

Integration with Custom Payment System



- [1. Issues under consideration](#)
 - [2. Overview](#)
 - [Terminology](#)
 - [Setup](#)
 - [3. General protocol description](#)
 - [3.1. An example of interaction between Payer, Payment System, Contractor and OpenTrade Commerce when making payment.](#)
 - [3.2. An example of interaction between Payment System, Contractor and OpenTrade Commerce when notifying a customer about payment.](#)
 - [3.3 Tables](#)
 - [3.3.1 Payment parameters](#)
 - [3.3.2 NoticeErrorCode description](#)
 - [3.4. Rules for elaborating a digital signature when the contractor notifies OpenTrade Commerce](#)
-

1. Issues under consideration

If you do not find a payment system in the list of available ones, you can set up any other payment system using the OpenTrade Commerce services interaction protocol.

2. Overview

OpenTradeCommerce provides a protocol that enables the contractor to integrate payments methods unavailable to OpenTradeCommerce customers into the order and funds flow management system.

Terminology

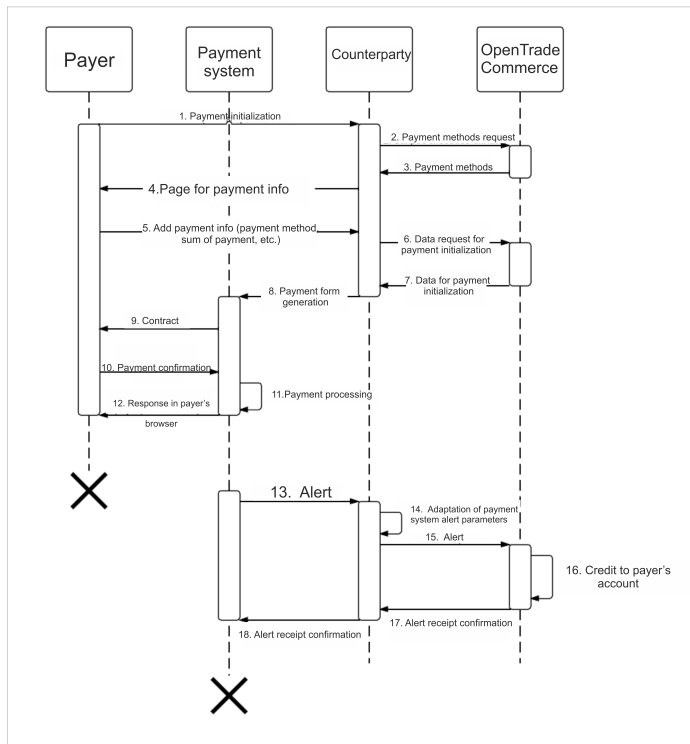
- **"OpenTrade Commerce"** means an order and funds flow management system. Billing.
- **"Contractor"** means an OpenTradeCommerce customer.
- **"Payer"** means a contractor's Web-resource user that makes a payment.
- **"PS"** means a payment system.

Setup

In order to set up a payment system using a protocol provided you will need to:

- Be an OpenTradeCommerce customer
- Submit a service request for custom PS integration to [OpenTrade Commerce technical support](#)
- Receive a payment secret (3.3.1). Subsequently the contractor is responsible for protecting information from disclosure.

3. General protocol description



A flowchart of interaction of all parties involved in payment process

The interaction process is a generalized example. There is a certain amount of flexibility in how these phases are implemented and when they occur in real-life situations of interaction.

Below is the step-by-step description of interaction process.

3.1. An example of interaction between Payer, Payment System, Contractor and OpenTrade Commerce when making payment.

Step 1. The payer initiates the payment process.

Step 2-4. The contractor's website will ask OpenTrade Commerce for payment methods using the GetPaymentModes method and show them to the payer.



In response the GetPayment Modes method will return the following identifier for a payment method intended for a custom PS : opentao.custom.

Step 5. The payer will then select a payment method and enter payment data such as amount of payment, etc.

Step 6. The contractor's website will request OpenTrade Commerce accounting information to initiate a payment through PS using the GetPaymentParameters method.

Step 7. OpenTrade Commerce will send back the payment data in response to the GetPaymentParameters method invocation.

