RUSSIAN STANDARD BANK

SPECIFICATION FOR INTEGRATION AND OPERATION OF *RSB_ECOMM* PAYMENT PLATFORM

MOSCOW, RUSSIAN FEDERATION

Index

Introduction

Payment platform RSB_ECOMM is dedicated in order to integrate Merchants with the Bank system to process transactions conducted on the Merchant web-sites in terms of the Internet-acquiring Agreement.

This documentation is designed and suitable for technical specialists.
In this documentation you could find principal stages of connection and system settings.

1 Main functional schemes.

Following integration schemes and functions are available:

Payment page hosted by Bank.	Direct API integration to ECOMM payment platform, card info is collected on Bank-hosted page. PCI DSS is NOT required for Merchant. Link.
Payment page hosted by Merchant.	Direct API integration to ECOMM payment platform, Merchant collects card info on its own page. Merchant is required to be compliant with PCI DSS and to submit verifying documents to the Bank annually Link.
Sending Invoices from private account on Bank's Ecomm Portal website.	No API integration is required. Send Invoice function is available from private account interface on Bank's Ecomm Portal website. This function sends e-mail or SMS with link to payment page for specified payment. Such link can be used to complete payment in 7 calendar days (the link will expire after this amount of time). Merchant can request access to private account from Bank by email ecom@rsb.ru
Sending Invoices via REST API.	Integration of REST API to service of sending Invoices via e-mail/SMS REST API is available for download via external link (available only in Russian).
Online fiscalization - Sending data necessary for fiscal receipt.	RSB_ECOMM is integrated to fiscalization service provided by our partner – Chek Online (http://chekonline.ru). For more info about this service contact e-commerce@rsb.ru Link to technical details
Apple Pay on Merchant's website. Token is decrypted by Bank.	Button is placed on Merchant's website, though Apple Pay Token will be decrypted by Bank. For documentation about this option, contact ecom@rsb.ru
Samsung Pay on Bank's payment page.	Samsung Pay button is placed on Bank's payment page. Option can be enabled by request to ecom@rsb.ru
Apple Pay / Samsung Pay / Google Pay - Merchant decrypts the Token	For Merchants integrated to relevant Pay service and able to decrypt tokens. Link.

2 Setting up the service

2.1 Methods of integration to RSB_ECOMM

For Direct API integration to RSB_ECOMM several methods exist: using HTTP request by POST, or using premade Integrated Merchant Agent module on Java for Merchant's server side (module is supplied by request to ecom@rsb.ru)

2.1.1 HTTP request by POST method

In order to transfer a request from Merchant with all necessary parameters POST method is used, the authentication process is implemented by means of SSL certificates.
The .pem certificate, chain-ecomm-ca-root-ca root certificate and .key private key received from the Bank should be saved in a separate folder — this will provide the possibility to easily reinstall the certificate in case of failure/expiry of the validity period.
SSL installation process – connections are depended on the software used on the Merchant side, in which it is necessary to specify (physical) paths to certificates.
SSL certificate is used only when connecting to (Merchant Handler URL).

If it is necessary, Bank can provide the example when using CURL library on PHP.

SSL certificates are valid for 1 year from the date of issue.
Merchant is held responsible for controlling validity of SSL certificates.
Connections only with TLS 1.2 or better are supported.

Certificate placement example:

WEB-server domestic directory:
/home/www/
or
/www/

Directory with SSL certificates beyond WEB-server structure:
/home/ssl/
or
/ssl/

Directory with SSL certificates should NOT be placed inside the WEB-server directories structure. In this case private key will be available for download, for example:
/home/www/ssl/
or
/www/ssl/

2.1.2 Connecting to RSB_ECOMM payment platform using IMA module

Connection to RSB_ECOMM payment platform can be also established using premade Integrated Merchant Agent module on Java for Merchant's server side (module is supplied by request to ecom@rsb.ru)

After receiving from Bank certificate .pem (this certificate has limited lifetime), root certificates ecomm-ca.crt and root-ca.crt and private key .key should be added to truststore in JKS container.

SSL certificates are valid for 1 year from the date of issue.
Merchant is held responsible for controlling validity of SSL certificates.
Connections only with TLS 1.1 or 1.2 are supported.

Example of creating JKS container:

1. Files necessary:
   - root-ca.crt
   - ecomm-ca.crt
   - MerchantID.key
   - MerchantID.pem
2. Generating container in PKCS12 format, where private key and certificate will be stored:
   openssl pkcs12 -export -inkey MerchantID.key -in MerchantID.pem -out MerchantID.pkcs12 -name MerchantID
3. Generating JKS container and adding root certificate:
   /usr/java/jdk1.6.0_25/bin/keytool -import -v -noprompt -trustcacerts -alias CA -file ecomm-ca.crt -keystore MerchantID.jks -keypass MerchantID -storepass MerchantID
4. Adding intermediate certificate to JKS container:
   /usr/java/jdk1.6.0_25/bin/keytool -import -v -noprompt -trustcacerts -alias root-CA -file root-ca.crt -keystore MerchantID.jks -keypass MerchantID -storepass MerchantID
5. Adding PKCS12 container to JKS container:
   /usr/java/jdk1.6.0_25/bin/keytool -v -importkeystore -srcalias MerchantID -destalias MerchantID -srckeystore MerchantID.pkcs12 -srcstoretype PKCS12 -destkeystore MerchantID.jks -deststoretype JKS
6. Verifying JKS container:
   /usr/java/jdk1.6.0_25/bin/keytool -v -list -keystore MerchantID.jks

Call methods:

1. Calling Java archive "ecomm_merchant.jar" from Command line. Examples can be found in "example" catalogue of the module;
2. Directly calling service methods of class "lv.tietoenator.cs.ecomm.merchant.Merchant". Class Merchant should be named after configuration file at the moment of class creation. Filename will allow initiation of IMA, in case of error "ConfigurationException" will be returned.

Example:

```
Merchant merchant;
try
{
        merchant = new Merchant(propFile);
```

```
} catch (ConfigurationException e)
{
        System.err.println("error: " + e.getMessage());
        return;
}
String result = merchant.sendTransData(amount, currency, client_ip,
description);
```

Additional examples can also be found in "example" catalogue.

2.2 Preparing for work.

Merchant must prepare information according to the list below and send it to Bank via email – ecom@rsb.ru:

1. Certificate request, created according to instruction (available for download by link). Certificate is used in order to create a secured (protected) connection based on SSL protocol between site web-server and server of the Bank, provides protection of the transmitted information. Merchant is identified on the Bank side by information in SSL certificate.
2. Information regarding RETURN_URL, in order to redirect the Client back to the Merchant page (test and production RETURN_URL can be different). For redirecting the POST method is used, can be changed to GET.
3. Merchant server IP addresses from which the requests will be received (test and production IP addresses can be different).

When changing/using additional IP addresses, it is necessary to inform the Bank beforehand and send the actual IP addresses. Works on the Bank side will be organized dealing with changing/adding the relevant IP addresses.

When contacting the Bank, please include your Merchant ID (ID will be issued to the Merchant after registration of Internet-acquiring Agreement) and Merchant's Legal name in Subject field of the e-mail.

At first it is necessary to determine the Merchant's on-line store payment model.
Bank supports following models:

Purchase (SMS)	Single transaction, which provides the possibility to transfer money funds from Cardholder's account to yours. This means that payment authorization and its clearing are implemented in the frame of one operation.
Authorization/Capture (DMS)	It is required two operations — Authorization, followed by Capture. This enables creating a hold of the necessary money funds on Cardholder's account with the following Capture (for example if you implement capture in accordance with the results of additional checking, availability of goods, goods dispatch, etc.). Please pay attention that Capture is the basis for debiting the Client's account (financial transaction) however the Authorization means only hold of the relevant amount on the Client's account.

Recurring operations	Operation aimed on presenting the periodic services to Cardholder; the operation is implemented by Merchant which provides services to the Clients on a permanent (regular) basis; rendering of this service is based on the singed Agreement which gives Merchant a possibility to implement periodic financial operations impacted on Cardholder's bank account.

Description of the necessary commands for each payment model is specified below. All of them are sent to Merchant Handler URL. Please note requirements for redirecting Client to payment page – Client Handler URL.

Merchant Handler URL (test) – used to receive transaction_id
https://testsecurepay.rsb.ru:9443/ecomm2/MerchantHandler

ClientHandler URL (test) – used to redirect Client to payment page, where card info will be collected and 3-D Secure check initiated:
https://testsecurepay.rsb.ru/ecomm2/ClientHandler?trans_id=<transaction_id>

It is important to send Transaction ID variable within the redirection process.
This variable contains the identifier of the transaction to be paid (please pay attention that trans_id could contain such symbols as '+', '=' and '/', which should be changed to URL encoding before sending the identifier (for example change '=' to the '%3D' group) :

+	/	!	"	#	%	&	'	*	,	:	;	<	=	>	?
%2B	%2F	21%	22%	23%	25%	26%	27%	%2a	%2c	%3a	%3b	%3c	%3d	%3e	%3f

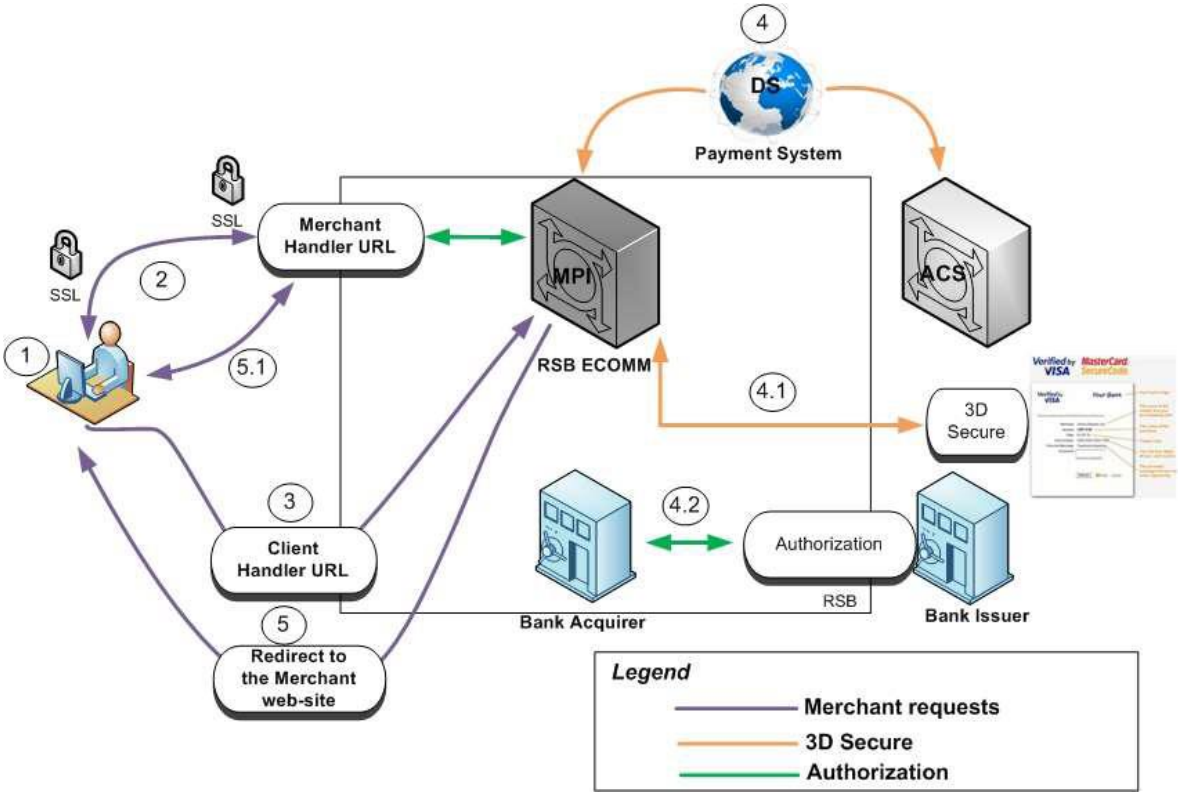
[	]	^	`	{	|	}	space
%5b	%5d	%5e	60%	%7b	%7c	%7d	20%

Attention:
If parameter trans_id was changed incorrectly, Client will receive error message: error: failed to get payment status

3 Integration schemes.

Below are described two main integration schemes to RSB_ECOMM payment platform:

3.1 Payment page hosted by Bank.



1. Client chooses a product and is ready to pay for it. When pressing «Checkout»/«Pay» button/link, the control is transferred to Merchant.
2. Merchant registers the transaction in RSB_ECOMM system with the use of the relevant request, which is forwarded by the encrypted channel to the address (Merchant Handler URL). In response you will get Transaction ID (trans ID).
3. Merchant transfers the Client to the RSB_ECOMM payment page with the specified Transaction ID in order to enter card data (Client Handler URL).
4. Depending on whether it is supported 3D Secure or not after entering card data it should be implemented the following requests (Merchant Plug-in Interface – Directory Server – Access Control Server):
    a. 3D Secure:

It is implemented a request on conducting the relevant authentication. The Client is transferred to the Issuing bank page (ACS) for additional 3D Secure authentication by entering a password (SMS or code from the list given out by the Issuing bank).

In case the password confirmation passed successfully the Issuing bank sends a response regarding the possibility of conducting this transaction to the Bank-acquirer.

In case if the Client's card doesn't participate in 3D Secure, the redirection to the Issuing bank page won't be implemented.

b. Payment authorization process:

Bank-acquirer sends an authorization request to Payment System. Payment System transfers the request to the Issuing bank. Issuing bank authorizes money funds and sends a response regarding successful authorization to Payment System. Payment System transfers a response regarding payment success to Bank-acquirer.

5. Client is transferred to the Merchant web-site (RETURN_URL) specifying the Transaction ID.

3.1.1 Main commands

Below are featured main commands for integration scheme with Bank-hosted payment page.
Before implementing please refer to section **2. Setting up the service**

3.1.1.1 Registration of SMS transactions

SMS payment model - debiting from the Client's card is implemented simultaneously and doesn't require additional confirmation. It means that payment Authorization and Capture are implemented in the frame of one transaction.

Variable	Field type	Number of symbols (max)	Description
-v	**M**	**1**	**Identifies a transaction registration request.**
amount	M	12	Transaction amount in integral units, last two symbols – kopecks.
currency	M	3	Transaction currency code (ISO 4217). In Russian Federation the only used code is 643.
client_ip_addr	M	15/39	Client's IP address (IPv4/IPv6)
description	O	125 (in Latin letters)	Payment description (encoding – UTF 8). Each Russian letter – counts as 2 (two) symbols. You should not use double quotes: ". We don't recommend to use such symbols as: &, %, №, @, $, * (for these symbols you should use values in accordance with the URL encoding table).
mrch_transaction_id	O	225 (in Latin letters)	Payment description (encoding – UTF 8). Each Russian letter counts as 2 (two) symbols. It is used in order to provide additional information regarding the order.
language	O	32 (in Latin letters, case sensitive)	Payment page language identifier. Mandatory in case it is necessary to use payment page in different languages.

server_version	O	4	It is used in order to return additional details, it should be specified „2.0”. In case the parameter is not specified, additional details won’t be returned.

Abbreviations:
M (Mandatory)
O (Optional)

Call to Bank, HTTP POST parameters:
command=v&amount=<amount>&currency=<currency>&client_ip_addr=<ip>&description=<desc>&mrch_transaction_id=<mrch_tx_id>&language=<language>&server_version=<2.0>&<property_name>=<property_value>

Call using IMA:
```
public String
startSMSTrans(String amount, String currency, String ip, String desc, String language, Properties properties)

//old calls for reverse compatibilty
public String
startSMSTrans(String amount, String currency, String ip, String desc, String language)

public String
sendTransData(String amount, String currency, String ip, String desc, String language)
```

Result:
TRANSACTION_ID: <trans_id>

Redirect Client to ClientHandlerURL for card info input and 3DS authentication:
https://testsecurepay.rsb.ru/ecomm2/ClientHandler?trans_id=<trans_id>

3.1.1.2 Status request.

After Client is redirected to RESULT_URL merchant must **request transaction status**.
If, for any reasons, Client didn’t return to RESULT_URL, Merchant can send Status request after 10 minutes from receiving transaction_id (transactions not completed in 10 minutes are failed by timeout).

Important:
Status requests for a single transaction should be sent only after any actions regarding transaction, i.e. capture, refund, etc. Excessive status requests do not return new information and create needless load for our servers.

