Yandex.Money to Merchant Integration Protocol: HTTP and Email Notifications.

Protocol 3.0, revised on May 29, 2014

**Table of Contents**

[1. General information 2](#_Toc390249569)

[1.1. The purpose of this document 2](#_Toc390249570)

[1.2. The Counterparty’s integration 2](#_Toc390249571)

[2. Interaction: a general overview 3](#_Toc390249572)

[3. The payment form 4](#_Toc390249573)

[4. HTTP-notifications about payments 6](#_Toc390249574)

[4.1. Format for interaction 6](#_Toc390249575)

[4.2. Checking orders (сheckOrder) 6](#_Toc390249576)

[4.2.1. Format of the Operator’s request 7](#_Toc390249577)

[4.2.2. Format of the Counterparty’s response 8](#_Toc390249578)

[4.3. Notification of payment (paymentAviso) 9](#_Toc390249579)

[4.3.1. Format of the Operator’s request 9](#_Toc390249580)

[4.3.2. Format of the Counterparty’s response 10](#_Toc390249581)

[4.4. Rules for the Counterparty when processing HTTP-notifications 10](#_Toc390249582)

[5. Email-notifications about payments 11](#_Toc390249583)

[6. Appendices 12](#_Toc390249584)

[6.1. The Counterparty’s integration parameters 12](#_Toc390249585)

[6.2. Particulars of interaction when payment method is cash via payment kiosks 14](#_Toc390249586)

[6.3. Daily payment reports 15](#_Toc390249587)

[6.4. Payment methods 17](#_Toc390249588)

[6.5. Types of data 17](#_Toc390249589)

# General information

## The purpose of this document

This document describes the protocol by which Yandex.Money (hereinafter ‘Operator’) will interact with the information system (IS) of the Counterparty to notify the Counterparty in real time about complete transactions (hereinafter ‘payments’) made in the interest of the Counterparty.

The Operator shall ensure receipt of payments made by various methods: bank cards, electronic wallets (including Yandex.Money e-wallets), cash via payment kiosks, and mobile phone balances. The payment methods that are available to a particular Counterparty will depend on the provisions of the agreement and are further controlled by the Operator’s settings.

If simplified, the interaction process can be presented as a few successive steps:

1. The Operator transmits information about the order and the payment method (performed using the payer’s browser).
2. The Counterparty receives notification about the payment (performed by the Operator either by HTTP or email notification).
3. A report of the payments accepted in the interest of the Counterparty is generated (sent by the Operator via email).
4. Funds are transferred to the Counterparty’s bank account.
5. If necessary, refunds of successful payments are made to payers as initiated by the Counterparty.

Steps 1-3 are described below. Steps 4-5 are beyond the scope of this document. To work with refunds, the Counterparty will also need to issue a certificate and implement the Merchant Web Services (MWS) protocol. The procedure for exchanging MWS certificates and the MWS protocol are described in separate documents.

## The Counterparty’s integration

The Counterparty has the option of integrating with Yandex.Money according to one of two integration schemes:

* the scheme wherein the Operator sends payment notifications to the Counterparty via the HTTP protocol (hereinafter - HTTP-scheme, a detailed description of interactions is presented in Section 4 ‘HTTP-notifications about payments’);
* the scheme wherein the Operator sends payment notifications to the Counterparty via email (hereinafter – email scheme, a detailed description is presented in Section 5 ‘Email-notifications about payments’).

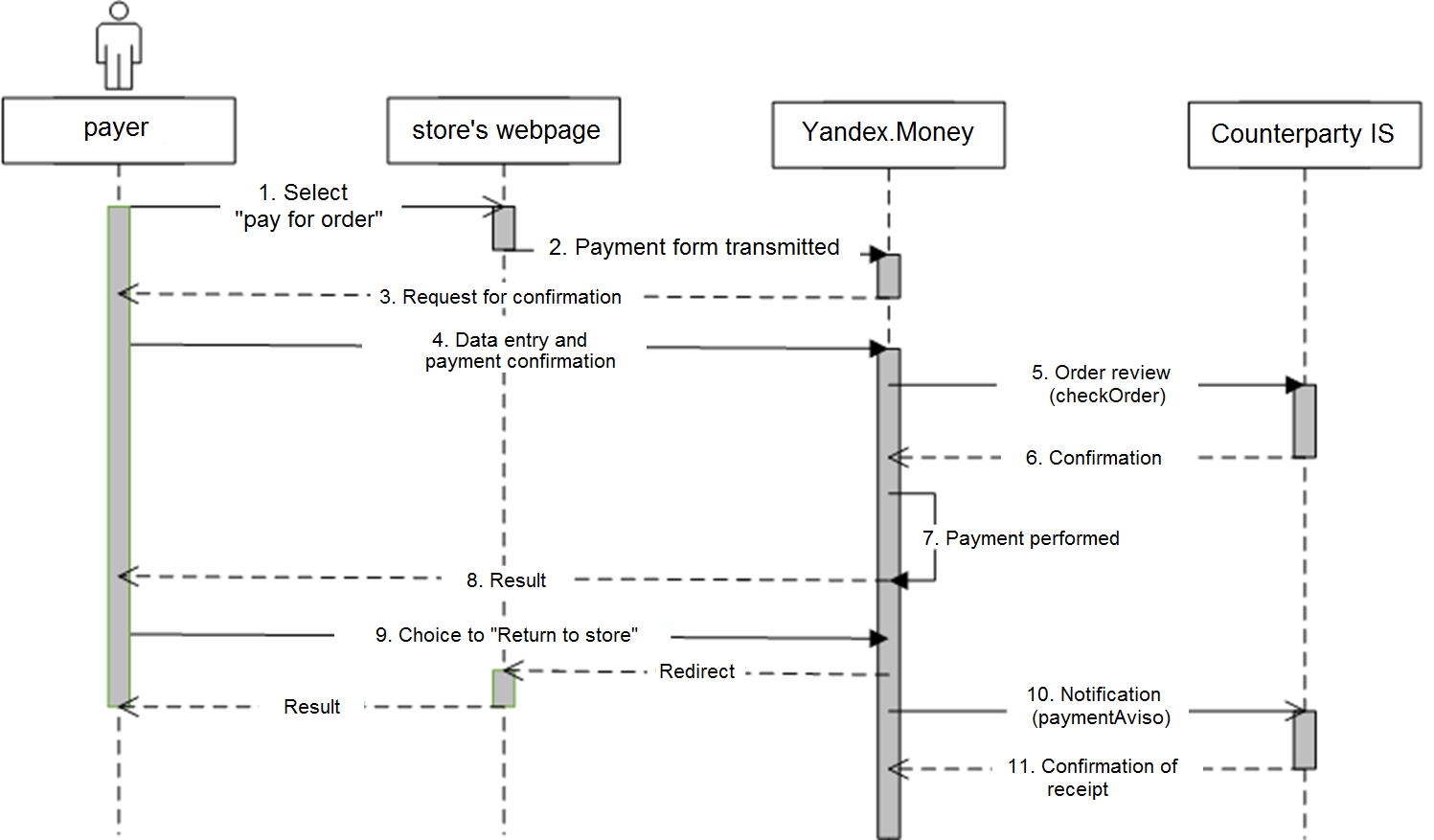
The main difference between the two schemes is that the email scheme does not allow for feedback, whereas the HTTP scheme allows the Counterparty to perform online review of order parameters during the payment process. If the Counterparty needs to show the payer on its website and in real time that the goods or services have been paid for, the Counterparty must use the HTTP-scheme to integrate with Yandex.Money. Both schemes cannot be used simultaneously. The number of available payment methods does not depend on the scheme.

