



BPC Banking Technologies

Merchant integration

manual

SATIM



Date: 19 june 2018
V0.8

Table of contents	
About the document	3
Terms	3
Introduction	4
Connection	4
Interaction between systems during a payment	5
<i>Scheme of implementation on the merchant's side</i>	7
Algorithm of connecting to the payment gate	7
REST interface	8
<i>Testing of REST requests in an Internet browser</i>	9
<i>One-phase payment requests</i>	9
Order registration request	9
SATIM specific. Confirm order request	16
<i>Two-phase payment requests</i>	12
Preauthorized order registration request	12
Order completion request	14
<i>Additional methods</i>	16
Order reversal request	16
Refund request	19
Order state request	20
Payment statistics for a specified period	25
Address data for connections	29



About the document

This document describes principles of connecting and program interfaces of the payment gate.

Terms

- 3D Secure (Verified by Visa) is a technology developed by Visa that allows to perform additional authorization for a client on the side of the issuing bank.
- ACS is acronym for Access Control Server which is an element of 3D Secure infrastructure that ensures validation of the payer by the issuing bank.
- Merchant Plugin Interface (MPI) is a component of 3D Secure technologies that may be implemented in the payment gate or on the merchant's side.
- SecureCode is a technology developed by MasterCard that allows to perform additional authorization for a client on the side of the issuing bank. Practically the same as the 3-D Secure technology. Further in the document both technologies will be referred to as 3D Secure.
- Bank card is a plastic card linked to one or more bank accounts and is used to pay for goods and services including payments through the Internet.
- Acquiring bank is the bank that implements and runs the payment gate.
- Issuing bank is the bank that issued a client's bank card.
- Refund is a partial or full refund to the payer's card in case of refusal from goods (services) or refund of goods. The operation of money return is performed after money withdrawal from the payer's account.
- Two-phase payment is operation on goods/services payment carried out through the Internet with use of bank card, which requires additional confirmation. Two-phase mechanism allows to divide a process into checking of bank card paying capacity (authorization) and money withdrawal (financial confirmation). At the first step of two-phase payment bank card paying capacity is checked and money on the Client's account are blocked.
- Order – basic entity of the system, it describes an order in a certain internet store or an analogue thereof. Each order has an amount.
- Merchant – Trade and Service Company (TSC), selling goods or providing services through the Internet.
- Mobile payment page - functionality of a mobile and ordinary payment pages is the same. Mobile version differs in the layout, that takes into account the need for a compact and high-quality display of the page on the screens of mobile devices.
- IPS is acronym for International Payment System (e.g. Visa or MasterCard).
- One-phase payment -- operation on goods/services payment carried out through the Internet with use of bank card, which does not require additional confirmation.
- Reversal is the unlocking of funds on the payer's account. This feature is available for a limited time, the exact terms to be specified in the bank.
- Payment form is an HTML page that clients use to enter payment properties.
- Payment details – details used by a user to pay an order. Generally, details include card number, expiration date, CVC
- Payment gate of Bank-acquirer (PG) – automated system allowing a Merchant to receive and a Client to send payments through the Internet using bank cards.



Introduction

Internet-acquiring is up-to-date way of goods or services selling through the public Internet using bank cards.

Due to peculiarities of purchasing through the Internet there is a necessity to ensure the safe interaction of the parties involved in the operation of goods/services selling – client, merchant and bank-acquirer.

The merchant planning to sell goods or services through the Internet by bank cards should carry out certain procedures to ensure payment security:

- At the moment of key data transferring (personal data, bank cards details) the interaction with the client should be performed with use of cryptographic means (SSL/TLS).
- Information on the current payment (amount, currency, order description) and the result of the payment should be effectively protected from interference of intruders.
- At the moment of payment procedures of checking of card belonging to the client should be applied.

To meet these requirements the Bank-acquirer uses a specialized technology 3DSecure, developed by International Payment Systems Visa International and MasterCard (Verified by Visa and MasterCard SecureCode) to ensure the safety of Internet payments.

As a technology platform a payment gate is used. It allows a merchant to carry out necessary safety procedures without a significant restructuring of the Internet store site and business processes existing.

Connection

For connection to the system the merchant provides:

1. HTML-page (with graphics and CSS and other pluggable objects) that displays the payment form. Requirements to this page are described in "Payment page design" document.

As a result of connection the store receives:

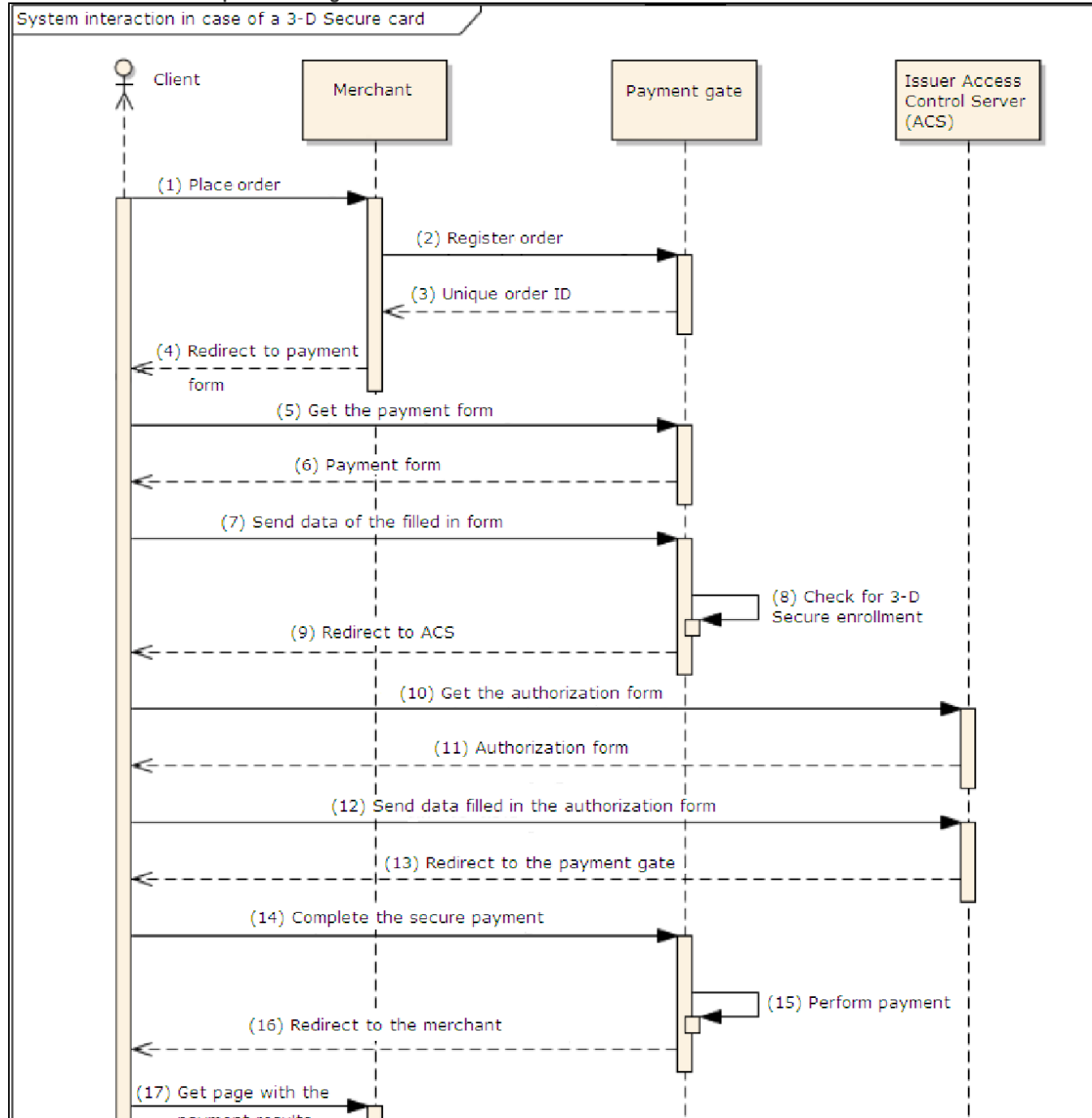
1. Login – name of the merchant in the payment gate. It is used only for operations performed by API.
2. Password – password of the merchant in the payment gate. It is used only for operations performed by API.