Variable	Field type	Number of symbols (max)	Description
-c	**M**	**1**	**Identifies the transaction result request**
trans_id	M	28	Transaction identifier
client_ip_addr	M	15/39	Client’s IP address (IPv4/IPv6)
server_version	O	4	It is used in order to return additional details, it should be specified „2.0”

Abbreviations:
M (Mandatory)
O (Optional)

Call to Bank, HTTP POST parameters:

command=c&trans_id=<trans_id>&client_ip_addr=<ip>&server_version=<2.0>&<property_name>=<property_value>

Call using IMA:

public String
getTransResult(String trans_id, String ip)

public String
getTransResult(String trans_id, String ip, Properties properties)

Result:

RESULT: <result>
RESULT_PS: <result_ps>
RESULT_CODE: <result_code>
3DSECURE: <3dsecure>
RRN: <rrn>
APPROVAL_CODE: <app_code>
CARD_NUMBER: <pan>
MRCH_TRANSACTION_ID: <mrch_tx_id>

Example:

Call to Bank, HTTP POST parameters. Transaction registration:

command=v&amount=12300&currency=643&client_ip_addr=10.0.20.30&description=Order N123&
mrch_transaction_id=SMS transaction&language=ru&server_version=2.0

Result:

TRANSACTION_ID: rEsfhyIk8s9ypxkcS9fj/3C8FqA=

Redirect Client to payment page
https://testsecurepay.rsb.ru/ecomm2/ClientHandler?trans_id=rEsfhyIk8s9ypxkcS9fj%2F3C8FqA%3D

Verifying transaction result (on RETURN_URL page)

Call to Bank, HTTP POST parameters:

command=c&trans_id=rEsfhyIk8s9ypxkcS9fj/3C8FqA=&client_ip_addr=10.0.20.30&server_version=2.0

Result:

RESULT: OK
RESULT_PS: FINISHED
RESULT_CODE: 000
3DSECURE: AUTHENTICATED
RRN: 331711380059
APPROVAL_CODE: 327593
CARD_NUMBER: 5**********2372
MRCH_TRANSACTION_ID: SMS TRANSACTION

3.1.1.3 Status request using merchant's transaction ID.

In RSB ECOMM system all commands that create new transaction are supplied with new special parameter **merch_trans_id** in which you can send your own unique transaction ID. ID is saved and can be used afterwards for payment status request (-c) by sending this ID in **trans_id** parameter.

There are checks that this ID is:

1. UUID format.
2. URL-safe;
3. Unique and wasn't used before.

In current version is available for following commands: -v -a -j -i -u -n -q -m –k.

Example of generating your unique UUID for use in merch_trans_id:

```
import java.util.UUID;
public class UUIDGen {
public static void main(String args[]) {
UUID uuid = UUID.randomUUID();
System.out.println("Generated UUID: " + uuid.toString());
}
}
Generated UUID: b38c087e-99b4-4901-a495-f06ce71a5146
```

Example of payment request:

command=q&amount=100&currency=643&pan=4172500967168405&expiry=2205&cvc2=240&cardname=TEST&**merch_trans_id=54b06e70-9b4c-4cc7-80f0-9b1ee02b391e**&server_version=2.0&client_ip_addr=1.1.1.1

Result:

TRANSACTION_ID: 54b06e70-9b4c-4cc7-80f0-9b1ee02b391e
RESULT: OK
RESULT_PS: FINISHED
RESULT_CODE: 000
3DSECURE: FAILED
RRN: 104112404761

Status request:

command=c&**trans_id=54b06e70-9b4c-4cc7-80f0-9b1ee02b391e**&server_version=2.0&client_ip_addr=1.1.1.1

Result:

RESULT: OK
RESULT_PS: FINISHED
RESULT_CODE: 000
3DSECURE: FAILED

RRN: 104112404761

Refund request:

command=k&**trans_id=54b06e70-9b4c-4cc7-80f0-9b1ee02b391e**&**merch_trans_id=166988b0-d1a6-4d69-a233-006267aedb87**&mrch_transaction_id=refnd&amount=100&server_version=2.0&client_ip_addr=1.1.1.1

Result:

RESULT: OK
RESULT_CODE: 000
REFUND_TRANS_ID: **166988b0-d1a6-4d69-a233-006267aedb87**

Refund status request:

command=c&**trans_id=166988b0-d1a6-4d69-a233-006267aedb87**&server_version=2.0&client_ip_addr=1.1.1.1

Result:

RESULT: OK
RESULT_PS: FINISHED
RESULT_CODE: 000
RRN: 104112404803
APPROVAL_CODE: 169665
CARD_NUMBER: 4**********8405
MRCH_TRANSACTION_ID: refnd
AMOUNT: 100
TYPE: REFND
ISS_CCY: RUS

3.1.1.4 Registration of DMS transactions (authorization)

DMS payment model – debiting from the Client's card is implemented in two stages:

1. Authorization – money funds are held on the Client's card.

2. Capture of the transaction (financial operation).

For example, Capture upon the fact of availability of goods, goods dispatch, after conducting the additional checking, etc. Capture is the basis for debiting the Client's account however the Authorization means only hold of the relevant amount on the Client's account.

Variables description:

Variable	Field type	Number of symbols (max)	Description
-a	**M**	**1**	**Identifies a transaction registration request.**
client_ip_addr	M	15/39	Client's IP address (IPv4/IPv6)

amount	M	12	Transaction amount in integral units, last two symbols – kopecks.
currency	M	3	Transaction currency code (ISO 4217). In Russian Federation the only used code is 643.
description	O	125 (in Latin letters)	Payment description (encoding – UTF 8). Each Russian letter – counts as 2 (two) symbols. You should not use double quotes: ". We don't recommend to use such symbols as: &, %, №, @, $, * (for these symbols you should use values in accordance with the URL encoding table).
mrch_transaction_id	O	225 (in Latin letters)	Payment description (encoding – UTF 8). Each Russian letter counts as 2 (two) symbols. It is used in order to provide additional information regarding the order.
language	O	32 (in Latin letters, case sensitive)	Payment page language identifier. Mandatory in case it is necessary to use payment page in different languages.
server_version	O	4	It is used in order to return additional details, it should be specified „2.0"

Abbreviations:

M (Mandatory)
O (Optional)

Call to Bank, HTTP POST parameters:

command=a&amount=<amount>&currency=<currency>&client_ip_addr=<ip>&description=<desc>&mrch_transaction_id=<mrch_tx_id>&language=<language>&server_version=<2.0>&<property_name>=<property_value>

Call using IMA:

```
public String
startDMSAuth(String amount, String currency, String ip, String desc, String language, Properties properties)

// old calls for reverse compatibilty
public String
startDMSAuth(String amount, String currency, String ip, String desc, String language)

public String
startDMSAuth(String amount, String currency, String ip, String desc, String language)
```

Result:

TRANSACTION_ID: <trans_id>

Redirect Client to ClientHandlerURL for card info input and 3DS authentication:

https://testsecurepay.rsb.ru/ecomm2/ClientHandler?trans_id=<trans_id>

After redirection of Client to Return_URL, result of authorization can be requested by 3.1.1.2 Status request.

3.1.1.5 Capture/Finishing of DMS transaction

Variables description:

Variable	Field type	Number of symbols (max)	Description
-t	**M**	**1**	**Identifies a transaction capture request.**
trans_id	M	28	Transaction identifier of authorization to be captured
amount	M	12	Transaction amount in integral units, last two symbols – kopecks.
currency	M	3	Transaction currency code (ISO 4217). In Russian Federation the only used code is 643.
client_ip_addr	M	15/39	Client's IP address (IPv4/IPv6)
description	O	125 (in Latin letters)	Payment description (encoding – UTF 8). Each Russian letter – counts as 2 (two) symbols. You should not use double quotes: ". We don't recommend to use such symbols as: &, %, №, @, $, * (for these symbols you should use values in accordance with the URL encoding table).
language	O	32 (in Latin letters, case sensitive)	Payment page language identifier. Mandatory in case it is necessary to use payment page in different languages.

Abbreviations:

M (Mandatory)
O (Optional)

Call to Bank, HTTP POST parameters:

command=t&trans_id=<trans_id>&amount=<amount>&currency=<currency>&client_ip_addr=<ip>&description=<desc>&language=<language>&<property_name>=<property_value>

Call using IMA:

```
public String
makeDMSTrans(String auth_id, String amount, String currency, String ip, String desc, String language, Properties properties)

// old calls for reverse compatibilty
public String
makeDMSTrans(String auth_id, String amount, String currency, String ip, String desc, String language)

public String
makeDMSTrans(String auth_id, String amount, String currency, String ip, String desc)
```

Result:

RESULT: <result>
RESULT_CODE: <result_code>
RRN: <rrn>
APPROVAL_CODE: <app_code>
CARD_NUMBER <pan>

Example:

Capture/Finishing of DMS transaction

Call to Bank, HTTP POST parameters:

command=t&trans_id=rnSC7dvU73Wt9WTxE6TDXOeal/o&amount=12300&currency=643&client_ip_addr=10.0.20.30&description=order N123&language=ru

Result:

RESULT: OK
RESULT_CODE: 000
RRN: 331813509357
APPROVAL_CODE: 864745
CARD_NUMBER: 5**********2372

3.1.1.6 Reversal/Refund transaction

Transaction reversal/refund is initiated in case it is necessary to reimburse money funds to the Client.
Recommendations on using the commands in case if it is necessary to refund money funds to the Client:

Operation type	Conditions	
SMS	Before the operation date is closed	After the operation date was closed
	-r (reverse) Only full reversal	-k (refund) full / partial
DMS (authorization)	-r (reverse) only for a full amount	
DMS (capture)	-k (refund) full / partial	

Note:
When conducting «reversal» - money funds may be available for the Client within 1 day, when conducting «refund» - from 3 business days. Both periods may be increased on discretion of Issuer to a maximum of 30 days.

In case if your Client has a foreign currency account the reimbursement on the reversal/refund will be recalculated at the current rate of exchange as of the date of conducting the original transaction.

Variables description for reversal

Variable	Field type	Number of symbols (max)	Description
-r	M	1	Identifies the transaction reversal request.
trans_id	M	28	Transaction identifier.
suspected_fraud	O	3 (Latin letters)	Parameter – flag which shows that the reversal is conducted due to the suspected fraud. In such cases the value of this parameter should be set as “yes”.

Abbreviations:

M (Mandatory)
O (Optional)

Call to Bank, HTTP POST parameters:

command=r&trans_id=<trans_id>&<property_name>=<property_value>
command=r&trans_id=<trans_id>&suspected_fraud=yes&<property_name>=<property_value>

Call using IMA:

public String
reverse(String trans_id)

public String
reverse(String trans_id, Properties properties)

Result:

RESULT: <result>
RESULT_CODE: <result_code>

Example:

Call to Bank, HTTP POST parameters:

command=r&trans_id=HbetBxfr87TvObsevaRhAIdWWqM=&server_version=2.0

Result:

RESULT: OK
RESULT_CODE: 400

Merchant can also make a reversal from private account on Ecomm Portal website, access can be requested by email to ecom@rsb.ru

Only one reversal for full amount can be made.
SMS transaction can be reversed only before business day is closed, DMS transaction can be reversed only if it wasn't Captured.

Variables description for refund

Variable	Field type	Number of symbols (max)	Description
-k	**M**	**1**	**Identifies the transaction refund request**
trans_id	M	28	Transaction identifier.
amount	O	12	Refund amount in integral units, last two symbols – kopecks. If not specified, full amount of original transaction will be refunded.

Abbreviations:

M (Mandatory)
O (Optional)

Important:
Refund is an independent transaction which is opposite to the original transaction and has its own Transaction ID.

Refund can be made only for transaction in FINISHED status.
Total amount of several Refunds cannot exceed the amount of original transaction.

Call to Bank, HTTP POST parameters:

command=k&trans_id=<trans_id>&amount=<amount>&<property_name>=<property_value>

Call using IMA:

public String
refund(String trans_id)

public String
refund(String trans_id, Properties properties)

public String
refund(String trans_id, String amount, Properties properties)

Result:

RESULT: <result>
RESULT_CODE: <result_code>
REFUND_TRANS_ID: <refund_trans_id>

Example:

Call to Bank, HTTP POST parameters:

command=k&trans_id=M1pUcZowyEKaM5wrrlzyPbqDooU=&server_version=2.0

Result:

RESULT: OK
RESULT_CODE: 000
REFUND_TRANS_ID: 6ru/d9kr9u1vtqATyNdlP8KX4Kc=

Status of original transaction won't change after refund.

Merchant can also make a refund from private account on Ecomm Portal website, access can be requested by email to ecom@rsb.ru

3.1.1.7 Closing the business day

The procedure of Business day closing MUST be initiated at least once per twenty-four hours.

In accordance with the results of the conducted procedure the Bank processes the received operations; then reimburses money funds to the Merchant by transferring it to the Merchant account in the frame of the signed Agreement.

Описание переменных:

Variable	Field type	Number of	Description

		symbols (max)	
-b	**M**	**1**	**Identifies the Business day closing request.**

Call to Bank, HTTP POST parameters:

command=b&<property_name>=<property_value>

Call using IMA:

public String
closeDay()

public String
closeDay(Properties properties)

Result:

RESULT: <result>
RESULT_CODE: <result_code>
FLD_075: <fld_075>
FLD_076: <fld_076>
FLD_087: <fld_087>
FLD_088: <fld_088>

Example:

Call to Bank, HTTP POST parameters:

command=b&server_version=2.0

Result:

RESULT: OK
RESULT_CODE: 500
FLD_075: 12
FLD_076: 31
FLD_087: 3201
FLD_088: 10099

3.1.1.8 Regular (recurring) payments

Recurring payment – a payment which doesn't require re-entering card details.

The Client conducts a payment just once, agree with conditions of the regular debiting, next debiting will be implemented independently from the Client. Recurring payment is registered when conducting the first payment, herewith it is registered a recurring payment template in the database. There will be a number referred to the template which will be known to the Merchant. The recurring payment is initiated upon the request of the template number.

Recurring payments may be initiated by merchant or by consumer:

1. Consumer-Initiated – payment operation initiated by cardholder for payment using previously saved card credentials.

Transactions are identified as Consumer-Initiated by default.

2. Merchant-Initiated – payment operation initiated by merchant using previously stored card credentials. Payment may be either non-regular (i.e. automatic billing for services rendered to customer) or regular (i.e. payment for subscriptions). Before making such payment operations merchant must obtain specific agreement from customer to make payment operations in his stead.

To mark transaction as Merchant-Initiated in request to make a recurring transaction a detail **fld_126_MIRT** must be added, possible values are described below.

Согласие держателя карты:

В целях проведения операции Оплата товаров (работ, услуг), инициированной ТСП по ранее сохранным реквизитам карты, ТСП обязано получить согласие держателя карты, предусматривающее право ТСП использовать реквизиты для проведения операции Merchant-Initiated. Согласие должно содержать следующие сведения:

- название платежной системы;
- наименование и адрес местонахождения ТСП;
- реквизиты карты (маскированный номер и срок действия);
- цель дальнейшего использования реквизитов;
- срок, в течение которого предполагается использовать реквизиты, и, если это заранее известно, даты или периодичность проведения последующих операций Merchant-Initiated;
- наименование валюты последующих операций Merchant-Initiated;
- суммы последующих операций Merchant-Initiated или, если значения сумм на момент оформления Согласия не известны, порядок определения этих сумм и график их списаний (для оплаты по графику);
- размер комиссии ТСП по предстоящим операциям Merchant-Initiated или уведомление об отсутствии таковой;
- порядок отмен и возвратов по предстоящим операциям Merchant-Initiated;
- порядок отзыва Согласия.

Please note, this functionality may require additional Addendum to your Agreement. Please contact your key-account manager or e-commerce@rsb.ru to sign up for this functionalty.

3.1.1.8.1 Regular (recurring) SMS payment registration

Variables description:

Variable	Field type	Number of symbols (max)	Description
-z	**M**	**1**	**Regular (recurring) SMS payment registration request**
amount	M	12	Transaction amount in integral units, last two symbols – kopecks.
currency	M	3	Transaction currency code (ISO 4217). In Russian Federation the only used code is 643.
client_ip_addr	M	15/39	Client's IP address (IPv4/IPv6)
description	O	125 (in Latin letters)	Payment description (encoding – UTF 8). Each Russian letter – counts as 2 (two) symbols. You should not use double quotes: ". We don't recommend to use such symbols as: &, %, №, @, $, *

			(for these symbols you should use values in accordance with the URL encoding table).
mrch_transaction_id	O	225 (in Latin letters)	Payment description (encoding – UTF 8). Each Russian letter counts as 2 (two) symbols. It is used in order to provide additional information regarding the order.
ask_save_card_data	**O**	**4**	If not specified, template will be saved as normal If «true», payment page will show checkbox for Client to agree with saving his card info in template. If checkbox isn't checked payment will be finished, but template will be saved. To enable this variable contact ecom@rsb.ru.
language	O	32 (in Latin letters, case sensitive)	Payment page language identifier. Mandatory in case it is necessary to use payment page in different languages.
server_version	O	4	It is used in order to return additional details, it should be specified „2.0"
perspayee_gen=1	M	1	Used in order to generate a new regular (recurring) payment template.
biller_client_id	M	49	The recurring payment identifier chosen by the Merchant. Final value of the recurring payments identifier is formed with the use of Merchant ID and the value of the specified rec_pmnt_id identifier.
perspayee_expiry	O	4	The deadline validity period of the recurring payment in format MMYY. If not specified or set higher than expiry date of the card will be automatically set to expiry date of the card.

Abbreviations:

M (Mandatory)
O (Optional)

Call to Bank, HTTP POST parameters:

command=z&amount=<amount>&currency=<currency>&client_ip_addr=<ip>&description=<desc>&mrch_transaction_id=<mrch_tx_id>&language=<language>&biller_client_id=<recc_pmnt_id>&perspayee_expiry=<expiry>&perspayee_gen=1

Call using IMA:

public String
startSMSTransRP(String amount, String currency, String ip, String desc, String language, String recc_pmnt_id, String expiry, Properties properties)

Result:

TRANSACTION_ID: <trans_id>

After redirection of Client to Return_URL, result of authorization can be requested by 3.1.1.2 Status request. Result of status request will include additional details:

Result:

RESULT: <result>
RESULT_PS: <result_ps>
RESULT_CODE: <result_code>
3DSECURE: <3dsecure>
RRN: <rrn>
APPROVAL_CODE: <app_code>

CARD_NUMBER: <pan>
RECC_PMNT_ID: < biller_client_id >
RECC_PMNT_EXPIRY: < perspayee_expiry >
MRCH_TRANSACTION_ID: <mrch_tx_id>

Editing recurring payment template (i.e. Client changes his payment info):

Previously created payment template (without changing RECC_PMNT_ID) can be changed using special parameter perspayee_overwrite=1 (available for command=z).
Existing recurring payment template will be overwritten with new Client's card details. Without this parameter only new template can be created (or, if template with specified biller_client_id already exists, nothing will happen).