Depending on the scheme, the Counterparty must inform the Operator of the integration parameters: URLs (email address) for delivery of notifications, URLs for redirecting payers after completion of payment, email address for delivery of daily payment reports, etc. (detailed information can be found in section 6.1 ‘The Counterparty’s integration parameters’).

In response, the Operator will provide the settings for accessing Yandex.Money’s testing environment. After the testing procedure is complete, the Operator will provide the Counterparty with settings for information exchange in the production environment. Integration to MWS is performed separately.

# Interaction: a general overview

0. The Counterparty places a ‘payment form’ on the order payment page with the order information and payment method options (in some cases the ‘payment form’ may be placed on the Operator’s site: <https://money.yandex.ru/shops.xml>).



|  |  |
| --- | --- |
| **HTTP integration scheme** | **Email integration scheme** |
| 1-2. The payer’s browser transmits the completed form to the Operator IS. | |
| 3. The Operator uses the data received to determine the payment method and displays the payment confirmation page to the payer. | |
| 4. The payer enters additional information (e.g. bank card details) and confirms the payment. | |
| 5. Before the payment is completed, the Operator sends the Counterparty IS a request to review the order (‘сheckOrder’). | 5. This step is skipped. |
| 6. The Counterparty confirms the accuracy of the order or declines the payment. | 6. This step is skipped. |
| 7-8. If the Counterparty IS gives an affirmative response to the ‘Check order’ request, the Operator debits money from the payer and displays the payment result to the payer. | 7-8. The Operator debits money from the payer and displays the payment result to the payer. |
| 9. On the payment result page the link ‘Return to store’ is displayed.  The Counterparty determines the URL where the payer will be redirected. | |
| 10-11. If the payment is completed successfully, the Operator will send the Counterparty IS a ‘Notification of payment’ (paymentAviso) request. | 10. If the payment is completed successfully, the Operator will send the Counterparty IS an email-notification about the payment. |

**Please note:** if the payment method is cash at a payment kiosk, interaction between the Operator and Counterparty involves a few special differences. A description of this scenario can be found in section 6.2 ‘Particulars of interaction when payment method is cash via payment kiosks’.

Once a day, the Operator sends the Counterparty a **daily payment report** via email. The Counterparty should check that the list of successful payments in its IS’ data matches the list of operations in the payment report and inform the Operator of any discrepancies. The format of the payment report is described in section 6.3 ‘Daily payment reports’.

# The payment form

The Counterparty places the payment form on the payment page. It determines the order parameters and the payment method. The act of sending via the payer’s browser the payment form at the standard address (<http://money.yandex.ru/eshop.xml>) initiates the creation and processing of an order for transfer on the side of the Operator.

Sample payment form:

<!—Values for all fields are arbitrary and provided only as examples -->

<form action="https://money.yandex.ru/eshop.xml" method="post">

<!—Required fields -->

<input name="shopId" value="1234" type="hidden"/>

<input name="scid" value="4321" type="hidden"/>

<input name="sum" value="100.50" type="hidden">

<input name="customerNumber" value="abc000" type="hidden"/>

<!—Optional fields -->

<input name="shopArticleId" value="567890" type="hidden"/>

<input name="paymentType" value="AC" type="hidden"/>

<input name="orderNumber" value="abc1111111" type="hidden"/>

<input name="cps\_phone" value="79110000000" type="hidden"/>

<input name="cps\_email" value="user@domain.com" type="hidden"/>

<input type="submit" value="Pay"/>

</form>

When transmitting information about the order and payment method, the Counterparty should use the parameters from the table below. All parameters are **case sensitive**.

