

Guidelines for the programmatic interaction with

«MoneXy» payment system

Technical description of communication protocol

CONTENT

Introduction	3
General information about the exchange protocol.....	4
Requirements for the use of this API.....	5
General information about the messages format.....	6
Message structure.....	6
Data format.....	7
Format of customer queries.....	8
General structure of queries to the server.....	8
General structure of the server replies.....	8
Information on authentication.....	9
Creating an authentication token.....	9
Errors processing.....	10
Security requirements.....	10
Description of the supported queries.....	11
Request for transfer from the user's wallet to the corporate wallet.....	11
Request for transfer confirmation with an SMS code.....	13
Request for transfer (transfer possibility check) from a corporate to a private wallet.....	15
Queries sequence.....	16
Request for transfer status using an OrderID.....	19
Payment queries sequence:.....	20
Authorization request.....	21
Authentication request.....	22
Request for balance of the user's session.....	23
Request for balance using a BCode.....	24
Request for wallet-to-wallet transfer.....	25
Transaction history request.....	27
Generating a MoneXy voucher.....	29
Transfer from MoneXy voucher to customer wallet.....	30
Transfer from MoneXy voucher to a corporate wallet.....	31
Checking the balance of MoneXy voucher.....	32
Balance of a corporate wallet linked to an API (Merchant).....	33
Balance of a corporate wallet linked to an API (Distributor).....	34
Checking the status of payment (or voucher creation) using an unique OrderID.....	35
Request for transaction cancellation.....	36

Introduction

The presented document contains a complete description of the XML API interaction protocol between web-based clients, and the payment system MoneXy server, including use features of all of its functions.

XML API is a software interface that allows you to automate the login process, sending payments, receiving the accounts balance or the history of operations performed. The format and structure of data in XML API is designed on the concept of XML open standard.

The document is intended for all developers who aim at integration of Internet payment procedures of web resources with MoneXy software interface.

You will need knowledge and experience in the following domains:

- . Preparing and sending HTTP queries;
- . Receiving and analyzing of the HTTP response.

The following skills are recommended:

- . Understanding the HTTPS protocol features;
- . Working with XML data (<http://en.wikipedia.org/wiki/XML>);
- . Basic knowledge of encryption systems and hashing methods.

General info about exchange protocol

Software interaction with the payment system via the Internet network is performed using TCP / IP as the transport layer protocol. Data transfer is carried out by messaging within an application layer protocol HTTPS, ie HTTP (RFC 2068, RFC 2616), with the mandatory use of SSL / TLS layer. The connection to the server is installed on port 443.

There are the following types of system queries:

- . Request for making a payment;
- .. Payment confirmation;
- Request to check the payment status.

Each of query types uses the same URL.

Requirements for the use of this API

To start working with the API provided the following steps are required:

You have to register the API linked to your personal account and also to configure additional settings:

- API name. This name will be used in a variety of queries, as a part of the authentication process.
- IP addresses from which queries will be sent to API. This should be a list of IP- addresses that you authorize to make requests for your XML API. You can leave it blank if you do not want to restrict access by IP-address. To learn more about this topic, please see the security considerations section..
- Secret keyword. This code word will be used in all queries as part of the authentication process.
- You should link your wallets to the API name.

You can create an unlimited number of API, linked to the same account.

It should be understood that access protection via indicating IP-addresses is possible only if you have static IP addresses.

To work with this API, you will need the following::

- Implementation of the interaction using a programming language that allows you to create XML documents and pass them through HTTPS POST queries.
- If you plan to restrict access to your account with one or more static IP-addresses, please make sure that the computer that sends queries MoneXy API has one of these IP-addresses.

General information about message format

Message structure

XML API is synchronous interface that provides information only on request and in accordance with its correct wording. Each request consists of MoneXy XML API commands. After each request, the system sends a response with the result of the requested action or an error code. In the case of returning an error code, a textual description is also returned, indicating the nature of the error

Each request and the response contains data in a well-defined XML API format.

Successful completion of the XML API queries requires the following steps:

- Data collection for the operation.
- Creating XML document in accordance with the format of requested operation.
- Creating a HTTP POST request with pre-prepared XML document.
- Sending HTTP POST request to MoneXy server via HTTPS.
- Receiving the server's response.
- Processing of the data returned from the MoneXy server.

Messages contain a "flat" text in UTF-8 - encoding, each character is encoded with 8 bits. For the Russian character code page is used: UTF-8. The message body may consist of one or more lines. Newline characters are encoded with the pair of symbols with the codes 0x0D, 0x0A.

Data format

The following table lists the types and format of the data used in MoneXy XML API.

Data type	Format	Examples
API Name	API name, up to 32 characters (specified during registration)	TestApiName
Transaction ID	The long integer (64-bit integer)	121992579522263 9994456683762355345868
Username (Wallet)	+ and 12 digits	+380671231212
Amount	Fractional up to 4 digits in the denominator, a dot (.) as a separator	1.26 456.7895
Date	YYYYMMDD HHMMSS	20072503 104555
Currency	Currency in the following formats: USD, EUR, UAH, RUR	UAH
Commission (fee)	Transaction Fees. Fractional up to 4 digits in the denominator, a dot (.) as a separator.	0.26 6.75
Payment description	Payment description. A string of text up to 100 characters (varchar (100))	Refilling the wallet 380671231212
Recipient's account	12 digitss	380671231212
Transaction amount	Fractional up to 4 digits in the denominator, a dot (.) as a separator. (-) Can reflect withdrawal transaction	-0.5000 1.5000
API Hash	40 characters	961fdc35a590cad460305140fa68bf 63a9e9fcb6

Customers' queries format.

The overall structure of the server query.