Example:

Regular (recurring) SMS payment registration

Call to Bank, HTTP POST parameters:

command=z&amount=12300&currency=643&client_ip_addr=10.0.20.30&description=Заказ N123 &mrch_transaction_id=recurrent&language=ru&biller_client_id=recurrent321&perspayee_expiry=0516&perspayee_gen=1&perspayee_overwrite=1&server_version=2.0

Result:

TRANSACTION_ID: Ipnu4iHy4UpGAB8aqc9Mwti58dE=

Redirect Client to ClientHandlerURL for card info input and 3DS authentication

https://testsecurepay.rsb.ru/ecomm2/ClientHandler?trans_id=Ipnu4iHy4UpGAB8aqc9Mwti58dE=

Verifying transaction result (on RETURN_URL page)

Call to Bank, HTTP POST parameters:

command=c&trans_id=Ipnu4iHy4UpGAB8aqc9Mwti58dE=&client_ip_addr=10.0.20.30&server_version=2.0

Result:

RESULT: OK
RESULT_PS: FINISHED
RESULT_CODE: 000
3DSECURE: AUTHENTICATED
RRN: 332518927710
APPROVAL_CODE: 418900
CARD_NUMBER: 5**********3361
RECC_PMNT_ID: recurrent321
RECC_PMNT_EXPIRY: 0516
MRCH_TRANSACTION_ID: RECURRENT

3.1.1.8.2 Repeated debiting of the recurring SMS payment

Important:

It is impossible to make a repeated debiting of SMS type for template created with DMS type and vice versa.

According to requirements of VISA Bank limits recurring payments in case of receiving following 4 declines in the span of 16 calendar days for single template:

100 Decline (general, no comments)
116 Decline, not sufficient funds
121 Decline, exceeds withdrawal amount limit
123 Decline, exceeds withdrawal frequency limit

All following attempts of repeated debiting of this template will result in decline with result code 103.
Repeated debiting of this recurring payment will be available again after 16 calendar days from the last declined attempt of repeated debiting.

Variables description:

Variable	Field type	Number of symbols (max)	Description
-e	**M**	**1**	**Request for the repeated debiting for the earlier registered regular (recurring) payment.**
-f	O	1	If, instead of **-e**, command **-f** is used, Client will need to be redirected to ClientHandler to enter CVV and make 3DS authentication. To use this command contact ecom@rsb.ru
amount	M	12	Transaction amount in integral units, last two symbols – kopecks.
currency	M	3	Transaction currency code (ISO 4217). In Russian Federation the only used code is 643.
client_ip_addr	M	15/39	Client's IP address (IPv4/IPv6)
description	O	125 (in Latin letters)	Payment description (encoding – UTF 8). Each Russian letter – counts as 2 (two) symbols. You should not use double quotes: ". We don't recommend to use such symbols as: &, %, №, @, $, * (for these symbols you should use values in accordance with the URL encoding table).
mrch_transaction_id	O	225 (in Latin letters)	Payment description (encoding – UTF 8). Each Russian letter counts as 2 (two) symbols. It is used in order to provide additional information regarding the order.
language	O	32 (in Latin letters, case sensitive)	Payment page language identifier. Mandatory in case it is necessary to use payment page in different languages.
biller_client_id	M	49	The recurring payment identifier chosen by the Merchant.
fld_126_MIRT	O	1	**Marks transaction that was initiated by merchant. If missing, the initiator of transaction is supposed to be a cardholder. Possible values:** **1 – "No show" charge or payment for uncancelled guaranteed booking;** **2 – Recurring payment by agreement with cardholder without schedule;** **3 – Recurring payment by agreement with cardholder by schedule.**

Abbreviations:

M (Mandatory)
O (Optional)

Call to Bank, HTTP POST parameters:

command=e&amount=<amount>&currency=<currency>&client_ip_addr=<ip>&description=<desc>&mrch_transaction_id=<mrch_tx_id>&language=<language>&biller_client_id=<recc_pmnt_id>

Call using IMA:

```
public String
makeRP(String recc_pmnt_id, String amount, String currency, String ip, String desc, Properties properties)

//public String
//startRP(String recc_pmnt_id, String amount, String currency, String ip, String desc, Properties properties)
//In this case Client will need to be redirected to ClientHandler to enter CVV and make 3DS authentication. To use this command contact ecom@rsb.ru
```

Result:

TRANSACTION_ID: <trans_id>
RESULT: <result>
RESULT_CODE:<result_code>
RRN:<rrn>
APPROVAL_CODE:<appr_code>

Example:

Repeated debiting for the earlier registered regular (recurring) payment.

Call to Bank, HTTP POST parameters:

command=e&amount=12300&currency=643&client_ip_addr=10.0.20.30&description=Заказ N123&mrch_transaction_id=recurrent&language=ru&biller_client_id=recurrent 321&server_version=2.0

Result:

TRANSACTION_ID: XHsmi/w7RmoirHpyPDGENBzBsMY=
RESULT: OK
RESULT_CODE: 000
RRN: 332518927712
APPROVAL_CODE: 598667

3.1.1.8.3 Regular (recurring) DMS payment registration (authorization)

DMS payment model – debiting from the Client's card is implemented in two stages:
1. Authorization – money funds are held on the Client's card.
2. Capture of the transaction (financial operation).

For example, Capture upon the fact of availability of goods, goods dispatch, after conducting the additional checking, etc. Capture is the basis for debiting the Client's account while the Authorization means only hold of the relevant amount on the Client's account.

Variables description:

Variable	Field type	Number of symbols (max)	Description
-d	**M**	**1**	**Regular (recurring) DMS payment registration request**
amount	M	12	Transaction amount in integral units, last two symbols – kopecks.
currency	M	3	Transaction currency code (ISO 4217). In Russian Federation the only used code is 643.
client_ip_addr	M	15/39	Client's IP address (IPv4/IPv6)
template_type=DMS	**M**	**3**	**template_type=DMS is mandatory parameter for creation of DMS-type template**
description	O	125 (in Latin letters)	Payment description (encoding – UTF 8). Each Russian letter – counts as 2 (two) symbols. You should not use double quotes: ". We don't recommend to use such symbols as: &, %, №, @, $, * (for these symbols you should use values in accordance with the URL encoding table).
mrch_transaction_id	O	225 (in Latin letters)	Payment description (encoding – UTF 8). Each Russian letter counts as 2 (two) symbols. It is used in order to provide additional information regarding the order.
ask_save_card_data=true	**O**	**4**	If not specified, template will be saved as normal If «true», payment page will show checkbox for Client to agree with saving his card info in template. If checkbox isn't checked payment will be finished, but template will be saved. To enable this variable contact ecom@rsb.ru.
language	O	32 (in Latin letters, case sensitive)	Payment page language identifier. Mandatory in case it is necessary to use payment page in different languages.
server_version	O	4	It is used in order to return additional details, it should be specified „2.0"
biller_client_id	M	49	The recurring payment identifier chosen by the Merchant. Final value of the recurring payments identifier is formed with the use of Merchant ID and the value of the specified rec_pmnt_id identifier.
perspayee_expiry	O	4	The deadline validity period of the recurring payment in format MMYY. If not specified or set higher than expiry date of the card will be automatically set to expiry date of the card.
perspayee_gen=1	M	1	Used in order to generate a new regular (recurring) payment template.

Abbreviations:

M (Mandatory)
O (Optional)

Call to Bank, HTTP POST parameters:

command=d&amount=<amount>&currency=<currency>&client_ip_addr=<ip>&description=<desc>&mrch_transaction_id=<mrch_tx_id>&language=<language>&**template_type=DMS**&biller_client_id=<recc_pmnt_id>&perspayee_expiry=<expiry>&perspayee_gen=1

Call using IMA:

public String
startSMSTransRP(String amount, String currency, String ip, String desc, String language, String recc_pmnt_id, String expiry, **template_type=DMS,** Properties properties)

Result:

TRANSACTION_ID: <trans_id>

After redirection of Client to Return_URL, result of authorization can be requested by 3.1.1.2 Status request.

Editing recurring payment template (i.e. Client changes his payment info):

Previously created payment template (without changing RECC_PMNT_ID) can be changed using special parameter perspayee_overwrite=1 (available for command=d).
Existing recurring payment template will be overwritten with new Client's card details. Without this parameter only new template can be created (or, if template with specified biller_client_id already exists, nothing will happen).

Example:

Regular (recurring) DMS payment registration

Call to Bank, HTTP POST parameters:

command=d&amount=12300&currency=643&client_ip_addr=10.0.20.30&description=Заказ N123 &mrch_transaction_id=recurrent&language=ru&**template_type=DMS**&biller_client_id=recurrent321&perspayee_expiry=0516&perspayee_gen=1&perspayee_overwrite=1&server_version=2.0

Result:

TRANSACTION_ID: /ZxP0bhyqtR8BURNjsEVnQh4kRI=

Redirect Client to ClientHandlerURL for card info input and 3DS authentication

https://testsecurepay.rsb.ru/ecomm2/ClientHandler?trans_id=%2FZxP0bhyqtR8BURNjsEVnQh4kRI=

Verifying transaction result (on RETURN_URL page)

Call to Bank, HTTP POST parameters:

command=c&trans_id=/ZxP0bhyqtR8BURNjsEVnQh4kRI=&client_ip_addr=10.0.20.30&server_version=2.0

Result:

RESULT: OK
RESULT_PS: ACTIVE
RESULT_CODE: 000
3DSECURE: AUTHENTICATED
RRN: 332518927710
APPROVAL_CODE: 418900
CARD_NUMBER: 5**********3361

RECC_PMNT_ID: recurrent321
RECC_PMNT_EXPIRY: 0516
MRCH_TRANSACTION_ID: RECURRENT

Capture/Finishing of DMS transaction

Call to Bank, HTTP POST parameters:

command=t&trans_id=/ZxP0bhyqtR8BURNjsEVnQh4kRI=&amount=12300&currency=643&client_ip_addr=10.0.20.30&description=Заказ N123&language=ru

Result:

RESULT: OK
RESULT_CODE: 000
RRN: 332518927710
APPROVAL_CODE: 418900
CARD_NUMBER: 5**********3361

IF Capturing/Finishing is not completed (command -t), DMS-type recurring payment template will not be saved.

3.1.1.8.4 Repeated debiting of the recurring DMS payment (authorization)

Important:

It is impossible to make a repeated debiting of SMS type for template created with DMS type and vice versa.

According to requirements of VISA Bank limits recurring payments in case of receiving following 4 declines in the span of 16 calendar days for single template:

100 Decline (general, no comments)
116 Decline, not sufficient funds
121 Decline, exceeds withdrawal amount limit
123 Decline, exceeds withdrawal frequency limit

All following attempts of repeated debiting of this template will result in decline with result code 103.
Repeated debiting of this recurring payment will be available again after 16 calendar days from the last declined attempt of repeated debiting.

Variables description:

Variable	Field type	Number of symbols (max)	Description
-f	**M**	**1**	**Request for the repeated debiting for the earlier registered regular (recurring) DMS payment.**
amount	M	12	Transaction amount in integral units, last two symbols – kopecks.
currency	M	3	Transaction currency code (ISO 4217). In Russian Federation the only used code is 643.

template_type	**M**	**3**	**template_type=DMS is mandatory parameter for using DMS-type template.**
savedcard=Y	**O**	**1**	savedcard=Y – if used, Client will need to be redirected to ClientHandler to enter CVV and make 3DS authentication. To use this parameter contact ecom@rsb.ru
client_ip_addr	M	15/39	Client's IP address (IPv4/IPv6)
description	O	125 (in Latin letters)	Payment description (encoding – UTF 8). Each Russian letter – counts as 2 (two) symbols. You should not use double quotes: ". We don't recommend to use such symbols as: &, %, №, @, $, * (for these symbols you should use values in accordance with the URL encoding table).
mrch_transaction_id	O	225 (in Latin letters)	Payment description (encoding – UTF 8). Each Russian letter counts as 2 (two) symbols. It is used in order to provide additional information regarding the order.
language	O	32 (in Latin letters, case sensitive)	Payment page language identifier. Mandatory in case it is necessary to use payment page in different languages.
biller_client_id	M	49	The recurring payment identifier chosen by the Merchant.
fld_126_MIRT	O	1	**Marks transaction that was initiated by merchant. If missing, the initiator of transaction is supposed to be a cardholder.** **Possible values:** **1 – "No show" charge or payment for uncancelled guaranteed booking;** **2 – Recurring payment by agreement with cardholder without schedule;** **3 – Recurring payment by agreement with cardholder by schedule.**

Abbreviations:

M (Mandatory)
O (Optional)

Call to Bank, HTTP POST parameters:

command=f&amount=<amount>&currency=<currency>&client_ip_addr=<ip>&description=<desc>&mrch_transaction_id=<mrch_tx_id>&language=<language>&**template_type=DMS**&biller_client_id=<recc_pmnt_id>

Call using IMA:

public String
startRP(String recc_pmnt_id, String amount, String currency, String ip, String desc, **template_type=DMS,** Properties properties)

Result:

TRANSACTION_ID: <trans_id>

Example:

Repeated debiting for the earlier registered regular (recurring) DMS payment (authorization)

Call to Bank, HTTP POST parameters

command=f&amount=12300&currency=643&client_ip_addr=10.0.20.30&description=Order N123&mrch_transaction_id=recurrent&language=ru&template_type=DMS&biller_client_id=recurrent321&server_version=2.0

Result:

TRANSACTION_ID: bKbv8UrPg+Y0RFLyv+b2WTnfIbg=

Verifying transaction result (on RETURN_URL page)

Call to Bank, HTTP POST parameters

command=c&trans_id=bKbv8UrPg+Y0RFLyv+b2WTnfIbg=&client_ip_addr=10.0.20.30&server_version=2.0

Result:

RESULT: OK
RESULT_PS: ACTIVE
RESULT_CODE: 000
3DSECURE: AUTHENTICATED
RRN: 332518927710
APPROVAL_CODE: 418900
CARD_NUMBER: 5**********3361
RECC_PMNT_ID: recurrent321
RECC_PMNT_EXPIRY: 0516
MRCH_TRANSACTION_ID: RECURRENT

Capture/Finishing of DMS transaction

Call to Bank, HTTP POST parameters

command=t&trans_id=bKbv8UrPg+Y0RFLyv+b2WTnfIbg=&amount=12300&currency=643&client_ip_addr=10.0.20.30

Result:

RESULT: OK
RESULT_CODE: 000
RRN: 332518927710
APPROVAL_CODE: 418900
CARD_NUMBER: 5**********3361

3.1.1.8.5 Deleting recurring payment template.

Variables description:

Variable	Field type	Number of symbols (max)	Description
-x	**M**	**1**	**Recurring payment template deletion request**
biller_client_id	M	49	The recurring payment identifier chosen by the Merchant.

Call to Bank, HTTP POST parameters:

command=x&biller_client_id=<rec_payment_id>(&<property_name>=<property_value>)*

Call using IMA:

public String
deleteRecurring(String recc_pmnt_id, Properties properties)

Result:

RESULT: <result>

3.1.2 Additional payment methods.

3.1.2.1 Apple Pay on Merchant's website. Token is decrypted by Bank.

Functionality of ECOMM for integration Apple Pay on the Web. Button is placed on Merchant's website, though Apple Pay Token will be decrypted by Bank.

For integration of Apple Pay to your website you may use Apple Pay JS API:
https://developer.apple.com/documentation/apple_pay_on_the_web/apple_pay_js_api
This demo page may help with integration of Apple Pay to website:
https://applepaydemo.apple.com/
Important information for working with Apple Pay in test environment:
https://developer.apple.com/apple-pay/sandbox-testing/

Before beginning:

- You must have Apple Developer Account developer.apple.com.
- Read following information from Apple website:
  - Apple Pay on the Web Developer Documentation
  - Apple Pay on the Web Human Interface Guidelines
  - Apple Pay on the Web Acceptable Use Guidelines
  - Configuring Your Developer Account for Apple Pay
- Inform the Bank (at ecom@rsb.ru) of merchant identifier assigned to you at developer.apple.com
- Inform the Bank of Merchant ID used in ECOMM (in format 929…).
- Inform the Bank of IP address of your server.
- Receive from Bank employee certificate requests for **Apple Pay payment processing certificate** and **Apple merchant identity certificate.**
- Generate certificates at developer.apple.com using requests received from Bank.
- Forward Apple Pay certificates to Bank.
- At developer.apple.com register and verify the domain Apple Pay.

Making calls to ECOMM:
Calls to ECOMM are made using REST API. API is based on JSON API specification. Detailed description of JSON API is available at jsonapi.org.
Authentication is made using standard OpenID. Call must contain http header Authorization of Bearer type that must be followed by JWT token.
Prior to the call JWT token must be received from actual authorization server (values necessary for token reception are supplied by Merchant's request at ecom@rsb.ru).

