Yandex.Money Merchant Web Services

Version 3.0 from 05.08.2013

CONTENTS

[1. Overview 2](#_Toc335310486)

[2. Counterparty connection parameters 3](#_Toc335310487)

[2.1. Counterparty Certificate 3](#_Toc335310488)

[3. Order statements 3](#_Toc335310489)

[3.1. Orders statement request (listOrders) 3](#_Toc335310490)

[3.1.1. Request parameters 3](#_Toc335310491)

[3.1.2. Response format 4](#_Toc335310492)

[3.2. Refunds statement request (listReturns) 9](#_Toc335310493)

[3.2.1. Request parameters 9](#_Toc335310494)

[3.2.2. Response format 9](#_Toc335310495)

[4. Financial operations 11](#_Toc335310496)

[4.1. Request format 11](#_Toc335310497)

[4.2. Response format 12](#_Toc335310498)

[4.3. Refund operation request (returnPayment) 12](#_Toc335310499)

[4.3.1. Refund operation requests processing rules 13](#_Toc335310500)

[5. Data Types 13](#_Toc335310501)

[6. Error codes 15](#_Toc335310502)

[7. Funds transfer notification status codes 17](#_Toc335310503)

*Last updated: 13.06.2013*

# Overview

Yandex.Money payment service Operator (hereinafter - “Operator”) provides interface for secure interactions through the Internet between merchants (hereinafter - “Counterparty” and Operator, called “Merchant Web Services” (hereinafter – MWS).

MWS provides two types of queries:

* Informational, receipt of orders and statements, etc.
* Financial operations, full or partial refunds

Communication channel is protected by SSL. Counterparty has to obtain the client certificate for MWS authentication. Query result is returned in response to an HTTP request.

Table 1 Possible HTTP response codes

|  |  |
| --- | --- |
| **HTTP Code** | **Description** |
| 200 OK | Request has accepted for processing. Response sent in accordance with this protocol. |
| 400 Bad Request | Request has refused. Request body has corrupted, server unable to parse client request.  Possible reasons:   * Unparseable request * Wrong MIME-type (Content-Type header value) |
| 403 Forbidden | Counterparty has no access rights to execute the operation requested |
| 404 Not Found | Invalid operation URI in request, either at the moment the operation is not available. |
| 500 Internal Server Error | Server has technical errors. Contact to technical support. |
| 501 Not Implemented | HTTP request sent with method different from POST. |

Table Common response parameters

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Type** | **Description** |
| status | xs:int | Operation execution result, shows operation status. By the value of this field, Counterparty should make decision on the status of the operation. See. Table 3 Request Status Codes |
| error | xs:int | Error code (see. Table 12 Funds transfer notification status codes). Additional description to the status field. Optional field. |
| clientOrderId | ClientTransactionNumber | Copy of the clientOrderId request parameter. |
| processedDT | xs:dateTime | Request processed timestamp by Operator time. |
| techMessage | xs:string | Optional field. May contain additional descriptive text to the server's response. This text contains technical information and should not be displayed in any user interface. |

Table Request Status Codes

|  |  |
| --- | --- |
| **Status Code** | **Description** |
| **0** | **Success.** Final state. Request has processed successfully. |
| **1** | **In Progress.** Request processing in progress. Returns if request processing has timed out. Counterparty is required to repeat the request to refine the final status. |
| **3** | **Refused.** Final state. Request has refused. The reason of refusal shown in the “error” parameter. |

# Counterparty connection parameters

## Counterparty Certificate

Counterparty has to obtain an X.509 certificate for SSL client authentication within MWS. See document «Certificate exchange procedure».

# Order statements

## Orders statement request (listOrders)

This query yields a list of orders and their properties, request parameters described in Table 4 listOrders request parameters.

Operation address: <https://server:port/webservice/mws/api/listOrders>

### Request parameters

Counterparty creates a POST-query using HTTP/1.1 protocol (<http://www.ietf.org/rfc/rfc2616.txt>, <http://www.ietf.org/rfc/rfc2618.txt> ). Request parameters have to be packed into parameters of a POST request as “name-value” pairs. MIME-type: application/x-www-form-urlencoded. Character encoding is UTF-8.

Some parameters are optional, if missing they use the default value. Empty value is not equal to default value. If the mandatory parameter is missing, an error “Invalid parameter value NNNN” will be occurred.