Below is a basic XML-template of request to the XML API server. Requests for all operations can have a different set of parameters; but the following part remains always constant and mandatory:

```
<monexyApi type="REQUEST NAME" mtime="DataTimeMS">
  <Auth>
    <ApiName>API name</ApiName>
    <ApiHash>authentication token</ApiHash>
    <ApiLogin>authentication login</ApiLogin>
  </Auth>
  <Operation name>
    ... operation data ...
  </Operation name>
  ...
  <Operation name>
    ... operation data ...
  </Operation name>
</monexyApi>
```

In the root element, the TYPE parameter determines the type of query, and MTIME parameter determines current date of the query including time in milliseconds.

The displayed contents of the authentication unit is optional, but can be extended depending on the type of query.

The overall structure of the server responses

This section provides a basic template in XML format of a typical server response. All responses have the same structure. The root element of the document contains parameters defining query type (TYPE parameter), current date of the query including time in milliseconds (MTIME) as well as authentication blocks and descriptions of errors.

```
<monexyApi type="RESPONSE NAME" mtime="DataTimeMS">
  <Auth>
    <ApiName> API name </ApiName>
    <ApiHash> authentication token </ApiHash>
  </Auth>
  <Error ErrorId="ERROR ID">
  <ErrorDesc> ERROR DESCRIPTION</ErrorDesc>
  </Error>
  <Element name>
    ... response data ...
  </Element name>
  ...
  <Element name>
    ... response data ...
  </Element name>
</monexyApi>
```

Authentication information

XML-document should always contain information that will enable XML API to check user identity. This document will contain the user name and other information, which will help the server to identify accurately a user who sends a request for a transaction. Each request to the XML API server must contain the following authentication information:

- API name.
- The authentication token, which consists of an encrypted line of text, formed from the concatenation of the API secret word and the time stamp in UTC format.

As mentioned above, the contents of the block can be extended depending on the type of request.

Creating an authentication token.

To initiate the exchange of information via the API, you must create an authentication token using the following procedure:

- To get concatenation of the following line:
«API name»:« Secret word»:« The current date and time UTC in milliseconds»
- To get the SHA1 hash of the above line.

Example of creating an authentication token:

API name - test, Secret word - MySecWord, The current date and time UTC in milliseconds - 125890135061147

Concatenation of the parameters:

Test:MySecWord:125890135061147

Hash SHA1 of the above line:

32295ae9f3b7988a57ce1893d227486815884c77

Example of creating an authentication token in the PHP programming language:

```
// Creation of MkTime and ApiHash
```

```
function MkTime ()
{
    $MkTime = $this->getmicrotime ();
    $this->ApiHash = sha1 ($this->ApiName . ':' . $this->SecretKey . ':' .
    $MkTime);
    return $MkTime;
}
```

Error processing

Errors can occur during the processing of XML API queries the server or in the analysis of your requests. Bcero существует три вида ошибок. They are listed below:

- Authentication / authorization errors. Such mistakes can occur if the authentication token contains the wrong data, if the API input is disabled or if server XML API interface is unavailable..
- Errors resulting from incorrect formation of structure or using incorrect data format in query elements. The source of these errors is usually located in your software.
- The errors on the server side. These errors may be due to technical problems on the server.

Whenever XML API server encounters an error, it will include information about it in the response. So you always get the results of processing codes and information about what went wrong in the response to every request that is completed with an error.

Description of available error codes (<http://monexy.com/xml/index.php?mx=PaymentCode>)

If you receive a code with a value of "0" - this means that the query was successful. If it's a different meaning - the query was unsuccessful.

Security requirements

To ensure the most optimal and secure access to the MoneXy API XML interfaces , it is necessary to implement the following recommendations:

- When you connect your API, it is recommended to specify the IP-address of access.. This will prevent attempts to obtain access to your account from a computer that has an IP address other than within the range you have specified. Please remember that only static IP-addresses are permitted You can use a dynamic IP address is only for testing.
- Secret word (Secret Keyword) should not be easily guessable or very short, and it should never be the same as your original password to the account.
- Never distribute your passwords, secret words, and other confidential data. Never transfer this information via open communication channels..
- Do not use your account password or other sensitive information in the source code of the software. It is highly desirable to keep such sensitive information in a file with limited access, or in a separate database in encrypted form.

Description of supported queries

All requests must be sent to the URL in the following format:

<https://www.moneyx.com/xml/server.php?req=<request body>>, where <request body> - query XML document

All requests must be transmitted over HTTPS via GET or POST command. The body of the request should be transferred in the URL-encoded form.

Request for transfer from a personal wallet to a corporate wallet

Parameters	Name	Value
<i>Incoming</i>	type	Type of Request, «payment-req» - a request for transfer from one account to another
	mtime	The current date and time in milliseconds
	ApiName	API name
	ApiHash	Hash from the API name, the secret word and the current time
	ApiLogin	Username (phone number)
	ApiSess	Session ID UserSession
	OrderId	Number of transfer document
	OrderDesc	Transfer purpose
	PayeeCard	Recipient (corporate wallet – merchant ID)
	PayeeCurrency	Currency of the recipient
	PayerCurrency	Currency of the sender
	Amount	Transfer amount
	AmountType	Transfer currency type (0-in sender's currency, 1-in recipient's currency)
	Status	Query type (0-test, 1 – transfer)
<i>Outgoing</i>	ErrorId	Error code
	ErrorDesc	Error code description
	PaymentId	Transfer ID
	TransId	Transaction ID
	PayeeAmount	The amount of transfer to the recipient
	PayeeCurrency	Recipient's currency
	PayerAmount	The amount of transfer from the sender
	PayerCurrency	Sender's currency
	TransTax	Transfer fee
	TransDate	Date and time of the transfer
	OrderId	Number of the transfer document
	Status	Query type (0-test, 1 – transfer)