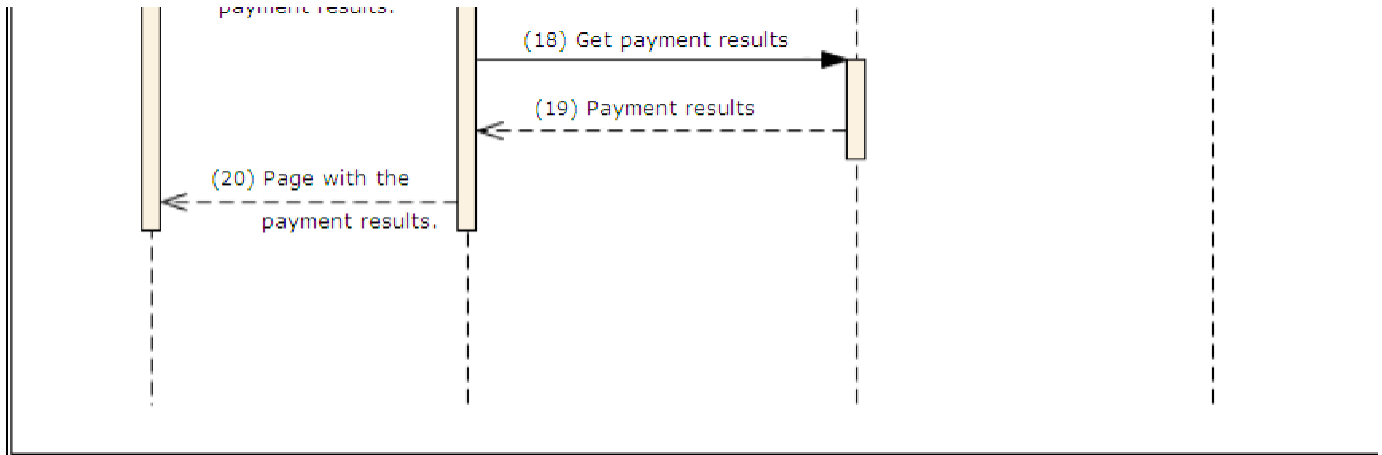


Interaction between systems during a payment

There are two possible cases for payment: the first case is with the card being involved with a 3D Secure technology, and the second one is with the card being not involved in 3D.

Interaction scheme of the process using 3D Secure





Description:

1. The client forms an order using the online shop of the merchant.
2. When the order is confirmed by the client, the merchant system registers the order in the payment gate. The following parameters are used for order registering: amount to be charged and currency of the expected payment, internal order number and merchant URL for client redirecting after payment.
3. The payment gate responds with a unique identifier of the registered order and with a URL of the payment form that the client should be redirected to.
4. Merchant system redirects the client to the URL generated in the previous step.
5. Client's browser opens the URL.
6. As a result, the browser opens the payment form.
7. The client fills in the form and sends the data to the payment gate server.
8. The payment gate checks if the specified card is involved into 3D Secure (SecureCode).
9. The gate redirects the client's browser to the ACS page of the issuer bank (this step is mandatory for 3D Secure processing).
10. Client's browser requests authorization form from the ACS (each issuer implements this step in it's own way).
11. ACS sends the form to the client.
12. Client fills in this form and sends the data back to ACS.
13. ACS processes the data and (regardless of the result) redirects the client's browser to a page of the payment gate. Along with it the URL of the page browser receives enciphered authorization results.
14. Client's browser opens the page, passing authorization results to the payment gate.
15. The payment gate performs the payment.
16. When the payment is processed, the payment gate sends the return URL to the client's browser (specified by the merchant during the order registration).
17. Client's browser requests the payment results page from the merchant.
18. The merchant system requests the confirmation and payment state from the payment gate using the order identifier assigned by the gate earlier.
19. The payment gate returns the payment status.
20. The merchant system sends the page with payment results to the client's browser.



Scheme of implementation on the merchant's side

There are several automatic interactions between a merchant and the payment gate:

- steps 2, 3
- steps 18, 19

These interactions are synchronous, i.e. requesting side must wait for a response or timeout to continue. Interactions are also unilateral: the merchant system sends requests to the payment gate, not the other way. To implement these interactions developers can use REST API provided by the payment gate, which consists of 2 requests:

1. Register order.
2. Confirm order (Get information about order status)

Note: The client has limited time for payment - 20 minutes from order registration. Attempt to perform the payment after 20 minutes causes displaying of the error page.

Algorithm of connecting to the payment gate

1. Receive logins and passwords for the test server.
2. Design the payment page.
3. Upload archived payment page files to the test server.
4. Test the payment page:
 - using REST interface;
 - using administrative console.
5. When integration and the page are ready, ask the support group to verify your payment page. If the verification is successful, support group will transfer the payment page to the production server.
6. Receive login and password pair (or pairs, for multiple users) for the production server.
7. Reconfigure the merchant's system for using the production server as the payment gate.
8. Try paying for an order with a real card (it is recommended to use both 3DS and SSL cards for testing to be thorough).
9. Reverse and refund these payments via the console.
10. Send order identifiers (mdOrder) from steps 8-9 to the support group for verification.
11. Receive confirmation of successful verification and possibility of switching to production.