Table listOrders request parameters

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Data Type** | **Description** |
| requestDT | xs:dateTime | Request timestamp by Counterparty time. Mandatory parameter. |
| outputFormat | xs:normalizedString | Output format, possible values is «XML» or «CSV». Optional parameter, default value is XML. |
| csvDelimiter | xs:string, 1 character | CSV format values delimiter. Cannot be «”» (quote). Optional parameter, default value is «;». |
| shopId | xs:long | Counterparty’s ID assigned by Operator. If not specified, orders of all shopId’s allowed to this MWS user will be shown. |
| orderCreatedDatetimeGreaterOrEqual | xs:dateTime | The lower boundary of the order creation time. Selects orders which is creation time is greater or equal to the parameter value. |
| orderCreatedDatetimeLessOrEqual | xs:dateTime | The upper boundary of the order creation time. Selects orders which is creation time is less or equal to the parameter value. |
| paid | xs:boolean | If the value is “true”, then returned only paid orders. If the value is “false”, then returned only non-paid orders. If not specified, then returned as paid and non-paid orders. |
| paymentDatetimeGreaterOrEqual | xs:dateTime | Lower boundary of the order payment time. Selects orders, which is payment time is greater or equal to the parameter value. |
| paymentDatetimeLessOrEqual | xs:dateTime | Upper boundary of the order payment time. Selects orders, which is payment time is less or equal to the parameter value. |
| invoiceId | xs:long | Unique transaction number of the order. |
| orderNumber | xs:string, up to 64 characters | Order number in Counterparty’s system. Usage is possible when Counterparty specifies orderNumber on the payment form. |
| outputFields | xs:string, up to 4000 characters | List of order properties to be shown in the output. Value delimiter is «;».  Default value:  «shopId;shopName;articleId;articleName;invoiceId;orderNumber;paymentSystemOrderNumber; customerNumber; createdDatetime;paid;orderSumAmount; orderSumCurrencyPaycash;orderSumBankPaycash;paidSumAmount; paidSumCurrencyPaycash;paidSumBankPaycash;receivedSumAmount; receivedSumCurrencyPaycash;receivedSumBankPaycash; shopSumAmount;shopSumCurrencyPaycash;shopSumBankPaycash; paymentDatetime;paymentAuthorizationTime;payerCode; payerAddress;payeeCode;paymentSystemDatetime; avisoReceivedDatetime;avisoStatus».  See the full list of order properties in the Table 6 Order properties may be requested |

At least one of the following conditions must be specified in the request parameters:

* unique transaction number (invoiceId) and Counterparty’s ID (shopId);
* Counterparty’s order number (orderNumber) and Counterparty’s ID (shopId);
* order creation time interval (orderCreatedDatetimeGreaterOrEqual and/or orderCreatedDatetimeLessOrEqual);
* funds transfer time interval (paymentDatetimeGreaterOrEqual and/or paymentDatetimeLessOrEqual).

When querying for transaction number (invoiceId) or Counterparty’s order number (orderNumber), the Counterparty ID (shopID) must be specified.

These rules are applicable for queries by time interval (order creation and payment time):

* If only one time boundary specified, the second boundary takes the default value. Default value for upper boundary is current Operator time (“now”), for lower boundary is current Operator time minus one day (“now – 24hours”);
* If order payment time specified, request parameters must contain parameter “paid” with value “true”;
* Maximum allowed time interval is 31 days.

### Response format

Response format is defined by request parameters outputFormat and csvDelimiter. In case of error, error code and description will be returned in the specified response format.

Example 1 Response about an error in XML format

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

<listOrdersResponse

status="3" error="111"

processedDT="2011-07-02T20:38:01.000+04:00"

techMessage="Invalid value for the requestDT"

/>

Example 2 Response about an error in CSV format

|  |
| --- |
| status=3;error=111;processedDT=2011-07-21T13:20:14.869+04:00;"techMessage=Invalid value for the requestDT" |

If the listOrders result contains more than 10 000 records, the error 216 will be returned. If case of error 216, Counterparty have to change request parameters to reduce the time interval or specify additional conditions.

Table 5Additional response header properties

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Description** | **Data Type** | **Comments** |
| orderCount | Number of records in response. | xs:int | Present on success of listOrders request. |

Full list of the order properties may be requested by listOrders described below.