Query

```

<monexyApi type="payment-req" mtime="DataTimeMS">
  <Auth>
    <ApiName> API name </ApiName>
    <ApiHash> Authentication token </ApiHash>
    <ApiLogin> Login (phone number) </ApiLogin>
    <ApiSess> PWS </ApiSess>
  </Auth>
  <Payment>
    <OrderId> OrderId </OrderId>
    <OrderDesc> OrderDesc </OrderDesc>
    <PayeeCard> PayeeCard </PayeeCard >
    <PayeeCurrency> PayeeCurrency </PayeeCurrency>
    <PayerCurrency> PayerCurrency </PayerCurrency>
    <Amount> Amount </Amount>
    <AmountType> AmountType </AmountType>
    <Status> 1 </Status>
  </Payment>
</monexyApi>

```

Response

```

<monexyApi type="payment-req" mtime="DataTimeMS">
  <Auth>
    <ApiName> API name </ApiName>
    <ApiHash> Authentication token </ApiHash>
  </Auth>
  <Error ErrorId="0">
    <ErrorDesc> Error message </ErrorDesc>
  </Error>
  <Payment>
    <PaymentId> PaymentId </PaymentId>
    <TransId>TransactionID</TransId>
    <PayeeAmount> PayeeAmount </PayeeAmount>
    <PayeeCurrency> PayeeCurrency </PayeeCurrency>
    <PayerAmount> PayerAmount </PayerAmount>
    <PayerCurrency> PayerCurrency </PayerCurrency>
    <TransTax> Transaction fee </TransTax>
    <TransDate> Transaction Date & time (YYYYMMDD HHMMSS) </TransDate>
    <OrderId> OrderId </OrderId>
    <Status> 1 </Status>
  </Payment>
</monexyApi>

```

Request for transfer confirmation with SMS code

Parameters	Name	Value
<i>Incoming</i>	type	Type of query, «payment-conf» - a request for transfer confirmation with an SMS code
	mtime	The current date and time in milliseconds
	ApiName	API name
	ApiHash	Hash from the API name, the secret word and the current time
	ApiLogin	Username (phone number)
	ApiSess	Session ID UserSession
	PaymentId	Transfer ID
	PaymentSMS	Transfer confirmation code, received in SMS
<i>Outgoing</i>	ErrorId	Error code
	ErrorDesc	Error code description
	TransId	Transaction ID
	PayeeAmount	The amount of transfer to the recipient
	PayeeCurrency	Recipient's currency
	PayerAmount	The amount of transfer from the sender
	PayerCurrency	Sender's currency
	TransTax	Transfer fee
	TransDate	Date and time of the transfer
	OrderId	Number of the transfer document
	Status	Query type (0-test, 1 – transfer)

Query

```
<monexyApi type="payment-conf" mtime="DateTimeMS">
  <Auth>
    <ApiName> API name </ApiName>
    <ApiHash> Authentication token </ApiHash>
    <ApiLogin> Login (phone number) </ApiLogin>
    <ApiSess> PWS </ApiSess>
  </Auth>
  <Payment>
    <PaymentId> PaymentId </PaymentId>
    <PaymentSms> PaymentSms </PaymentSms>
  </Payment>
</monexyApi>
```

Response

```
<moneyApi type="payment-conf" mtime="DateTimeMS">
  <Auth>
    <ApiName> API name </ApiName>
    <ApiHash> Authentication token </ApiHash>
  </Auth>
  <Error ErrorId="0">
    <ErrorDesc> Error message </ErrorDesc>
  </Error>
  <Payment>
    <TransId> TransactionID </TransId>
    <PayeeAmount> PayeeAmount </PayeeAmount>
    <PayeeCurrency> PayeeCurrency </PayeeCurrency>
    <PayerAmount> PayerAmount </PayerAmount>
    <PayerCurrency> PayerCurrency </PayerCurrency>
    <TransTax> Transaction fee </TransTax>
    <TransDate> Transaction Date & time (YYYYMMDD HHMMSS) </TransDate>
    <OrderId> OrderId </OrderId>
    <Status> 1 </Status>
  </Payment>
</moneyApi>
```

Request for transfer (transfer possibility check) from a corporate to a private wallet

Parameters	Name	Value
<i>Incoming</i>	type	Type of Request, «transfer-api» - a request for transfer from one account to another
	mtime	The current date and time in milliseconds
	ApiName	API name
	ApiHash	Hash from the API name, the secret word and the current time
	ApiSess	Session ID UserSession
	OrderId	Number of transfer document
	OrderDesc	Transfer purpose
	PayeePhone	Recipient
	PayeeCurrency	Currency of the recipient
	Amount	Transfer amount
	AmountType	Transfer currency type (0-in sender's currency, 1-in recipient's currency)
	VerifyOld	Query type (1 – activates OrderID coincidence check, new transaction will not be executed if the transfer with this OrderId has previously been made, 0 – deactivates coincidence check)
	Status	Query type (0-test, 1 – transfer)
<i>Outgoing</i>	ErrorId	Error code
	ErrorDesc	Error code description
	TransId	Transaction ID
	PayerAmount	The amount of transfer from the sender
	PayerCurrency	Sender's currency
	TransTax	Transfer fee
	TransDate	Transaction Date & time (YYYYMMDD HHMMSS)
	OrderId	Number of the transfer document
	MinSum	The minimum transfer amount (0,39 UAH for Ukraine)
	MaxSum	The maximum possible amount for the transfer for the given user
	TransSum	The maximum amount of the transfer within the system (8000 UAH for Ukraine)
	Status	Query type (0-test, 1 – transfer)

The sequence of queries.

