

# IBOX PAYMENT API

## A. Requirements for merchant interface

1.1. The interface should make requests under the HTTP or HTTPS with the IP-address subnet:

**production: 213.160.149.229**

**Test: 213.160.149.230**

1.2. The interface must process parameters passed by GET

1.3. The interface must generate a response system in XML encoded in UTF-8 (if the response contains the symbols of national alphabets)

1.4. Information exchange is conducted using the request-response, and the response delay should not exceed 60 seconds, otherwise the system breaks the connection to time out.

1.5. If the expected number of payments for the services of receiver that connects expected heavy (more than 10 payments per minute), you need to supporting interface with many communication flow to 10-15 simultaneous connections.

## 2. Basic principles of interface

2.1. Each payment in IBox system has a unique ID that is sent to the merchant in variable **txn\_id** - an integer of up to 20 characters. With this ID is held further reconciliation and mutual resolve controversial issues.

2.2. Total amount of payment to the merchant made by Payer and transferring to the receiver in UAH in variable **sum** - a fractional number to hundredths, as a separator is used "." (Dot). If the sum is an integer, it still may be supplemented by a point and zeros, for example - "**152.00**"

2.3. A request to add new payment, system transfers the payment date (date when the payment system refers to the date of receipt of request from the terminal system IBox) in variable **txn\_date** - date format **HHHHMMDDCHCHMMSS**.

- 2.4. Merchant identifies payers for a unique identifier (personal account number, phone number, login, etc.). Before sending to merchant, identifier accuracy is checked in accordance with the regular expression, which must provide the merchant. Payer ID is transmitted in variable account - a string that contains letters, numbers and special characters, up to 200 characters.
- 2.5. If necessary, use online verification, possible presence of an additional stage of checking with the transfer of user account and the minimum-possible amount of payment to merchant. In this case, the transfer of payment information to the merchant system is conducted in two stages - verification of payer and the payment directly. Type of Request is transmitted in the system variable command - string that takes the value of «check» and «pay». When checking the status (request «check»), through the billing system. When a payment (request «pay»), Recipient shall make prepaid balance Payer.
- 2.6. If any of the requests the receiver fails with an error, the recipient returns the error code, according to the table below (Appendix A). All errors are a sign of fatal error. For a system fatal error means that the re-send your request with the same parameters will result in 100% repeat the same mistakes, so the system stops processing the client request and ends with his mistake. Not a fatal error means for system repeat the query with the same parameters over time may lead to success. The system will repeat requests that end with no fatal error, constantly increasing the interval until the operation is completed successfully or a fatal error, or until you end the life of the request - 24 hours. No link to the service receiver is nonfatal error. Lack of response element <result> (incorrect XML, page Service temporarily unavailable, etc.) is a fatal mistake. Interpreted as a waiver of error 300 - Another error receiver.
- 2.7. With a receiver should not contain two of successful payments with the same number txn\_id. If the system repeatedly requests from already existing in the receiver txn\_id, then the recipient must return the result of processing the previous request.

Receiver returns the response to requests from the system in XML format with the following structure:

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
<ibox_txn_id></ibox_txn_id>
<prv_txn></prv_txn>
```

```
<prv_txn_date></prv_txn_date>
<sum></sum>
<result></result>
<comment></comment>
</response>
```

Where:

<response> - body response

<ibox\_txn\_id> - transaction number in the system, which is transmitted to the receiver in variable txn\_id.

<prv\_txn> - Unique number of operations to replenish the balance of the payer (in a receiver), an integer of up to 20 characters. This element should return a recipient after a request to replenish the balance (request «PAY»). In response to a request for review of Payer (request «CHECK»), his return is required - not processed.

<prv\_txn\_date> - Stock replenishment operations balance the payer (in a receiver) in the format yy-MM-DD hh: MM: SS. This element should return a recipient after a request to replenish the balance (request «pay»). In response to a request for review of Payer (request «CHECK») its return is required - not processed.

<sum> - Amount of payments in favor of the receiver that is transmitted to the receiver, a fractional number to the nearest hundredths, as a separator is used "." ( point). If the sum is an integer, it will still be supplemented by a point and zeros, for example «152.00»-

<result> -code completion query result.

<comment> - Optional element - the comments of the transaction.

### 3. Example request for review of the payer's account and registration payment

3.1. Paying applet receiver payment\_app.cgi, located at service.someprv.ua, server supports HTTPS connections on port 8443. To check the status of the payer, the system generates a query of the form:

```
https://service.someprv.ua:8443/payment_app.cgi?  
command=check&txn_id=1234567&account=4957835959&sum = 10.45
```

Request contains variables:

**command**= check - request for review of Payer

**txn\_id**= 1234567 - extension number in the payment system IBox

**account**= 4957835959 - Payer ID in the information system receiver

**sum**= 10.45 - amount credited to the personal account of the payer (sum payment to the receiver)

Possible transfer of these additional fields in the query string:  
**pay\_type**= 1 - ID service provided by a recipient, integer length to 5 characters. (Used if Receiver provides services to more than 1.)  
**Prv\_id**= 999 - internal identifier receiver in the system IBox, integer, long, up to 4 characters. (Used for consolidators.)  
**Trm\_id**= 8792525 - id terminal, integer length to 20 characters. (Used for branches IBox.)  
**Agt\_id**= 4151231 - id agent, integer length to 20 characters. (Used for branches IBox.)  
**Sum\_from**= 12.00 - sum payment of additional. Commission, the fractional number to the nearest hundredths, as a separator is used "." (dot). (Used for branches IBox.)  
**Data1, data2, ..., dataN** - Additional fields that transmits Receiver - rows that contain letters, numbers and special characters.  
In the query string can be transferred one, several, all additional fields.

### 3.2.Ryadok query might look like

```
https://service.someprv.ua:8443/payment\_app.cgi?  
command=check&txn\_id=1234567&account=4957835959&sum=10.45&pay\_type=1&prv\_id=999&trm\_  
id=4151200&agt\_id=4151231&sum\_from=11.00&data1=ibox
```

If online verification is possible with previous generation query lowest possible amount.

A receiver should look like this:

```
<?xml version="1.0" encoding="UTF-8"?>  
<response>  
<ibox_txn_id>1234567</ibox_txn_id>  
<result>0</result>  
<comment></comment>  
</response>
```

Return result = 0 at the request «check» indicates that the personal account of a taxpayer with the appropriate number txn\_id it can be replenished in an amount specified in the request. After successful verification of the payer's account system goes to the form and send your request to replenish the balance (request «pay»).

In the optional comment field contains the official comment.

3.3. If necessary, additional information about payment can be transferred in the fields.

A receiver should then look like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
<ibox_txn_id>1234567</ibox_txn_id>
<result>0</result>
<fields>
<field1 name="name1"> value1</field1>
<field2 name="name2"> value2</field2>
...
<fieldN name="nameN">valueN</fieldN>
</fields>
<comment></comment>
</response>
fields>
<comment>comment>
response >
```

In the optional fields field1, field2, fieldN contains information that should be transferred to the system. This information can be displayed to the user when making payment if you use online validation.

3.4. If necessary, information gateway, which will be fixed payment, you can send in tag pay\_id.

A receiver should then look like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
<ibox_txn_id>1234567</ibox_txn_id>
<result>0</result>
<comment></comment>
<pay_id>111</pay_id>
</response>
```

Gateways are available for selection specified in the test system. A receiver may contain any additional fields or only part of them.

## 4. Example request for updating personal account

4.1. To confirm the payment, the system generates the following query

```
https://service.someprv.ua:8443/payment_app.cgi?  
command=pay&txn_id=1234567&  
txn_date=20050815120133&account=4957835959&sum=10.45
```

Request contains variables:

**Command** = pay - request to replenish the balance of Payer  
**txn\_id** = 1234567 - extension number in the payment system IBox  
**txn\_date** = 20050815120133 - date of payment records in the system IBox  
**account** = 4957835959 - Payer ID in the information system receiver  
**sum** = 10.45 - the amount to transfer the personal account of the payer (sum payment to the receiver).

Possible transfer of these additional fields in the query string:

**pay\_type** = 1 - ID service provided by a recipient, an integer, up to 5 characters. (Used if Receiver provides services to more than 1).

**Prv\_id** = 999 - internal identifier receiver in the system IBox, integer, long, up to 4 characters. (Used for consolidators).

**Trm\_id** = 8792525 - id terminal, integer length to 20 characters. (Used for branches IBox).

**Agt\_id** = 4151231 - id agent, integer length to 20 characters. (Used for branches IBox).

**Sum\_from** = 12.00 - sum payment of additional commission, a fractional number to the nearest hundredths, as a separator is used "." (Dot). (Used for branches IBox).

**Data1, data2, ..., dataN** - Additional fields transmitted recipient - string containing letters, numbers and special characters.

In the query string can be transferred one, several, all additional fields.

4.2. Query string might look like

```
https://service.someprv.ua:8443/payment_app.cgi?  
command=pay&txn_id=1234567&account=4957835959&sum=10.45&txn_date=20050815120133&pay_  
_type=1&prv_id=999&trm_id=4151200&agt_id=4151231&sum_from=11.00&data1=ibox
```

Example answer:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<response>
<ibox_txn_id>1234567</ibox_txn_id>
<prv_txn>2016</prv_txn>
<prv_txn_date>2005-08-15 12:01:40</ prv_txn_date>
<sum>10.45</sum>
<result>0</result>
<comment>OK</comment>
</response>
```

Returning result = 0 at the request «PAY», Receiver has successfully balance replenishment operations. The system closes the processing of The transactions.optional comment field contains the official comment.

## **Appendix A: List of code completion**

When processing requests from the system, Receiver must compare all the errors that arise in its application, from the list below and return the appropriate code in the element <result>. Sign "+" in column fatality - shows how the system will interpret this error.

In an optional comment field contains the official comment errors.