Table 6 Order properties may be requested

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Description** | **Data Type** | **Comments** |
| shopId | Counterparty ID assigned by Operator | xs:long |  |
| articleId | Article ID assigned by Operator | xs:long |  |
| shopName | Counterparty Title in Yandex.Money system | xs:string, up to 64 chars |  |
| articleName | Article Title in Yandex.Money system | xs:string, up to 128 chars |  |
| invoiceId | Unique transaction number which identifies the order. | xs:long |  |
| orderNumber | Order number in Counterparty system | xs:string, up to 64 chars | Available when Counterparty specifies orderNumber on payment form, otherwise this property contains invoiceId value. |
| paymentSystemOrderNumber | Funds transfer ID, in Yandex.Money system | xs:string, up to 40 chars | Available only for paid orders. |
| customerNumber | Customer ID in Counterparty system.  Cell phone number, service contract number etc. Specified by Counterparty. | xs:string, up to 64 chars |  |
| createdDatetime | Order registration time on Operator side | xs:dateTime |  |
| paid | «true» if order paid, otherwise «false» | xs:boolean |  |
| paymentDatetime | Funds transfer time. | xs:dateTime | Available only for paid orders. |
| paymentAuthorizationTime | Funds transfer ID, in Yandex.Money system | xs:long | Available only for paid orders. |
| payerCode | Payer account | xs:string, до 33 символов | Available only for paid orders. |
| payerAddress | Payer IP-address, if determined | xs:string, до 33 символов | Available only for paid orders. |
| payeeCode | Counterparty’s account | xs:string, до 33 символов | Available only for paid orders. |
| paymentSystemDatetime | Funds transfer time. | xs:dateTime | Available only for paid orders. |
| avisoReceivedDatetime | Funds transfer notification received by Counterparty timestamp. | xs:dateTime | Available when Counterparty has received a funds transfer notification. |
| avisoStatus | Funds transfer notification status | xs:int | See Table 12 Funds transfer notification status codes |
| avisoRegistryId | Register number which contains this order | xs:long |  |
| orderSumAmount | Amount of the order | CurrencyAmount |  |
| orderSumCurrencyPaycash | Currency code for order amount. | CurrencyCode |  |
| orderSumBankPaycash | Yandex.Money Processing Center code for order amount. | CurrencyBank |  |
| contractAmount | Amount to be paid in payer’s account currency | CurrencyAmount |  |
| contractCurrency | Currency code for contractAmount | CurrencyCode |  |
| paidSumAmount | Amount actually paid by payer | CurrencyAmount | Available only for paid orders. |
| paidSumCurrencyPaycash | Currency code for paidSumAmount | CurrencyCode | Available only for paid orders. |
| paidSumBankPaycash | Processing Center code for paidSumAmount | CurrencyBank | Available only for paid orders. |
| shopSumAmount | Net amount has received by Counterparty after Operator commission fees. | CurrencyAmount |  |
| shopSumCurrencyPaycash | Currency code for shopSumAmount | CurrencyCode |  |
| shopSumBankPaycash | Processing Center code shopSumAmount | CurrencyBank |  |
| receivedSumAmount | Amount has received by Operator from payer. | CurrencyAmount | Available only for paid orders. |
| receivedSumCurrencyPaycash | Currency code for receivedSumAmount | CurrencyCode | Available only for paid orders. |
| receivedSumBankPaycash | Payment Processing Center code for receivedSumAmount | CurrencyBank | Available only for paid orders. |
| paymentFormParams | Payment Form parameters | xs:string | Available only to Counterparties with “store payment form parameters” option. |
| paymentType | Payment method | PaymentType |  |

If CSV output format has requested, response document header and body will be set of string rows, delimited with carriage return/line feed symbols (\r\n). The first row of response contains request execution result status value. The second row contains list of names of properties in order that the manner in which the subsequent lines will contain the value of the order property. Each row of response document, starting from the third, contains the properties of a single order.

Order properties values delimited by symbol specified by request parameter csvDelimiter. If order property value contains the delimiter symbol (specified by csvDelimiter) or quote symbol (“), the whole value will be escaped by quotes (“) and quote symbols within value will be double-quoted (“”).

If the order property value is not available (payment time for unpaid order for example), the corresponding CSV field will be empty.

Example 3 Response example in CSV format

(with default order properties set, the long strings has been wrapped to fit into the document)

|  |
| --- |
| status=0;error=0;processedDT=2011-08-02T18:28:08.541+04:00;orderCount=2 |
| shopId;articleId;invoiceId;orderNumber;paymentSystemOrderNumber;customerNumber;createdDatetime;paid;orderSumAmount;orderSumCurrencyPaycash;orderSumBankPaycash;paidSumAmount;paidSumCurrencyPaycash;paidSumBankPaycash;receivedSumAmount;receivedSumCurrencyPaycash;receivedSumBankPaycash;shopSumAmount;shopSumCurrencyPaycash;shopSumBankPaycash;paymentDatetime;paymentAuthorizationTime;payerCode;payerAddress;payeeCode;paymentSystemDatetime;avisoReceivedDatetime;avisoStatus;avisoRegistryId |
| 1;2;2000000294394;12345670;;;2011-08-02T18:28:08.269+04:00;false;2.08;10643;1003;;;;;;; 2.00;10643;1003;;;;;;;;;; |
| 2;3;2000000294386;12345671;334074060091013004;;2011-08-02T18:20:58.562+04:00;true;350.00;10643;1002;350.00;10643;1003;350.00;10643;1003;300.00;10643;1002; 2011-08-02T18:21:00.091+04:00;334074060091013004;41003321020;192.168.1.127;410031234567; 2011-08-02T18:21:00.091+04:00;2011-08-02T18:27:59.448+04:00;81; |

If order property value is not available (payment time for unpaid order for example), the corresponding XML attribute will be absent.