REST interface

Interactions are implemented as HTTP requests with GET or POST methods to certain URLs, separate for each individual request type (see "Address data for connections" section for exact addresses). Parameters of interactions are sent as parameters of GET or POST requests; values should be URL encoded.

Result of processing the request returns as a JSON object. Example:

```
{"errorCode":"12","errorMessage":"Empty amount"}
```

For authorization purposes each request should include login and password of the merchant that were produced during its registration. Login and password values are sent as the following parameters:

Name	Type	Mandatory	Description
userName	AN..30	yes	Login of a user received during registration.
password	AN..30	yes	Password of a user received during registration.

Requests for one- and two-phase schemes may differ. Requests for each scheme are described below.

All text fields must have UTF-8 encoding.

Special characters in REST requests must be screened in accordance with the URL-code. The table of characters is represented here - <http://web-developer.name/urlcode/>. For example, password "qwe?rt%y" should be transferred as "qwe%0Frt%25y".

If the Error code = 0, the request was processed by the Payment gate without system errors (error code does not show the status of the order).



To get the status of an order should use the request `getOrderStatus` or `getOrderStatusExtended`.

Testing of REST requests in an Internet browser

Testing of REST requests with simultaneous work with administrative console in the same Internet browser will entail errors in the console. If you login in the console, then execute one or more REST requests in the other tab, any try to perform an action in the console after that cause an error. To make administrative console work correctly:

- In some cases it is enough to logout and login again;
- Sometimes it is necessary to delete cookie files or to wait until the session timeout.

Such situation may be avoided due to one of the following ways:

- Use incognito mode for REST requests, while working in the administrative console in the regular window;
- Use different Internet browsers for REST requests and administrative console.

One-phase payment requests

Order registration request

To register an order use `register.do` request (see "Address data for connections" section). This request is designed for registering orders in the payment gate.

Request parameters:

Name	Type	Mandatory	Description
userName	AN..30	yes	Merchant's login received during registration.
password	AN..30	yes	Merchant's password received during registration.
orderNumber	AN..10	yes	Number (identifier) of the order in the merchant's system.
amount	N..20	yes	Order amount in minimal currency units (e.g. cents).
currency	N3	yes	Code of the payment currency, according to ISO 4217.
returnUrl	AN..512	yes	Web address that the client should be redirected to after successful payment.
failUrl	AN..512	no	Web address that the client should be redirected if the payment failed.
description	AN..512	no	Free-formed description of the order.
language	A2	no	Language according to ISO 639-1. If not specified, the system uses the default language from the merchant's settings.
pageView	A..7	no	This parameter indicates a type of device used -- mobile device (pageView=MOBILE) or desktop (pageView=DESKTOP). Depending on this parameter value the response will contain a link to a mobile payment page or an ordinary payment page. The default value is DESKTOP.
clientId	AN..255	no	Client number (ID) in the shop system. It is used for bundles functionality implementation. Presents only if a shop is allowed to create links.
jsonParams	AN..1024	no	Fields of additional information for storage, the type is {"param":value,"param2":value 2}. These fields may be transferred to the bank processing for their representation in registers. * This functionality can be switched on during the integration period at the agreement with bank. See SATIM specific parameters below.
sessionTimeoutSecs	N...9	no	Duration of the session in seconds. If the parameter is not specified, then the value from merchant settings is used, otherwise - the default value (1200 seconds - 20 minutes). If the request contains expirationDate parameter, the parameter sessionTimeoutSecs is ignored.
expirationDate	ANS	no	Order expiration date. Format: yyyy-MM-dd'T'HH:mm:ss. If this parameter is not specified, the sessionTimeoutSecs parameter is used to determine an order life time.



bindingId	AN..255	no	<p>Identifier of the binding, that was created earlier. May be used only if the merchant has permission to work with bindings. If this parameter is sent in the <code>registerOrder</code> request, that means:</p> <ol style="list-style-type: none"> 1. This order may be payed only by binding; 2. The payer will be redirected to a payment page, where entering of CVC only is required.
-----------	---------	----	---

* Fields "orderNumber" and its "description" are sent to the bank processing by default (not more than 99 simbols, the following simbols are forbidden to use - %, +, \r, \n)

SATIM specific Json request field

Field	Format	Example	Description
"force_terminal_id"	AN..16	E0123456789	<p>Mandatory parameter.</p> <p>Terminal ID which is assigned by bank to Merchant.</p> <p>One merchant might have several terminals</p>
"feTransType"	N3	698	<p>Optional Parameter:</p> <p>Specific Transactions used by SATIM:</p> <p>698 – Bill Payment</p> <p>676 – Balance Inquiry</p>
"transaction_type"	ANS..16	"balance_inquiry"	<p>This is mandatory parameter for Balance Inquiry transaction</p> <p>To indicate Balance Inquiry request put parameters:</p> <ul style="list-style-type: none"> - "feTransType" with value "676" - And "transaction_type" field with value "balance_inquiry". <p>Note amount should be 0. See balance inquiry request example below.</p>
srvID	AN..8	PMTBILL01	<p>This is mandatory parameter for Bill Payment transaction.</p> <p>This parameter identifies the ID of biller. List of biller ID will be provided by SATIM to merchant.</p> <p>To indicate bill payment request put parameters:</p> <ul style="list-style-type: none"> - "feTransType" with value "698" - "srvID" with ID of billers (list of ID will be provided by SATIM)
udf1 udf2 udf3 udf4 udf5	AN..20	Udf1 Mandatory udf2 udf3 udf4 udf5 Mandatory	<p>Mandatory parameter.</p> <p>These are SATIM specific values.</p> <p>Merchant may register up to 5 additional parameters.</p> <p>udf5=01 for Bill Payment transaction.</p> <p>udf5=00 for Payment transaction.</p>

Response parameters:

Name	Type	Mandatory	Description
orderId	ANS20	no	Order number in the payment gate. Unique within the gate. Absent if order registration failed (error is described in ErrorCode).
formUrl	AN..512	no	URL of the payment form that the client should be redirected to. Absent if order registration failed (error is described in ErrorCode).
errorCode	N3	no	Code of the error that occurs during registration.
errorMessage	AN..512	no	Description of the error in the language sent in the Language parameter of the request.

Error codes (values of the errorCode field) Errors classification:

Value	Description
-------	-------------



0	No system error.
1	Order number is duplicated, order with given order number is processed already
3	Unknown currency
4	Mandatory request parameter was not specified
5	Erroneous value of a request parameter.
7	System error.