**Table 3.1.** Payment form parameters

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Type** | **Description** |
| *Reserved parameters:* | | |
| shopId | xs:long, required field | Counterparty’s ID. Issued by the Operator. |
| shopArticleId | xs:long, optional field | Article ID. Issued by the Operator. Applies if the Counterparty uses various payment forms for different articles (goods). |
| scid | xs:long, required field | Counterparty's payment form ID. Issued by the Operator. |
| sum | CurrencyAmount, required field | Order total. |
| customerNumber | xs:normalizedString, up to 64 characters, required field | Payer ID in the Counterparty IS. The ID can be the payer’s contract number, login, etc.  Multiple payments may be applied for a single customerNumber. |
| orderNumber | xs:normalizedString, up to 64 characters, optional field | Unique order number in the Counterparty IS. The uniqueness is verified by the Operator in combination with the parameter shopId.  If a payment with the same order number has been successfully completed already, repeated attempts to pay will be rejected by the Operator. |
| shopSuccessURL | xs:string, up to 250 characters, optional field | URL for payer redirection when payment is successful. Applies only if the corresponding Counterparty integration option is selected (see section 6.1 ‘The Counterparty’s integration parameters’). |
| shopFailURL | xs:string, up to 250 characters, optional field | URL for payer redirection if an error occurs. Applies only if the corresponding Counterparty integration option is selected. |
| cps\_email | xs:string, up to 100 characters, optional field | Payer’s email address. If it is present, the corresponding field on the payment confirmation page will be filled in automatically (step 3 on the flow chart above). |
| cps\_phone | xs:string, up to 15 characters, numbers only, optional field | Payer’s phone number. If it is present, the corresponding field on the payment confirmation page will be filled in automatically (step 3 on the flow chart above). The phone number is used for cash payments via payment kiosks. |
| paymentType | xs:normalizedString,  up to 5 characters, optional field | Payment method. Example:   * РС – payment from a Yandex.Money wallet; * АС – payment by any bank card.   For the full list of values, see Table 6.4.1.  **Please note**:   * The absence of the paymentType parameter is interpreted as payment from a Yandex.Money wallet; * If the payment form contains a payment method that the Counterparty is not authorized to accept, the payer won’t be able to make the payment. |
| *Parameters added by the Counterparty:* | | |
| Any names other than listed above | xs:string | Parameters added by the Counterparty to the payment form will be saved and transmitted to the Counterparty IS with the ‘Check order’ (checkOrder) and ‘Notification of payment’ (paymentAviso) requests.  The total length of all parameters added by the Counterparty shall not exceed 4,096 characters.  **Please note**: email notifications about payments do not feature parameters added by the Counterparty. To transmit additional payment details, the Counterparty can use the standard parameters listed below. |
| *Reserved parameters used in email notifications about payments:* | | |
| сustName | xs:string, optional field | Payer’s full name. |
| сustAddr | xs:string, optional field | Delivery address or payer’s residential address. |
| сustEMail | xs:string, optional field | Payer’s email address, for email notifications only. |
| orderDetails | xs:string, optional field | Order details: the list of purchased goods, their quantity, purpose of payment, etc. |

# HTTP-notifications about payments

## Format for interaction

When integrating according to the HTTP-scheme, the Counterparty should determine the address where it will receive HTTP-notifications from the Operator.

Information from the Operator is transmitted to the Counterparty IS via HTTP/1.1 protocol using POST request method. The message’s parameters are packed as a set of POST-request parameters in the form of the pairs ‘name=value’. Content-Type: application/x-www-form-urlencoded, charset – UTF-8.

The ‘md5’ request parameter contains the result of the hash function from the concatenation of the message parameters with a secret word, which the Counterparty will specify during the integration process. The Counterparty should check the result of the ‘md5’ parameter (the algorithm can be found in section 4.4 ‘Rules for the Counterparty when processing HTTP-notifications’) and decline to process the request if the check is unsuccessful. A successful hash check will confirm:

* that the request was sent by the Operator;
* the information in the request is authentic.

We also recommend checking the IP-addresses from which the Counterparty IS receives requests (a list of the Operator’s IP-addresses can be provided during the integration process).

In order to ensure the security of payment information during the interaction between the Operator and Counterparty IS, one of the conditions listed below must be met:

1. the transmission of information takes place via a secure channel (the Counterparty uses the HTTPS protocol for receiving messages from the Operator);
2. the Operator’s messages are encrypted prior to being sent(\*).

\* *This requires that the Counterparty integrate according to the scheme XML/PKSC#7. In this case messages are transmitted by the Operator in the form of XML-documents embedded in PKCS#7 cryptographic message containers. Data are signed by the Operator’s SSL-certificate. Contact your account manager for detailed information about the XML/PKCS#7 integration scheme.*

The Counterparty should return the result of processing the Operator’s request in the form of an XML-document in the body of the response to the HTTP-request. A document is created according to the standard XML 1.0 (Fifth Edition), which can be found at the following address: <http://www.w3.org/TR/xml/>. Names of elements and attributes are case sensitive. Content-Type: application/xml, charset – UTF-8.

## Checking orders (сheckOrder)

This section covers the request for validation of the order parameters. This step will help the Counterparty prevent errors that can occur when the payment form is being sent through the payer’s browser.

If the response from the Counterparty is successful, the Operator will invite the payer to pay for the order. If the payment is successful, the Operator will send the Counterparty a ‘Notification of payment’.

**Please note:**

1. A ‘Check order’ request is usually created before money is debited from the payer’s account. During this step, the Counterparty **can refuse** to accept the payment.
2. When the payment method is a bank card, payment authorization occurs before the ‘Check order’ request is created. If the Counterparty refuses the payment, money will automatically be returned to the card.
3. When the payment method is anything other than a Yandex.Money e-wallet, third-party systems may charge additional commission. In that case, if the Counterparty refuses the payment, funds will be returned to the payer less the third-party’s commission.

### Format of the Operator’s request

**Table 4.2.1.1.** checkOrder request parameters

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Type** | **Description** |
| requestDatetime | xs:dateTime | The time when the request is generated in the Operator IS. |
| action | xs:normalizedString, up to 16 characters | Request type. Value: ‘checkOrder’ (without quotation marks). |
| md5 | xs:normalizedString, exactly 32 hexadecimal characters, all caps. | MD5 hash of the payment form parameters. Rules for generating it are described in section 4.4 ‘Rules for the Counterparty when processing HTTP-notifications’. |
| shopId | xs:long | Counterparty’s ID assigned by the Operator. |
| shopArticleId | xs:long | Article ID assigned by the Operator. |
| invoiceId | xs:long | Unique transaction number in the Operator IS. |
| orderNumber | xs:normalizedString, up to 64 characters | Order number in the Counterparty IS. Present only if specified on the payment form. |
| customerNumber | xs:normalizedString, up to 64 characters | Payer ID (sent in the payment form) with the Counterparty: contract number, mobile phone number, etc. |
| orderCreatedDatetime | xs:dateTime | The time when the order is registered in the Operator IS. |
| orderSumAmount | CurrencyAmount | Order total. Can differ from the payment amount if the user paid in a currency other than the one specified on the payment form. In this case, the Operator assumes all conversion transactions. |
| orderSumCurrencyPaycash | CurrencyCode | Currency code for the order total. |
| orderSumBankPaycash | CurrencyBank | The Operator's processing center code for the order total. |
| shopSumAmount | CurrencyAmount | Amount to be received on the Counterparty's bank account: the order total less the Operator's commission. |
| shopSumCurrencyPaycash | CurrencyCode | Currency code for shopSumAmount. |
| shopSumBankPaycash | CurrencyBank | The Operator's processing center code for shopSumAmount. |
| paymentPayerCode | YMAccount | Number of the account in the Operator IS where the payment is made from. |
| paymentType | xs:normalizedString | Payment method used. The list of possible values can be found in Table 6.4.1. |
| Any names other than listed above | xs:string | Parameters added to the payment form by the Counterparty. |