1. **Call to authentication server to receive JWT token.**
   URL:
   https://testsecurepay.rsb.ru:8143/auth/realms/RussianStandardBank/protocol/openid-connect/token

   Example:
   POST /auth/realms/RussianStandardBank/protocol/openid-connect/token HTTP/1.1

Host: testsecurepay.rsb.ru:8143
Accept: */*
Content-Length: 134
Content-Type: application/x-www-form-urlencoded

json: username=merchantid&password=**********&client_id=server3ds-app&
client_secret=**********&grant_type=password

Where values are:

*username=merchantid – your Merchant ID in format 929*******
password=********** - must be requested from ecom@rsb.ru,
client_id=server3ds-app
client_secret=********** - must be requested from ecom@rsb.ru,
grant_type=password

Result:

{"access_token":"***JWT token value***","token_type":"bearer","not-before-policy":0,"session_state":"***","scope":"***"}

2. **Apple Pay Payment Session call.**
Ecomm is capable of making server-to-server request which is answered by Apple Pay Server with **merchant session object,** object is used in method completeMerchantValidation.

For this you must make a POST request to ECOMM, which must contain header attribute "Content-Type" set to "application/json", body of request must contain JSON object.

URL for such call:
https://testsecurepay.rsb.ru:8445/applePay/validateMerchant/**merchantid** (merchantid = 929*******)

Example:

POST /applePay/validateMerchant/merchantid HTTP/1.1
Host: testsecurepay.rsb.ru:8445
Accept: */*
Authorization: bearer + ***JWT token value***
Origin: merchantdomain
Content-Length: ...
Content-Type: application/json; charset=utf-8

json: {"data":{"type":"applepay_session","attributes":{"url":"https://***applepayserverurl***/paymentservices/startSession"}}}

Where values are:

applepayserverurl *(sandbox or production) – determined automatically according to Apple ID that is used to make the payment.*

Result:

{"data":{"type":"applepay_session","attributes":{"session":"{"epochTimestamp":***,"expiresAt":***,"merchantSessionIdentifier":"

...","nonce":"...","merchantIdentifier":"***","domainName":"merchantdomain","displayName":"***","signature":"..."}"},"links":{}}}

3. **Call to execute Apple Pay payment.**
   To make Apple Pay payment you must make POST request, which must contain header attribute "Content-Type" set to "application/json", body of request must contain JSON object. Transaction type (SMS or DMS), depends on URL that request is sent to.

   In response Ecomm returns transaction result.

   https://testsecurepay.rsb.ru:8445/applePay/sms/execute/merchantid - URL for making SMS payment (single message).

   https://testsecurepay.rsb.ru:8445/applePay/dms/register/merchantid - URL for making DMS payment (dual message). Capture must be finished with command –t.

Example:

POST /applePay/sms/execute/merchantid HTTP/1.1
Host: testsecurepay.rsb.ru:8445
Accept: */*
Authorization: bearer + ***JWT token value***
Origin: merchantdomain
Content-Length: …
Content-Type: application/json; charset=utf-8

json:
{"data":{"type":"payment","attributes":{"amount":"1000","ccy_code":"643","client_ip":"10.230.2.18","apple_pay":{"token":{"paymentData":{"version":"EC_v1","data":"***","header":{"ephemeralPublicKey":"***","publicKeyHash":"***","transactionId":"***"}},"paymentMethod":{"displayName":"MasterCard 1471","network":"MasterCard","type":"debit"},"transactionIdentifier":"***"},"shipping_contact":{"emailAddress":"***","familyName":"","givenName":"***"}}}}}

Where values are:

amount *– transaction amount in kopecks, calculated by Merchant's server based on products or services that Client decided to purchase; Merchant's server should not send amount received from browser without checking.*
ccy_code *– currency code (ISO 4217);*
client_ip *– IP address of the Client;*
description *– misc. transaction details (optional);*
language *– Payment page language identifier (optional);*
apple_pay:
- ***token*** *- Apple Pay payment token, received from browser after authorization;*
- ***shipping_contact*** *– contact info for delivery, received from browser after authorization;*
- ***application_data*** *– additional data;*

Result:

{"data":{"id":"QXCnmPM75E9pYB3TbTV6UwemcPA=","type":"payment","attributes":{"apple_pay":{"transaction_id":"***"},"amount":"1000","ccy_code":"643","payment_type":"SMS","agent_i

d":"929*******","cardname":"TEST ","acceptor_id":"929*******","created_at":"2019-02-07T17:22:10","pan":"520424******0010","payment_method":"apple_pay","status":"FINISHED"},"links":{"self":"https://testsecurepay.rsb.ru/ecomm/v1.0/payments/QXCnmPM75E9pYB3TbTV6UwemcPA="}}}

3.1.2.2 Samsung Pay on Bank's payment page.

Button is located on Bank's payment page. This option can be enabled/disabled by sending request to ecom@rsb.ru

3.1.2.3 Google Pay™ on Bank's payment page.

Introduction

Google Pay is the fast, simple way that allows you to make card payments without entering card details for each payment. The card data is safely stored by Google. This payment method is available for all devices (mobile phones and computers), irrespective of the operating system and web browser.

Preconditions:

You have to adhere to Google Pay APIs Acceptable Use Policy.
You have to agree with Google Pay Terms of Service.

Description:

This payment method does not require additional integration. For creating payment use standard API calls per **3.1**. Google Pay™ button will be shown to customer on Russian Standard Bank's payment page. After customer authentication our server receives from Google Pay encrypted payment data, makes authorization and redirects customer back to your page for payment confirmation.

Integration:

To enable Google Pay payment method you may write us an email to ecom@rsb.ru or contact your key account manager.

Supported card networks:

Visa and **MasterCard**. All countries.

Bank's payment page in iFrame:

If you wish to open Bank's payment page on your website in iFrame, you should also add the **allowpaymentrequest** attribute to the **<iframe>** element.

Example of <iframe> for your website:

```
<iframe
    src="
https://testsecurepay.rsb.ru/ecomm2/ClientHandler?trans_id=<trans_id>"
    allowpaymentrequest
    title="Checkout page"
    width="400"
    height="300">
</iframe>
```

If you add the **sandbox** attribute with **allow-popups** value to the<iframe> element, it will allow the Google Pay session to open in new window.
Example:

```
<iframe
    src="
https://testsecurepay.rsb.ru/ecomm2/ClientHandler?trans_id=<trans_id>"
    allowpaymentrequest
    sandbox="allow-popups"
    title="Checkout page"
    width="400"
    height="300">
</iframe>
```

Authentication methods:

By default, all authentication methods are enabled (PAN_ONLY and CRYPTOGRAM_3DS), you may ask us at ecom@rsb.ru to restrict some of them:

- PAN_ONLY - This authentication method is associated with payment cards stored on file with the user's Google Account. Returned payment data includes personal account number (PAN) with the expiration month and the expiration year.
  For enabling/disabling of 3-D Secure please contact us at ecom@rsb.ru
- CRYPTOGRAM_3DS - This authentication method is associated with cards stored as Android device tokens. Returned payment data includes a 3-D Secure (3DS) cryptogram generated on the device.

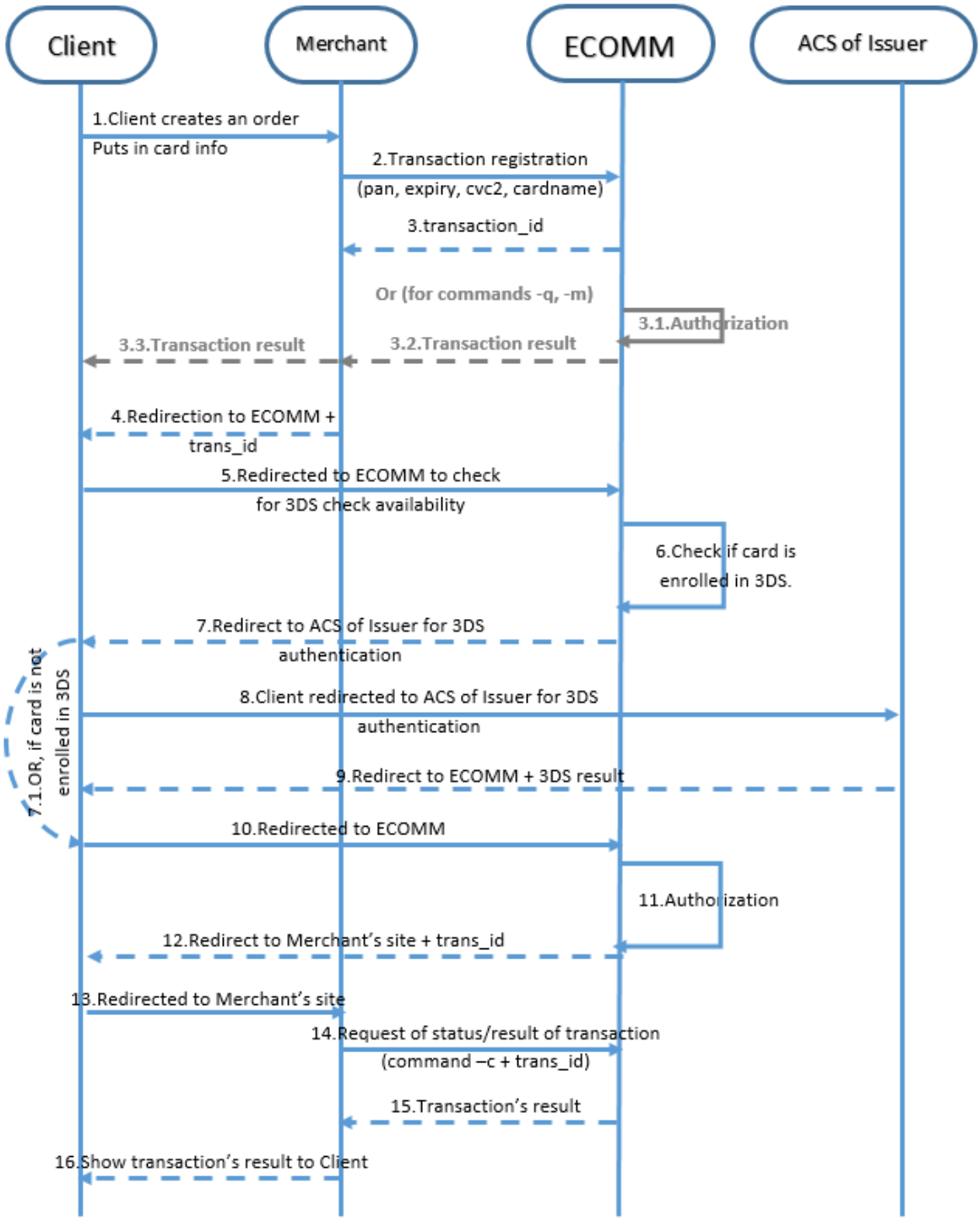
Billing address:

Billing address is not required for processing.

3.1.2.4 Google Pay™. Merchant side integration.

Google Pay method merchant side Web Integration and Android integration with decryption of Payment Data on Bank's side is **not** supported. For integration with decryption of Payment Data on the merchant's side please refer to 3.2.2.1

3.2 Payment page hosted by Merchant.



1. Client creates an order and fills his card's details on Merchant's website (or provider's payment page)
2. Merchant makes a transaction registration request by sending relevant command complete with card data to MerchantHandler URL.
3. ECOMM responds with **transaction_id** (same as trans_id)
   3.1. OR if Merchant uses command **-q** or **-m**, we skip directly to authorization. *3DS step is skipped, Merchant can guide client through 3DS flow themselves, using own MPI or*

directly connecting to our MPI, prior to step 2. In this case in step 2 Merchant must also send results of 3DS check alongside card data. Details in command descriptions below.

3.2. ECOMM responds with **transaction_id** and transaction result.
3.3. Merchant supplies transaction result to the Client. Skip all steps below.
4. Merchants redirects Client to ECOMM using URL ClientHandler+trans_id.
5. Client is sent to ECOMM using URL ClientHandler+trans_id.
6. ECOMM requests MPI (MPI > PS) to determine if client's card participate in 3DS.
7. If card is participating in 3DS, ECOMM redirects Client to Issuer's ACS URL.
7.1. If card is not participating in 3DS, skip to step 10.
8. Client is sent to Issuer's ACS URL for 3DS authentication process.
9. ACS redirects Client to ECOMM (MPI) with result of 3DS authentication.
10. Client is sent to ECOMM (MPI > ECOMM).
11. ECOMM sends an authorization (ECOMM > PS > Issuer).
12. ECOMM redirects Client to Return URL of the Merchant and sends along trans_id, either in POST or GET.
13. Client is sent to Merchant's website.
14. Merchant makes a status request, using received trans_id, to URL MerchantHandler.
15. ECOMM responds with transaction result.
16. Merchant supplies transaction result to the Client.

3.2.1 Main commands

Below are featured main commands for integration scheme with Merchants-hosted payment page. Before implementing please, refer to section **2. Setting up the service.**

3.2.1.1 Registration of SMS transactions

SMS payment model - debiting from the Client's card is implemented simultaneously and doesn't require additional confirmation. It means that payment Authorization and Capture are implemented in the frame of one transaction.

Variable	Field type	Number of symbols (max)	Description
-i	**M**	**1**	**Identifies a transaction registration request.**
amount	M	12	Transaction amount in integral units, last two symbols – kopecks.
currency	M	3	Transaction currency code (ISO 4217). In Russian Federation the only used code is 643.
client_ip_addr	M	15/39	Client's IP address (IPv4/IPv6)
pan	M	19	Card number.
expiry	M	4	Card expiry date (only in format **YYMM**).
cvc2	M/O	4	CVC2/CVV2 value. Usually is Mandatory.
cardname	M	-	Cardholder's name.
description	O	125 (in Latin letters)	Payment description (encoding – UTF 8). Each Russian letter – counts as 2 (two) symbols. You should not use double quotes: ". We don't recommend to use such symbols as: &, %, №, @, $, * (for these symbols you should use values in accordance with the URL encoding table).
mrch_transaction_id	O	225 (in Latin letters)	Payment description (encoding – UTF 8). Each Russian letter counts as 2 (two) symbols. It is used in order to provide additional information regarding the order.

language	O	32 (in Latin letters, case sensitive)	May be used to change the language of transaction confirmation email, if applicable.
server_version	O	4	It is used in order to return additional details, it should be specified „2.0”. In case the parameter is not specified, additional details won’t be returned.

3-D Secure authentication result may be supplied in following additional parameters:

Variable	Field type	Number of symbols (max)	Description
md_status	O	1	3-D Secure authentication status. Possible values: 0 – Not authenticated, 1 – Fully authenticated, 2 – Not enrolled, 4 – Attempted, 5 – U received, 6 – Error, 7 – Our Error. Also check our MPI API document for details.
cavv	O	28	3-D Secure authentication value for Visa or if cavvAlgorithm = 1, 2 or 4
aav	O	28	3-D Secure authentication value for MasterCard 3DS v1.0 or if cavvAlgorithm = 0 or 3
xid	O	28	Transaction ID from MPI or 3DS Server.
3ds_trans_status	O	1	3-D Secure 2.0 authentication status. Possible values: Y – Authenticated. N - not authenticated. A - not authenticated, but a proof of authentication attempt (Authentication Value) was generated. U - not authenticated because authentication could not be performed due to a technical or other problem. R - not authenticated because the Issuer is rejecting authentication and requesting that authorization not be attempted. Check full list of possible values here: https://www.emvco.com/terms-of-use/?u=wp-content/uploads/documents/EMVCo_3DS_Spec_210_1017_0318.pdf
3ds_protocol	O	-	3D Secure protocol version in format "3DS", for example "3DS2.1.0"
3ds_ds_id	O	36	Directory server transaction ID (3D Secure v2.0)

Abbreviations:

M (Mandatory)
O (Optional)

Call to Bank, HTTP POST parameters:

command=i&amount=<amount>&currency=<currency>&client_ip_addr=<ip>&description=<desc>&language=<language>&msg_type=<SMS>&cardname=<cardname>&pan=<pan>&expiry<expiry>&cvc2=<cvc2>(&<property_name>=<property_value>)*

Call using IMA:

public String

startCardSMSTrans(String amount, String currency, String ip, String desc, String cardname, String pan, String expiry, String cvc2, String language, Properties properties)

public String
startCardSMSTrans(String amount, String currency, String ip, String desc, String cardname, String pan, String expiry, String cvc2, String language)

public String
startCardSMSTrans(String amount, String currency, String ip, String desc, String cardname, String pan, String expiry, String cvc2)

Result:

TRANSACTION_ID: <trans_id>

Redirect Client to ClientHandlerURL for 3DS authentication:

https://testsecurepay.rsb.ru/ecomm2/ClientHandler?trans_id=<trans_id>

3.2.1.2 Status request.

After Client is redirected to RESULT_URL merchant must **request transaction status**.
If, for any reasons, Client didn't return to RESULT_URL, Merchant can send Status request after 10 minutes from receiving transaction_id (transactions not completed in 10 minutes are failed by timeout).

Important:
Status requests for a single transaction should be sent only after any actions regarding transaction, i.e. capture, refund, etc. Excessive status requests do not return new information and create needless load for our servers.