Possible error messages:

Value	Description
0	No system error
1	Order with this number was already processed.
1	Order with this number was registered, but was not paid off.
1	Invalid orderNumber
3	Unknown currency.
4	Order number is empty
4	Empty merchant user name
4	Empty amount
4	Return URL cannot be empty
4	Password cannot be empty
5	Wrong amount.
5	Wrong value of the Language parameter.
5	Access denied
5	The user must change his password
5	Invalid [jsonParams]
7	System error

Example: when the 4 error is returned ("mandatory request parameter was not specified"), it means that the order registration request lacked one of the following parameters: language, order number, login, amount, return URL, or password.

GET Request example:

```
https://server/application_context/rest/register.do?
amount=100&currency=810&language=ru&orderNumber=87654321&password=testPwd&
returnUrl=finish.html%3Flogin%3Dtest%26password%3DtestPwd&
userName=test&jsonParams={"orderNumber":1234567890}&pageView=MOBIL
&expirationDate=2014-09-08T14:14:14
```

POST Request example:

```
amount=100&currency=810&language=ru&orderNumber=87654321&
returnUrl=finish.html%3Flogin%3Dtest%26password%3DtestPwd&
jsonParams={"orderNumber":1234567890}&MOBILE
&expirationDate=2014-09-08T14:14:14
```



Response example:

```
{ "orderId": "123456789AZERTYUIOPL", "formUrl":  
  "https://server/application_context/merchants/test/mobile_payment_ru.html?  
  mdOrder=123456789AZERTYUIOPL" }
```

Balance inquiry request:

```
https://server/application_context/rest/register.do? amount=0&currency=810&language=ru&orderNumber=87654321&password=testPwd&  
returnUrl=finish.html%3Flogin%3Dtest%26password%3DtestPwd& userName=test&jsonParams={  
  "force_terminal_id": "1111111111111111", "orderNumber": "1234567890",  
  "feTransType": "676", "transaction_type": "balance_inquiry", "udf1": "111", "udf2": "222", "udf3": "333", "udf4": "444", "udf5": "555"}&page  
View=MOBIL &expirationDate=2014-09-08T14:14:14
```

Response example:

```
{ "orderId": " 123456789AZERTYUIOPL", "formUrl":  
  "https://server/application_context/merchants/test/mobile_payment_ru.html?  
  mdOrder= 123456789AZERTYUIOPL" }
```

Two-phase payment requests

Next 2 methods can be used by merchants to perform Pre-Authorization and Completion transactions

Preauthorized order registration request

Use `registerPreAuth.do` request for order registration in two-phase payment scheme (see "Address data for connections" section).

Request parameters:

Name	Type	Mandatory	Description
userName	AN..30	yes	Merchant's login received during registration.
password	AN..30	yes	Merchant's password received during registration.
orderNumber	AN..10	yes	Number (identifier) of the order in the merchant's system.
amount	N..20	yes	Order amount in minimal currency units (e.g. cents).
currency	N3	yes	Code of the payment currency, according to ISO 4217.
returnUrl	AN..512	yes	Web address that the client should be redirected to after successful payment.
failUrl	AN..512	no	Web address that the client should be redirected if the payment failed.
description	AN..512	no	Free-formed description of the order.
language	A2	no	Language according to ISO 639-1. If not specified, the system uses the default language from the merchant's settings.
pageView	A..7	no	This parameter indicates a type of device used -- mobile device (pageView=MOBILE) or desktop (pageView=DESKTOP). Depending on this parameter value the response will contain a link to a mobile payment page or an ordinary payment page. The default value is DESKTOP.



clientId	AN..255	no	Client number (ID) in the shop system. It is used for bundles functionality implementation. Presents only if a shop is allowed to create links.
jsonParams	AN..1024	no	Fields of additional information for storage, the type is {"param":value,"param2":value 2}. These fields may be transferred to the bank processing for their representation in registers. * This functionality can be switched on during the integration period at the agreement with bank.
sessionTimeoutSecs	N...9	no	Duration of the session in seconds. If the parameter is not specified, then the value from merchant settings is used, otherwise - the default value (1200 seconds - 20 minutes). If the request contains expirationDate parameter, the parameter sessionTimeoutSecs is ignored.
expirationDate	ANS	no	Order expiration date. Format: yyyy-MM-dd'T'HH:mm:ss. If this parameter is not specified, the sessionTimeoutSecs parameter is used to determine an order life time.
bindingId	AN..255	no	Identificator of the binding, that was created earlier. May be used only if the merchant has permission to work with bindings. If this parameter is sent in the <code>registerOrder</code> request, that means: 1. This order may be payed only by binding; 2. The payer will be redirected to a payment page, where entering of CVC only is required.

* Fields "orderNumber" and its "description" are sent to the bank processing by default (not more than 99 simbols, the following simbols are forbidden to use - %, +, \r, \n)

Response parameters:

Name	Type	Mandatory	Description
orderId	ANS20	no	Order number in the payment gate. Unique within the gate. Absent if order registration failed (error is described in ErrorCode).
formUrl	AN..512	no	URL of the payment form that the client should be redirectd to. Absent if order registration failed (error is described in ErrorCode).
errorCode	N3	no	Code of the error that occurs during registration.
errorMessage	AN..512	no	Description of the error in the language sent in the Language parameter of the request.

Error codes (values of the errorCode field) Errors classification:

Value	Description
	No system error.
1	Order number is duplicated, order with given order number is processed already
3	Unknown currency
4	Mandatory request parameter was not specified
5	Erroneous value of a request parameter.
7	System error.

Possible error messages:

Value	Description
0	No system error
1	Order with this number was already processed.
1	Order with this number was registered, but was not paid off.
1	Invalid orderNumber



3	Unknown currency.
4	Order number is empty
4	Empty merchant user name
4	Empty amount
4	Return URL cannot be empty
4	Password cannot be empty
5	Wrong amount.
5	Wrong value of the Language parameter.
5	Access denied
5	The user must change his password
5	Invalid [jsonParams]
5	Pre-authorization payment is restricted
7	System error

GET request example:

```
https://server/application_context/rest/registerPreAuth.do?
amount=100&currency=810&language=ru&orderNumber=87654321&password=testPwd&
returnUrl=finish.html%3Flogin%3Dtest%26password%3DtestPwd&userName=test&pageView=MOBILE&
expirationDate=2014-09-08T14:14:14
```

POST request example:

```
amount=100&currency=810&language=ru&orderNumber=87654321&
returnUrl=finish.html%3Flogin%3Dtest%26password%3DtestPwd&pageView=MOBILE&
expirationDate=2014-09-08T14:14:14
```

Response example:

```
{"orderId":"123456789AZERTYUIOPL","formUrl":
"https://server/application_context/mobile_payment_ru.html?
mdOrder=123456789AZERTYUIOPL"}
```

Order completion request

Use deposit.do request to complete preauthorized payment.

This operation can be performed if there is a corresponding permission in the system. Request parameters:



Name	Type	Mandatory	Description
userName	AN..30	yes	Merchant's login received during registration.
password	AN..30	yes	Merchant's password received during registration.
orderId	ANS20	yes	Order number in the payment gate. Unique within the gate.
amount	N..20	yes	Order amount in minimal currency units (e.g. cents).