**Please note:** the Operator’s requests may contain parameters that are not described in this document. The Counterparty should ignore such parameters.

Sample ‘сheckOrder’ request parameters:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | requestDatetime | 2011-05-04T20:38:00.000+04:00 | | action | checkOrder | | md5 | 8256D2A032A35709EAF156270C9EFE2E | | shopId  shopArticleId | 13  456 | | invoiceId | 1234567 | | customerNumber | 8123294469 | | orderCreatedDatetime | 2011-05-04T20:38:00.000+04:00 | | orderSumAmount | 87.10 | | orderSumCurrencyPaycash | 643 | | orderSumBankPaycash | 1001 | | shopSumAmount | 86.23 | | shopSumCurrencyPaycash | 643 | | shopSumBankPaycash | 1001 | | paymentPayerCode | 42007148320 | | paymentType | AC | | MyField | Field added by the Counterparty | |

### Format of the Counterparty’s response

**Table 4.2.2.1.** checkOrder response parameters

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Type** | **Description** |
| performedDatetime | xs:dateTime | The time the request is processed according to the time of the Counterparty IS. |
| code | xs:int | Processing result code. The list of possible values can be found in the table below. |
| shopId | xs:long | Counterparty ID. Should be identical to the request's ‘shopId’ field. |
| invoiceId | xs:long | Transaction ID in the Operator IS. Should be identical to the request's ‘invoiceId’ field. |
| orderSumAmount | CurrencyAmount | Order total in the currency determined by the ‘orderSumCurrencyPaycash’ request parameter. |
| message | xs:string, up to 255 characters | Message if the payment is declined. |
| techMessage | xs:string, up to 64 characters | Additional comment to the Counterparty’s response. Generally used as extra information about errors. Optional field. |

**Table 4.2.2.2.** checkOrder request processing result codes

|  |  |  |
| --- | --- | --- |
| **Code** | **Value** | **Description** |
| 0 | Successful | The Counterparty has given its permission and is ready to accept the payment. |
| 1 | Authorization error | md5 parameter value does not match the hash function calculation result. The Operator considers the error final and will not perform the payment. |
| 100 | Payment declined | The payment with the given parameters cannot be accepted. The Operator considers the error final and will not perform the payment. |
| 200 | Bad Request | The Counterparty IS is unable to process the request. The Operator considers the error final and will not perform the payment. |

Sample response to checkOrder request when the processing is successful:

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <checkOrderResponse performedDatetime="2011-05-04T20:38:01.000+04:00"  code="0" invoiceId="1234567"  shopId="13"/> |

Sample response to checkOrder request when an error occurs: the IS rejected the payment at the order review stage:

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <checkOrderResponse performedDatetime="2011-05-04T20:38:01.000+04:00"  code="100" invoiceId="1234567"  shopId="13"  message="The phone number specified does not exist"  techMessage="Invalid phone number"/> |

## Notification of payment (paymentAviso)

This section covers how the Counterparty is notified that a payment has been performed. This request confirms that the funds transfer between the payer and the Counterparty was successfully completed and that the Counterparty is now obligated to issue the payer the goods he or she ordered.

**Please note:** during this step the Counterparty cannot refuse to accept the payment.

### Format of the Operator’s request

The parameters of the ‘Notification of payment’ request match the parameters in the ‘Check order’ request (see the description in section 4.2.1). The parameters specific to the operation paymentAviso are outlined in the table below:

**Table 4.3.1.1.** paymentAviso request parameters

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Type** | **Description** |
| action | xs:normalizedString, up to 16 characters | Request type, value: paymentAviso. |
| paymentDatetime | xs:dateTime | The time the payment for an order is registered with the Operator IS. |

**Please note:** the Operator’s requests may contain parameters that are not described in this document. The Counterparty should ignore such parameters.

Sample paymentAviso request parameters:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | requestDatetime | 2011-05-04T20:38:00.000+04:00 | | action | paymentAviso | | md5 | 45125C95A20A7F25B63D58EA304AFED2 | | shopId  shopArticleId | 13  456 | | invoiceId | 1234567 | | customerNumber | 8123294469 | | orderCreatedDatetime | 2011-05-04T20:38:00.000+04:00 | | orderSumAmount | 87.10 | | orderSumCurrencyPaycash | 643 | | orderSumBankPaycash | 1001 | | shopSumAmount | 86.23 | | shopSumCurrencyPaycash | 643 | | shopSumBankPaycash | 1001 | | paymentDatetime | 2011-05-04T20:38:10.000+04:00 | | paymentPayerCode  paymentType | 42007148320  AC | | MyField | Field added by the Counterparty | |

### Format of the Counterparty’s response

The parameters for the Counterparty’s response to the ‘Notification of payment’ request match the parameters for the ‘Check order’ operation (see the description in section 4.2.2).