1. To perform transfer transaction, it's necessary to send a query of "transfer-api" type and with parameter "status"=0 (check for possibility to make a payment); the query transfers transaction details. Response will provide information on the minimum and the possible transfer amount.
2. After the previous query, "transfer-api" query is sent with parameter "status"=1; as a response to this query, transaction ID is received, which will be used for transaction confirmation.
3. If an error occurs in the previous request, "transfer-api" query is used to check the status of the transaction, which indicates the transaction ID, and transaction status comes as a response to this query
4. If you have not received a response to the request (violation of the packet, etc.), we recommend that you request a review of the payment status - "status-api"

Query

```

<monexyApi type="transfer-api" mtime="124863913146052">
  <Auth>
    <ApiName>test</ApiName>
    <ApiHash>961fdc35a590cad468305140fa68bf63a9e9fcb6</ApiHash>
  </Auth>
  <Transfer>
    <OrderId>BS-5434</OrderId>
    <OrderDesc>Teapot № s1-6785</OrderDesc>
    <PayeePhone>380672301119</PayeePhone>
    <Amount>0.40</Amount>
    <AmountType>1</AmountType>
    <PayeeCurrency>UAH</PayeeCurrency>
    <VerifyOID>1</VerifyOID>
    <Status>0</Status>
  </Transfer>
</monexyApi>

```

Response (if parameter <Status>0</Status>)

```

<monexyApi type="transfer-api" mtime="125967703228626">
  <Auth>
    <ApiName>test</ApiName>
    <ApiHash>ac3586873101529d77030cd6b728f67f817ac451</ApiHash>
  </Auth>
  <Error ErrorId="0">
    <ErrorDesc></ErrorDesc>
  </Error>
  <Transfer>
    <TransId>Blank</TransId>
    <PayeeAmount>0.40</PayeeAmount>
    <PayeeCurrency>UAH</PayeeCurrency>
    <PayerAmount>0.40</PayerAmount>
    <PayerCurrency>UAH</PayerCurrency>
    <TransTax>0.01</TransTax>
    <TransDate>Blank</TransDate>
    <OrderId>BS-5434</OrderId>
    <Status>0</Status>
  </Transfer>
  <Variable>
    <MinSum>0.3900</MinSum>
    <MaxSum>7998.0900</MaxSum>
    <TransSum>8000.0000</TransSum>
  </Variable>
</monexyApi>

```

Response (if parameter <Status>1</Status>)

```

<monexyApi type="transfer-api" mtime="125967703228626">
  <Auth>
    <ApiName>test</ApiName>
    <ApiHash>ac3586873101529d77030cd6b728f67f817ac451</ApiHash>
  </Auth>
  <Error ErrorId="0">
    <ErrorDesc></ErrorDesc>
  </Error>
  <Transfer>
    <TransId>1001001316</TransId>
    <PayeeAmount>0.40</PayeeAmount>
    <PayeeCurrency>UAH</PayeeCurrency>
    <PayerAmount>0.40</PayerAmount>
    <PayerCurrency>UAH</PayerCurrency>
    <TransTax>0.01</TransTax>
    <TransDate>20111205 152307</TransDate>
    <OrderId>BS-5434</OrderId>
    <Status>0</Status>
  </Transfer>
  <Variable>
    <MinSum>0.3900</MinSum>
    <MaxSum>7998.0900</MaxSum>
    <TransSum>8000.0000</TransSum>
  </Variable>
</monexyApi>

```

Error

```

monexyApi type="transfer-api" mtime="132379208130241">
  <Auth>
    <ApiName>test</ApiName>
    <ApiHash>003e720e2db00f8032a7de0cf12f3dbd9dc2acf0</ApiHash>
  </Auth>
  <Error ErrorId="152">
    <ErrorDesc>Incorrect amountt .</ErrorDesc>
  </Error>
</monexyApi>

```

Request for transfer status using an OrderID

Query

```
<monexyApi type="status-api" mtime="13237922229652">
  <Auth>
    <ApiName>test</ApiName>
    <ApiHash>cd167ca259e9e3c795cf569ba793a63b1d0ac787</ApiHash>
  </Auth>
  <Status>
    <OrderId>10003051911</OrderId>
  </Status>
</monexyApi>
```

Response

```
<monexyApi type="status-api" mtime="132379222238477">
  <Auth>
    <ApiName>tests</ApiName>
    <ApiHash>c2e4bbc021ffdb99fe9fca2e332b2d278ba78721</ApiHash>
  </Auth>
  <Error ErrorId="0">
    <ErrorDesc></ErrorDesc>
  </Error> <Status>
    <TransId>0</TransId>
    <TransAmount></TransAmount>
  </Status>
</monexyApi>
```

Note: If TransId = 0, the transfer doesn't exist, if there's a transaction, it's amount and ID will be returned.

Sequence of queries to process payments:

1. It is necessary to authorize sending an authorization request, the query type - "auth".
2. After the first stage, you must perform registration with the system, by sending a query of "login" type; the session ID which will be used during the session, will be transferred in the response.
3. To perform the transfer operation, you should send a query of "payment-req" type with the value of parameter "status" = 0; the query transmits details of the transaction to be processed. In response to the request, a transaction identifier and transaction fees are returned if the transaction is successful..
4. After a previous query, "payment-req" is send with the "Status" parameter = 1; as a result of this query, you'll get a transaction ID that will be used to confirm the transaction..
5. Next, the user who initiated the transaction should confirm processing of this transaction. To do this, it'ss necessary to send a query of "payment-conf" type; this query also transmits a confirmation code received via SMS, and a transaction identifier, obtained from the previous query. After successful completion of the transaction, the user receives a confirmation along with an SMS message with the transaction results.
6. If an error occurs in the previous request, "payment-status" query indicating the transaction ID is used to check the status of the transaction. As a response, we get the transaction status.