Attention!!! If you do not specify zero in "DepositAmount" parameter, completion occurs for the entire pre-authorized amount.

Response parameters

Name	Type	Mandatory	Description
errorCode	N3	no	Error code.
errorMessage	AN..512	no	Description of the error.

Error codes (values of the errorCode

field) Error classification:

Value	Description
0	No system error.
5	Erroneous value of a request parameter.
6	Unregistered OrderId.
7	System error.

Possible error messages:

Value	Description
0	No system error
5	Access denied
5	The user must change his password
5	Invalid amount
5	Deposit amount must be zero, or more than 1 currency unit (e.g. 1 euro).
6	Wrong order number.
7	Payment must be in a correct state.
7	System error.

GET Request example:

```
https://server/application_context/rest/deposit.do?amount=100&currency=643&language=ru&orderId=e5b59d3d-746b-4828-9da4-06f126e01b68&password=xxxxx&userName=testtest
```

POST Request example:

```
amount=100&currency=643&language=ru&orderId=e5b59d3d-746b-4828-9da4-06f126e01b68
```



Response example:

```
{"errorCode":0}
```

Additional methods

SATIM specific. Confirm order request

The request should be used to confirm that a merchant has successfully handled a client redirection back to merchant page after successful payment attempt. Returns order status details. If no confirmation request received by payment gateway then an order will be automatically reversed after some delay.

Request parameters:

Name	Type	Mandatory	Description
userName	AN..30	yes	Merchant's login received during registration.
password	AN..30	yes	Merchant's password received during registration.
orderId	ANS20	yes	Order number in the payment gate. Unique within the gate.
language	A2	no	Language according to ISO 639-1. If not specified, the system uses the default language from the merchant's settings. Error messages returned will be written in this very language.

Response parameters:

Name	Type	Mandatory	Description
orderStatus	N2	no	This parameter shows order status in the payment gate. Possible values are listed in the table below. Parameter is absent if no order with the specified ID was found.
errorCode	N3	no	Error code.
errorMessage	AN..512	no	Description of the error in the language sent in the Language parameter of the request.
orderNumber	AN..20	yes	Number (identifier) of the order in the merchant's system.
actionCode	N3	yes	Processing system authorization code.
actionCodeDescription	AN..512	yes	Action code description on the language passed in the parameter "Language".
pan	N..19	no	Masked number of the card that was used in payment. Specified only for paid orders.
expiration	N6	no	Card expiration date in the YYYYMM format. Specified only for paid orders.
cardholderName	A..64	no	Cardholder name. Specified only for paid orders.
amount	N..20	yes	Order amount in minimal currency units (e.g. cents).
currency	N3	no	Code of the payment currency, according to ISO 4217.
approvalCode	AN6	no	IPS authorization code. The field has fixed length of six characters, it can contain both numbers and letters.
authCode	N3	no	This parameter is deprecated. Its value is always "2", regardless of the order status and processing system authorization code.
ip	AN..20	no	IP address of the user who paid for the order.
bindingInfo (element consists of clientId and bindingId):			
clientId	AN..255	no	Number of the client (identifier) in the merchant's system. Used to implement a functional of ligament. May be present, if the merchant is allowed to create the ligament.



bindingId	AN..255	no	Identifier of the ligament created while paying an order or used to pay for. Present only if creation of ligaments is allowed for merchant.
-----------	---------	----	---

Possible codes for errors (the errorCode field):

Значение	Описание
0	Successfully confirmed
1	Empty order id
2	Already confirmed
3	Order is not confirmed due to order's state
5	Access denied
6	Unknown order
7	System error

GET Request example:

```
https://server/application_context/rest/confirmOrder.do?language=en&orderId=5e97e3fd-1d20-4b4b-a542-f5995f5e8208&password=xxxxx&username=testtest
```

POST Request example:

```
language=ru&orderId=123456789AZERTYUIOPL password=xxxxx&username=testtest
```

Response example:

```
{"expiration":"201810","cardholderName":"JOHN DOE","depositAmount":1002,"currency":"012","approvalCode":"058462","authCode":2,"params":{"respCode_desc":"Your payment was accepted","respCode":"00"},"actionCode":0,"actionCodeDescription":"Request processed successfully","errorCode":"3","errorMessage":"Order is not confirmed due to order's state","OrderStatus":3,"OrderNumber":"201851912556922","Pan":"502145**0917","Amount":1002,"Ip":"0:0:0:0:0:0:1","SvfeResponse":"00"}
```

In case when order marked using jsonParams as balance inquiry then amount field will be present in response params field:

Balance response example:

```
{"expiration":"201712","cardholderName":"JOHN DOE","depositAmount":100,"currency":"012","authCode":2,"errorCode":"0","errorMessage":"Success","OrderStatus":2,"OrderNumber":"201771117539439","Pan":"628070**0000","Amount":100,"Ip":"10.7.4.244","SvfeResponse":"00","params":{"host_balance":"500","host_resp":"00","host_resp_desc":"Successful transaction","host_date":"170901","host_time":"135212"}}
```

Order reversal request

To cancel payment of the order, use `reverse.do` request. This functionality is available for a limited period, that is specified by the Bank. The reversal operation may be performed only once. If the reversal request caused an error, the next try will not be successful.

The reversal operation is available for merchants only under agreement with the Bank. To perform the reversal request a user must have an appropriate permission.

Request parameters:

Name	Type	Mandatory	Description
------	------	-----------	-------------



userName	AN..30	yes	Merchant's login received during registration.
password	AN..30	yes	Merchant's password received during registration.
orderId	ANS36	yes	Order number in the payment gate. Unique within the gate.
language	A2	no	Language according to ISO 639-1. If not specified, the system uses the default language from the merchant's settings. Error messages returned will be written in this very language.

Response parameters:

Name	Type	Mandatory	Description
errorCode	N3	no	Error code.
errorMessage	AN..512	no	Description of the error.

Error codes (values of the errorCode field)

Error classification:

Value	Description
0	No system error.
5	Erroneous value of a request parameter.
6	Unregistered OrderId.
7	System error.

Possible error messages:

Value	Description
0	No system error
5	Access denied
5	The user must change his password
5	[orderId] is empty
6	Wrong order number.
7	Payment must be in a correct state.
7	Reversal is impossible. Reason: wrong internal values, check hold and deposited amounts.
7	System error.

GET Request example:

```
https://server/application_context/rest/reverse.do?
language=en&orderId=123456789AZERTYUIOPL&password=password&userName=login
```

POST Request example:

```
language=en&orderId=123456789AZERTYUIOPL
```



Response example:

```
{"errorCode":"0","errorMessage":"Success"}
```

Refund request

Use `refund.do` request to refund deposited money.