Possible codes for the results of processing the ‘Notification of payment’ request are outlined in the table below:

**Table 4.3.2.1.** paymentAviso request processing result codes

|  |  |  |
| --- | --- | --- |
| **Code** | **Value** | **Description** |
| 0 | Successful | Success, even when the Operator has sent the given request multiple times. |
| 1 | Authorization error | md5 parameter value doesn’t match the hash function calculation result. The Operator will not repeat the request and will mark the order as ‘Notification not delivered to the Counterparty’. |
| 200 | Bad Request | The Counterpartys IS is unable to process the request. The Operator will not repeat the request and will mark the order as ‘Notification not delivered to the Counterparty’. |

Sample response to paymentAviso request when the processing is successful:

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <paymentAvisoResponse  performedDatetime ="2011-05-04T20:38:11.000+04:00"  code="0" invoiceId="1234567"  shopId="13"/> |

## Rules for the Counterparty when processing HTTP-notifications

1. The Counterparty should review the value for the md5 parameter in order to verify the completeness and authenticity of the requests. If the value for md5 does not match the result of the calculated MD5 hash function, the Counterparty must refuse to process the request.

MD5 hashing is applied to the text assembled as a sequence of the values for several request parameters, which are delimited by semicolons - ‘;’. The parameters will appear in the following order:

|  |
| --- |
| action;orderSumAmount;orderSumCurrencyPaycash;orderSumBankPaycash;shopId;invoiceId;customerNumber;shopPassword |

Sample:

|  |  |
| --- | --- |
| **Original line** (without breaks) | **Hashing result** |
| checkOrder;87.10;643;1001;13;55;8123294469;s<kY23653f,{9fcnshwq | 1B35ABE38AA54F2931B0C58646FD1321 |

1. The Counterparty IS should respond to the Operator’s requests within 10 seconds.
2. If there is no response from the Counterparty to the ‘Check order’ request or if the response is anything besides ‘successful’, the Operator will inform the payer that he or she may not pay.
3. If, for a long period, there are no responses from the Counterparty to repeated ‘Notification of payment’ requests (or technical errors are returned repeatedly), the Operator IS will repeat attempts to deliver the notification within a 24-hour period: the first repeat attempt will be sent one minute after the initial request, the next five in intervals from 5-30 minutes. After that the payment will be given a final status. Whether the status is ‘successful’ or ‘unsuccessful’ depends on the Counterparty’s integration parameters (more detailed information can be found in the section 6.1 ‘The Counterparty’s integration parameters’).
4. The Operator assigns a unique number to each payment (invoiceId). The Counterparty should be prepared for the case that several ‘Notification of payment’ requests may be delivered for the same invoiceId (due to problems with the connection or an error in the Counterparty’s response to the request). The Counterparty IS should respond ‘successful’ (code=”0”) to duplicate notifications.

# Email-notifications about payments

If the Counterparty selects the email scheme for integration, it should provide the Operator with the email address where it will receive email notifications from the Operator.

Notifications are sent in the body of an email message and are signed with the Operator’s certificate (S/MIME signature).

The Operator generates a separate notification for every successful payment to the Counterparty. The format for notifications is outlined below:

**Table 5.1.** Fields for email notifications about payments

|  |  |
| --- | --- |
| **Field** | **Value** |
| Извещение № (Notification #) | Number of the email notification about the payment made to the Counterparty. Numbering is continuous. |
| Получатель  (Recipient) | Legal name of the Counterparty specified at the integration stage. |
| Время перевода  (Date and time of payment) | Date and time of payment in the format dd.mm.yyyy hh:mm:ss according to the time of the Operator IS. |
| Сумма  (Payment amount) | Amount of payment. Decimal delimiter — a decimal point, always exactly two characters after the decimal point, thousands place delimiter is absent. |
| Номер транзакции (Payment number) | Unique transaction ID in the Operator IS. |
| Идентификатор плательщика  (Customer ID) | Payer ID in the Operator IS. ‘customerNumber’ parameter value on the payment form (for here and below, see Section 3 ‘Payment form’). |
| Номер у контрагента  (Counterparty’s order number) | Unique order ID in the Counterparty IS. ‘orderNumber’ parameter value on the payment form. If the field is not filled out, the value from ‘Transaction number’ field is inserted. |
| ФИО (Full name) | First name, last name, and patronymic of the payer. ‘custName’ parameter value on the payment form. |
| Адрес доставки  (Address for delivery) | Delivery address or the payer’s residential address. ‘custAddr’ parameter value on the payment form. |
| Email | Email address of the payer. ‘custEMail’ parameter value on the payment form. |
| Содержание заказа (Contents of order) | Order details: the list of purchased goods, their quantity, purpose of payment destination, etc. ‘orderDetails’ parameter value on the payment form. |

**Please note:** some fields may be empty if the Counterparty has not included the corresponding parameter in the payment form or the payer did not fill in the field.

Sample email notification:

|  |
| --- |
| Subject: Yandex.Dengi payment for Наименование\_Контрагента #87  Извещение № 87  Получатель: ООО «Наименование\_Контрагента»  Время перевода: 18.01.2008 16:32:37  Сумма: 12.00 RUB  Номер транзакции: 1099511628638  Идентификатор плательщика: 4637937  Номер у Контрагента: 1099511628638  Заполнено плательщиком в платежной форме Контрагента:  ФИО: Ivanov Iva Ivanovich  Адрес доставки: Moscow, ul. Moskovskaya 3-45  Email: ivanovii@domain.com  Содержание заказа: some description of the order |

# Appendices

## The Counterparty’s integration parameters

To complete integration with the Operator IS, the Counterparty must inform the Operator of its selections for the following settings (parameters 3-6 are only required if the HTTP-scheme for integration is selected, parameter 9 is only required for the email-scheme for integration):