Example 4 Response example in XML format

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <listOrdersResponse status="0" error="0"  processedDT="2011-08-02T20:38:01.000+04:00"  orderCount="2">  <order shopId="1" articleId="2" invoiceId="2000000294393"  shopName="ЗАО &quot;Мобильник&quot;"  articleName="Оплата мобильного телефона"  orderNumber="12345678"  paymentSystemOrderNumber="334074426144011004"  createdDatetime="2011-08-02T18:27:05.568+04:00"  paid="true"  paymentDatetime="2011-08-02T18:27:06.144+04:00"  paymentAuthorizationTime="334074426144011004"  payerCode="41003122233"  payerAddress="192.168.1.127"  payeeCode="41003987654"  paymentSystemDatetime="2011-08-02T18:27:06.144+04:00"  avisoReceivedDatetime="2011-08-02T18:27:06.144+04:00"  avisoStatus="1000"  orderSumAmount="2.08"  orderSumCurrencyPaycash="643"  orderSumBankPaycash="1001"  paidSumAmount="2.08"  paidSumCurrencyPaycash="643"  paidSumBankPaycash="1001"  shopSumAmount="2.07"  shopSumCurrencyPaycash="643"  shopSumBankPaycash="1001"  receivedSumAmount="2.07"  receivedSumCurrencyPaycash="643"  receivedSumBankPaycash="1001"  />  <order shopId="2" articleId="3" invoiceId="2000000294394"  shopName="ЗАО &quot;Виртуальная реальность&quot;"  articleName="Футбол навсегда"  orderNumber="12345679"  createdDatetime="2011-08-02T18:28:08.269+04:00"  paid="false"  orderSumAmount="2.08"  orderSumCurrencyPaycash="643"  orderSumBankPaycash="1001"  shopSumAmount="2.07"  shopSumCurrencyPaycash="643"  shopSumBankPaycash="1001"  />  </listOrdersResponse> |

Payment form parameters within order properties display is specified by outputFormat request parameter:

* In CSV format, payment form parameters put into paymentFormParams order property as set of “key=value” pairs, delimited by semicolon (;). If value have quotes (“), the whole value will be escaped with double quotes (“”);
* In XML format, the payment form parameters put into child XML elements of XML element paymentFormParams. For each payment form parameter, the XML element have two attributes – “key” contains parameter name, “val” contains parameters value.

Example 5 Payment form parameters in CSV format

|  |
| --- |
| “PROPERTY1=123;PROPERTY2=1234567;Sum=10.00;NetSum=0;ShopArticleId=2;ShowCaseId=99901; BankId=1003;CurrencyId=10643;TargetBankId=1003;TargetCurrencyId=10643; BuyButton=Оплатить;””ArticleName=””””Оплата мобильного телефона””””””” |

Example 6 Payment form parameters in XML format

|  |
| --- |
| <paymentFormParams>  <param key="Sum" val="10.00"/>  <param key="NetSum" val="0"/>  <param key="ShopArticleId" val="3"/>  <param key="PROPERTY1" val="123"/>  <param key="PROPERTY2" val="1234567"/>  </paymentFormParams> |

## Refunds statement request (listReturns)

This query yields a list of operations of refunds, request parameters described in Table 7 listReturns request parameters. See also Refund operation request (returnPayment).

Operation address: <https://server:port/webservice/mws/api/listReturns>

### Request parameters

Counterparty creates POST-query using HTTP/1.1 protocol (<http://www.ietf.org/rfc/rfc2616.txt>, <http://www.ietf.org/rfc/rfc2618.txt> ). Request parameters are packed into parameters of POST request as the “name-value” pairs. MIME-type: application/x-www-form-urlencoded. Character encoding is UTF-8.

Some parameters are optional, if missing they use the default value. Empty value is not equal to default value. If the mandatory parameter is missing, an error “Invalid parameter value NNNN” is occurred.

Table listReturns request parameters

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Data Type** | **Description** |
| requestDT | xs:dateTime | Request timestamp by current Counterparty time. Mandatory parameter. |
| shopId | xs:long | Counterparty ID, assigned by Operator |
| from | xs:dateTime | Lower time boundary, list FROM time |
| till | xs:dateTime | Upper time boundary, list TILL time |
| status | xs:int | List operations only with status (Optional parameter) |
| partial | xs:boolean | Optional parameter with possible values:   * true – only partial refunds will be listed * false – only refunds with full order amount will be listed * parameter absent – all refund operations will be listed   Default value: absent |
| outputFormat | xs:normalizedString | Output format, can be “XML” or “CSV”. Default value: XML |
| csvDelimiter | xs:string, 1 символ | CSV format values delimiter. Cannot be «”» (quote). Optional parameter, default value is «;». |

**Note:** request parameters “from” and “till” are applied to “createdDT” field (refund request registration time on Operator side).

### Response format

Response format is defined by request parameters outputFormat and csvDelimiter. In case of error, error code and description will be returned in specified response format.