This request returns money paid for the order back to the client. Request causes an error if the client is not charged. The payment gate allows multiple refunds, but their total amount cannot exceed the amount that was deposited from the client's account as a result of the order.

A user must have a corresponding permission in the system to perform this operation.

Request parameters:

Name	Type	Mandatory	Description
userName	AN..30	yes	Merchant's login received during registration.
password	AN..30	yes	Merchant's password received during registration.
orderId	ANS20	yes	Order number in the payment gate. Unique within the gate.
amount	N..20	yes	Order amount in minimal currency units (e.g. cents).

Response parameters:

Name	Type	Mandatory	Description
errorCode	N3	no	Error code.
errorMessage	AN..512	no	Description of the error.

Error codes (values of the `errorCode` field)

Classification:

Value	Description
0	No system error.
5	Erroneous value of a request parameter.
6	Unregistered OrderId.
7	System error.

Possible error messages:

Value	Description
0	No system error
5	Access denied
5	The user must change his password
5	[orderId] is empty



6	Wrong order number.
7	Payment must be in a correct state.
7	Wrong deposit amount (less than 1 currency unit, e.g. 1 euro).
7	System error.

GET Request example:

```
https://server/application_context/rest/refund.do?amount=500&currency=643&language=ru&orderId=123456789AZERTYUIOPL&password=xxxxx&userName=testtest
```

POST Request example:

```
amount=500&currency=643&language=ru&orderId=123456789AZERTYUIOPL
```

Response example:

```
{"errorCode":0}
```

Order state request

To get the current state of a registered order send data with `getOrderStatus` method (GET or POST) to the corresponding URL (see "Address data for connections" section).

The order status should be determined by the value of `orderStatus` parameter.

`authCode` field is deprecated.

Request parameters:

Name	Type	Mandatory	Description
userName	AN..30	yes	Merchant's login received during registration.
password	AN..30	yes	Merchant's password received during registration.
orderId	ANS20	yes	Order number in the payment gate. Unique within the gate.
language	A2	no	Language according to ISO 639-1. If not specified, the system uses the default language from the merchant's settings. Error messages returned will be written in this very language.

Response parameters:



Name	Type	Mandatory	Description
orderStatus	N2	no	This parameter shows order status in the payment gate. Possible values are listed in the table below. Parameter is absent if no order with the specified ID was found.
errorCode	N3	no	Error code.
errorMessage	AN..512	no	Description of the error in the language sent in the Language parameter of the request.
orderNumber	AN..10	yes	Number (identifier) of the order in the merchant's system.
pan	N..19	no	Masked number of the card that was used in payment. Specified only for paid orders.
expiration	N6	no	Card expiration date in the YYYYMM format. Specified only for paid orders.
cardholderName	A..64	no	Cardholder name. Specified only for paid orders.
amount	N..20	yes	Order amount in minimal currency units (e.g. cents).
currency	N3	no	Code of the payment currency, according to ISO 4217.
approvalCode	AN6	no	IPS authorization code. The field has fixed length of six characters, it can contain both numbers and letters.
authCode	N3	no	This parameter is deprecated. Its value is always "2", regardless of the order status and processing system authorization code.
ip	AN..20	no	IP address of the user who payed for the order.
bindingInfo (element consists of clientId and bindingId):			
clientId	AN..255	no	Number of the client (identifier) in the merchant's system. Used to implement a functional of ligament. May be present, if the merchant is allowed to create the ligament.
bindingId	AN..255	no	Identifier of the ligament created while paying an order or used to pay for. Present only if creation of ligaments is allowed for merchant.

The **orderStatus** field may have the following values:

State number	Description
0	Order registered, but not paid off.
1	Pre-authorisation amount was held (for two-phase payment)
2	The amount was deposited successfully
3	Authorization reversed
4	Transaction was refunded.
5	Authorization through the issuer's ACS initiated
6	Authorization declined

Error codes (values of the **errorCode** field)

Value	Description
0	No system error
2	The order is declined because of an error in the payment credentials.
5	Access denied
5	The user must change his password
5	[orderId] is empty
6	Unregistered OrderId



Json response fields

Field	Format	Example	Description
host_balance	N	500	Balance returned by host for balance inquiry request
host_resp	AN2	00	Host response code
host_resp_desc	TEXT	Successful transaction	Host response description. Description language depends on request "language" parameter
host_date	YYMMDD	170901	Host transaction date
udf1	AN..20	Udf1 Mandatory	Mandatory parameter.
udf2		udf2	These are SATIM specific values.
udf3		udf3	Merchant may register up to 5 additional parameters.
udf4		udf4	udf5=01 for Bill Payment transaction.
udf5		udf5 Mandatory	udf5=00 for Payment transaction.

GET Request example:

```
https://server/application_context/rest/getOrderStatus.do?language=ru&orderId=123456789AZERTYUIOPL&password=xxxxx&u
serName=testtest
```

POST Request example:

```
language=ru&orderId=123456789AZERTYUIOPL
```

Response example:

```
{"expiration":"201512","approvalCode":"000000","Pan":"123456**6740", "Amount":1000,"cardholderName":"yrey
rgre","OrderStatus":4,"authCode":2,"OrderNumber":"132353464","depositAmount":500,"Ip":"127.0.0.1",
"params":{"host_resp":"00","host_resp_desc":"Successful transaction",
"host_date":"170901","host_time":"135212","udf1":"111","udf2":"222","udf3":"333","udf4":"444","udf5":"555"}}
```

In case when order marked using jsonParams as balance inquiry then amount field will be present in response params field:

Balance response example:

```
{"expiration":"201512","approvalCode":"000000","Pan":"123456**6740", "Amount":1000,"cardholderName":"yrey
rgre","OrderStatus":4,"authCode":2, "OrderNumber":"132353464","depositAmount":0,"Ip":"127.0.0.1", "params":{"host_balance ":500",
"host_resp":"00","host_resp_desc":"Successful transaction", "host_date":"170901","host_time":"135212",
"udf1":"111","udf2":"222","udf3":"333","udf4":"444","udf5":"555"}}
```

Extended order state request

To inquire about state of a registered order, use the getOrderStatusExtended

request. Request parameters

Name	Type	Mandatory	Description
userName	AN..30	yes	Merchant's login received during registration.
password	AN..30	yes	Merchant's password received during registration.
orderId	ANS20	yes*	Order number in the payment gate. Unique within the gate.



orderNumber	AN..32	yes*	Number (identifier) of the order in the merchant's system.
language	A2	no	Language according to ISO 639-1. If not specified, the system uses the default language from the merchant's settings. Error messages returned will be written in this very language.