Authorization request

A request for SMS sending a code to the specified username (phone number), which will be used in the authentication operation

Parameters	Name	Value
<i>Incoming</i>	type	Type of Request, «auth» - request for authorization
	mtime	The current date and time in milliseconds
	ApiName	API name
	ApiHash	Hash from the API name, the secret word and the current time
	ApiLogin	Username (phone number)
	ApiSess	Session ID UserSession (left blank)
<i>Outgoing</i>	ErrorId	Error code
	ErrorDesc	Error code description

Query

```
<monexyApi type="auth" mtime="DateTimeMS">
  <Auth>
    <ApiName> API name </ApiName>
    <ApiHash> Authentication token </ApiHash>
    <ApiLogin> Login (phone number) </ApiLogin>
    <ApiSess></ApiSess>
  </Auth>
</monexyApi>
```

Response

```
<monexyApi type="auth" mtime="DateTimeMS">
  <Auth>
    <ApiName> API name </ApiName>
    <ApiHash> Authentication token </ApiHash>
  </Auth>
  <Error ErrorId="0">
    <ErrorDesc> Error message </ErrorDesc>
  </Error>
</monexyApi>
```

Authentication request

A request for sending a code to the specified username (phone number), which will be used in the authentication operation

Parameters	Name	Value
<i>Incoming</i>	type	Type of query, «login» - a request for authentication
	mtime	The current date and time in milliseconds
	ApiName	API name
	ApiHash	Hash from the API name, the secret word and the current time
	ApiLogin	Username (phone number)
	PaymentSMS	A code, received by the user in SMS after “auth” query
<i>Outgoing</i>	ErrorId	Error code
	ErrorDesc	Error code description
	ApiSess	Session ID
	ApiBCode	A key with which you can then view this user's balance without re-authorization

Query

```
<monexyApi type="login" mtime="DateTimeMS">
  <Auth>
    <ApiName> API name </ApiName>
    <ApiHash> Authentication token </ApiHash>
    <ApiLogin> Login (phone number)</ApiLogin>
  </Auth>
  <Login>
    <SmsCode>30521</SmsCode>
  </Login>
</monexyApi>
```

Response

```
<monexyApi type="login" mtime="DateTimeMS">
  <Auth>
    <ApiName> API name </ApiName>
    <ApiHash> Authentication token </ApiHash>
  </Auth>
  <Error ErrorId="0">
    <ErrorDesc> Error message </ErrorDesc>
  </Error>
  <Login>
    <ApiSess> UserSession </ApiSess>
    <ApiBCode> UserApiBCode </ApiBCode>
  </Login>
</monexyApi>
```

Request for balance of the user's session

Request for obtaining the balance of all the MoneXy wallets. Note: Depending on your needs send a query using parameter {ApiSess | ApiBCode} (either one parameter or another)

Parameters	Name	Value
<i>Incoming</i>	type	Type of query, «balans» - a request for balance of the user's session
	mtime	The current date and time in milliseconds
	ApiName	API name
	ApiHash	Hash from the API name, the secret word and the current time
	ApiLogin	Username (phone number)
	ApiSess	UserSession ID
<i>Outgoing</i>	ErrorId	Error code
	ErrorDesc	Error code description
	Currency	Currency of balance
	Amount	A balance displayed in above currency

Query

```
<monexyApi type="balans" mtime="DateTimeMS">
  <Auth>
    <ApiName> API name </ApiName>
    <ApiHash> Authentication token </ApiHash>
    <ApiLogin> Login (phone number) </ApiLogin>
    <ApiSess> UserSession </ApiSess>
  </Auth>
</monexyApi>
```

Response

```
<monexyApi type="balans" mtime="DateTimeMS">
  <Auth>
    <ApiName> API name </ApiName>
    <ApiHash> Authentication token </ApiHash>
  </Auth>
  <Error ErrorId="0">
    <ErrorDesc> Error message </ErrorDesc>
  </Error>
  <Balans>
    <Currency id=VAL>
      <Amount>0.0000</Amount>
    </Currency>
    ...
  </Balans>
</monexyApi>
```


Request for balance using a BCode

Request for obtaining the balance of all the MoneXy wallets without SMS

Parameters	Name	Value
<i>Incoming</i>	type	Type of query, «balans-bcode» - a request for balance without SMS
	mtime	The current date and time in milliseconds
	ApiName	API name
	ApiHash	Hash from the API name, the secret word and the current time
	ApiLogin	Username (phone number)
	ApiBCode	UserApiBCode - a key for displaying balance without SMS
<i>Outgoing</i>	ErrorId	Error code
	ErrorDesc	Error code description
	Currency	Currency of balance
	Amount	A balance displayed in above currency

Query

```
<monexyApi type="balans-bcode" mtime="DataTimeMS">
  <Auth>
    <ApiName> API name </ApiName>
    <ApiHash> Authentication token </ApiHash>
    <ApiLogin> Login (phone number)</ApiLogin>
    <ApiBCode> UserApiBCode </ApiBCode>
  </Auth>
</monexyApi>
```

Response

```
<monexyApi type="balans-bcode" mtime="DataTimeMS">
  <Auth>
    <ApiName> API name </ApiName>
    <ApiHash> Authentication token </ApiHash>
  </Auth>
  <Error ErrorId="0">
    <ErrorDesc> Error message </ErrorDesc>
  </Error> <Balans>
    <Currency id=VAL>
      <Amount>0.0000</Amount>
    </Currency>
    ...
  </Balans>
</monexyApi>
```

