      "is_refunded": false,
      "payment_status": null
    }
  ]
}
```

Handling 3-D Secure transactions

To enable 3D secure, in the terminal editing, select the 3D flag.

If the card supports 3d secure, the gateway will send a link to confirm the transaction. You need to redirect the user to the link "redirect_url" or the gateway will redirect automatically, if this function is supported.

It is necessary that in the requests there were 2 links success url and failed url. After the redirect, the client will return to a successful link or a failed link.

Error response

Error response example:

```
{
  "guid": "f206dc60b27d11e995d63a87a064daed",
  "status": "Failed",
  "failed_url": "https://www.yourshop.com/search?q=failed",
  "error": "GatewayError",
  "description": {
    "code": "invalidPaymentStatus",
    "message": "Invalid invoice payment status"
  }
}
```

Types of errors:

- AntifraudError
- GatewayError
- InternalError
- InvalidDataError
- LimitReacheError
- MysqlError
- PrivilegesError
- ProcessorError
- ShopAuthorizationError
- ValidationError

If gateway send error it mark as Gateway error. Code and message sends from gateway, for more information about this error, check gateways documention. Error is generated by the platform and has a description in the table below

Error	Description
Unauthorized	make sure that shop secret and shop hash key are correct
Type, terminalguid doesn't presented	request doesn` t have type or shop guid
Any gateway not found	shop doesn't have any gateways
Any account not found	shop doesn't have any accounts
Currency {currency} doesn't find in database	this currency is not in the database

Error	Description
Transaction isn't have template	transaction is not supported
Validation in step params doesn't presented	transaction template error
Options in step params doesn't presented	transaction template error
Step {step_name} not found in db.stepProcessing for {transactionId}	transaction template error
Params for {savedStep} not found in db.stepParamProcessing	transaction template error
Part of params for {savedStep} not found in db.stepParamProcessing	transaction template error
Option isParentTransactionGuidWrite is true, but transactionId is undefined	transaction id not specified
Uncorrect data - currency or amount is undefined	this or previous transaction does not have amount or currency
Failed starting transaction: {key} is undefined	make sure that shop gateways params are correct
Any shop gateway not found	the gateway is not tied to the store
Payment limit reached	payment amount reached the limit
Any gateway processed amount not found	
Shop monthly payment limit reached	payments amount for month reached the shop limit
Any merchant limit not found	merchant doesn't have limit
Any merchant processed amount limit not found	
Merchant monthly payment limit reached	payments amount for month reached the merchnat limit
Shops hashKey doesn't generated	generate new hash key
Blacklist blocked request cause of: card	this card is blocked
Blacklist blocked request cause of: ip	this ip is blocked
Blacklist blocked request cause of: country	this country is blocked
Blacklist blocked request cause of: country(ip)	this country is blocked
Blacklist blocked request cause of: country(credit card)	this country is blocked
Blacklist blocked request cause of: card(mask)	this card is blocked
Blacklist blocked request cause of: card(bin)	this bin is blocked

Error	Description
Invalid settings for {type} with gateway guid {gatewayGuid}	invalid transaction template

If you get an error make sure you use the correct data.

Code example

This example shows how to do request to base banking platform. To generate an authorization header, you must use Basic access authentication. To generate signature, you must use sha256 hash from a request body with a terminal hash key concatenating.

```
let data = {
  succeed_url: "http://localhost:3001/success",
  failed_url: "http://localhost:3001/failed",
  terminalGuid: {{terminalGuid}},
  type: "Authorization",
  amount: {{amount}},
  currency: "RUB",
  description: "Test transaction",
  billing_address: {
    first_name: "John",
    last_name: "Doe",
    country: "US",
    city: "New York",
    address: "Groove st.",
    zip: "646464"
  },
  customer: {
    email: "mail@mail.com",
    ip: "200.200.200.200"
  }
}

const signature =
crypto.createHash("sha256").update(JSON.stringify(data) +
terminalHashkey).digest("base64");
data.signature = signature;

axios.post('http://{{host}}/api/v1/transactions/create', data, {
  auth: { "username": {{terminalGuid}}, "password": {{terminalSecret}} } })
  .then((response) => {
    console.log(response);
    newUrl = "http://{{host}}/api/v1" + response.data.redirect_to.url
+ "?auth_key=" + response.data.redirect_to.params.auth_key;
  })
  .catch(function (error) {
    console.log(error);
  });
```

To connect the payment form, you can use the iframe that the platform sends. Or connect checkout script.

```
checkoutBBP.init({
  terminalGuid: "1d1a1ea2664c11e98eab0242ac110002",
  terminalSecret:
    "22f080880cc58e54a2334f768b201beacd661957ab7cf18e39bde7b7c0f2ad4fc23647e5c8
    5095eb10360edd9fdb7670e23c48e76fc87e9dd83c5423b8ba9186",
  hashKey:
    "07cc44b691aebd560905e5e91f8bc05f4e751f1f10f60bb9cf64fdb59222c54382746a6d52
    eae86d59d3acd7aab352efd79a217bc0c1993c36ca826626e0bcb9",
  amount: `${localStorage.getItem("total")}`,
  currency: "RUB",
  description: "Test checkout widget",
  type: "Payment",
  billing_address: {
    first_name: "John",
    last_name: "Doe",
    country: "US",
    city: "New York",
    address: "Groove st.",
    zip: "646464"
  },
  customer: {
    email: "mail@klta.net",
    ip: "190.190.190.190"
  },
  redirect: {
    succeed_url: "http://localhost:3001/success",
    failed_url: "http://localhost:3001/failed"
  }
});
checkoutBBP.func();
```

If you have any questions or require assistance, please do not hesitate to contact [Payseven Technical Team](#)