* It is necessary to specify whether orderId or orderNumber in the request. If the request contains both parameters, orderId is a priority. There are 3 sets of response parameters. These sets depend on "getOrderStatusExtended version", specified in merchant settings.

Response parameters for versions 01, 02,03:

The following parameters are returned regardless getOrderStatusExtended version.

Name	Type	Mandatory	Description
orderNumber	AN..10	yes	Number (identifier) of the order in the merchant's system.
orderStatus	N2	no	Order status in the payment gate. The value is selected from the variants listed below. Absent, if no matching order was found.
actionCode	N3	yes	Processing system authorization code.
actionCodeDescription	AN..512	yes	Action code description on the language passed in the parameter "Language".
errorCode	N3	no	Error code.
errorMessage	AN..512	no	Description of the error in the language sent in the Language parameter of the request.
amount	N..20	yes	Order amount in minimal currency units (e.g. cents).
currency	N3	no	Code of the payment currency, according to ISO 4217. If not specified, the default value is used.
date	ANS	yes	Date of order registration.
orderDescription	AN..512	no	Free-formed description of the order.
ip	N..	yes	IP address of the user who paid for the order.
cardAuthInfo (element has structure which includes the list of elements secureAuthInfo type and maskedPan, expiration, cardholderName and approvalCode attributes):			
pan	N..19	no	Masked number of the card that was used in payment. Specified only for paid orders.
expiration	N6	no	Card expiration date in the YYYYMM format. Specified only for paid orders.
cardholderName	A..64	no	Cardholder name. Specified only for paid orders.
approvalCode	AN6	no	IPS authorization code. The field has fixed length of six characters, it can contain both numbers and letters.
secureAuthInfo (element includes eci element and elements of threeDSInfo type, which is a list of cavv and xid):			
eci	N..4	no	Electronic Commerce Indicator. Set just after the payment order in the case of a permit.
cavv	ANS..200	no	Cardholder Authentication Verification Value. Set just after the payment order in the case of a permit
xid	ANS..80	no	Electronic Commerce Transaction Identifier.Set just after the payment order in the case of a permit.
BindingInfo (element consists of clientId and bindingId):			



clientId	AN..255	no	Client number (ID) in the shop system, transferred during the order registration. Presents only if a shop is allowed to create links.
bindingId	AN..255	no	The identifier of a link created during order payment or used for payment. Presents only if a shop is allowed to create links.

Response parameters for versions 02, 03:

If getOrderStatusExtended version is 02 or higher, the following parameters are also returned along with parameters described above.

Name	Type	Mandatory	Description
authDateTime	ANS	no	Authorisation date and time
authRefNum	AN..24	no	Reference number
terminalId	AN..10	no	Terminal Id

Response parameters for version 03:

If getOrderStatusExtended version is 03, the following parameters are also returned along with parameters described above.

Name	Type	Mandatory	Description
paymentAmountInfo element (consists of approvedAmount, depositedAmount, refundedAmount and paymentState parameters):			
approvedAmount	N..20	no	Amount hold on the client's card (only for two-phase payments)
depositedAmount	N..20	no	Amount confirmed for deposit
refundedAmount	N..20	no	Refund amount
paymentState	N2	no	Payment state
bankInfo element (consist of bankName, bankCountryCode and bankCountryName parameters):			
bankName	AN..200	no	Name of issuing bank
bankCountryCode	AN..4	no	Code of issuing bank country
bankCountryName	AN..160	no	Country of issuing bank in language specified in language parameter or in user language if language parameter is empty

The OrderStatus field may have the following statuses:

State number	Description
0	Order registered, but not paid off
1	Pre-authorisation amount was held (for two-phase payment)
2	The amount was deposited successfully
3	Authorization reversed
4	Transaction was refunded
5	Authorization through the issuer's ACS initiated.
6	Authorization declined

Error codes (values of the errorCode field)



Value	Description
0	No system error
1	Expected [orderId] или [orderNumber]
2	The order is declined because of an error in the payment credentials
5	Access denied
5	The user must change his password
6	Unregistered OrderId
7	System error

GET Request example:

```
https://server/application_context/rest/getOrderStatusExtended.do?
userName=login&password=password&orderId=123456789AZERTYUIOPL&language=ru
```

POST Request example:

```
orderId=285b2973-4d02-4980-a54e-57c4d0d2xxx9&language=ru
```

Response example:

```
{
  "attributes": [],
  "date": "1342007119386",
  "currency": "643",
  "amount": "100",
  "actionCode": "0",
  "orderNumber": "1212x31334z15",
  "orderDescription": "test",
  "orderStatus": "2",
  "ip": "217.12.97.50",
  "actionCodeDescription": "The payment processed successfully",
  "merchantOrderParams": [],
  "cardAuthInfo": {
    "expiration": "201512",
    "pan": "411111**1111",
    "approvalCode": "123456",
    "cardholderName": "dsd qdqd",
    "secureAuthInfo": {
      "eci": "5",
      "threeDSInfo": {
        "cavv": "AAABCPeAUBNCAHEgBQAAAAAAAAA=",
        "xid": "MDAwMDAwMDEzNDIwMDcxMTk3Njc="
      }
    }
  }
}
```

Payment statistics for a specified period

`getLastOrdersForMerchants.do` method allows you to obtain statistics on payments for a certain period. Request parameters:

Name	Type	Mandatory	Description
userName	AN..30	yes	Merchant's login received during registration..
password	AN..30	yes	Merchant's login received during registration..
language	A2	no	Language according to ISO 639-1. If not specified, the system uses the default language from the merchant's settings. Error messages returned will be written in this very language.
page	N	no	The result of the request processing is a report divided into pages (each page contains <code>size</code> records). The response will return a page at number specified in this parameter. Page numbering starts at 0. If <code>page</code> parameter is not specified in the request, the response will return page number 0.
size	N..3	yes	Number of records on one page (maximum value is 200).
from	ANS	yes	Date and time of the report period beginning. Format is YYYYMMDDTHHmss.
to	ANS	yes	Date and time of the report period end. Format is YYYYMMDDTHHmss.



transactionStates	A..9	yes	This block is designed to specify order states you need to be included into the report. Several values should be separated by comma. Possible values: CREATED, APPROVED, DEPOSITED, DECLINED, REVERSED, REFUNDED.
merchants	ANS	yes	List of merchant Logins. Only transactions of specified merchants will be included into the report. Several values should be separated by comma. Leave this field empty to receive transactions of all available merchants.

Response parameters:

Name	Type	Mandatory	Description
errorCode	N..2	yes	Code of the error that occurred in the course of the report processing. Description of possible codes is represented in the table below - "Possible codes for errors (the <code>errorCode</code> field)".
errorMessage	AN..512	no	Description of the error. Present only in case of error (<code>errorCode</code> is not 0).
orderStatuses			Block containing data on orders in the report. See the table below - "Parameters of <code>orderStatuses</code> tags".
totalCount	N	yes	Total account of orders in the report (on all pages).
page	N	yes	The number of the current page (it is equal to the <code>page</code> parameter, transferred in the request).
pageSize	N..3	yes	Maximum number of orders on one page of the report (it is equal to the <code>size</code> parameter, transferred in the request).