PaymentFormAnswer response sample

```

<?xml version="1.0" encoding="utf-8"?>
<PaymentFormAnswer>
  <ErrorCode>Ok</ErrorCode>
  <Result>
    <RequestMethod>POST</RequestMethod>
    <RequestUrl>http://example.com/test.php</RequestUrl>
    <Parameters>
      <Parameter>
        <Name>paymentId</Name>
        <Value>222</Value>
      </Parameter>
      <Parameter>
        <Name>userId</Name>
        <Value>0000000001</Value>
      </Parameter>
      <Parameter>
        <Name>amount</Name>
        <Value>500.15</Value>
      </Parameter>
      <Parameter>
        <Name>currency</Name>
        <Value>643</Value>
      </Parameter>
      <Parameter>
        <Name>description</Name>
        <Value>Top up the account USR-0000000001 (user 0000000001, payment 222)</Value>
      </Parameter>
    </Parameters>
  </Result>
</PaymentFormAnswer>

```

where

Parameter	Description	
RequestUrl	Test page address for testing a custom payment system to be implemented	
RequestMethod	Communication method	GET/POST
Parameter	Payment parameter which is KeyValuePair	
Parameter.Name	Parameter name	
Parameter.Value	Parameter value	

Variants of parameters names (for description see 3.3.1)

Parameter name	Optional
orderID	+
PaymentID	
UserID	
Amount	
Currency	
Description	
SuccessUrl	
failUrl	
resultUrl	

Step 8. The contractor will generate a payment form in accordance with the payment system requirements on the basis of the data provided in Step 7.

- Step 9.** After you have provided all the required data by filling out the form the payment system will form a contract, send it to the payer and ask to confirm the payment.
- Step 10.** The payer will confirm the payment.
- Step 11.** The payment system will accept and verify the payment.
- Step 12.** The payment system will return a response to the payer’s browser with the result of the payment (successful/failed).

3.2. An example of interaction between Payment System, Contractor and OpenTrade Commerce when notifying a customer about payment.

- Step 13. The payment system notifies the contractor of completed payment.
- Step 14. The contractor will adapt the data provided in Step 13 so that they match the OpenTrade Commerce notification parameters.
- Step 15. The contractor will notify OpenTrade Commerce of the completed payment.

It is necessary to pass the following parameters to the notification address (resultUrl provided to OpenTrade Commerce in Step7) using the HTTP transport.

Parameters (for description see 3.3.1):

Parameter Name	Optional
instancekey	
orderId	+
payment ID	
userID	
amount	
currency	
status	
signature	

- Step 16.** OpenTrade Commerce will verify the notification parameters. If everything checks out and the payment status is considered “Completed”, OpenTrade Commerce will enter a sum to the payer’s account.
- Step 17.** OpenTrade Commerce will confirm the contractor that the payment notification has been received.

Error-free notification processing response

```
<?xml version="1.0" encoding="utf-8"?>
<NoticeAnswer>
  <PaymentId>222</PaymentId>
  <ErrorCode>Ok</ErrorCode>
</NoticeAnswer>
```

Response with error code

```
<?xml version="1.0" encoding="utf-8"?>
<NoticeAnswer>
  <PaymentId>222</PaymentId>
  <ErrorCode>VerificationError</ErrorCode>
  <ErrorDescription>Unknown notification status: 'Overpaid'</ErrorDescription>
</NoticeAnswer>
```

Where

Parameter	Description
Payment ID	Unique ID identifying payment at OpenTrade Commerce internal billing system
ErrorCode	Error code. For details refer to 3.3.2
ErrorDescription	Error description

Step 18. The contractor will confirm the PS that the payment notification has been received

3.3 Tables

3.3.1 Payment parameters

Parameter	Description	Allowed Values	Note
Instancekey	Unique ID identifying contractor		
secret	Payment secret		Required for creation of a digital signature 3.4. Provided to the Contractor by Opentrade Commerce
orderId	Unique ID identifying order for payment		Available if a customer pays for an order Unavailable when replenishing
paymentId	Unique ID identifying payment at OpenTrade Commerce internal billing system		
userId	Unique ID identifying OpenTrade Commerce user		
Amount	Amount of payment		The point as a decimal mark and two digits after it
Currency	ISO 4217 (3-letter currency code)	643 - RUB	http://en.wikipedia.org/wiki/ISO_4217
Status	Payment status	Completed Canceled	
Signature	MD5-hash of digital signature		32 hexadecimal uppercase symbols
description	Payment description		

3.3.2 NoticeErrorCode description

Code	Description
OK	A notification has been processed.
VerificationError	Incoming data verification error
SignatureVerification Error	There is a mismatch between the digital signature received from the contractor and the one calculated by OpenTrade Commerce
InternalError	OpenTrade Commerce service internal error. Unhandled error

3.4. Rules for elaborating a digital signature when the contractor notifies Opentrade Commerce

The MD5-hash algorithm is applied to a text represented as a sequence of parameter values separated by semicolon.

OrderId; paymentId; amount; currency; status; secret

Example:

String	Result
111;222;0000000001;500.15;643;Completed;secret	11AE0ABC8F0CF443F950D84C278F1C51