CyberPlat

Application Protocol Interface Guide

This document contains the protocol of interaction with the CyberPlat® payment system used for effecting of payments and checking of the payment status.

Contents

1. General information about data communication protocol

Program interaction with CyberPlat® payment services is organized over the open Internet communication channels using TCP/IP. Data transmission comes down to the exchange of messages within the limits of applied level protocol of HTTP (RFC2068,RFC2616). Application of the SSL layer (HTTPS) is available. In the first case connection with a server is established through the port 80, in the second case - through the 443.

POST method should be used for The Requests to the payment system server. The Content-Type HTTP header should be set to "application/x-www-form-urlencoded". The Content-Length header should contain the correct value.

The request message body is transferred in the same way, as the data, transformed in the HTML form.

There are three basic request types:

- Reception of payment authorization (request for checking the phone number / accuracy of the account)
- Payment
- Checking payment status

An URI, uniform resource identifier, is used for each of 3 basic request types. Three unique URL addresses are used for each Service Provider (sometimes one group of addresses is used for several providers).

In case the payment amount is unknown at the stage of checking the phone number, it is necessary to make 2 phone number checkings (the 1st is for fictive amount, the 2nd is for real one). The payment session code should coincide with a session code of the second number check.

2. Message format

2.1. Message structure

Messages contain a “plain” text. Each symbol is coded by 8 bits. The code page is Windows-1251. A message consists of one or more lines. Line conversion is coded by pair of symbols with the codes 0x0D, 0x0A ("\r\n" in C). Each line has a following form: "PARAMETER=VALUE", e.g., "PHONE=0951234567".

Before sending of request messages, a digital signature (DS) of a message is created. The message body with the signature is packed in a specific way. The operation data (signature and packing) are carried out by means of specialized function library. Library versions are available for C/C ++, C#, Java, and Delphi. The library binaries are provided for the following platforms: Linux (gcc 2.x, 3.x), FreeBSD 4.x/5.x (gcc 2.x/3.x), and Microsoft Windows (msvc). This specification does not provide detailed description of the library.

The request message is transferred in the form of a singular CGI parameter "inputmessage". The message should be signed by API outlet’s digital signature.

The reply comes back in the form of a plain text. It is signed by server DS and has the same structure, as the request message.

Three URL addresses are designated for each Service Provider (checking phone number, payment, checking payment status).

The format of request messages is identical for all mobile communication operators.

New fields can be added to request messages for other service operators. The sequence of fields can be changed.

More detailed information on the request message format for each Service Provider is available at:

Country	**URL**
India	https://in.cyberplat.com/cgi-bin/view_stat/help.cgi

2.2. Validation check (request for checking the phone number and amount)

URL: http://ru-demo.cyberplat.com/cgi-bin/DealerSertification/de_pay_check.cgi

Request:

SD=XXX – Dealer's code

AP=XXX – Acceptance outlet code

OP=XXX – Operator's (Outlet user's) code

SESSION=XXXXXXXXXXXXXXXXXXXX – unique dealer transaction ID (not more than 20 symbols, X means either a Latin letter, or a digit).

NUMBER=XXXXXXXXXX – Subscriber's phone number (X means a digit)

ACCOUNT= – Payer's personal account number (it is blank for mobile operators)

AMOUNT=XXX.XX – payment amount (not less than 1 ruble, comma stands for the separating character)

TERM_ID=XXXXX – code of the real outlet which sent the payment. It is used only by the aggregators for making payments to Beeline.

COMMENT=….. – commentary (payment assignment, only letters, digits, and blank spaces up to 64 symbols)

The abovementioned fields are mandatory to be filled in, except for the REQ_TYPE and COMMENT fields.

The request message has the following form, after its signature and packing:

```
0000035401SM0000009700000097000000121
api99                00000990
                     00000000
BEGIN
SD=199
AP=72
OP=990
SESSION=56567567100010000000
NUMBER=9998887766
ACCOUNT=
AMOUNT=500.00
TERM_ID=12345
COMMENT=test

END
BEGIN SIGNATURE
iQBRAwkBAAAD3j2r2NwBAeevAf4nvAG4rGAyAePHkyVKTt7wffzURhOckd3ctgmG
yQkKWkXh3CLpsbrExsllVUBlO6ih8qHozk2uttXApzHXQXoO
=+pch
END SIGNATURE
```

Answer:

DATE=DD.MM.YYYY HH:MM:SS – date and time of payment request message

SESSION= XXXXXXXXXXXXXXXXXXXX – unique payment number

ERROR=X – error code *

RESULT=X (0 – success, 1 – error)

OPNAME =SSSSSSSSSSSSSSSSSSS

ACCOUNT=XXXXXXXXXX

ERRMSG=SSSSSSSSSSSSSSSSSSSSS

If ERROR = 0, then the phone number exists and the payment is allowed.