Parameters of `orderStatuses` tags:

Name	Type	Mandatory	Description
orderNumber	AN..10	yes	Order number (identifier) in the merchant's system
orderStatus	N..2	yes	State of the order in the payment gate. The value is selected from the list below. Possible values are represented in the table below - "The <code>orderStatus</code> field".
actionCode	N..3	yes	Processing system authorization code.
actionCodeDescription	AN..512	yes	Action code description in the language specified in the parameter <code>language</code> of the request.
amount	N..20	yes	Order amount in minimal currency units.
currency	N3	yes	Code of the payment's currency (according to ISO 4217). If not specified, assumed to be default currency.
date	ANS	yes	Date of order registration.
orderDescription	AN..512	no	Free-formed description of the order from the order registration request.
ip	AN..20	no	IP address of the payer.
errorCode	N..2	yes	Error code.
merchantOrderParams		no	Tag with attributes, used to pass merchant additional parameters. See the table below - "Parameters of <code>merchantOrderParams</code> tag".
attributes		yes	Order attributes in the payment system (order number). See the table below - "Parameters of <code>attributes</code> tag".
cardAuthInfo		no	Tag with payment attributes. See the table below - "Параметры блока <code>cardAuthInfo</code> ".
bindingInfo		no	Tag with information on binding used for payment. See the table below - "Parameters of <code>bindingInfo</code> tag".
authDateTime	ANS	no	Date and time of payment authorization.
terminalId	AN..10	no	Terminal ID.
authRefNum	AN..24	no	Reference number
paymentAmountInfo		no	Tag with information on approved amount, deposited amount and refund. See the table below - "Parameters of <code>paymentAmountInfo</code> tag".
bankInfo		no	Tag with information on Issuing Bank. See the table below - "Parameters of <code>bankInfo</code> tag".

Parameters of `merchantOrderParams` tag:

Name	Type	Mandatory	Description
------	------	-----------	-------------



name	AN..20	yes	Name of the merchant additional parameter.
value	AN..1024	yes	Value of the merchant additional parameter.

Parameters of `attributes` tag:

Name	Type	Mandatory	Description
name	A7	yes	Attribute name - "mdOrder".
value	ANS36	yes	Attribute value - order identifier in the payment system (unique in the system).

Parameters of `cardAuthInfo` tag:

Name	Type	Mandatory	Description
pan	N..19	no	Masked number of the card that was used for payment.
expiration	N6	no	Card expiration date in the YYYYMM format.
cardholderName	A..64	no	Cardholder name.
approvalCode	AN6	no	IPS authorization code. The field has fixed length of six characters, it can contain both numbers and letters.

Parameters of `bindingInfo` tag:

Name	Type	Mandatory	Description
clientId	AN..255	no	Client number (ID) in the merchant system.
bindingId	AN..255	no	Identifier of the binding used for payment.

Parameters of `paymentAmountInfo` tag:

Name	Type	Mandatory	Description
paymentState	N..9	no	Payment state.
approvedAmount	N..20	no	Amount confirmed for deposit.
depositedAmount	N..20	no	Deposited amount.
refundedAmount	N..20	no	Refund amount.

Parameters of `bankInfo` tag:

Name	Type	Mandatory	Description
bankName	AN..200	no	Name of the Issuing Bank .
bankCountryCode	AN..4	no	Code of the Issuing Bank country.
bankCountryName	AN..160	no	Country of the Issuing Bank in language specified in <code>language</code> parameter or in user language if <code>language</code> parameter is empty.

The `orderStatus` field may have the following statuses:

Value	Description
0	The order is registered, but it is not paid yet.
1	Pre-authorisation amount is held (for two-phase payment)
2	The amount was deposited successfully



3	Authorization reversed
4	Transaction was refunded
5	Authorization through the Issuer's ACS initiated.
6	Authorization declined

Possible codes for errors (the `errorCode` field):

Value	Description
0	No system error
5	Mandatory field is empty
5	Value of transactionStates parameter is invalid
5	Access denied
7	System error
10	Parameter <code>size</code> exceeds maximum value

GET request example:

```
https://server/application_context/rest/getLastOrdersForMerchants.do?userName=987&password=111111&language=en&page=0&size=100&fro
```

POST request example:

```
language=en&page=0&size=100&from=20141009160000&to=20141111000000&transactionStates=DEPOSITED,REVERSED&merchants=Seve
```

Response example:

```
{ "errorCode": 0,
  "orderStatuses": [
    { "errorCode": "0", "orderNumber": "58drs0Pes459Hdsddd0567a0", "orderStatus": 2, "actionCode": 0, "actionCodeDescription": "Request processed successfully", "amount": 250000, "currency": "643", "date": 1414485649233, "orderDescription": "Opisanie", "ip": "212.5.125.194", "merchantOrderParam": "CARD", "bankCountryCode": "RU", "bankCountryName": "Russian Federation" },
    { "errorCode": "0", "orderNumber": "57drs0Pes459Hdsddd0567a0", "orderStatus": 2, "actionCode": 0, "actionCodeDescription": "Request processed successfully", "amount": 250000, "currency": "643", "date": 1414485277286, "orderDescription": "Opisanie", "ip": "212.5.125.194", "merchantOrderParam": "CARD", "bankCountryCode": "RU", "bankCountryName": "Russian Federation" }
  ],
  "totalCount": 2, "page": 0, "pageSize": 100 }
```



Address data for connections

Employees of a merchant can access the console or make API requests using logins and passwords of a user associated with the merchant. URLs for access to REST methods:

Method name	URL
Order registration	<a href="https://<TBP>/rebpayment/rest/register.do">https://<TBP>/rebpayment/rest/register.do
Reversal request	<a href="https://<TBP>/rebpayment/rest/reverse.do">https://<TBP>/rebpayment/rest/reverse.do
Request for refund of an order	<a href="https://<TBP>/rebpayment/rest/refund.do">https://<TBP>/rebpayment/rest/refund.do
Confirm order request	<a href="https://<TBP>/rebpayment/rest/confirmOrder.do">https://<TBP>/rebpayment/rest/confirmOrder.do
Order state request	<a href="https://<TBP>/rebpayment/rest/getOrderStatus.do">https://<TBP>/rebpayment/rest/getOrderStatus.do
Payment statistics for a specified period	<a href="https://<TBP>/rebpayment/rest/getLastOrdersForMerchants.do">https://<TBP>/rebpayment/rest/getLastOrdersForMerchants.do