Variable	Field type	Number of symbols (max)	Description
-c	**M**	**1**	**Identifies the transaction result request**
trans_id	M	28	Transaction identifier
client_ip_addr	M	15/39	Client's IP address (IPv4/IPv6)
server_version	O	4	It is used in order to return additional details, it should be specified „2.0"

Abbreviations:

M (Mandatory)
O (Optional)

Call to Bank, HTTP POST parameters:

command=c&trans_id=<trans_id>&client_ip_addr=<ip>&server_version=<2.0>&<property_name>=<property_value>

Call using IMA:

public String
getTransResult(String trans_id, String ip)

public String
getTransResult(String trans_id, String ip, Properties properties)

Result:

RESULT: <result>
RESULT_PS: <result_ps>
RESULT_CODE: <result_code>
3DSECURE: <3dsecure>
RRN: <rrn>
APPROVAL_CODE: <app_code>
CARD_NUMBER: <pan>
MRCH_TRANSACTION_ID: <mrch_tx_id>

3.2.1.3 Status request using merchant's transaction ID.

In RSB ECOMM system all commands that create new transaction are supplied with new special parameter **merch_trans_id** in which you can send your own unique transaction ID. ID is saved and can be used afterwards for payment status request (-c) by sending this ID in **trans_id** parameter.

There are checks that this ID is:

1. UUID format.
2. URL-safe;
3. Unique and wasn't used before.

In current version is available for following commands: -v -a -j -i -u -n -q -m –k.

Example of generating your unique UUID for use in merch_trans_id:

```
import java.util.UUID;
public class UUIDGen {
public static void main(String args[]) {
UUID uuid = UUID.randomUUID();
System.out.println("Generated UUID: " + uuid.toString());
}
}
```

Generated UUID: b38c087e-99b4-4901-a495-f06ce71a5146

Example of payment request:

command=q&amount=100&currency=643&pan=4172500967168405&expiry=2205&cvc2=240&cardname=TEST&**merch_trans_id=54b06e70-9b4c-4cc7-80f0-9b1ee02b391e**&server_version=2.0&client_ip_addr=1.1.1.1

Result:

TRANSACTION_ID: 54b06e70-9b4c-4cc7-80f0-9b1ee02b391e
RESULT: OK
RESULT_PS: FINISHED
RESULT_CODE: 000
3DSECURE: FAILED
RRN: 104112404761

Status request:

command=c&**trans_id=54b06e70-9b4c-4cc7-80f0-9b1ee02b391e**&server_version=2.0&client_ip_addr=1.1.1.1

Result:

RESULT: OK
RESULT_PS: FINISHED
RESULT_CODE: 000
3DSECURE: FAILED
RRN: 104112404761

Refund request:

command=k&**trans_id=54b06e70-9b4c-4cc7-80f0-9b1ee02b391e**&**merch_trans_id=166988b0-d1a6-4d69-a233-006267aedb87**&mrch_transaction_id=refnd&amount=100&server_version=2.0&client_ip_addr=1.1.1.1

Result:

RESULT: OK
RESULT_CODE: 000
REFUND_TRANS_ID: **166988b0-d1a6-4d69-a233-006267aedb87**

Refund status request:

command=c&**trans_id=166988b0-d1a6-4d69-a233-006267aedb87**&server_version=2.0&client_ip_addr=1.1.1.1

Result:

RESULT: OK
RESULT_PS: FINISHED
RESULT_CODE: 000
RRN: 104112404803
APPROVAL_CODE: 169665
CARD_NUMBER: 4**********8405
MRCH_TRANSACTION_ID: refnd
AMOUNT: 100
TYPE: REFND
ISS_CCY: RUS

3.2.1.4 Registration of SMS transaction without redirect to ClientHandler and status request.

Unlike command for SMS transaction "-i", command "-q" allows to complete payment with a single request. Redirection to ClientHandler and status request "–c" are not required. **Attention! Network problems may result in successful payments not reflected on your side, no idempotency supported!**

Variable	Field type	Number of symbols (max)	Description
-q	**M**	**1**	**Identifies a transaction registration request.**
amount	M	12	Transaction amount in integral units, last two

			symbols – kopecks.
currency	M	3	Transaction currency code (ISO 4217). In Russian Federation the only used code is 643.
client_ip_addr	M	15/39	Client's IP address (IPv4/IPv6)
pan	M	19	Card number.
expiry	M	4	Card expiry date (only in format **YYMM**).
cvc2	M/O	4	CVC2/CVV2 value. Usually is Mandatory.
cardname	M	-	Cardholder's name.
description	O	125 (in Latin letters)	Payment description (encoding – UTF 8). Each Russian letter – counts as 2 (two) symbols. You should not use double quotes: ". We don't recommend to use such symbols as: &, %, №, @, $, * (for these symbols you should use values in accordance with the URL encoding table).
mrch_transaction_id	O	225 (in Latin letters)	Payment description (encoding – UTF 8). Each Russian letter counts as 2 (two) symbols. It is used in order to provide additional information regarding the order.
language	O	32 (in Latin letters, case sensitive)	May be used to change the language of transaction confirmation email, if applicable.
server_version	O	4	It is used in order to return additional details, it should be specified „2.0". In case the parameter is not specified, additional details won't be returned.

3-D Secure authentication result may be supplied in following additional parameters:

Variable	Field type	Number of symbols (max)	Description
md_status	O	1	3-D Secure authentication status. Possible values: 0 – Not authenticated, 1 – Fully authenticated, 2 – Not enrolled, 4 – Attempted, 5 – U received, 6 – Error, 7 – Our Error. Also check our MPI API document for details.
cavv	O	28	3-D Secure authentication value for Visa or if cavvAlgorithm = 1, 2 or 4
aav	O	28	3-D Secure authentication value for MasterCard 3DS v1.0 or if cavvAlgorithm = 0 or 3
xid	O	28	Transaction ID from MPI or 3DS Server.
3ds_trans_status	O	1	3-D Secure 2.0 authentication status. Possible values: Y – Authenticated. N - not authenticated. A - not authenticated, but a proof of authentication attempt (Authentication Value) was generated. U - not authenticated because authentication could not be performed due to a technical or other problem. R - not authenticated because the Issuer is rejecting authentication and requesting that authorization not be attempted. Check full list of possible values here: https://www.emvco.com/terms-of-use/?u=wp-content/uploads/documents/EMVCo_3DS_Spec_210_1017_0318.pdf

3ds_protocol	O	-	3D Secure protocol version in format "3DS", for example "3DS2.1.0"
3ds_ds_id	O	36	Directory server transaction ID (3D Secure v2.0)

Call to Bank, HTTP POST parameters:

command=q&amount=<amount>&currency=<currency>&client_ip_addr=<ip>&description=<desc>&language=<language>&msg_type=<SMS>&cardname=<cardname>&pan=<pan>&expiry<expiry>&cvc2=<cvc2>(&<property_name>=<property_value>)*

Call using IMA:

public String
startCardSMSTrans(String amount, String currency, String ip, String desc, String cardname, String pan, String expiry, String cvc2, String language, Properties properties)

public String
startCardSMSTrans(String amount, String currency, String ip, String desc, String cardname, String pan, String expiry, String cvc2, String language)

public String
startCardSMSTrans(String amount, String currency, String ip, String desc, String cardname, String pan, String expiry, String cvc2)

Result:

TRANSACTION_ID: <trans_id>
RESULT: <result>
RESULT_PS: <result_ps>
RESULT_CODE: <result_code>
3DSECURE: <3dsecure>
RRN: <rrn>
APPROVAL_CODE: <app_code>
CARD_NUMBER: <pan>
MRCH_TRANSACTION_ID: <mrch_tx_id>

3.2.1.5 Registration of DMS transactions (authorization)

DMS payment model – debiting from the Client's card is implemented in two stages:
1. Authorization – money funds are held on the Client's card.
2. Capture of the transaction (financial operation).
For example, Capture upon the fact of availability of goods, goods dispatch, after conducting the additional checking, etc. Capture is the basis for debiting the Client's account however the Authorization means only hold of the relevant amount on the Client's account.

Variable	Field type	Number of symbols (max)	Description
-j	**M**	**1**	**Identifies a transaction registration request.**
amount	M	12	Transaction amount in integral units, last two symbols – kopecks.
currency	M	3	Transaction currency code (ISO 4217). In Russian Federation the only used code is 643.
client_ip_addr	M	15/39	Client's IP address (IPv4/IPv6)
pan	M	19	Card number.
expiry	M	4	Card expiry date (only in format **YYMM**).

cvc2	M/O	4	CVC2/CVV2 value. Usually is Mandatory.
cardname	M	-	Cardholder's name.
description	O	125 (in Latin letters)	Payment description (encoding – UTF 8). Each Russian letter – counts as 2 (two) symbols. You should not use double quotes: ". We don't recommend to use such symbols as: &, %, №, @, $, * (for these symbols you should use values in accordance with the URL encoding table).
mrch_transaction_id	O	225 (in Latin letters)	Payment description (encoding – UTF 8). Each Russian letter counts as 2 (two) symbols. It is used in order to provide additional information regarding the order.
language	O	32 (in Latin letters, case sensitive)	May be used to change the language of transaction confirmation email, if applicable.
server_version	O	4	It is used in order to return additional details, it should be specified „2.0"

3-D Secure authentication result may be supplied in following additional parameters:

Variable	Field type	Number of symbols (max)	Description
md_status	O	1	3-D Secure authentication status. Possible values: 0 – Not authenticated, 1 – Fully authenticated, 2 – Not enrolled, 4 – Attempted, 5 – U received, 6 – Error, 7 – Our Error. Also check our MPI API document for details.
cavv	O	28	3-D Secure authentication value for Visa or if cavvAlgorithm = 1, 2 or 4
aav	O	28	3-D Secure authentication value for MasterCard 3DS v1.0 or if cavvAlgorithm = 0 or 3
xid	O	28	Transaction ID from MPI or 3DS Server.
3ds_trans_status	O	1	3-D Secure 2.0 authentication status. Possible values: Y – Authenticated. N - not authenticated. A - not authenticated, but a proof of authentication attempt (Authentication Value) was generated. U - not authenticated because authentication could not be performed due to a technical or other problem. R - not authenticated because the Issuer is rejecting authentication and requesting that authorization not be attempted. Check full list of possible values here: https://www.emvco.com/terms-of-use/?u=wp-content/uploads/documents/EMVCo_3DS_Spec_210_1017_0318.pdf
3ds_protocol	O	-	3D Secure protocol version in format "3DS", for example "3DS2.1.0"
3ds_ds_id	O	36	Directory server transaction ID (3D Secure v2.0)

Call to Bank, HTTP POST parameters:

command=j&amount=<amount>&currency=<currency>&client_ip_addr=<ip>&
email_client=<email>&description=<desc>&cardname=<cardname>&pan=<pan>&expiry=<expiry>&cvc2=<cvc2>&language=<language>&(&<property_name>=<property_value>)*

Call using IMA:

public String
startCardDMSAuth(String amount, String currency, String ip, String desc, String cardname, String pan, String expiry, String cvc2, String language, Properties properties)

public String
startCardDMSAuth(String amount, String currency, String ip, String desc, String cardname, String pan, String expiry, String cvc2)

public String
startCardDMSAuth(String amount, String currency, String ip, String desc, String cardname, String pan, String expiry, String cvc2, String language)

Result:

TRANSACTION_ID: <trans_id>

Redirect Client to ClientHandlerURL for card info input and 3DS authentication:

https://testsecurepay.rsb.ru/ecomm2/ClientHandler?trans_id=<trans_id>

After redirection of Client to Return_URL, result of authorization can be requested by 3.2.1.2 Status request.

3.2.1.6 Registration of DMS transaction (authorization) without redirect to ClientHandler and status request.

Unlike command for SMS transaction "-j", command "-m" allows to complete payment with a single request. Redirection to ClientHandler and status request "–c" are not required. **Attention! Network problems may result in successful payments not reflected on your side, no idempotency supported!**

Variable	Field type	Number of symbols (max)	Description
-m	**M**	**1**	**Identifies a transaction registration request.**
amount	M	12	Transaction amount in integral units, last two symbols – kopecks.
currency	M	3	Transaction currency code (ISO 4217). In Russian Federation the only used code is 643.
client_ip_addr	M	15/39	Client's IP address (IPv4/IPv6)
pan	M	19	Card number.
expiry	M	4	Card expiry date (only in format **YYMM**).
cvc2	M/O	4	CVC2/CVV2 value. Usually is Mandatory.
cardname	M	-	Cardholder's name.
description	O	125 (in Latin letters)	Payment description (encoding – UTF 8). Each Russian letter – counts as 2 (two) symbols. You should not use double quotes: ". We don't recommend to use such symbols as: &, %, №, @, $, * (for these symbols you should use values in accordance with the URL encoding table).

mrch_transaction_id	O	225 (in Latin letters)	Payment description (encoding – UTF 8). Each Russian letter counts as 2 (two) symbols. It is used in order to provide additional information regarding the order.
language	O	32 (in Latin letters, case sensitive)	May be used to change the language of transaction confirmation email, if applicable.
server_version	O	4	It is used in order to return additional details, it should be specified „2.0"

3-D Secure authentication result may be supplied in following additional parameters:

Variable	Field type	Number of symbols (max)	Description
md_status	O	1	3-D Secure authentication status. Possible values: 0 – Not authenticated, 1 – Fully authenticated, 2 – Not enrolled, 4 – Attempted, 5 – U received, 6 – Error, 7 – Our Error. Also check our MPI API document for details.
cavv	O	28	3-D Secure authentication value for Visa or if cavvAlgorithm = 1, 2 or 4
aav	O	28	3-D Secure authentication value for MasterCard 3DS v1.0 or if cavvAlgorithm = 0 or 3
xid	O	28	Transaction ID from MPI or 3DS Server.
3ds_trans_status	O	1	3-D Secure 2.0 authentication status. Possible values: Y – Authenticated. N - not authenticated. A - not authenticated, but a proof of authentication attempt (Authentication Value) was generated. U - not authenticated because authentication could not be performed due to a technical or other problem. R - not authenticated because the Issuer is rejecting authentication and requesting that authorization not be attempted. Check full list of possible values here: https://www.emvco.com/terms-of-use/?u=wp-content/uploads/documents/EMVCo_3DS_Spec_210_1017_0318.pdf
3ds_protocol	O	-	3D Secure protocol version in format "3DS", for example "3DS2.1.0"
3ds_ds_id	O	36	Directory server transaction ID (3D Secure v2.0)

Call to Bank, HTTP POST parameters:

command=m&amount=<amount>&currency=<currency>&client_ip_addr=<ip>&description=<desc>&language=<language>&msg_type=<SMS>&cardname=<cardname>&pan=<pan>&expiry<expiry>&cvc2=<cvc2>(&<property_name>=<property_value>)*

Call using IMA:

public String
makeCardDMSTrn(String amount, String currency, String ip, String desc, String cardname, String pan, String expiry, String cvc2, String language, Properties properties)

public String
makeCardDMSTrn(String amount, String currency, String ip, String desc, String cardname, String pan, String expiry, String cvc2, String language)

public String
makeCardDMSTrn(String amount, String currency, String ip, String desc, String cardname, String pan, String expiry, String cvc2, String language)

Result:

TRANSACTION_ID: <trans_id>
RESULT: <result>
RESULT_PS: <result_ps>
RESULT_CODE: <result_code>
3DSECURE: <3dsecure>
RRN: <rrn>
APPROVAL_CODE: <app_code>
CARD_NUMBER: <pan>
MRCH_TRANSACTION_ID: <mrch_tx_id>

3.2.1.7 Capture/Finishing of DMS transaction

Variable	Field type	Number of symbols (max)	Description
-t	**M**	**1**	**Identifies a transaction capture request.**
trans_id	M	28	Transaction identifier of authorization to be captured
amount	M	12	Transaction amount in integral units, last two symbols – kopecks.
currency	M	3	Transaction currency code (ISO 4217). In Russian Federation the only used code is 643.
client_ip_addr	M	15/39	Client's IP address (IPv4/IPv6)
description	O	125 (in Latin letters)	Payment description (encoding – UTF 8). Each Russian letter – counts as 2 (two) symbols. You should not use double quotes: ". We don't recommend to use such symbols as: &, %, №, @, $, * (for these symbols you should use values in accordance with the URL encoding table).
language	O	32 (in Latin letters, case sensitive)	May be used to change the language of transaction confirmation email, if applicable.

Abbreviations:

M (Mandatory)
O (Optional)

Call to Bank, HTTP POST parameters:

command=t&trans_id=<trans_id>&amount=<amount>&currency=<currency>&client_ip_addr=<ip>&description=<desc>&language=<language>&<property_name>=<property_value>

Call using IMA:

public String
makeDMSTrans(String auth_id, String amount, String currency, String ip, String desc, String language, Properties properties)

```
//old methods for reverse compatibilty
public String
makeDMSTrans(String auth_id, String amount, String currency, String ip, String desc, String language)

public String
makeDMSTrans(String auth_id, String amount, String currency, String ip, String desc)
```

Result:

RESULT: <result>
RESULT_CODE: <result_code>
RRN: <rrn>
APPROVAL_CODE: <app_code>
CARD_NUMBER <pan>

3.2.1.8 Reversal/Refund transaction

Transaction reversal/refund is initiated in case it is necessary to reimburse money funds to the Client.
Recommendations on using the commands in case if it is necessary to refund money funds to the Client:

Operation type	Conditions	
SMS	Before the operation date is closed	After the operation date was closed
	-r (reverse) Only full reversal	-k (refund) full / partial
DMS (authorization)	-r (reverse) only for a full amount	
DMS (capture)	-k (refund) full / partial	

Note:
When conducting «reversal» - money funds may be available for the Client within 1 day, when conducting «refund» - from 3 business days. Both periods may be increased on discretion of Issuer to a maximum of 30 days.