**Table 6.1.1.** Counterparty integration parameters

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Value** | **Comments** |
| 1. Name of the Counterparty | Up to 128 characters | Name of the Counterparty’s store, which should be visible to the payer during the payment process. |
| 2. URL of Counterparty’s website |  |  |
| 3. checkURL | * For testing * For production   \* up to 200 characters | URL where the Counterparty IS will be available for the Operator’s ‘Check order’ requests. For interaction, the HTTPS protocol must be used. |
| 4. paymentAvisoURL | * For testing * For production   \* up to 200 characters | URL where the Counterparty IS will be available for the Operator’s ‘Notification of payment’ requests. For interaction, the HTTPS protocol must be used. |
| 5. Counterparty's secret word | We suggest you use a randomly generated set of symbols with a length of no less than 20 characters. | Used to create the md5 hash transmitted in the ‘Check order’ and ‘Notification of payment’ requests. |
| 6. Status of payments when notification of payment cannot be delivered | * 6.1 Consider unsuccessful * 6.2 Consider successful | These settings account for the mutual behavior of the Counterparty and the Operator when the ‘Notification of payment’ cannot be delivered (if the Counterparty does not respond to repeated requests from the Operator over a long period of time or the Counterparty IS returns multiple technical errors).  For description of options, see Table 6.1.2 below. |
| 7. Procedure for redirecting payers after payment is complete | * 7.1 To static URLs of the Counterparty:  |  |  |  | | --- | --- | --- | | articleId | successURL | (\*) | |  | failURL | (\*) |  |  |  |  | | --- | --- | --- | | articleId | successURL | (\*) | |  | failURL | (\*) |   \* up to 200 characters; addresses for testing and for production   * 7.2 To URLs transmitted by the Counterparty on the payment form | These settings determine where the payer will be redirected after the payment is completed. The payer is redirected from the payment result page by clicking the ‘Return to store’ link.  For a description of redirect options, see Table 6.1.3 below. |
| 8. Email address for payment reports |  | Email address for delivery of daily payment reports of accepted by the Operator in the interest of the Counterparty. |
| 9. Email address for notifications about payments |  | Email address for delivery of notifications. |

**Table 6.1.2.** Status of payments when ‘Notification of payment’ cannot be delivered

|  |  |
| --- | --- |
| **Option** | **Comments** |
| ‘Consider unsuccessful’ (by default) | The Operator ceases all attempts to deliver the notification, marks the payment as not delivered to the Counterparty and does not include it in the daily payment report. The amount of the unsuccessful payment will be automatically returned to the payer. The Counterparty can find the ‘lost notifications’ by reviewing the list on Merchant Web Services (MWS). |
| ‘Consider unsuccessful’ | The Operator ceases all attempts to deliver the notification and marks the payment as successful. The payment will be included in the daily payment report as of the time of the last attempt to deliver the ‘Notification of payment’. The Counterparty can find the ‘lost notifications’ by reviewing the list on MWS (**\***). |

\**To view the list of operations through the web interface MWS, the Counterparty just needs to issue a certificate. No programming is necessary.*

*The Counterparty will need to implement the MWS protocol if it needs the ability to issue refunds. Contact your account manager for the necessary documentation.*

**Table 6.1.3.** Options for redirecting the payer after completion of the payment

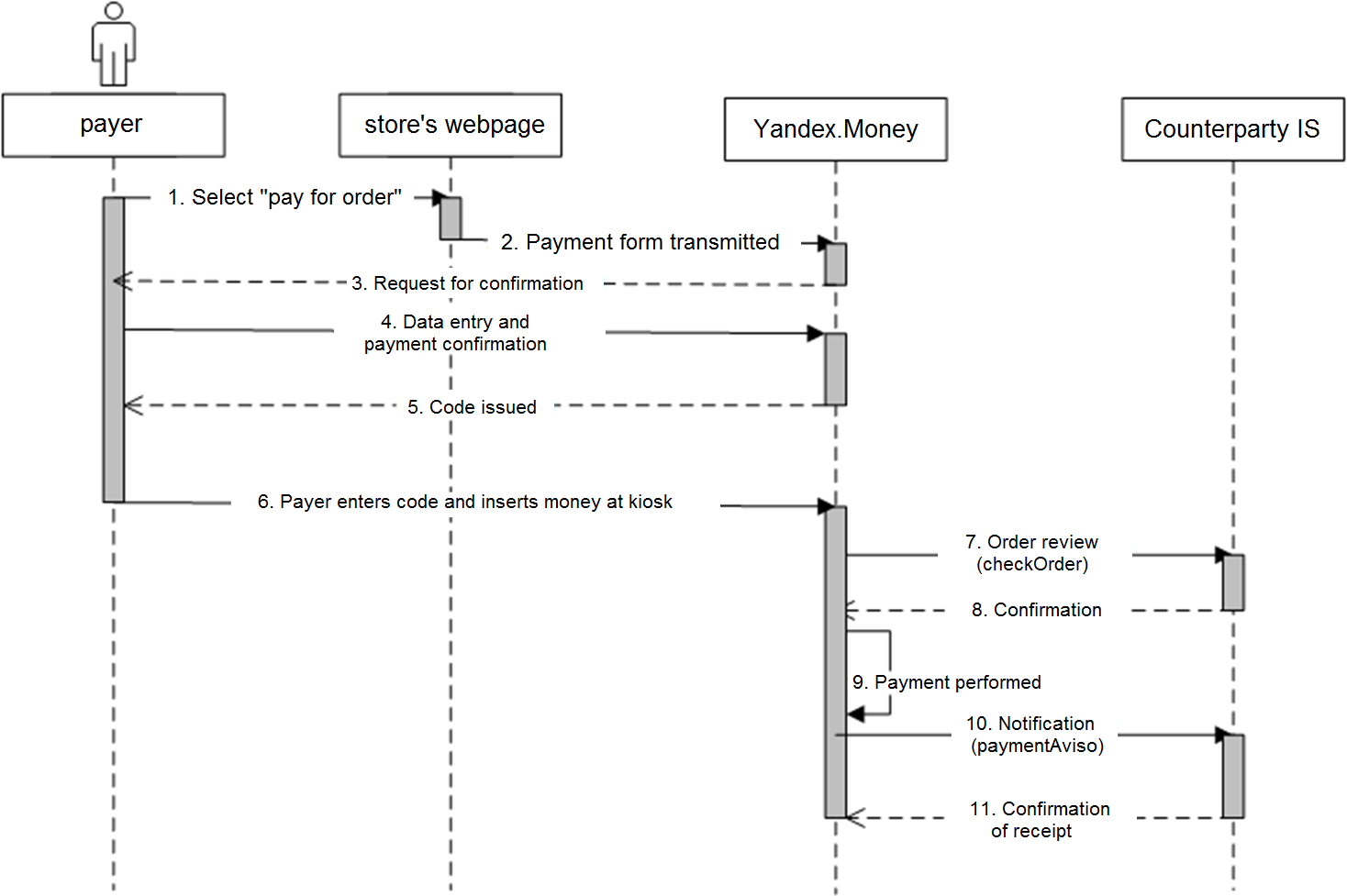
|  |  |
| --- | --- |
| **Option** | **Comments** |
| ‘To static URLs of the Counterparty’ (by default) | The addresses used for redirect are fixed ones and are determined by the following settings (individually for each good):   * successURL * failURL |
| ‘To URLs transmitted by the Counterparty on the payment form’ | The addresses used for redirect are transmitted by the Counterparty with the payment form parameters (individually for each good):   * shopSuccessURL * shopFailURL |