Request for wallet-to-wallet transfer

Parameters	Name	Value
<i>Incoming</i>	type	Type of Request, «payment-req» - a request for transfer from one wallet to another
	mtime	The current date and time in milliseconds
	ApiName	API name
	ApiHash	Hash from the API name, the secret word and the current time
	ApiLogin	Username (phone number)
	ApiSess	Session ID UserSession
	OrderId	Number of transfer document
	OrderDesc	Transfer purpose
	PayeeLogin	Recipient
	PayeeCurrency	Currency of the recipient
	PayerCurrency	Currency of the sender
	Amount	Transfer amount
	AmountType	Transfer currency type (0-in sender's currency, 1-in recipient's currency)
	Status	Query type (0-test, 1 – transfer)
<i>Outgoing</i>	ErrorId	Error code
	ErrorDesc	Error code description
	PaymentId	Transfer ID
	TransId	Transaction ID
	PayeeAmount	The amount of transfer to the recipient
	PayeeCurrency	Recipient's currency
	PayerAmount	The amount of transfer from the sender
	PayerCurrency	Sender's currency
	TransTax	Transfer fee
	TransDate	Date and time of the transfer
	OrderId	Number of the transfer document
	Status	Query type (0-test, 1 – transfer)

Query

```
<monexyApi type="payment-req" mtime="DateTimeMS">
  <Auth>
    <ApiName> API name </ApiName>
    <ApiHash> Authentication token </ApiHash>
    <ApiLogin> Login (phone number) </ApiLogin>
    <ApiSess> UserSession </ApiSess>
  </Auth>
  <Payment>
    <OrderId> OrderId </OrderId>
    <OrderDesc> OrderDesc </OrderDesc>
    <PayeeLogin> PayeeLogin </PayeeLogin>
    <PayeeCurrency> PayeeCurrency </PayeeCurrency>
    <PayerCurrency> PayerCurrency </PayerCurrency>
    <Amount> Amount </Amount>
    <AmountType> AmountType </AmountType>
    <Status> 1 </Status>
  </Payment>
</monexyApi>
```

Response

```
<monexyApi type="payment-req" mtime="DateTimeMS">
  <Auth>
    <ApiName> API name </ApiName>
    <ApiHash> Authentication token </ApiHash>
  </Auth>
  <Error ErrorId="0">
    <ErrorDesc> Error message </ErrorDesc>
  </Error>
  <Payment>
    <PaymentId> PaymentId </PaymentId>
    <TransId> TransactionID </TransId>
    <PayeeAmount> PayeeAmount </PayeeAmount>
    <PayeeCurrency> PayeeCurrency </PayeeCurrency>
    <PayerAmount> PayerAmount </PayerAmount>
    <PayerCurrency> PayerCurrency </PayerCurrency>
    <TransTax> Transaction fee </TransTax>
    <TransDate> Transaction Date & time (YYYYMMDD HHMMSS) </TransDate>
    <OrderId> OrderId </OrderId>
    <Status> 1 </Status>
  </Payment>
</monexyApi>
```

Transaction history request

Parameters	Name	Value
<i>Incoming</i>	type	Type of Request, «history» - a request for transaction history
	mtime	The current date and time in milliseconds
	ApiName	API name
	ApiHash	Hash from the API name, the secret word and the current time
	ApiLogin	Username (phone number)
	ApiSess	Session ID UserSession
	DateFrom	Transaction history, starting from the date
	DateTo	Transaction history, ending with the date
	Page	Page number
	Listing	Number of records per page
	Currency	Transaction history of the account in specified currency
<i>Outgoing</i>	ErrorId	Error code
	ErrorDesc	Error code description
	ID	Block ID for which the transaction history is displayed
	TransId	Transaction ID
	TransPayer	Payer
	TransPayee	Recipient
	TransSum	The amount of the transaction
	TransBalans	Account balance after the transaction is completed
	TransDesc	Transfer purpose
	TransOrderId	Number of the transfer document
	TransDate	Date and time of the transfer
	Pages	Number of pages
	Count	Number of transactions

Query

```

<monexyApi type="history" mtime="DateTimeMS">
  <Auth>
    <ApiName> API name </ApiName>
    <ApiHash> Authentication token </ApiHash>
    <ApiLogin> Login (phone number)</ApiLogin>
    <ApiSess> UserSession </ApiSess>
  </Auth>
  <History>
    <DateFrom> DateFrom </DateFrom>
    <DateTo> DateTo </DateTo>'
    <Page> page </Page>
    <Listing> rows </Listing> // Кол-во на страницу
    <Currency> Currency </Currency>
  </History>
</monexyApi>

```

Response

```

<monexyApi type="history" mtime="DateTimeMS">
  <Auth>
    <ApiName> API name </ApiName>
    <ApiHash> Authentication token </ApiHash>
  </Auth>
  <Error ErrorId="0">
    <ErrorDesc> Error message </ErrorDesc>
  </Error>
  <History>
    <Trans id=TransactionID>
      <TransId> TransactionID </TransId>
      <TransPayer> Payer </TransPayer>
      <TransPayee> Payee </TransPayee>
      <TransSum> Transaction sum </TransSum>
      <TransBalans> Balance after transaction </TransBalans>
      <TransDesc> Transaction description </TransDesc>
      <TransOrderId> Transaction order ID </TransOrderId>
      <TransDate> Transaction date </TransDate>
    </TransId>
    ...
  </History>
  <Pager>
    <Pages> Page </Pages>
    <Count> Count transactions </Count>
  </Pager>
</monexyApi>