Table 8 Refund operation properties shown in response

|  |  |  |
| --- | --- | --- |
| **Property** | **Data Type** | **Description** |
| returnId | xs:long | Unique operation ID, assigned by Operator |
| invoiceId | xs:long | Source order transaction number |
| shopId | xs:long | Counterparty ID, assigned by Operator |
| amount | CurrencyAmount | Amount to refund |
| currency | CurrencyCode | Currency code for amount |
| cause | xs:string, up to 255 chars | Refund cause description |
| status | xs:int | Operation status |
| error | xs:int | Operation error code |
| createdDT | xs:dateTime | Refund operation request registration time on Operator side |
| processedDT | xs:dateTime | Actual time of refund operation. Present only on success. |
| sender | xs:string | Refund order sender name. Typically contains CN field of X.509 certificate has used to make digital signature on refund order request. |
| articleAmount | CurrencyAmount | Article amount to refund |
| articleCurrency | CurrencyCode | Currency code for article amount |
| orderNumbe | xs:string, up to 64 chars | Order number in Counterparty system |

Example 7 Success response example in XML format

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

<listReturnsResponse

status="0" error="0"

processedDT="2011-07-02T20:38:01.000Z">

<returnPayment

returnId="123"

status="0" error="0"

invoiceId="2000000123"

shopId="6689"

amount="10.00"

currency="643"

createdDT="2011-07-02T20:38:01.000Z"

processedDT="2011-07-02T20:38:01.000Z"

cause="the buyer refused to accept goods"

sender="shopName"

articleAmount="10.00"

articleCurrency="643"

orderNumber="12345"

/>

<returnPayment

returnId="124"

status="3" error="506"

invoiceId="2000000125"

shopId="6689"

amount="12.00"

currency="643"

createdDT="2011-07-02T20:38:01.000Z"

cause="Buyer has refused to accept goods"

sender="shopName"

articleAmount="12.00"

articleCurrency="643"

orderNumber="12346"

/>

</listReturnsResponse>

Example 8 Success response example in CSV format

status=0;error=0;processedDT=2011-07-02T20:38:01.000Z

123;0;0;2000000123;6689;10.00;643;2011-07-02T20:38:01.000Z;2011-02-02T20:38:01.000Z;"Buyer has refused to accept goods";shopName;10.00;643;12345

124;3;506;2000000125;6689;12.00;643;2011-07-02T20:38:01.000Z;;"the buyer refused to accept the goods";shopName;10.00;643;12346

**Note:** if operation property value is not available the corresponding CSV field will be empty.

Example 9 Response example on error in XML format

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

<listReturnsResponse

status="3" error="113"

processedDT="2011-07-02T20:38:01.000Z"/>

Example 10 Response example on error in CSV format

status=3;error=113;processedDT=2011-07-02T20:38:01.000Z

# Financial operations

Financial operations are designed to perform actions with payments, such as full or partial refund of successful order to the payer.

## Request format

To make a request to financial operation, the following steps should be made:

1. Create operation request document in accordance with XML 1.0 standard (Fifth Edition), published here: <http://www.w3.org/TR/xml/>. Character encoding is UTF-8 in accordance with standard: <http://www.ietf.org/rfc/rfc2279.txt>.

2. Put XML document into cryptographic message in PKCS#7 format in accordance with standard: <http://www.ietf.org/rfc/rfc5652.txt>. PKCS#7 message must contain digital signature of request document. PKCS#7 message should not contain any certification authority chains. Data compression is not used. Data encryption is not used. PKCS#7 message must be encoded in PEM format (used by OpenSSL). X.509 certificate used to make digital signature must comply X.509 Version 3 standard (<http://www.ietf.org/rfc/rfc2459.txt>).

3. Create POST request using HTTP/1.1 protocol (<http://www.ietf.org/rfc/rfc2616.txt>, <http://www.ietf.org/rfc/rfc2618.txt> ). PKCS#7 message can be transferred by one of these methods:

1. PKCS#7 message put into POST-body, MIME type is: application/pkcs7-mime
2. PKCS#7 message put into multipart-data attach. MIME type is: application/pkcs7-mime. POST request must have only one 'part', PKCS#7 message must be attached as file. For example, you can create that request by HTML File Upload Form. See. <http://www.ietf.org/rfc/rfc2388.txt>

Operator authorizes requests by verifying its digital signatures.

Protection from mistaken double-operations is guaranteed by unique «client order ID» request parameter, assigned by Counterparties.