If the phone number exists, one should can send the payment request (step 2).

OPNAME – domestic Service Provider name, which owns the phone number (not more than 20 symbols). ACCOUNT – Subscriber's personal account number.

ERRMSG – error decoding (blank in case of successfully effected payment). Parameters OPNAME, ACCOUNT, ERRMSG are optional. They can be omitted in the answer.

An example of a reply from the server signed by the CyberPlat DS:

```
0000030301SM000000460000004600000121
0J0005              00064182
                    00000000
BEGIN
DATE=04.07.2002 12:34:12
SESSION=56567567100010000000
ERROR=0
RESULT=0
OPNAME=OAO "MTS" Moscow
ACCOUNT=3335566
ERRMSG=

END
BEGIN SIGNATURE
iQBRAwkBAAD6tj1BJ10BAYKxAfsHlQsEFnO2k6ry++W8O8AiJuv4gT+ZVCfZHsKk
c0CbZpP/W3vkljG3xNzMLiqjbwkNuIdwR9Dq7gHmH+ZQMhbT
=LOnP
END SIGNATURE
```

2.3. Payment request

URL: http://ru-demo.cyberplat.com/cgi-bin/DealerSertification/de_pay.cgi

Payment request should be sent only after a successful validation check (i.e. after Step 1 has been successfully performed).

Request:

SD=XXX – Dealer's code

AP=XXX – Acceptance outlet code

OP=XXX – Operator's (Outlet user's) code

SESSION=XXXXXXXXXXXXXXXXXXXX – unique operation number (the same as in 2.2.).

NUMBER=XXXXXXXXXX – Payer's phone number (X means a digit)

ACCOUNT= – Payer's personal account number

AMOUNT=XXX.XX – payment amount (not less than 1 ruble and not more than 30000 rubles, where the period is the separating character)

The above-mentioned fields are mandatory.

If an error occurs at this step, the payment must be repeated from the first step using new session number.

In case there has not been any unambiguous reply (e.g., communication error), it is necessary to check the payment status until unambiguous reply is received (Step. 3).

The request message has the following form, after its signature and packing:

```
0000035401SM0000009700000097000000121
api99                00000990
                     00000000
BEGIN
SD=199
AP=72
OP=990
SESSION=56567567100010000000
NUMBER=9998887766
ACCOUNT=
AMOUNT=500.00
TERM_ID=12345

END
BEGIN SIGNATURE
iQBRAwkBAAAD3j2r2NwBAeevAf4nvAG4rGAyAePHkyVKTt7wffzURhOckd3ctgmG
yQkKWkXh3CLpsbrExsllVUBlO6ih8qHozk2uttXApzHXQXoO
=+pch
END SIGNATURE
```

Answer:

DATE=DD.MM.YYYY HH:MM:SS – date and time of payment request message

SESSION= XXXXXXXXXXXXXXXXXXXX – unique payment number

ERROR=X – error code *

RESULT=X (0 – success, 1 – error)

TRANSID=XXXXX – a unique transaction number in the payment system.

An example of a reply from the server signed by the CyberPlat DS:

```
0000030301SM0000004600000046000000121
0J0005               00064182
                     00000000
BEGIN
DATE=04.07.2002 12:34:12
SESSION=56567567100010000000
ERROR=0
RESULT=0


END
BEGIN SIGNATURE
iQBRAwkBAAD6tj1BJ10BAYKxAfsHlQsEFnO2k6ry++W8O8AiJuv4gT+ZVCfZHsKk
c0CbZpP/W3vkljG3xNzMLiqjbwkNuIdwR9Dq7gHmH+ZQMhbT
=LOnP
END SIGNATURE
```

2.4. Checking payment status (status request)

URL: http://ru-demo.cyberplat.com/cgi-bin/DealerSertification/de_pay_status.cgi

Status request should be sent only after a successful payment requisites check (i.e. after Step 2 has been successfully performed).