**Please note:**

1. During a redirect to a URL, the parameter “?action=PaymentSuccess” (“?action=PaymentFail”) is added, as well as the request parameters from the Operator to the Counterparty IS (payment form parameters). Redirects are performed via the GET method (an **exception** is unsuccessful payment from a Yandex.Money e-wallet, in which case a redirect will be performed via the POST method).
2. If the payment status cannot be determined, the payer will be redirected to the Counterparty’s homepage (the URL specified during integration: ‘2. URL of Counterparty’s website’), additional parameters will not be added to the URL.
3. If the Counterparty plans on displaying personal information intended for a specific payer, then it should authorize that payer using its own resources. This may be a standard authorization on the Counterparty’s website (e.g. via cookies) or through use of the Counterparty’s session keys placed on the payment form.
4. When the **payment method is cash** at a payment kiosk or **payment from a mobile phone balance**, the payer will be redirected to the homepage of the Counterparty’s website and additional parameters will not be added to the URL.
5. When the payment method is a **WebMoney** e-wallet, the payer will be redirected straight from the WebMoney system to the Counterparty’s site. In this case, WebMoney may add its own additional parameters to the URL for redirect.

## Particulars of interaction when payment method is cash via payment kiosks

The interaction between the Operator and the Counterparty when the payment method is cash at a payment kiosk differs from the basic scenario (as described in section 2 ‘Interaction: a general overview’). These particulars must be taken into consideration:



3-4. After payment form parameters are received and the payment method is determined, the payer is additionally asked to provide a phone number and email address.

If the Counterparty transmitted the payer’s phone number (cps\_phone) and/or email address (cps\_email) among the payment form parameters, the payment confirmation form will be pre-filled with these data.

5. The payer is issued a special code and instructions for paying at a payment kiosk or retailer. The Operator sends this code, as well as the payment total, via SMS to the phone number specified by the payer.

When the payer clicks on the link ‘Return to store’, found on the page where the code is displayed, the payer will be redirected to the webpage the Counterparty specified during the integration process (‘2. URL of Counterparty’s website’). The parameters (shop)successURL, (shop)failURL are not used in this case.

6. The payer can use the payment code received in step 5 to indicate the order he or she wants to pay for at any payment kiosk or ATM where users can add money to their Yandex.Money e-wallets.

7-11. After the payment kiosk network transmits confirmation that the payer inserted the full payment amount, the Operator sends the successive requests ‘Check order’ (checkOrder) and ‘Notification of payment’ (paymentAviso).

**Please note:**

* If the Counterparty refuses to accept the payment, the Operator will return money to the payer independent of the Counterparty;
* If the amount of money the payer inserts into the payment kiosk exceeds the order amount, the change will be automatically credited to the balance of the mobile phone specified by the payer during the order process;
* If the amount of money the payer inserts into the payment kiosk is less than the order amount, the Operator will send him or her an SMS with information about the insufficient amount. To complete the payment, the payer must insert the insufficient amount.

11. After the Counterparty IS responds to the ‘Notification of payment’ request, the Operator sends a message to the payer about the payment result to the email address indicated by the payer earlier.

## Daily payment reports

Once a day, the Operator generates a report of payments accepted in the interest of the Counterparty. The payment report is sent in the body of an email message to the email address (\*) the Counterparty indicates during the integration process. (‘8. Email address for payment reports’). The payment report is signed with the Operator’s certificate (S/MIME signature). The daily payment report contains all payments made on the date indicated in the report.

**\*** *Payment reports can also be sent using (s)ftp. Contact your account manager for more details.*

Subject lines of email messages are generated by the following model (continuous numbering):

|  |
| --- |
| YANDEX.MONEY DAILY PAYMENT REPORT FOR <*Name\_of\_Counterparty*>. № <*number*> |

The body of email messages are generated as follows:

|  |
| --- |
| YANDEX.MONEY DAILY PAYMENT REPORT FOR <*Name\_of\_Counterparty*>. № <*number*>  Date of payments: <*dd.mm.yyyy*>  Transaction number; Customer ID; Gross payment amount; Payment currency; Payment amount less fee; Date and time of payment; E-wallet number; Description; Payment type  <*Payment details*>  Total gross amount of payments accepted of type <*Payment type*>: <gross amount of payments accepted of given type on the given date>  Total amount of payments less fees of type <*Payment type*>: <total *amount of payments accepted less Operator’s fees of given type*>  Number of payments of type <*Payment type*>: <*quantity of payments of the given type*>  Total gross amount of payments accepted: <gross amount of payments accepted on the given date>  Total amount of payments less fees: <total *amount of payments accepted less Operator’s fees>*  Number of payments: <*quantity of payments*>  To: < *Name\_of\_Counterparty* >  (Under Agent agreement <*number of agreement between Counterparty and Operator*>) |

A description of the fields with the payment information is outlined in the table below.