In case if your Client has a foreign currency account the reimbursement on the reversal/refund will be recalculated at the current rate of exchange as of the date of conducting the original transaction.

Variables description for reversal

Variable	Field type	Number of symbols (max)	Description
-r	M	1	**Identifies the transaction reversal request.**
trans_id	M	28	Transaction identifier.
suspected_fraud	O	3 (латиницей)	Parameter – flag, which shows that the reversal is conducted due to the suspected fraud. In such cases the value of this parameter should be set as “yes”.

Call to Bank, HTTP POST parameters:

command=r&trans_id=<trans_id>&<property_name>=<property_value>
command=r&trans_id=<trans_id>&suspected_fraud=yes&<property_name>=<property_value>

Call using IMA:

public String
reverse(String trans_id)

public String
reverse(String trans_id, Properties properties)

Result:

RESULT: <result>
RESULT_CODE: <result_code>

Merchant can also make a reversal from private account on Ecomm Portal website, access can be requested by email to ecom@rsb.ru

Only one reversal for full amount can be made.
SMS transaction can be reversed only before business day is closed, DMS transaction can be reversed only if it wasn't Captured.

Variables description for refund

Variable	Field type	Number of symbols (max)	Description
-k	**M**	**1**	**Identifies the transaction refund request**
trans_id	M	28	Transaction identifier.
amount	O	12	Refund amount in integral units, last two symbols – kopecks. If not specified, full amount of original transaction will be refunded.

Important:

Refund is an independent transaction which is opposite to the original transaction and has its own Transaction ID.
Refund can be made only for transaction in FINISHED status.
Total amount of several Refunds cannot exceed the amount of original transaction.

Call to Bank, HTTP POST parameters:

command=k&trans_id=<trans_id>&amount=<amount>&<property_name>=<property_value>

Call using IMA:

public String
refund(String trans_id)

public String
refund(String trans_id, Properties properties)

public String

refund(String trans_id, String amount, Properties properties)

Result:

RESULT: <result>
RESULT_CODE: <result_code>
REFUND_TRANS_ID: <refund_trans_id>

Status of original transaction won't change after refund.

Merchant can also make a refund from private account on Ecomm Portal website, access can be requested by email to ecom@rsb.ru

3.2.1.9 Closing the business day

The procedure of Business day closing MUST be initiated at least once per twenty-four hours.

In accordance with the results of the conducted procedure the Bank processes the received operations; then reimburses money funds to the Merchant by transferring it to the Merchant account in the frame of the signed Agreement.

Variable	Field type	Number of symbols (max)	Description
-b	**M**	**1**	**Identifies the Business day closing request.**

Call to Bank, HTTP POST parameters:

command=b&<property_name>=<property_value>

Call using IMA:

public String
closeDay()

public String
closeDay(Properties properties)

Result:

RESULT: <result>
RESULT_CODE: <result_code>
FLD_075: <fld_075>
FLD_076: <fld_076>
FLD_087: <fld_087>
FLD_088: <fld_088>

3.2.1.10 Regular (recurring) payments

Recurring payments are payments that do not require input of card details by cardholder. Both subscription and "binding card to account" scenarios are resolved via commands below.

Client makes payment as usual for first time and accepts terms of recurring payments. Subsequent payments do not need input of card details. Recurring payment is registered at first payment, by assigning template ID provided by merchant. By using this template ID you can initiate recurring payment.

Recurring payments may be initiated by merchant or by consumer:

1. Consumer-Initiated – payment operation initiated by cardholder for payment using previously saved card credentials.

Transactions are identified as Consumer-Initiated by default.

2. Merchant-Initiated – payment operation initiated by merchant using previously stored card credentials. Payment may be either non-regular (i.e. automatic billing for services rendered to customer) or regular (i.e. payment for subscriptions). Before making such payment operations merchant must obtain specific agreement from customer to make payment operations in his stead.

To mark transaction as Merchant-Initiated in request to make a recurring transaction a detail **fld_126_MIRT** must be added, possible values are described below.

Согласие держателя карты:

В целях проведения операции Оплата товаров (работ, услуг), инициированной ТСП по ранее сохранным реквизитам карты, ТСП обязано получить согласие держателя карты, предусматривающее право ТСП использовать реквизиты для проведения операции Merchant-Initiated. Согласие должно содержать следующие сведения:

- название платежной системы;
- наименование и адрес местонахождения ТСП;
- реквизиты карты (маскированный номер и срок действия);
- цель дальнейшего использования реквизитов;
- срок, в течение которого предполагается использовать реквизиты, и, если это заранее известно, даты или периодичность проведения последующих операций Merchant-Initiated;
- наименование валюты последующих операций Merchant-Initiated;
- суммы последующих операций Merchant-Initiated или, если значения сумм на момент оформления Согласия не известны, порядок определения этих сумм и график их списаний (для оплаты по графику);
- размер комиссии ТСП по предстоящим операциям Merchant-Initiated или уведомление об отсутствии таковой;
- порядок отмен и возвратов по предстоящим операциям Merchant-Initiated;
- порядок отзыва Согласия.

Please note that you Internet-acquiring agreement may not cover this functionality. Please contact your manager or e-commerce@rsb.ru to sign additional agreement enabling this functionality.

3.2.1.10.1 Creating recurrent template (SMS transaction)

Variable	Field type	Number of symbols (max)	Description
-u	**M**	**1**	**Запрос на регистрацию регулярного (рекуррентного) SMS платежа**
amount	M	12	Transaction amount in integral units, last two symbols – kopecks.
currency	M	3	Transaction currency code (ISO 4217). In Russian Federation the only used code is 643.
client_ip_addr	M	15/39	Client's IP address (IPv4/IPv6)
pan	M	19	Card number.
expiry	M	4	Card expiry date (only in format **YYMM**).
cvc2	M/O	4	CVC2/CVV2 value. Usually is Mandatory.
cardname	M	-	Cardholder's name.
description	O	125 (in Latin letters)	Payment description (encoding – UTF 8). Each Russian letter – counts as 2 (two) symbols. You should not use double quotes: ". We don't recommend to use such symbols as: &, %, №, @, $, * (for these symbols you should use values in accordance with the URL encoding table).
mrch_transaction_id	O	225 (in Latin letters)	Payment description (encoding – UTF 8). Each Russian letter counts as 2 (two) symbols. It is used in order to provide additional information regarding the order.
language	O	32 (in Latin letters, case sensitive)	May be used to change the language of transaction confirmation email, if applicable.
template_type	O	3	template_type=DMS. Use this detail to save template as DMS type with SMS payment. For recurring payments use DMS type command (-f) as per 3.2.1.9.5
server_version	O	4	It is used in order to return additional details, it should be specified „2.0". In case the parameter is not specified, additional details won't be returned.
biller_client_id	M	49	The recurring payment identifier chosen by the Merchant. Final value of the recurring payments identifier is formed with the use of Merchant ID and the value of the specified rec_pmnt_id identifier.
perspayee_expiry	O	4	The deadline validity period of the recurring payment in format MMYY. If not specified or set higher than expiry date of the card will be automatically set to expiry date of the card.
perspayee_gen=1	M	1	Used in order to generate a new regular (recurring) payment template.

3-D Secure authentication result may be supplied in following additional parameters:

Variable	Field type	Number of symbols (max)	Description
md_status	O	1	3-D Secure authentication status. Possible values: 0 – Not authenticated, 1 – Fully authenticated, 2 – Not enrolled, 4 – Attempted, 5 – U received, 6 – Error,

			7 – Our Error. Also check our MPI API document for details.
cavv	O	28	3-D Secure authentication value for Visa or if cavvAlgorithm = 1, 2 or 4
aav	O	28	3-D Secure authentication value for MasterCard 3DS v1.0 or if cavvAlgorithm = 0 or 3
xid	O	28	Transaction ID from MPI or 3DS Server.
3ds_trans_status	O	1	3-D Secure 2.0 authentication status. Possible values: Y – Authenticated. N - not authenticated. A - not authenticated, but a proof of authentication attempt (Authentication Value) was generated. U - not authenticated because authentication could not be performed due to a technical or other problem. R - not authenticated because the Issuer is rejecting authentication and requesting that authorization not be attempted. Check full list of possible values here: https://www.emvco.com/terms-of-use/?u=wp-content/uploads/documents/EMVCo_3DS_Spec_210_1017_0318.pdf
3ds_protocol	O	-	3D Secure protocol version in format "3DS", for example "3DS2.1.0"
3ds_ds_id	O	36	Directory server transaction ID (3D Secure v2.0)

Call to Bank, HTTP POST parameters:

command=u&amount=<amount>&currency=<currency>&client_ip_addr=<client_ip_addr>&description=<desc>&biller_client_id=<rec_payment_id>&perspayee_expiry=<rec_payment_expiry>&cardname=<cardname>&pan=<pan>&expiry=<expiry>&cvc2=<cvc2>&language=<language>&perspayee_gen=1(&<property_name>=<property_value>)*

Call using IMA:

public String
startCardSMSRecurringTrans(String amount, String currency, String ip, String desc, String rec_payment_id, String rec_payment_expiry, String cardname, String pan, String expiry, String cvc2)

public String
startCardSMSRecurringTrans(String amount, String currency, String ip, String desc, String rec_payment_id, String rec_payment_expiry, String cardname, String pan, String expiry, String cvc2, String language)

public String
startCardSMSRecurringTrans(String amount, String currency, String ip, String desc, String rec_payment_id, String rec_payment_expiry, String cardname, String pan, String expiry, String cvc2, String language, Properties properties)

Result:

TRANSACTION_ID: <trans_id>

3.2.1.10.2 Repeated debiting of the recurring SMS payment

Important:

It is impossible to make a repeated debiting of SMS type for template created with DMS type and vice versa.

According to requirements of VISA Bank limits recurring payments in case of receiving following 4 declines in the span of 16 calendar days for single template:
100 Decline (general, no comments)
116 Decline, not sufficient funds
121 Decline, exceeds withdrawal amount limit
123 Decline, exceeds withdrawal frequency limit
All following attempts of repeated debiting of this template will result in decline with result code 103.
Repeated debiting of this recurring payment will be available again after 16 calendar days from the last declined attempt of repeated debiting.

Variables description:

Variable	Field type	Number of symbols (max)	Description
-e	**M**	**1**	**Request for the repeated debiting for the earlier registered regular (recurring) payment.**
amount	M	12	Transaction amount in integral units, last two symbols – kopecks.
currency	M	3	Transaction currency code (ISO 4217). In Russian Federation the only used code is 643.
client_ip_addr	M	15/39	Client's IP address (IPv4/IPv6)
biller_client_id	M	49	The recurring payment identifier chosen by the Merchant. Final value of the recurring payments identifier is formed with the use of Merchant ID and the value of the specified rec_pmnt_id identifier.
description	O	125 (in Latin letters)	Payment description (encoding – UTF 8). Each Russian letter – counts as 2 (two) symbols. You should not use double quotes: ". We don't recommend to use such symbols as: &, %, №, @, $, * (for these symbols you should use values in accordance with the URL encoding table).
mrch_transaction_id	O	225 (in Latin letters)	Payment description (encoding – UTF 8). Each Russian letter counts as 2 (two) symbols. It is used in order to provide additional information regarding the order.
language	O	32 (in Latin letters, case sensitive)	May be used to change the language of transaction confirmation email, if applicable.
fld_126_MIRT	O	1	**Marks transaction that was initiated by merchant. If missing, the initiator of transaction is supposed to be a cardholder. Possible values:** **1 – "No show" charge or payment for uncancelled guaranteed booking;** **2 – Recurring payment by agreement with cardholder without schedule;** **3 – Recurring payment by agreement with cardholder by schedule.**

Call to Bank, HTTP POST parameters:

command=e&amount=<amount>&currency=<currency>&client_ip_addr=<ip>&description=<desc>&language=<language>&biller_client_id=<rec_payment_id>(&<property_name>=<property_value>)*

Call using IMA:

public String
makeRP(String recc_pmnt_id, String amount, String currency, String ip, String desc, Properties properties)

Result:

TRANSACTION_ID: <trans_id>
RESULT: <result>
RESULT_CODE:<result_code>
RRN:<rrn>
APPROVAL_CODE:<appr_code>

3.2.1.10.3 Regular (recurring) DMS payment registration (authorization)

Please note, that recurring payment template can be used for recurring payments only after Capture/Finishing of this transaction as per normal DMS flow.

Variable	Field type	Number of symbols (max)	Description
-n	**M**	**1**	**Regular (recurring) DMS payment registration request**
amount	M	12	Transaction amount in integral units, last two symbols – kopecks.
currency	M	3	Transaction currency code (ISO 4217). In Russian Federation the only used code is 643.
client_ip_addr	M	15/39	Client's IP address (IPv4/IPv6)
pan	M	19	Card number.
expiry	M	4	Card expiry date (only in format **YYMM**).
cvc2	M/O	4	CVC2/CVV2 value. Usually is Mandatory.
cardname	M	-	Cardholder's name.
template_type	M	**3**	template_type=DMS. Use this detail to save template as DMS type. For recurring payments use DMS type command (-f) as per 3.2.1.9.5
description	O	125 (in Latin letters)	Payment description (encoding – UTF 8). Each Russian letter – counts as 2 (two) symbols. You should not use double quotes: ". We don't recommend to use such symbols as: &, %, №, @, $, * (for these symbols you should use values in accordance with the URL encoding table).
mrch_transaction_id	O	225 (in Latin letters)	Payment description (encoding – UTF 8). Each Russian letter counts as 2 (two) symbols. It is used in order to provide additional information regarding the order.
language	O	32 (in Latin letters, case sensitive)	May be used to change the language of transaction confirmation email, if applicable.
server_version	O	4	It is used in order to return additional details, it should be specified „2.0". In case the parameter is not specified, additional details won't be returned.
biller_client_id	M	49	The recurring payment identifier chosen by the Merchant. Final value of the recurring payments identifier is formed with the use of Merchant ID and the value of the specified rec_pmnt_id identifier.
perspayee_expiry	M	4	The deadline validity period of the recurring payment in format MMYY. If not specified or set higher than expiry date of the card will be automatically set to expiry date of the card.
perspayee_gen=1	M	1	Used in order to generate a new regular (recurring) payment template.

3-D Secure authentication result may be supplied in following additional parameters:

Variable	Field type	Number of symbols (max)	Description
md_status	O	1	3-D Secure authentication status. Possible values: 0 – Not authenticated, 1 – Fully authenticated, 2 – Not enrolled, 4 – Attempted, 5 – U received, 6 – Error, 7 – Our Error. Also check our MPI API document for details.
cavv	O	28	3-D Secure authentication value for Visa or if cavvAlgorithm = 1, 2 or 4
aav	O	28	3-D Secure authentication value for MasterCard 3DS v1.0 or if cavvAlgorithm = 0 or 3
xid	O	28	Transaction ID from MPI or 3DS Server.
3ds_trans_status	O	1	3-D Secure 2.0 authentication status. Possible values: Y – Authenticated. N - not authenticated. A - not authenticated, but a proof of authentication attempt (Authentication Value) was generated. U - not authenticated because authentication could not be performed due to a technical or other problem. R - not authenticated because the Issuer is rejecting authentication and requesting that authorization not be attempted. Check full list of possible values here: https://www.emvco.com/terms-of-use/?u=wp-content/uploads/documents/EMVCo_3DS_Spec_210_1017_0318.pdf
3ds_protocol	O	-	3D Secure protocol version in format "3DS", for example "3DS2.1.0"
3ds_ds_id	O	36	Directory server transaction ID (3D Secure v2.0)

Call to Bank, HTTP POST parameters:

command=n&amount=<amount>&currency=<currency>&client_ip_addr=<client_ip_addr>&description=<desc>&biller_client_id=<rec_payment_id>&**template_type=DMS**&perspayee_expiry=<rec_payment_expiry>&cardname=<cardname>&pan=<pan>&expiry=<expiry>&cvc2=<cvc2>&language=<language>&perspayee_gen=1(&<property_name>=<property_value>)*

Call using IMA:

public String
startCardDMSRecurringAuth(String amount, String currency, String ip, String desc, String rec_payment_id, String rec_payment_expiry, String cardname, String pan, String expiry, String cvc2, **template_type=DMS**)

public String
startCardDMSRecurringAuth(String amount, String currency, String ip, String desc, String rec_payment_id, String rec_payment_expiry, String cardname, String pan, String expiry, String cvc2, String language, **template_type=DMS**)

public String
startCardDMSRecurringAuth(String amount, String currency, String ip, String desc, String rec_payment_id, String rec_payment_expiry, String cardname, String pan, String expiry, String cvc2, String language, **template_type=DMS** Properties properties)

Result:
TRANSACTION_ID: <trans_id>

Example:
Creating DMS-type recurring template with DMS transaction.

Call to Bank, HTTP POST parameters:
command=n&amount=2000&description=Order123456&language=&currency=643&biller_client_id=DMS_n_f_00003&template_type=DMS&perspayee_expiry=0620&pan=5100476090795048&cardname=TEST&expiry=1906&cvc2=099&server_version=2.0&client_ip_addr=10.35.30.16

Result:
TRANSACTION_ID: Z2+ce//TEkNmCbOfQWPPkaF9I3M=

Redirect Client to ClientHandlerURL for 3DS authentication:
https://testsecurepay.rsb.ru/ecomm2/ClientHandler?trans_id=Z2%2Bce%2F%2FTEkNmCbOfQWPPkaF9I3M%3D

Status request:
command=c&trans_id=Z2+ce//TEkNmCbOfQWPPkaF9I3M=&server_version=2.0&client_ip_addr=10.35.30.16

Result:
RESULT: OK
RESULT_PS: FINISHED
RESULT_CODE: 000

//Without capture –t, recurring template won't be saved.