Request:

SESSION=XXXXXXXXXXXXXXXXXXXX – unique operation number (not more than 20 symbols, X – either Latin letter, or digit).

The request message has the following form, after its signature and packing:

```
0000035401SM0000009700000097000000121
api99                00000990
                     00000000
BEGIN
SESSION=56567567100010000000

END
BEGIN SIGNATURE
iQBRAwkBAAAD3j2r2NwBAeevAf4nvAG4rGAyAePHkyVKTt7wffzURhOckd3ctgmG
yQkKWkXh3CLpsbrExsllVUBlO6ih8qHozk2uttXApzHXQXoO
=+pch
END SIGNATURE
```

Answer:

DATE=DD.MM.YYYY HH:MM:SS – date and time of checking request payment status

SESSION= XXXXXXXXXXXXXXXXXXXX – unique payment number

ERROR=X – error code

RESULT=X – payment status

1 – there has only been a phone number checking (payment request has not been delivered yet)

3 – the payment has been transferred to the Service Provider (the transaction is in pending)

7 – the payment has been successfully effected (the ERROR field can now be checked, "0" means that the payment has been successfully effected)

AUTHCODE=XXXXXXXXXXX – transaction ID of the Service Provider.

TRANSID=XXXXX – a unique transaction number in the payment system. This parameter does not come back for all payment types.

The following combinations of ERROR and RESULT fields are possible in reply messages:

If the RESULT is an empty string, then ERROR contains code of an error in the payment status reception request . If ERROR field contains the error 11, this means that the payment has not been registered in the Cyberplat system. It is necessary to repeat from the first step using a new session number. All other values of the ERROR field mean that payment status is unknown (e.g., signature checking error or database failure). One should try again later.

If RESULT is smaller than 2, then the payment has not been effected. It is necessary to repeat from the first step using a new session number.

If RESULT is more than 1 and less than 7, then the payment is being processed, the result is yet unknown. One should try again later.

If RESULT is equal to 7, then the payment has been processed, the status has been defined; one may check the ERROR field value. If ERROR is equal to zero, then the payment has been successfully processed. Other values of the ERROR field mean that an error has occurred (payment failed).

The payment is considered to be successfully effected only if ERROR=0 and RESULT = 7

Status check should iterate until RESULT equals to 7.

If status check returns payment failure for a given payment – the payment should be sent one more time, all payment steps repeated with a new session number.

An example of a reply from the server signed by the CyberPlat DS:

```
0000030301SM000000460000004600000121
0J0005              00064182
                    00000000
BEGIN
DATE=04.07.2002 12:34:12
SESSION=56567567100010000000
ERROR=0
RESULT=7
AUTHCODE=5967596


END
BEGIN SIGNATURE
iQBRAwkBAAD6tj1BJwMBAVP6Af0XC+5LDcE/OED4jrKoJjmERFMKOV4NZ1MUkAiM
dKC6eQSXELkVY8fvv1HvTE2G07p4uGoIJ4yLpwYL5tHHPLs7
=mJ0H
END SIGNATURE
```

2.5. Wallet Balance request

A request described below can be used to retrieve dealer's account balance. Target URL is specific for each country:

Country	URL
India	https://in.cyberplat.com/cgi-bin/mts_espp/mtspay_rest.cgi

POST method should be used for account balance request. The request signed by API outlet's DS goes in parameter "inputmessage".

SD=XXX – Dealer's code

AP=XXX – Acceptance outlet code

OP=XXX – Operator's (Outlet user's) code

SESSION=XXXXXXXXXXXXXXXXXXXX – unique operation number.

The request message has the following form when signed and packed:

```
0000037001SM000000170000001700000217
api123456           00000040
                    00000000
BEGIN
OP=102
AP=10
SD=123
SESSION=asdfghjkl123Wderfgth

END
BEGIN SIGNATURE
iQCRAwkBAAAAKEvRqHcBAfC7A/9bni8dnCiLma0+BL3q0GWagxczjJnzuIgyfO2/
6H4+y31I/6Tr7jhldIYdsIomZxlfOWdKZjmTiKtz4bGSore8wc7lthtkQzlS/9SA
0Njw/pPKBcT8TeK0UogHwOOPvTOfWXxATDEM4tlNtwdiXXVe85Xd+cX4L252ef63
ZmMgSrABxw==
=sYlr
END SIGNATURE
```

Answer:

REST = X – dealer's account balance.

