Merchant API Reference

Contents

Chapter 1

Overview

Topics:

Interface Protocols and Standards

The interaction between MONETA.RU and external system is built on basis of SOAP or JSON and described via WSDL document.

SOAP

The WSDL defines services as collections of network endpoints. An endpoint is defined by associating a network address with a reusable binding. The collection of endpoints defines the service. Messages are abstract descriptions of the data being exchanged, and endpoint types are abstract collections of supported operations. The concrete protocol and data format specifications for a particular endpoint type constitutes a reusable binding, where the operations and messages are bound to a concrete network protocol and message format. In this way, WSDL describes the public interface to the web service.

WSDL is used in combination with SOAP and XML Schema to provide web services over the Internet. A client system connecting to a web service can read the WSDL to determine what functions are available on the server. Any special data types used are embedded in the WSDL file in the form of XML Schema. The client can then use SOAP to actually call one of the functions listed in the WSDL.

SOAP is protocol for exchanging XML-based messages over computer networks using HTTPS. SOAP forms the foundation layer of the web services protocol stack providing a basic messaging framework upon which abstract layers can be built.

WSDL URL: *https://service.moneta.ru/services.wsdl*

Service URL: *https://service.moneta.ru/services*

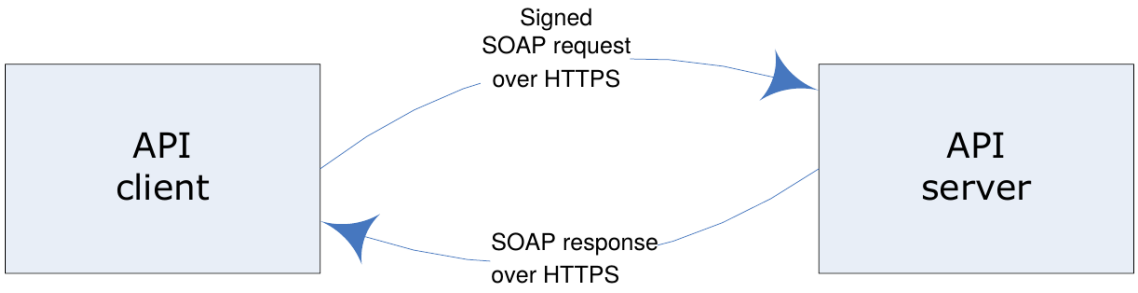
WSDL URL (x509): *https://service.moneta.ru:8443/services/x509.wsdl*

Service URL (x509): *https://service.moneta.ru:8443/services/x509*

HTTP method: POST

Content-type: text/xml;charset=UTF-8

SOAPAction: ""



JSON

JSON is an open standard format that uses human-readable text to transmit data objects consisting of attribute-value pairs.

WSDL URL: *https://service.moneta.ru/services.wsdl*

Service URL: *https://service.moneta.ru/services*

Service URL (x509): *https://service.moneta.ru:8443/services/x509*

HTTP method: POST

Content-type: application/json;charset=UTF-8

Chapter 2

Security

Topics:

The MONETA.MerchantAPI service is protected to make sure that only authorized “Moneta.ru” members use it.

There are two primary levels of security:

- Secure Sockets Layer (SSL) data transport
- API username/password or client certificate

A failure of authenticated security at any one of these levels denies access to the MONETA.MerchantAPI.

SSL Data Transport

All data must be transported over the Secure Hyper Text Transport Protocol (also known as HTTPS), which relies on the Secure Sockets Layer (SSL) data communications protocol.

UsernameToken Authentication

MONETA.RU can authenticate SOAP requests to the Merchant API by a username and password that are provided in SOAP Message Security headers.

For more information about SOAP security headers, read *Web Services Security UsernameToken Profile 1.1*.

MONETA.RU can authenticate JSON requests to the Merchant API by a username and password that are provided in request.

If you use security headers to authenticate your requests, use the following Merchant API URLs:

- WSDL URL: *https://service.moneta.ru/services.wsdl*
- Service URL: *https://service.moneta.ru/services*

Request Credentials

MONETA.RU authenticates each merchant's request with client certificate or username and password pair set in SOAP/JSON request header.

The following code shows a sample security header of a SOAP request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
    <wsse:Security soapenv:mustUnderstand="1"
            xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-
            200401-wss-wssecurity-secext-1.0.xsd">
      <wsse:UsernameToken wsu:Id="UsernameToken"
            xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-
            200401-wss-wssecurity-utility-1.0.xsd">
        <wsse:Username>MERCHANT'S USERNAME</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-
            200401-wss-username-token-profile-
            1.0#PasswordText">PLAIN PASSWORD</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soapenv:Header>
```