DMS capture/finish:
command=t&trans_id= Z2+ce//TEkNmCbOfQWPPkaF9I3M=&amount=2000&currency=643&description=Order123456&server_version=2.0&client_ip_addr=10.35.30.16

Result:
RESULT: OK
RESULT_CODE: 000

3.2.1.10.4 Regular (recurring) DMS payment registration (authorization and capture/finish in single request)

Creating DMS-type recurring template is possible with single request, separate capture/finish request isn't required, by using command **–u** from 3.2.1.9.1 and sending an additional detail **template_type=DMS.** Subsequent recurring payments are made as DMS –f from 3.2.1.9.5

Example:
Regular (recurring) DMS payment registration (authorization and capture/finish in single request).

Call to Bank, HTTP POST parameters:

command=u&amount=100&description=PaymentDescription&currency=643&biller_client_id=REG_DMS&perspayee_gen=1&perspayee_expiry=0620&pan=220077******2311&expiry=****&cvc2=***&cardname=TEST&**template_type=DMS**&server_version=2.0&client_ip_addr=10.35.30.16

Result:

TRANSACTION_ID: iOgLxqZnCn3ZtWR6uEIpsR9fWxg=

Redirect Client to ClientHandlerURL for 3DS authentication:

https://testsecurepay.rsb.ru/ecomm2/ClientHandler?trans_id=iOgLxqZnCn3ZtWR6uEIpsR9fWxg%3D

Status request:

command=c&trans_id=iOgLxqZnCn3ZtWR6uEIpsR9fWxg=&server_version=2.0&client_ip_addr=10.35.30.16

Result:

RESULT: OK
RESULT_PS: FINISHED
RESULT_CODE: 000
3DSECURE: AUTHENTICATED
RRN: 912810173731

3.2.1.10.5 Repeated debiting of the recurring DMS payment (authorization)

Important:

It is impossible to make a repeated debiting of SMS type for template created with DMS type and vice versa.

According to requirements of VISA Bank limits recurring payments in case of receiving following 4 declines in the span of 16 calendar days for single template:

100 Decline (general, no comments)
116 Decline, not sufficient funds
121 Decline, exceeds withdrawal amount limit
123 Decline, exceeds withdrawal frequency limit

All following attempts of repeated debiting of this template will result in decline with result code 103.
Repeated debiting of this recurring payment will be available again after 16 calendar days from the last declined attempt of repeated debiting.

Variables description:

Variable	Field type	Number of symbols (max)	Description
-f	**M**	**1**	**Regular (recurring) DMS payment registration request**
amount	M	12	Transaction amount in integral units, last two symbols – kopecks.
currency	M	3	Transaction currency code (ISO 4217). In Russian Federation the only used code is 643.

template_type	**M**	**3**	**template_type=DMS is mandatory parameter for use of DMS-type template**
client_ip_addr	M	15/39	Client's IP address (IPv4/IPv6)
description	O	125 (in Latin letters)	Payment description (encoding – UTF 8). Each Russian letter – counts as 2 (two) symbols. You should not use double quotes: ". We don't recommend to use such symbols as: &, %, №, @, $, * (for these symbols you should use values in accordance with the URL encoding table).
mrch_transaction_id	O	225 (in Latin letters)	Payment description (encoding – UTF 8). Each Russian letter counts as 2 (two) symbols. It is used in order to provide additional information regarding the order.
language	M	32 (in Latin letters, case sensitive)	May be used to change the language of transaction confirmation email, if applicable.
biller_client_id	M	49	The recurring payment identifier chosen by the Merchant. Final value of the recurring payments identifier is formed with the use of Merchant ID and the value of the specified rec_pmnt_id identifier.
fld_126_MIRT	O	1	**Marks transaction that was initiated by merchant. If missing, the initiator of transaction is supposed to be a cardholder.** **Possible values:** **1 – "No show" charge or payment for uncancelled guaranteed booking;** **2 – Recurring payment by agreement with cardholder without schedule;** **3 – Recurring payment by agreement with cardholder by schedule.**

Call to Bank, HTTP POST parameters:

command=f&amount=<amount>&currency=<currency>&client_ip_addr=<ip>&description=<desc>&mrch_transaction_id=<mrch_tx_id>&language=<language>&**template_type=DMS**&biller_client_id=<recc_pmnt_id>

Call using IMA:

public String
startRP(String recc_pmnt_id, String amount, String currency, String ip, String desc, **template_type=DMS,** Properties properties)

Result:

TRANSACTION_ID: <trans_id>

Redirection to ClientHandler URL is not required here, make status request for this TRANSACTION_ID

Example:

Repeated debiting of the recurring DMS payment (authorization)

Call to Bank, HTTP POST parameters:

command=f&amount=2000&currency=643&biller_client_id=DMS_n_f_00003&template_type=DMS&server_version=2.0&client_ip_addr=10.35.30.16

Result:

TRANSACTION_ID: ZovmwYE+XSa2zgykYWcuvdvc2IA=

Redirection to ClientHandler URL is not required

Status request:

command=c&trans_id=ZovmwYE+XSa2zgykYWcuvdvc2IA=&server_version=2.0&client_ip_addr=10.35.30.16

Result:

RESULT: OK

RESULT_PS: ACTIVE

RESULT_CODE: 000

Result:

command=t&trans_id=ZovmwYE+XSa2zgykYWcuvdvc2IA=&amount=2000&currency=643&description=Order 123456&server_version=2.0&client_ip_addr=10.35.30.16

Result:

RESULT: OK

RESULT_CODE: 000

3.2.1.10.6 Deleting recurring payment template.

Описание переменных:

Variable	Field type	Number of symbols (max)	Description
-x	**M**	**1**	**Recurring payment template deletion request**
biller_client_id	M	49	The recurring payment identifier chosen by the Merchant.

Call to Bank, HTTP POST parameters:

command=x&biller_client_id=<rec_payment_id>(&<property_name>=<property_value>)*

Call using IMA:

```
public String
deleteRecurring(String recc_pmnt_id, Properties properties)
```

Result:

RESULT: <result>

3.2.2 Additional payment methods.

3.2.2.1 Apple Pay, Samsung Pay and Google Pay, M4M and VTS.

To use Apple Pay, Samsung Pay, Google Pay, M4M or VTS through our Ecomm gateway you should use commands -q/-m (Type SMS/DMS).
Request to Ecomm must include variables listed in the table below.
For example, the table also has Apple Pay variables listed in relevance to variables of Ecomm protocol.

If you need, we can also introduce you to employees of Apple, Samsung or Google, to receive help in integration of Apple Pay, Samsung Pay and Google Pay.

Tieto Ecomm	Apple Pay	Description
amount	transactionAmount	Transaction amount in integral units, last two symbols – kopecks.
currency	currencyCode	Currency Code (ISO 4217). For Russia - 643 only.
client_ip_addr		IP address of the cardholder, mandatory detail
cardname	cardholderName	Cardholder Name
pan	applicationPrimaryAccountNumber	DPAN.
expiry	applicationExpirationDate	Card expiration date (in YYMM format).
description	-	Misc. transaction details, optional (max 125 symbols)
language	-	Language of transaction identifier, optional (max 32 symbols)
sli	-	Security Level Indicator **mandatory for Mastercard:** 242 – for Apple Pay, G Pay, Samsung Pay transactions 246 – for tokens of MDES for Merchants (M4M) 212 – for 3DS authenticated transactions
eci	-	**VISA and MIR.** Electronic Commerce Indicator: “05” – successful 3DS authentication “06” – attempted 3DS authentication “07” – all other cases
fld_122_DPCR	onlinePaymentCryptogram	Cardholder authentication information for Visa/Mastercard/MIR
fld_126_TWID	-	**Mandatory for MIR:** “W” – Digital channel indicator, wallet-initiated e-commerce

3-D Secure authentication result may be supplied in following additional parameters:

Variable	Field type	Number of symbols (max)	Description
md_status	O	1	3-D Secure authentication status. Possible values:

			0 – Not authenticated, 1 – Fully authenticated, 2 – Not enrolled, 4 – Attempted, 5 – U received, 6 – Error, 7 – Our Error. Also check our MPI API document for details.
cavv	O	28	3-D Secure authentication value for Visa or if cavvAlgorithm = 1, 2 or 4
aav	O	28	3-D Secure authentication value for MasterCard 3DS v1.0 or if cavvAlgorithm = 0 or 3
xid	O	28	Transaction ID from MPI or 3DS Server.
3ds_trans_status	O	1	3-D Secure 2.0 authentication status. Possible values: Y – Authenticated. N - not authenticated. A - not authenticated, but a proof of authentication attempt (Authentication Value) was generated. U - not authenticated because authentication could not be performed due to a technical or other problem. R - not authenticated because the Issuer is rejecting authentication and requesting that authorization not be attempted. Check full list of possible values here: https://www.emvco.com/terms-of-use/?u=wp-content/uploads/documents/EMVCo_3DS_Spec_210_1017_0318.pdf
3ds_protocol	O	-	3D Secure protocol version in format "3DS", for example "3DS2.1.0"
3ds_ds_id	O	36	Directory server transaction ID (3D Secure v2.0)

xPay recurring payment registration, saving token.

To create recurrent payment or to save token you need to send additional fields in the request. In case of successful payment, merchant receives Transaction ID (COF_ORIGINAL_TID) of initial transaction (COF_ORIGINAL_TID: **********), that must be saved for subsequent transactions.

Variable	Field type	Number of symbols (max)	Description
mperspayee_gen	M	1	Value “1” defines initial transaction of Apple Pay Recurring
initiator	M	8	Value “merchant” defines correct initiator of the payment
fld_122_DPCR	M	28	Cryptogram. Authentication value for Visa/Mastercard customer
mperiodic	M	1	Values: “y” for regular payment, “n” for non-regular payment.
mperiod	M/O	1	For regular payments, period between payments is defined here. Possible values: “D”, “W”, “M”, “Q”, “Y”.
eci	M/O	2	Electronic Commerce Indicator, for Visa transactions only. Values: “05” – successful 3DS authentication, “06” – attempted 3DS authentication, “07” – for any other cases.
sli	M/O	3	Security Level Indicator, for Mastercard transactions only.

			Values: “242” – for xPay transactions. “246” – for MDES for Merchants (M4M) token transactions.
fld_126_TWID	M/O	1	**Mandatory for MIR:** “W” – Digital channel indicator, wallet-initiated e-commerce

3-D Secure authentication result may be supplied in following additional parameters:

Variable	Field type	Number of symbols (max)	Description
md_status	O	1	3-D Secure authentication status. Possible values: 0 – Not authenticated, 1 – Fully authenticated, 2 – Not enrolled, 4 – Attempted, 5 – U received, 6 – Error, 7 – Our Error. Also check our MPI API document for details.
cavv	O	28	3-D Secure authentication value for Visa or if cavvAlgorithm = 1, 2 or 4
aav	O	28	3-D Secure authentication value for MasterCard 3DS v1.0 or if cavvAlgorithm = 0 or 3
xid	O	28	Transaction ID from MPI or 3DS Server.
3ds_trans_status	O	1	3-D Secure 2.0 authentication status. Possible values: Y – Authenticated. N - not authenticated. A - not authenticated, but a proof of authentication attempt (Authentication Value) was generated. U - not authenticated because authentication could not be performed due to a technical or other problem. R - not authenticated because the Issuer is rejecting authentication and requesting that authorization not be attempted. Check full list of possible values here: https://www.emvco.com/terms-of-use/?u=wp-content/uploads/documents/EMVCo_3DS_Spec_210_1017_0318.pdf
3ds_protocol	O	-	3D Secure protocol version in format "3DS", for example "3DS2.1.0"
3ds_ds_id	O	36	Directory server transaction ID (3D Secure v2.0)

Call to Bank, HTTP POST parameters. Registering regular payment with Mastercard example:

command=m&amount=100&currency=643&pan=5**********1208&expiry=****&cardname=TEST&**mperspayee_gen=1&initiator=merchant&fld_122_DPCR=**********&mperiodic=y&mperiod=M&sli=242**&server_version=2.0&client_ip_addr=231.234.11.23

Response example:

TRANSACTION_ID: MfJYPROGPPtCIn3746Z9KOiziCQ=
RESULT: OK
RESULT_PS: ACTIVE
RESULT_CODE: 000
RRN: 033508161111
APPROVAL_CODE: 006131
CARD_NUMBER: 5**********1208
COF_ORIGINAL_TID: MCC013GHW1130 – identifier of initial transaction

xPay subsequent payment.

For subsequent payments with saved token merchant sends ID of original payment in the request – cof_original_tid=********** and other additional details according to the table below.

Variable	Field type	Number of symbols (max)	Description
mperspayee_id	M	49	ID of regular payment. Any value at your discretion, but variable must be present in the request.
initiator	M	8	Value “merchant” defines correct initiator of the payment
cof_original_tid	M	-	ID of original payment
mperiodic	M	1	Values: “y” for regular payment, “n” for non-regular payment.
mperiod	M/O	1	For regular payments, period between payments is defined here. Possible values: “D”, “W”, “M”, “Q”, “Y”.
eci	M/O	2	Electronic Commerce Indicator, for Visa transactions only. Values: “05” – successful 3DS authentication, “06” – attempted 3DS authentication, “07” – for any other cases.
sli	M/O	3	Security Level Indicator, for Mastercard transactions only. Values: “247” – for regular payment, if cryptogram was sent in original transaction.
fld_126_TWID	M/O	1	**Mandatory for MIR:** “W” – Digital channel indicator, wallet-initiated e-commerce
fld_126_MIRT	O	1	**Marks transaction that was initiated by merchant. If missing, the initiator of transaction is supposed to be a cardholder.** **Possible values:** **1 – “No show” charge or payment for uncancelled guaranteed booking;** **2 – Recurring payment by agreement with cardholder without schedule;** **3 – Recurring payment by agreement with cardholder by schedule.**

Call to Bank, HTTP POST parameters. Subsequent regular payment with Mastercard example:

command=m&amount=100&currency=643&pan=5**********1208&expiry=****&cardname=TEST&**mperspayee_id=random&initiator=merchant&cof_original_tid=MCC013GHW1130&mperiodic=y&mperiod=M&sli=247**&server_version=2.0&client_ip_addr=231.234.11.23

Response example:

TRANSACTION_ID: 5BGyk4Da66JI0jD79qDAUF6ezxI=
RESULT: OK
RESULT_PS: ACTIVE
RESULT_CODE: 000
RRN: 033508161121
APPROVAL_CODE: 006132
CARD_NUMBER: 5**********1208
COF_ORIGINAL_TID: MCC013GHW1130

4 Optional features.

4.1 Changing transaction expiration time limit.

Default time limit to complete payment for cardholder is 10 minutes. During this time client must both finish entering his card details and succeed in 3DS authentication check.

This time limit can be changed for each transaction by additional variable **ecomm_payment_timeout.** Time limit is set in seconds. Please do not exceed 20 minutes for perfmormance reasons

Example, Call to Bank, HTTP POST parameters:

command=a&amount=12300&currency=643&client_ip_addr=10.0.20.30&description=Заказ N123&language=ru&ecomm_payment_timeout=1200&server_version=2.0

4.2 Information about flight route (Airline itinerary).

Function Airline itinerary is used for airline tickets purchases to reflect identification details of passenger: ticket number, passenger's name, agency's name, and flight details (aiports' names, etc.) Before using Airline itinerary function, please contact us at ecom@rsb.ru so we can make necessary settings.

Details mentioned below are common for all steps of travel.
Detail «n_legs» is mandatory and must contain value between 1 and 4.

Variable	Field type	Format	Number of symbols (max)	Description
n_legs	M	n	1	Amount of travel steps
ticket_number	M	an	15	ID number of the ticket
carrier_name	O	an	19	Name of the carrier company
travel_agency_code	O	an	8	Travel agency code
travel_agency_name	O	an	25	Travel agency name
plan_nr	O	an	2	Number of plan
invoice_number	O	an	6	Number of invoice
airline_orig_ccy	O	n	3	Code of original currency
passenger_name	M	ans	29	Name of the passenger
customer_ref	M	ans	20	Client's reference
original_amnt	O	n	12	Amount of original invoice
ticket_issue_addr	O	ans	16	Address of ticket's issue
ticket_issue_date	O	n	8	Date of ticket's issue
total_fare	O	n	12	Total amount
total_fees	O	n	12	Total amount of fees
total_tax	O	n	12	Total amount of taxes
restricted_ticket_indicator	O	ans	1	Indicator of restricted ticket

Following details describe each travel step.

Names of these details must have following format: <detail_name> <travel_step_number>.
In example, details “departure_airport1”, “carrier_code1”, etc. describe first travel step, while “departure_airport2”, “carrier_code2”, etc. describe second travel step and so on.