```

MoneXy voucher creation

Query

```
<monexyApi type="vaucher-api" mtime="124833913146052">
  <Auth>
    <ApiName>test</ApiName>
    <ApiHash>961fdc35a590cad460305140fa68bf63a9e9fcb6</ApiHash>
  </Auth>
  <Transfer>
    <Desc>Vaucher 0.10</Desc>
    <OrderId>325523</OrderId>
    <Amount>0.10</Amount>
    <VaucherType>1</VaucherType>
    <Status>1</Status>
  </Transfer>
</monexyApi>
```

Response

```
<monexyApi type="vaucher-api" mtime="129804179717337">
  <Auth>
    <ApiName>test</ApiName>
    <ApiHash>6daf65cce773062901dfe58283d62135d7e5718</ApiHash>
  </Auth>
  <Error ErrorId="0">
    <ErrorDesc></ErrorDesc>
  </Error>
  <Transfer>
    <TransId>1000218269</TransId>
    <PayeeAmount>0.10</PayeeAmount>
    <PayeeCurrency>UAH</PayeeCurrency>
    <PayerAmount>0.10</PayerAmount>
    <PayerCurrency>UAH</PayerCurrency>
    <TransTax></TransTax>
    <TransDate>20110218 150957</TransDate>
    <OrderId></OrderId>
    <Status>1</Status>
  </Transfer>
  <Vaucher>
    <Number>123123123412312</Number>
    <Pass>4321</Pass>
  </vaucher>
</monexyApi>
```

Transfer from MoneXy voucher to user wallet

Query

```
<monexyApi type="transfer" mtime="134289751949">
  <Auth>
    <ApiName>test</ApiName>
    <ApiHash>a162a0b996e433923ccf383e9e49dfca0d9c2</ApiHash>
  </Auth>
  <Transfer>
    <OrderId>13394</OrderId>
    <PayeeCurrency>USD</PayeeCurrency>
    <PayerCurrency>USD</PayerCurrency>
    <PayeePhone>380971234567</PayeePhone>
    <Amount>1</Amount>
    <AmountType>1</AmountType>
    <OrderDesc>Cash-in 1 USD to the phone number 380971234567</OrderDesc>
    <PayerCard>436220214474499</PayerCard>
    <PayerPass>1234</PayerPass>
    <Status>1</Status>
  </Transfer>
</monexyApi>
```

Response

```
<monexyApi type="transfer" mtime="137289555716">
  <Auth>
    <ApiName>test</ApiName>
    <ApiHash>c0b37e873471a487dd39c8ec41577790784</ApiHash>
  </Auth>
  <Error ErrorId="0">
    <ErrorDesc></ErrorDesc>
  </Error>
  <Transfer>
    <TransId>1005782925</TransId>
    <PayeeAmount>1.00</PayeeAmount>
    <PayeeCurrency>USD</PayeeCurrency>
    <PayerAmount>1.00</PayerAmount>
    <PayerCurrency>USD</PayerCurrency>
    <TransTax></TransTax>
    <TransDate>20120910 150916</TransDate>
    <OrderId>13394</OrderId>
    <Status>1</Status>
  </Transfer>
</monexyApi>
```

Error

```
<monexyApi type="transfer" mtime="133792813041">
  <Auth>
    <ApiName>test</ApiName>
    <ApiHash>003e20e2db00f803a7de0ef12fdbd9e2aef0</ApiHash>
  </Auth>
  <Error ErrorId="152">
    <ErrorDesc>Incorrect amount.</ErrorDesc>
  </Error>
</monexyApi>
```

Transfer from MoneXy voucher to a corporate wallet

Query

```
<monexyApi type="transfer" mtime="131652335771971">
  <Auth>
    <ApiName>test</ApiName>
    <ApiHash>6f6df74a095ca34dd5864656f0abe18bd150c6ec</ApiHash>
  </Auth>
  <Transfer>
    <OrderId>BS-5434</OrderId>
    <OrderDesc>Teapot № s1-6785</OrderDesc>
    <PayeeCard>recipient's wallet number</PayeeCard>
    <PayerCard>4022*****5924</PayerCard> voucher number
    <PayerPass>7**4</PayerPass> password
    <Amount>0.10</Amount>
    <AmountType>1</AmountType>
    <PayeeCurrency>UAH</PayeeCurrency>
    <Status>0</Status>
  </Transfer>
</monexyApi>
```

Response

```
<monexyApi type="transfer" mtime="132376548942891">
  <Auth>
    <ApiName>test</ApiName>
    <ApiHash>6f1ffccc6ab2c205a798765d2ab2cef85</ApiHash>
  </Auth>
  <Error ErrorId="0">
    <ErrorDesc></ErrorDesc>
  </Error>
  <Transfer>
    <TransId></TransId>
    <PayeeAmount>0.50</PayeeAmount>
    <PayeeCurrency>UAH</PayeeCurrency>
    <PayerAmount>0.50</PayerAmount>
    <PayerCurrency>UAH</PayerCurrency>
    <TransTax></TransTax>
    <TransDate></TransDate>
    <OrderId>BS-5434</OrderId>
    <Status>0</Status>
  </Transfer>
  <Variable>
    <MinSum>0.3900</MinSum>
    <MaxSum>0.0000</MaxSum>
    <TransSum>8000.0000</TransSum>
  </Variable>
</monexyApi>
```

Checking balance of MoneXy voucher

Query

```
<monexyApi type="balans-card" mtime="131653010419314">  
  <Auth>  
    <ApiName>test</ApiName>  
    <ApiHash>2b1b4d7a983ad61cd4031999ce4037ee7bec8bbf</ApiHash>  
  </Auth>  
  <Balans>  
    <CardNumber>4574*****168</CardNumber>  
    <CardPass>5**7</CardPass>  
  </Balans>  
</monexyApi>
```