Example HTTP request example

POST /webservice/mws/api/returnPayment HTTP/1.1

Content-Type: application/pkcs7-mime

Content-Length: 906

-----BEGIN PKCS7-----

MIAGCSqGSIb3DQEHAqCAMIACAQExCzAJBgUrDgMCGgUAMIAGCSqGSIb3DQEHAaCA

JIAEgbE8P3htbCB2ZXJzaW9uPSIxLjAiIGVuY29kaW5nPSJVVEYtOCI/Pg0KPG1h

a2VEZXBvc2l0aW9uUmVzcG9uc2UgY2xpZW50T3JkZXJJZD0iMTI5MTExNjIzNDUy

OCIgc3RhdHVzPSIwIiBlcnJvcj0iMCIgcHJvY2Vzc2VkRFQ9IjIwMTAtMTEtMzBU

MTE6MjM6NTQuNjI0WiIgYmFsYW5jZT0iNTQxNDYuNzMiIC8+DQoAAAAAAAAxggF8

MIIBeAIBATB3MGoxCzAJBgNVBAYTAlJVMQ8wDQYDVQQIEwZSdXNzaWExFjAUBgNV

BAcTDVN0LlBldGVyc2J1cmcxITAfBgNVBAoTGEludGVybmV0IFdpZGdpdHMgUHR5

IEx0ZDEPMA0GA1UEAxMGc2VydmVyAgkAy2xbdQckXjIwCQYFKw4DAhoFAKBdMBgG

CSqGSIb3DQEJAzELBgkqhkiG9w0BBwEwHAYJKoZIhvcNAQkFMQ8XDTEwMTEzMDEx

MjM1NVowIwYJKoZIhvcNAQkEMRYEFEYNh8glwqIXGR/n6oYrApa8DaO5MA0GCSqG

SIb3DQEBAQUABIGAHlgGsYK30RXWBvuQao0V73KIPQEx2hH/9GY6Iag/xlmZ3rBB

kFpszF/O2fB+t84pCHfV15ErZQEkAqIotkEYEgA3hAddEW5+RWUzp+3npHpW5OY7

h3niP5Pj+r0P8EDgHe2j0Zb3dzj2mbwOshZD+FP1IcR8AmiTV3u35C6KAEsAAAAA

AAA=

-----END PKCS7-----

## Response format

Operation response is returned as XML document with accordance to standard: <http://www.w3.org/TR/xml/> . MIME type is application/xml. Character encoding is UTF-8.

## Refund operation request (returnPayment)

This query issues refund to Payer account, request parameters are described in Table 9 Refund operation request parameters. All parameters are mandatory. If mandatory parameter is missing, an error “Invalid parameter value NNNN” will be occurred.

Operation address: [https://server:port/webservice/mws/api/returnPayment](https://server:port/webservice/mws/api/returnPayment%20)

Table 9 Refund operation request parameters

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Data Type** | **Description** |
| clientOrderId | ClientTransactionNumber | Unique Identifier, assigned by Counterparty for refund operation. Recommended value: incrementing integer number. |
| requestDT | xs:dateTime | Request timestamp by current Counterparty time |
| invoiceId | xs:long | Source order transaction number |
| shopId | xs:long | Counterparty ID assigned by Operator |
| amount | CurrencyAmount | Amount to refund |
| currency | CurrencyCode | Currency code for amount |
| сause | xs:string, до 255 символов | Refund cause description |

Operation response contains common operation status fields (see Table 2 Common response parameters). Errors described inTable 11 Error codes.

### Refund operation requests processing rules

1. Each Counterparty’s request must contain unique operation identifier, assigned by Counterparty (clientOrderId).
2. If Counterparty sent request with clientOrderId is equal to clientOrderId sent before and other request parameters has same values (except requestDT), Operator returns state of previous request.
3. If Counterparty sent request with clientOrderId is equal to clientOrderId sent before and any other request parameter has different value from previous operation, Operator will refuse that request and returns an error (status=3, error=405).
4. Operator processes requests immediately. In case of unable to process request immediately, Operator returns status «in progress» (status=1). Counterparty should repeat the request with same parameters after a while. The following repeats schedule is recommended: one attempt after 1 minute, second three attempts with 5 minute interval, next attempts with minimum 30 minutes interval.
5. In case of no response or unclear response (HTTP status 500 or status=1 for example) Counterparty should repeat the request with same parameters after a while. The following repeats schedule is recommended: one attempt after 1 minute, second three attempts with 5 minute interval, next attempts with minimum 30 minutes interval.
6. Operation status «in progress» (status=1) after a while will be changed to «success» or «refused».
7. If operation request has refused the response contains status=3 and error= with refusal cause description. In some cases, response may contain an additional text field techMessage with detailed technical description. This text contains technical details and should not be displayed in any user-interface.

Example Request example

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

<returnPaymentRequest

clientOrderId="12345"

requestDT="2011-07-02T20:38:00.000Z"

invoiceId="2000000123"

shopId="6689"

amount="10.00"

currency="643"

cause="Buyer has refused to accept goods"

/>

Example Response example

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

<returnPaymentResponse

clientOrderId="12345"

status="0" error="0"

processedDT="2011-07-02T20:38:01.000Z"