**Table 6.3.1.** Fields in a standard payment report

|  |  |
| --- | --- |
| **Field** | **Value** |
| Transaction number | Unique transaction number in the Operator IS (string, up to 32 characters). Operator’s notification ‘invoiceId’ parameter value. |
| Customer ID | Payer ID in the Counterparty IS (string, up to 64 characters). ‘customerNumber’ parameter value on the payment form. |
| Gross payment amount | Amount of transaction. Decimal delimiter — a decimal point, always exactly two characters after the decimal point. There is no thousands place delimiter. |
| Payment currency | Three-letter currency code (RUB – Russian Federation ruble). |
| Payment amount less fee | Amount to be transferred to the Counterparty's bank account. Decimal delimiter — a decimal point, always exactly two characters after the decimal point, thousands place delimiter is absent. |
| Date and time of payment | The time when the ‘Notification of payment’ is delivered to the Counterparty (for email-scheme for integration – the time when the payment for an order is registered with the Operator IS). Date and time of payment is displayed in the format dd.mm.yyyy hh:mm:ss according to the time of the Operator IS. |
| E-wallet number | Number of the account in the Operator IS where the payment was made from. |
| Description | Written name of the purchased good in the Operator IS. |
| Payment type | The method that was used to complete the payment. Values correspond to the ‘paymentType’ parameter values (see Table 6.4.1). Optional field. |

Sample payment report:

|  |
| --- |
| Subject: YANDEX.MONEY DAILY PAYMENT REPORT FOR Name\_of\_Counterparty. № 3355  YANDEX.MONEY DAILY PAYMENT REPORT FOR “Name\_of\_Counterparty”. № 3355  Date of payments: 14.03.2014  Transaction number; Customer ID; Gross payment amount; Payment currency; Payment amount less fee; Date and time of payment; E-wallet number; Description; Payment type  549755819524; 4956; 10.00; RUB; 9.50; 18.12.2007 17:46:58; 410038366898; online store payment of services; GP  549755819525; 4957; 15.00; RUB; 14.25; 18.12.2007 17:47:32; 410038366898; online store payment of services; PC  Total gross amount of payments accepted of type PC: 15.00 RUB  Total amount of payments less fees of type PC: 14.25 RUB  Number of payments of type PC: 1  Total gross amount of payments accepted of type GP: 10.00 RUB  Total amount of payments less fees of type GP: 9.50 RUB  Number of payments of type GP: 1  Total gross amount of payments accepted: 25.00 RUB  Total amount of payments less fees: 23.75 RUB  Number of payments: 2  To: Name of Counterparty  (Under Agent agreement 111.1111.11) |

## Payment methods

**Table 6.4.1.** ‘paymentType’ parameter values

|  |  |
| --- | --- |
| **Value** | **Comments** |
| PC | Payment from a Yandex.Money e-wallet. |
| АС | Payment by any bank card. |
| MC | Payment from a mobile phone balance. |
| GP | Payment in cash via retailers and payment kiosks. |
| WM | Payment from a WebMoney e-wallet. |
| SB | Payment via Sberbank Online. |
| MP | Payment via portable terminal (mPOS). |

## Types of data

**Table 7.1.** Definitions of protocol data types

|  |  |
| --- | --- |
| **Type** | **Description** |
| xs:int | 32-bit integer. Int32, as defined in the standard: <http://www.w3.org/TR/xmlschema-2/#int> |
| xs:long | 64-bit integer. Int64, as defined in the standard: <http://www.w3.org/TR/xmlschema-2/#long> |
| xs:decimal | Fixed-point decimal, as defined in the standard: <http://www.w3.org/TR/xmlschema-2/#decimal> |
| xs:boolean | Logical value (true/false), as defined in the standard:  <http://www.w3.org/TR/xmlschema-2/#boolean> |
| xs:string | Text string, as defined in the standard:  <http://www.w3.org/TR/xmlschema-2/#string> |
| xs:normalizedString | Text string, as defined in the standard:  <http://www.w3.org/TR/xmlschema-2/#normalizedString> |
| xs:dateTime | Time stamp of the format according to recommendations of:   * <http://www.w3.org/TR/xmlschema-2/#dateTime> * ISO8601:2004   The format is defined as:  ***YYYY-MM-DDThh:mm:ss.fZZZZZ***  Format breakdown:   |  |  | | --- | --- | | YYYY | year, precisely 4 digits | | *MM* | month, precisely 2 digits (01=January, etc.) | | *DD* | day of the month, precisely 2 digits (from 01 to 31) | | *T* | **Latin** letter ‘T’, must be upper case | | *hh* | hours, precisely two digits (24hrs format, from 00 to 23) | | *mm* | minutes, precisely two digits (from 00 to 59) | | *ss* | seconds, precisely two digits (from 00 to 59) | | *f* | fraction of a second (1 to 6 digits) can be omitted; in such cases the ‘.’ delimiter should be omitted as well. | | *ZZZZZ* | Time Zone descriptor, a universal parameter, can have values:  **Z** – UTC, the ‘Z’ symbol must be in upper case;  **+hh:mm** or -hh:mm – deviation from UTC (shows that the **local time** is displayed, which is the given number of hours and minutes ahead of or behind UTC). |   All the above elements are required; the only element that can be omitted is the fraction of a second (in such cases the ‘.’ delimiter should be omitted as well). If only the date is to be specified, the time of 00:00:00 needs to be entered.  **Examples:**   * 2011-07-24T19:00:00+04:00 – 19 hours 00 minutes 24 July 2011, the time zone is UTC + 4 hrs.; * 2004-07-24T15:00:00Z – the same moment of time on the canonic format; * 2004-07-24T15:00:00.666Z – the same moment of time plus 666 milliseconds. |
| YMAccount | Virtual account number in the Operator IS, a string of decimal digits from 11 to 33 symbols long.  <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 decimal fixed-point number with precisely two digits after the decimal point.  <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 — Russian Federation ruble; * 10643 — test currency (demo-rubles of the Yandex.Money demo-system).   <xs:simpleType name="CurrencyCode">  <xs:restriction base="xs:int">  </xs:restriction>  </xs:simpleType> |
| CurrencyBank | Operator’s processing center code. Possible values:   * 1001 – EcomBank; * 1003 – DemoBank of the Yandex.Money demo-system.   <xs:simpleType name="CurrencyBank">  <xs:restriction base="xs:int">  </xs:restriction>  </xs:simpleType> |