Variable	Field type	Format	Number of symbols (max)	Description
departure_airportN	M	an	5	Departure airport
carrier_codeN	**M**	**an**	**2**	Code of the carrier company
fare_bassisN	O	an	15	Code of base payment plan
service_classN	**O**	**an**	**2**	Class of flight service
stop_over_codeN	O	an	1	Stop over code
destination_airportN	**M**	**an**	**5**	Arrival airport
departure_dateN	M	n	8	Date of departure
departure_taxN	**O**	**n**	**12**	Amount of departure tax
conjunct_ticketN	O	an	15	Number of conjunct ticket
exchange_ticketN	**O**	**an**	**15**	Number of exchange ticket
coupon_numberN	O	ans	1	Number of coupon
trip_numberN	**O**	**ans**	**5**	Number of trip
departure_timeN	O	n	4	Time of departure
arrival_timeN	**O**	**n**	**4**	Date of arrival
fare_amntN	O	n	12	Amount
fees_amntN	**O**	**n**	**12**	Amount of fees
tax_amntN	O	n	12	Amount of taxes
endorsements_restrN	**O**	**ans**	**20**	Additional restrictions

Abbreviations:

M – Mandatory
O – Optional
n – Numerical
an – alphanumerical
ans – alphanumerical with symbols.

Example:

DMS payment (authorization)

Call to Bank, HTTP POST parameters:

command=a&amount=123400&currency=643&client_ip_addr=10.0.20.30&description=Заказ N123&mrch_transaction_id=Airlineitinerary&language=ru&server_version=2

Result:

TRANSACTION_ID: TEQzJ0AWm8lNChYUhpXTlrbh6Vo=
MRCH_TRANSACTION_ID: AIRLINE ITINERARI

Redirect Client to ClientHandlerURL for card info input and 3DS authentication

https://testsecurepay.rsb.ru/ecomm2/ClientHandler?trans_id=TEQzJ0AWm8lNChYUhpXTlrbh6Vo=

Verifying transaction result (on RETURN_URL page)

Call to Bank, HTTP POST parameters:

command=c&trans_id=TEQzJ0AWm8lNChYUhpXTlrbh6Vo=&client_ip_addr=10.0.20.30&server_version=2.0

Result:

RESULT: OK
RESULT_PS: ACTIVE
RESULT_CODE: 000
3DSECURE: AUTHENTICATED
RRN: 330912581996
APPROVAL_CODE: 005704
CARD_NUMBER: 676196********0498
MRCH_TRANSACTION_ID: AIRLINE ITINENARI
AUTH_TIME: 20131105124227
TYPE: DMS

Capture/Finishing of DMS transaction

Call to Bank, HTTP POST parameters:

command=t&trans_id=TEQzJ0AWm8lNChYUhpXTlrbh6Vo=&amount=123400&currency=643&client_ip_addr=10.0.20.30&language=ru&n_legs=2&ticket_number=12345678901234&passenger_name=VASILII PUPKIN&customer_ref=No money no funny&departure_airport1=SVO&carrier_code1=UN&departure_date1=20140131&destination_airport1=LED&departure_airport2=LED&carrier_code2=SU&departure_date2=20140201&destination_airport2=HEM&server_version=2.0

Result:

RESULT: OK
RESULT_CODE: 000
RRN: 330912581996
APPROVAL_CODE: 005704
CARD_NUMBER: 676196********0498

4.3 Sending client's email / phone

All requests to create SMS payment or DMS authorization should, if possible, contain variables **email_client, phone_client** with relevant information about client's email address and phone number.

This information improves your risk statistics and decreases transaction declines for fraud prevention reasons.

4.4 Creating online fiscal receipt.

Function is supported only for online cash desks provided by our partners:
Чек Онлайн (http://chekonline.ru)
Full up-to-date list of supported online cash desks vendors can be requested at ecom@rsb.ru
We can also introduce you to any of our partners for exclusive prices and conditions.

Following details are added to transaction register request to form and send fiscal receipt:

- email - (email or phone number, where receipt will be sent, sending receipt in phone message may not be available for your cash desk, test environment supports only sending to email) I.e. email=ecom@rsb.ru
- **G**roup - (ID of the merchant in online cash desks vendor's system, please note, that this variable's name starts with uppercase G, it is sensitive) I.e. Group=testgroup
- basket – (details of purchase, must have following format)
  - Lines :
    - "Qty": 1000 – product quantity. Quantity is indicated in millesimal amount, for example, 2,5 kilograms should be indicated as 2500 or singular product as 1000.
    - "Price": 1000 – Price **of single instance** of product, must be indicated in kopecks.
    - "PayAttribute": 4 – Type of clearing-off method, refer to the table 1 below for possible values.
    - "TaxId": 4 – Id of taxation type (1 – 6), refer to the table 2 below for possible values.
    - "Description": "Milk" – Name of the product. Cannot be null.

  Example:
  basket={"Lines":[{"Qty": 1000, "Price": 12000, "PayAttribute": 4, "TaxId": 4, "Description": "Milk"},{"Qty": 1500, "Price": 9500, "PayAttribute": 1, "TaxId": 1, "Description": "Apples"}]}
  In this example total amount is 262.50 roubles (120 roubles x 1 pack + 95 roubles x 1,5kg), in **amount** variable must be sent corresponding value: amount=26250. If amounts will not match, receipt won't be formed and error will be added to status request response RECEIPT: 77 (all error codes can be found in online cash desks vendor's documentation).
  Also check all parentheses, commas, cases, etc. in basket, as they are highly sensitive.
  If receipt was formed correctly, it will be added in RECEIPT line in status request response.

Example, Call to Bank, HTTP POST parameters:

command=v&amount=26250&currency=643&description=Заказ 54650&basket={"Lines":[{"Qty": 1000, "Price": 12000, "PayAttribute": 4, "TaxId": 4, "Description": "Milk"},{"Qty": 1500, "Price": 9500, "PayAttribute": 1, "TaxId": 1, "Description": "Apples"}]}&Group=testgroup&email=ecom@rsb.ru&client_ip_addr=10.35.30.16

In HTTP POST request, value in **basket** must be in URL Encoded format.
After successful payment Ecomm will automatically send details to online cash desk.

Result:

RESULT: OK
RESULT_CODE: 000
RRN: 816015093702
APPROVAL_CODE: 898429
CARD_NUMBER: 4**********5207
AMOUNT: 26250
RECEIPT: full receipt text or error code

If for any reason the payment was completed successfully, but receipt wasn't formed, you can create receipt manually on online cash desks vendor's website.

Reversal and refund.

In cases of reversal and refund corresponding receipts will be formed automatically if original payment ended with correct receipt formed.

You only need to supply basket in case of partial refund.

Example, Call to Bank, HTTP POST parameters:

command=k&trans_id=c95drFkfman3S0VkMtr2bthpyaw%3D&description=test&amount=12000&email=avivanov@rsb.ru&basket={"Lines":[{"Qty": 1000, "Price": 12000, "PayAttribute": 4, "TaxId": 4, "Description": "Milk"}]}&Group=testgroup&client_ip_addr=10.35.30.16

In response to refund and reversal commands the RECEIPT line will not be sent.

Result:

RESULT: OK
RESULT_CODE: 000
REFUND_TRANS_ID: 4zaRjoqmx0u+6OUaxKEkALyfC2E=

Following tables were not translated to keep the legal wording correct. Please consult with your legal department which values should be chosen for your business in accordance to Russian tax law.

Таблица "PayAttribute" - Признак способа расчёта:

Признак	Перечень оснований для присвоения реквизиту «признак способа расчёта» соответствующего значения реквизита
0	Для индивидуальных предпринимателей, являющихся налогоплательщиками, применяющими патентную систему налогообложения и упрощённую систему налогообложения, а также индивидуальных предпринимателей, применяющих систему налогообложения для сельскохозяйственных товаропроизводителей, систему налогообложения в виде единого налога на вменённый доход для отдельных видов деятельности при осуществлении видов предпринимательской деятельности, установленных пунктом 2 статьи 346.26 Налогового кодекса Российской Федерации, за исключением индивидуальных предпринимателей, осуществляющих торговлю подакцизными товарами, требование об обязательном включении в состав кассового чека и БСО реквизита применяется с 1 февраля 2021 года. Поле PayAttribute можно опустить.
1	Полная предварительная оплата до момента передачи предмета расчёта
2	Частичная предварительная оплата до момента передачи предмета расчёта
3	Аванс
4	Полная оплата, в том числе с учётом аванса (предварительной оплаты) в момент передачи предмета расчёта
5	Частичная оплата предмета расчёта в момент его передачи с последующей оплатой в кредит
6	Передача предмета расчёта без его оплаты в момент его передачи с последующей оплатой в кредит
7	Оплата предмета расчёта после его передачи с оплатой в кредит (оплата кредита). Этот признак должен быть единственным в документе и документ с этим признаком может содержать только одну строку.

Таблица "TaxId" - Код налога:

Код	Значение по умолчанию

1	НДС 20%
2	НДС 10%
3	НДС 0%
4	Без налога
5	Ставка 20/120
6	Ставка 10/110

5 Explanation of values in Bank's responses to your commands.

RESULT	Transaction result status
OK	Transaction was completed successfully
FAILED	Unsuccessful transaction
CREATED	Transaction has been registered in the system
PENDING	Transaction execution is currently in progress
DECLINED	Transaction was declined by RSB_ECOMM system as ECI is in the list of blocked ECI (configuration of the RSB_ECOMM server end)
REVERSED	Transaction has been reversed (referred only for reversed transactions)
AUTOREVERSED	Transaction has been automatically reversed by RSB_ECOMM system (referred only for reversed transactions)
TIMEOUT	The time allotted in order to conduct the transaction is up
RESULT_PS	Transaction result in the payment server interpretation (configured in case in the request it is passed a variable server_version=2.0)
ACTIVE	Transaction has been authorized but not finished (captured)
FINISHED	Transaction has been captured successfully
CANCELLED	Transaction has been reversed
RETURNED	Transaction has been refunded
RESULT_CODE	Transaction result code (3 digits) (see in table « Result codes explanation»). RESULT_CODE field is informative only.
3DSECURE	3D Secure authentication status
AUTHENTICATED	Successful 3D Secure authentication
NOT_AUTHENTICATED	Non-successful 3D Secure authentication
NOTPARTICIPATED	Client's card doesn't participate in 3D Secure v1.0
CHALLENGE	Active 3DSecure authentication is required, wait for result
ATTEMPTED	There was a 3D secure authentication attempt
UNAVAILABLE	3D secure authentication is not available
REJECTED	3D Secure authentication was rejected by issuer
SKIPPED	3D Secure authentication was bypassed due to dynamic 3DS2.0 rules
FAILED	Default value, no details about 3D Secure available yet
RRN	Retrieval reference number returned from RSB_ECOMM system. Number uniquely identifies a transaction (12 characters) and appears only for successful transactions, for informative purposes, and they facilitate tracking the transactions in the RSB_ECOMM system. The decision as to whether a transaction was successful or failed must be based on the value of RESULT and RESULT_CODE fields only.
APP_CODE	Confirmation code, returned by RSB_ECOMM system (maximum 6 symbols) which appears only for successful transactions, for informative purposes, and they facilitate tracking the transactions in the RSB_ECOMM system. The decision as to whether a transaction was successful or failed must be based on the value of RESULT and RESULT_CODE fields only.
PAN	Masked card number
TRANS_ID	Transaction identifier (28 symbols in base64 encoding)
REFUND_TRANSACTION_ID	Transaction identifier (trans ID) of the refund – used in order to get details of the refund operation
CLOSE DAY	Applied to the Business-day closing procedure
fld_075	Number of reversals (up to 10 digits), displayed only if the result_code begins from 5
fld_076	Number of the payment operations (up to 10 digits), displayed only if the result_code begins from 5

fld_087	Total amount of reversals (up to 16 digits), displayed only if the result_code begins from 5
fld_088	Total amount of payment operations (up to 16 digits), displayed only if the result_code begins from 5
RECURRING	Used when registering a recurring payment
rec_pmnt_id	Recurring payment identifier chosen by Merchant
expiry	The validity period of the recurring payment in format MMYY

The field RESULT_PS has informative value and may be not displayed. Fields RRN and APPROVAL_CODE are displayed only for successful transactions, have the informational purpose and make the process of transaction tracking in the Ecomm Portal system easier. The basis for making decision regarding successful or non-successful execution of transaction should be only the value of the field RESULT and RESULT_CODE.
In case of the error the returned string of symbols will begin from 'error:'.
In case of the warning the returned string of symbols will begin from 'warning:'

5.1 Disambiguation of response codes (RESULT_CODE)

Result Code (returned from RSB ECOMM system (3 digits))	**Description SHORT**	**Description FULL**
000	Approved	Approved
100	Decline	Decline (general, no comments)
101	Decline	Decline, expired card
102	Decline	Decline, suspected fraud
103	Decline	Decline, card acceptor contact acquirer
104	Decline	Decline, restricted card
105	Decline	Decline, card acceptor call acquirer's security department
106	Decline	Decline, allowable PIN tries exceeded
107	Decline	Decline, refer to card issuer
108	Decline	Decline, refer to card issuer's special conditions
109	Decline	Decline, invalid merchant
110	Decline	Decline, invalid amount
111	Decline	Decline, invalid card number
112	Decline	Decline, PIN data required
113	Decline	Decline, unacceptable fee
114	Decline	Decline, no account of type requested
115	Decline	Decline, requested function not supported
116	Decline, no funds	Decline, not sufficient funds
117	Decline	Decline, incorrect PIN
118	Decline	Decline, no card record
119	Decline	Decline, transaction not permitted to cardholder
120	Decline	Decline, transaction not permitted to terminal
121	Decline	Decline, exceeds withdrawal amount limit
122	Decline	Decline, security violation
123	Decline	Decline, exceeds withdrawal frequency limit
124	Decline	Decline, violation of law
125	Decline	Decline, card not effective
126	Decline	Decline, invalid PIN block
127	Decline	Decline, PIN length error

128	Decline	Decline, PIN kay synch error
129	Decline	Decline, suspected counterfeit card
180	Decline	Decline, by cardholders wish
181	Decline	Decline, Card is not active
182	Decline	Decline, Card is not active
183	Decline	Decline, Card is not active
184	Decline	Decline, Card is not active
185	Decline	Decline, Card is not active
186	Decline	Decline, Card is not active
187	Decline	Decline, Card is not active
188	Decline	Decline, Card is not active
189	Decline	Decline, Card is not active
190	Decline	Decline, Card is not active
191	Decline	Decline, Card is not active
192	Decline	Decline, Card is not active
193	Decline	Decline, Card is not active
194	Decline	Decline, Card is not active
195	Decline	Decline, Card is not active
196	Decline	Decline, Card is not active
197	Decline	Decline, Card is not active
198	Decline	Decline, Card is not active
199	Decline	Decline, Card is not active
200	Pick-up	Pick-up (general, no comments)
201	Pick-up	Pick-up, expired card
202	Pick-up	Pick-up, suspected fraud
203	Pick-up	Pick-up, card acceptor contact card acquirer
204	Pick-up	Pick-up, restricted card
205	Pick-up	Pick-up, card acceptor call acquirer's security department
206	Pick-up	Pick-up, allowable PIN tries exceeded
207	Pick-up	Pick-up, special conditions
208	Pick-up	Pick-up, lost card
209	Pick-up	Pick-up, stolen card
400	Accepted	Accepted (for reversal)
499	Approved	Approved, no original message data
500	Approved	Status message: reconciled, in balance
900	Accepted	Advice acknowledged, no financial liability accepted
901	Accepted	Advice acknowledged, finansial liability accepted
902	Call acquirer	Decline reason message: invalid transaction
903	Call acquirer	Status message: re-enter transaction
904	Call acquirer	Decline reason message: format error
905	Call acquirer	Decline reason message: acqiurer not supported by switch
906	Call acquirer	Decline reason message: cutover in process
907	Call acquirer	Decline reason message: card issuer or switch inoperative
908	Call acquirer	Decline reason message: transaction destination cannot be found for routing
909	Call acquirer	Decline reason message: system malfunction
910	Call acquirer	Decline reason message: card issuer signed off
911	Call acquirer	Decline reason message: card issuer timed out
912	Call acquirer	Decline reason message: card issuer unavailable
913	Call acquirer	Decline reason message: duplicate transmission
914	Call acquirer	Decline reason message: not able to trace back to original transaction

915	Call acquirer	Decline reason message: reconciliation cutover or checkpoint error
916	Call acquirer	Decline reason message: MAC incorrect
917	Call acquirer	Decline reason message: MAC key sync error
918	Call acquirer	Decline reason message: no communication keys available for use
919	Call acquirer	Decline reason message: encryption key sync error
920	Call acquirer	Decline reason message: security software/hardware error - try again
921	Call acquirer	Decline reason message: security software/hardware error - no action
922	Call acquirer	Decline reason message: message number out of sequence
923	Call acquirer	Status message: request in progress
950	Not accepted	Decline reason message: violation of business arrangement

5.2 System errors (RESULT_CODE 1001)

List of most common error messages and reasons for them:

Result Code*	Description
1001	**error: unable to process transaction request** Connection failed
	error: wrong transaction id Incorrect transaction id value
	error: no transaction id No Transaction_id specified in merchant request
	error: unregistered merchant.IP: 12.345.67.89 Request has come from merchant IP address not registered on the Bank side.
	error: failed to get payment status The symbol + is incorrectly processed on merchant side (trans ID)
	error: total refunds amount already exceeds original amount The total amount to be refunded exceeds the amount of purchase
	error: digest failed Two possible reasons: - Specified card expiration date is prior todays date - Description is longer than approved
	error: transaction not found Status request contains space instead of symbol “+”