/>

# Data Types

Table 10 Data types definition

|  |  |
| --- | --- |
| **Data Type** | **Description** |
| xs:int | 32-bit signed integer number. Int32 as defined by standard: <http://www.w3.org/TR/xmlschema-2/#int> |
| xs:long | 64-bit signed integer number. Int64, as defined by standard: <http://www.w3.org/TR/xmlschema-2/#long> |
| xs:decimal | [Fixed point decimal number](http://www.multitran.ru/c/m.exe?t=255417_1_2) as defined by standard: <http://www.w3.org/TR/xmlschema-2/#decimal> |
| xs:boolean | Logical value (true/false), as defined by standard: <http://www.w3.org/TR/xmlschema-2/#boolean> |
| xs:string | Character string as defined by standard: <http://www.w3.org/TR/xmlschema-2/#string> |
| xs:normalizedString | Character string as defined by standard: <http://www.w3.org/TR/xmlschema-2/#normalizedString> |
| xs:dateTime | Timestamp value in format conforms these specifications:   * <http://www.w3.org/TR/xmlschema-2/#dateTime> * ISO8601:2004   The format definition is:  ***YYYY-MM-DDThh:mm:ss.fZZZZZ***  Format description   |  |  | | --- | --- | | YYYY | 4 decimal digits, representing an Year | | *MM* | month, 2 decimal digits (01=January etc) | | *DD* | Day of month, 2 decimal digits (from 01 to 31) | | *T* | **Latin** character «T» in uppercase | | *hh* | hours, 2 decimal digits (24-hours format, from 00 to 23) | | *mm* | minutes, 2 decimal digits (from 00 to 59) | | *ss* | seconds, 2 decimal digits (from 00 to 59) | | *f* | second’s fraction (from 1 to 6 digits),  may be absent, in this case the separator character «.» shall be absent too | | *ZZZZZ* | Time Zone Offset, acceptable values:  **Z** – UTC, «Z» shall be in the uppercase  **+hh:mm** or -hh:mm –UTC (GMT) offset. (to indicate the **local time** offset from UTC) |   All parameters listed above shall be included, it is allowed to skip the seconds’ fractions only (together with “.” separator). Even in case only the date is to be set, the time shall be indicated as 00:00:00.  **Samples:**  2011-07-24T19:00:00+04:00 – 19 hours 00 minutes 24 july 2011, time zone – UTC + 4h.  2011-07-24T15:00:00Z – the same time moment in canonical representation.  2011-07-24T15:00:00.666Z – the same time moment plus 666 milliseconds. . |
| ClientTransactionNumber | Unique operation ID, assigned by Counterparty. Must be unique for any operation. Value is String with length from 1 up to 4 characters, with symbols allowed, according this pattern: 0-9 A-Z a-z . , \ | / - + = # ~ ( ) { } [ ] : ; Recommended value: incrementing integer number.  <xs:simpleType name="ClientTransactionNumber">  <xs:restriction base="xs:normalizedString">  <xs:minLength value="1"/>  <xs:maxLength value="24"/>  <xs:pattern value="[0-9A-Za-z.,\\|/\-+=#~(){}\[\]:;]+"/>  </xs:restriction>  </xs:simpleType> |
| YMAccount | An account number in Yandex.Money system, a string of [decimal digits](http://www.multitran.ru/c/m.exe?t=255417_1_2) (from 11 to 33 characters)  <xs:simpleType name="YMAccount">  <xs:restriction base="xs:normalizedString">  <xs:minLength value="11"/>  <xs:maxLength value="33"/>  <xs:pattern value="[0-9]+"/>  </xs:restriction>  </xs:simpleType> |
| CurrencyAmount | Amount. Positive fixed-point decimal number with precision is 2 digits, fraction point delimiter is “.”.  <xs:simpleType name="CurrencyAmount">  <xs:restriction base="xs:decimal">  <xs:minExclusive value="0"/>  <xs:maxInclusive value="9999999999999"/>  <xs:fractionDigits value="2"/>  </xs:restriction>  </xs:simpleType> |
| CurrencyCode | Currency code. Possible values:   * 643 — RUB (Russian Rouble); * 10643 — test currency (demo-roubles of Yandex.Money demo-system)   <xs:simpleType name="CurrencyCode">  <xs:restriction base="xs:int">  </xs:restriction>  </xs:simpleType> |
| CurrencyBank | Yandex.Money payment processing center code. Possible values:   * 1001 – EcomBank; * 1003 – DemoBank.   <xs:simpleType name="CurrencyBank">  <xs:restriction base="xs:int">  </xs:restriction>  </xs:simpleType> |
| PaymentType | Method of payment. Possible values:   * PC — payment from user’s Yandex.Money account or from a bank card, attached to the electronic wallet. * AC —payment from a bank card that is not attached to user’s electronic wallet in Yandex.Money. * GP —payment via cash-kiosk (terminal) * MC — payment from mobile phone balance.   Range of possible values can be extended.  <xs:simpleType name="PaymentType">  <xs:restriction base="xs:normalizedString">  <xs:minLength value="2"/>  <xs:maxLength value="2"/>  </xs:restriction>  </xs:simpleType> |