Response

```
<monexyApi type="balans-card" mtime="132345662813671">  
  <Auth>  
    <ApiName>test</ApiName>  
    <ApiHash>3923ece782027a26bb12345796b296681c8bca48</ApiHash>  
  </Auth>  
  <Error ErrorId="0">  
    <ErrorDesc></ErrorDesc>  
  </Error>  
  <Balans>  
    <Currency>UAH</Currency>  
    <Amount>3828.2345</Amount>  
  </Balans>  
</monexyApi>
```

Balance of a corporate wallet linked to an API (merchant)

Query

```
<monexyApi type="balans-card-api" mtime="131653395739355">  
  <Auth>  
    <ApiName>test</ApiName>  
    <ApiHash>049a0548fb0db547533df92f016a97551a7a9464</ApiHash>  
  </Auth>  
</monexyApi>
```

Response

```
<monexyApi type="balans-card-api" mtime="132343450035115">  
  <Auth>  
    <ApiName>test</ApiName>  
    <ApiHash>c1cf31f820b0148c34567c745b0e7d137e6f1f71</ApiHash>  
  </Auth>  
  <Error ErrorId="0">  
    <ErrorDesc></ErrorDesc>  
  </Error>  
  <Balans>  
    <Currency>UAH</Currency>  
    <Amount>19.5800</Amount>  
  </Balans>  
</monexyApi>
```

Balance of a corporate wallet linked to an API (distributor)

Query

```
<monexyApi type="balans-card-api" mtime="131653395739355">  
  <Auth>  
    <ApiName>test</ApiName>  
    <ApiHash>049a0548fb0db547533df92f016a97551a7a9464</ApiHash>  
  </Auth>  
</monexyApi>
```

Response

```
<monexyApi type="balans-card-api" mtime="132343450035115">  
  <Auth>  
    <ApiName>test</ApiName>  
    <ApiHash>c1cf31f820b0148c34567c745b0e7d137e6f1f71</ApiHash>  
  </Auth>  
  <Error ErrorId="0">  
    <ErrorDesc></ErrorDesc>  
  </Error>  
  <Balans>  
    <Currency>UAH</Currency>  
    <Amount>19.5800</Amount>  
  </Balans>  
</monexyApi>
```

Checking the status of payment (or voucher creation) using an unique OrderID

Query

```
<monexyApi type="status-api" mtime="" . $mkttime . "">
  <Auth>
    <ApiName>test</ApiName>
    <ApiHash>' . $ApiHASH . '</ApiHash>
  </Auth>
  <Status>
    <OrderId>4634611</OrderId>
  </Status>
</monexyApi>
```

Response

```
<monexyApi type="status-api" mtime="132343176376470">
  <Auth>
    <ApiName>test</ApiName>
    <ApiHash>881b3679b33cceba31766ae8a6e4f2f793ef5193</ApiHash>
  </Auth>
  <Error ErrorId="0">
    <ErrorDesc></ErrorDesc>
  </Error>
  <Status>
    <TransId>0</TransId>
    <TransAmount></TransAmount>
  </Status>
  <Vaucher>
    <Number>63406347604643666</Number>
    <Pass>4634</Pass>
  </Vaucher>
</monexyApi>
```

If TransId = 0, the payment doesn't exist, if there's a payment, its amount and transaction ID will be returned

Request for transaction cancellation

Parameters	Name	Value
<i>Incoming</i>	type	Type of Request, ««transfer-api-return»»
	mtime	The current date and time in milliseconds
	ApiName	API name
	ApiHash	Hash from the API name, the secret word and the current time
	TransId	Transaction ID
	AddDesc	Reasons for cancellation
	MaxTime	The time limit for a cancellation
	Status	0 – checking possibility for cancellation, 1 - cancellation
<i>Outgoing</i>	ErrorId	Error code
	ErrorDesc	Error code description
	TransId	Transaction ID
	TaxId	ID of transaction subject to a transactional fee (if there was any)
	ReturnTransId	ID of transaction subject to a transactional fee after cancellation
	ReturnTaxId	ID of transaction subject to a transactional fee after cancellation (if there was a fee)
	Status	0 – checking possibility for cancellation, 1 - cancellation

Query

```
<monexyApi mtime="1324552617257" type="transfer-api-return">
  <Auth>
    <ApiName>test</ApiName>
    <ApiHash>da25ce05258bc82086ca1fe769a9f97e78e2f5</ApiHash>
  </Auth>
  <Transfer>
    <TransId>1001222725</TransId>
    <AddDesc> Description of reasons for the cancellation </AddDesc>
    <MaxTime>15</MaxTime>
    <Status>1</Status>
  </Transfer>
</monexyApi>
```

Query

```
<monexyApi type="transfer-api-return" mtime="1324551625488">
  <Auth>
    <ApiName>test</ApiName>
    <ApiHash>df353a07ac65a94850c015524efdc502e78f2a</ApiHash>
  </Auth>
  <Error ErrorId="0">
    <ErrorDesc></ErrorDesc>
  </Error>
  <Transfer>
    <TransId>1001275648</TransId>
    <TaxId></TaxId>
    <ReturnTransId>1001276067</ReturnTransId>
    <ReturnTaxId></ReturnTaxId>
    <Status>1</Status>
  </Transfer>
</monexyApi>
```

Error

```
<monexyApi type="transfer-api-return" mtime="1324553451649">
  <Auth>
    <ApiName>test</ApiName>
    <ApiHash>42f5982e14c9c79443dc82dee140de4c02d6ac</ApiHash>
  </Auth>
  <Error ErrorId="850">
    <ErrorDesc>TransId error</ErrorDesc>
  </Error>
</monexyApi>
```