ERROR=X - error code (0 – success).

An example of a reply from the server signed by the CyberPlat DS:

```
0000037001SM000000170000001700000217
inist_in            00000040
                    00000000
BEGIN
ERROR=0
REST=124050

END
BEGIN SIGNATURE
iQCRAwkBAAAAKEvRqHcBAfC7A/9bni8dnCiLma0+BL3q0GWagxczjJnzuIgyfO2/
6H4+y31I/6Tr7jhldIYdsIomZxlfOWdKZjmTiKtz4bGSore8wc7lthtkQzlS/9SA
0Njw/pPKBcT8TeK0UogHwOOPvTOfWXxATDEM4tlNtwdiXXVe85Xd+cX4L252ef63
ZmMgSrABxw==
=sYlr
END SIGNATURE
```

2.6. ERROR field values

Error code	Description
0	Successfully completed.
1	Session with such a number already exists.
2	Incorrect Dealer's code (SD)
3	Incorrect acceptance outlet code (AP)
4	Incorrect Operator's (outlet user's) code (OP)
5	Incorrect session code format.
6	Incorrect EDS.
7	Incorrect amount
8	Incorrect phone number
9	Incorrect format of personal account number.
10	Incorrect request message format.
11	Session with such a number does not exist.
12	Request is made from an unregistered IP.
15	Payments in favor of this Service Provider are not supported by the system.
17	The phone number does not correspond to the previously entered number.
18	The payment amount does not correspond to the previously entered amount.
19	The account number does not correspond to the previously entered number.
20	The payment is being completed.
21	Not enough money on Dealer's account for effecting the payment.
22	The payment has not been accepted. Money transfer error.
23	Wrong phone number (such phone number does not exist).
24	No connection with Service Provider's server (routine break).
25	Effecting of this type of payments is suspended.
26	No connection with Service Provider's server (routine break).
30	General system failure.
32	Repeated payment within 60 minutes from the moment of payment effecting (CyberPlat).
33	Maximum interval between number verification and payment (24 hours).
34	Transaction with such number could not be found.

Error code	Description
35	Payment status alteration error.
39	Wrong account number.
41	Error in saving the payment in the system.
44	The Client cannot operate on this Trading Server.
45	The specific provider not activated for recharges
46	Could not complete the erroneous payment.
47	Temporary limitation of access rights has been activated.
50	It is temporarily impossible to effect payments in the system.
51	Data not found in the system.
52	The Dealer may be blocked. The Dealer's current status does not allow effecting payments.
53	The Acceptance Outlet may be blocked. The Acceptance Outlet's current status does not allow effecting payments.
54	The Operator (outlet user) may be blocked. The Operator's current status does not allow effecting payments.
55	The Dealer Type does not allow effecting payments.
56	An Acceptance Outlet of another type has been expected. This Acceptance Outlet type does not allow effecting payments.
57	An Operator (outlet user) of another type has been expected. This Operator type does not allow effecting payments.
81	Maximum payment amount has been exceeded.
82	Daily debit amount has been exceeded.
88	Duplicate Transaction (Same Mobile Number)
224	Operator Server Down

3. Software operation algorithm

- Request message is formed (see description).
- DS of outlet is applied to message (see relevant manual).
- URL-encoding for the request message is performed. Latin letters and digits remain without any changes; other symbols are transferred in hexadecimal notation with prefix '%', e.g. "%OD%OA" – line conversion. Blank intervals can be replaced by symbol '+'. Attention! Symbol '+' should be encoded as well since upon decoding it will be replaced with a blank space.
- "inputmessage=" should be placed in the beginning of the line.
- The message is transferred to the payment system server as part of the body of an HTTP request by POST method with the header "Content-Length" and "Content-Type" (see above).
- The HTTP response is sent in a plain text with a signed reply of the payment system (without URL-encoding).
- The signature verification should be done via a call to the library API.
- The content of the response message is parsed (see above).