# Error codes

Table Error codes

|  |  |  |
| --- | --- | --- |
| **Code** | **Description** | **Comments** |
| Common errors | | |
| 0 | No error |  |
| 10 | Unable to parse XML document.  Possible reasons:   * XML syntax is broken * Mandatory XML elements are missing |  |
| 50 | Unable to read PKCS#7 cryptographic message (usually means format error or data corruption). |  |
| 51 | Digital signature verification failed. (usually means the digital signature does not match the document within cryptographic message) |  |
| 53 | Digital signature made with unknown certificate |  |
| 55 | Counterparty’s X.509 certificate has expired |  |
| 110 | Counterparty has no access rights to execute the operation requested. | For example, Counterparty tries to see orders statement of another merchant. |
| Common request parameters errors | | |
| 111 | Invalid value for the requestDT parameter |  |
| 112 | Invalid value for the invoiceId parameter |  |
| 113 | Invalid value for the shopId parameter |  |
| 114 | Invalid value for the orderNumber parameter |  |
| 115 | Invalid value for the clientOrderId parameter |  |
| 117 | Invalid value for the status parameter |  |
| 118 | Invalid value for the from parameter |  |
| 119 | Invalid value for the till parameter |  |
| Informational requests parameters errors | | |
| 200 | Invalid value for the outputFormat parameter | Allowed values: XML, CSV |
| 201 | Invalid value for the csvDelimiter parameter |  |
| 202 | Invalid value for the orderCreatedDatetimeGreaterOrEqual parameter |  |
| 203 | Invalid value for the orderCreatedDatetimeLessOrEqual parameter |  |
| 204 | Invalid value for the paid parameter |  |
| 205 | Invalid value for the paymentDatetimeGreaterOrEqual parameter |  |
| 206 | Invalid value for the paymentDatetimeLessOrEqual parameter |  |
| 207 | Invalid value for the outputFields |  |
| 208 | Empty order creation time interval specified | Upper boundary (orderCreatedDatetimeLessOrEqual) is less or equals lower boundary (orderCreatedDatetimeGreaterOrEqual) |
| 209 | Too long order creation time interval specified | Time interval, specified by parameters orderCreatedDatetimeGreaterOrEqual and orderCreatedDatetimeLessOrEqual, is longer than 31 days |
| 210 | Empty payment time interval specified | Upper boundary (paymentDatetimeLessOrEqual) is less or equals lower boundary (paymentDatetimeGreaterOrEqual) |
| 211 | Too long payment time interval specified | Time interval, specified by parameters paymentDatetimeGreaterOrEqual and paymentDatetimeLessOrEqual, is longer than 31 days |
| 212 | Logical contradiction between the payment time interval and the “paid" parameter | Payment time interval specified in parameters, but “paid” parameter is missing or have value is not “true”. |
| 213 | No one selection condition specified in parameters |  |
| 214 | Missing shopId condition in selection by orderNumber |  |
| 215 | Missing shopId condition in selection by invoiceId |  |
| 216 | Result contains too many elements | Specify more detailed selection conditions (usually narrower time interval) |
| 217 | Invalid value for the partial parameter | Allowed values: true, false |
| Financial operations requests parameters errors | | |
| 402 | Invalid value for the amount parameter |  |
| 403 | Invalid value for the currency parameter |  |
| 404 | Invalid value for the cause parameter | Parameter missing or have too long value |
| 405 | Non-unique operation ID | Other operation with same clientOrderId,is already processed by system (usually means same clientOrderId, but different other parameters) |
| 410 | Unable to refund unpaid order | Order wasn’t paid |
| 411 | Unable to refund, because funds transfer is not delivered to Counterparty | Funds transfer notification is not delivered to the Counterparty, order marked is unsuccessful. |
| 412 | Currency code specified in refund request is different from funds transfer currency |  |
| 413 | Refund amount exceeds order amount. | In accordance with previous partial refunds by this order. |
| 414 | Funds transfer has already refunded |  |
| 415 | Order with specified invoiceId does not exists |  |
| 416 | Not enough funds on Counterparty’s account |  |
| 417 | Payer’s account closed. Refund isn’t possible. |  |
| 418 | Payer’s account blocked. Refund isn’t possible. |  |
| Other errors | | |
| 1000 | Technical error |  |

# Funds transfer notification status codes

Table 12 Funds transfer notification status codes

|  |  |
| --- | --- |
| **Status code** | **Description** |
| -1 | Fund transfer notification is not created (usually order is not paid) |
| 81 | Fund transfer notification has created, waiting for delivery to Counterparty |
| 100 | Notification delivered to Counterparty, but Counterparty has refused it. |
| 103 | Notification delivery cancelled. |
| 1000 | Notification delivery successful. |
| 1010 | Notification not delivered to Counterparty, but marked successful. |
| 1020, 1021 | Notification delivery successful. Funds transfer has refunded to the Payer. |
| 1022 | Notification delivery successful. Funds transfer has partially refunded to the Payer. |