```
  <soapenv:Body>
    ...
  </soapenv:Body>
</soapenv:Envelope>
```

If the request cannot be authenticated, a SOAP security fault is returned:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
 <SOAP-ENV:Header/>
 <SOAP-ENV:Body>
   <SOAP-ENV:Fault>
     <faultcode>SOAP-ENV:Client</faultcode>
     <faultstring xml:lang="en">com.sun.xml.wss.impl.WssSoapFaultException:
       Authentication of Username Password Token Failed; nested exception
       is com.sun.xml.wss.XWSSecurityException:
       com.sun.xml.wss.impl.WssSoapFaultException:
       Authentication of Username Password Token Failed</faultstring>
     <detail>
       <messages:faultDetail xmlns:messages="http://www.moneta.ru/schemas/
            messages.xsd">300.1</messages:faultDetail>
     </detail>
    </SOAP-ENV:Fault>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Important: HTTP status code is 500

The following code shows a sample security header of a JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "MERCHANT'S USERNAME",
                "Password": "PLAIN PASSWORD"
            }
        }
    },
    "Body": {
        ...
}}}
```

If the request cannot be authenticated, a JSON security fault is returned:

```
{"Envelope": {
    "Body": {
        "fault": {
            "detail": {
                "faultDetail": "300.1"
            },
            "faultcode": "Client",
            "faultstring": "Wrong credentials"
        }
    }
}}
```

Important: HTTP status code is 400

Certificate Authentication

You can obtain an SSL certificate from MONETA.RU to authenticate your requests to the Merchant API.

If you use a certificate to authenticate your requests, use the following Merchant API URLs:

- WSDL URL: *https://service.moneta.ru:8443/services/x509.wsdl*
- Service URL: *https://service.moneta.ru:8443/services/x509*

Obtaining a Certificate

If you want to use authentication by certificates, you must obtain a certificate from MONETA.RU.

1. To generate a certificate request, perform the following substeps:
   a) Install the OpenSSL toolkit.
   b) Generate a certificate request by running the following command:

      ```
      openssl req -new -newkey rsa:2048 -sha512 -nodes -out request.txt -keyout
       private.key
      ```

      OpenSSL generates the certificate request in the request.txt file. Use this certificate request to get your certificate from MONETA.RU.
2. Obtain a certificate from MONETA.RU. Perform the following substeps:
   a) Log in to the MONETA.RU web site.
   b) Navigate to the Security web page.

      Go to **My account** > **Security**.
   c) Click **Add certificate**.
   d) Copy the certificate request from the request.txt file that you generated in Step 1 to the **Certificate signing request** field and click **Next**.

      > **Important:** Copy the entire contents of the file, including the `BEGIN CERTIFICATE REQUEST` and `END CERTIFICATE REQUEST` lines.

   e) Review the certificate information and click **Save**.
      MONETA.RU generates the certificate based on your certificate request.
   f) To download the certificate, click the **Download certificate** link.
   g) Optionally, create a PKCS12 wallet with the certificate. Run the following command:

      ```
      openssl pkcs12 -export -clcerts -in certificate.pem -inkey private.key -
      out wallet_name.p12 -name user_name
      ```

   h) Optionally, import the PKCS12 wallet in your web browser to open MONETA.RU without entering your user name and password.

Chapter 3

Faults

Topics:

Overview of SOAP/JSON Faults

If an error occurs when processing a SOAP request, the MONETA.RU server returns HTTP status 500 and a SOAP Fault element that provides information about the error. If an error occurs when processing a JSON request, the MONETA.RU server returns HTTP status 400 and a fault element that provides information about the error.

The following example shows a generic SOAP Fault that MONERA.RU returns if an error occurs when processing a request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>Fault code</faultcode>
      <faultstring xml:lang="en">Error message</faultstring>
      <detail>
        <messages:faultDetail xmlns:messages="http://www.moneta.ru/schemas/
            messages.xsd">Error code</messages:faultDetail>
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The following example shows a generic JSON Fault that MONERA.RU returns if an error occurs when processing a request:

```
{"Envelope": {
    "Body": {
        "fault": {
            "detail": {
                "faultDetail": "ERROR CODE"
            },
            "faultcode": "FAULT CODE",
            "faultstring": "ERROR MESSAGE"
        }
    }
}}
```

Each Fault contains the following elements:

- **Fault Code**. Identifies the fault. The SOAP fault code consists of a prefix and a local name. The prefix is `SOAP-ENV`. The following table describes local name values:

Local name	Description
VersionMismatch	A invalid namespace for the SOAPEnvelope object.
MustUnderstand	The mustUnderstand attribute of the SOAPHeader child is set to true, but MONETA.RU either does not understand or ignores this attribute.
Client	The SOAP request was invalid or contained insufficient information.
Server	An error occurred when processing a valid SOAP request.

- **Fault string**. Provides a text description of the error.
- **Detail object**. Provides an error code from the Merchant API. For the complete list of error codes, see *Error Codes* on page 13.

The following example shows a SOAP Fault that was returned because of a request validation error:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <SOAP-ENV:Body>
      <SOAP-ENV:Fault>
         <faultcode>SOAP-ENV:Client</faultcode>
         <faultstring xml:lang="ru">Transaction is not found</faultstring>
         <detail>
            <messages:faultDetail xmlns:messages="http://www.moneta.ru/schemas/
messages.xsd">500.1.5</messages:faultDetail>
         </detail>
      </SOAP-ENV:Fault>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The following example shows a JSON Fault that was returned because of a request validation error:

```
{"Envelope": {
    "Body": {
        "fault": {
            "detail": {
                "faultDetail": "500.1.5"
            },
            "faultcode": "Client",
            "faultstring": "Transaction is not found"
        }
    }
}}
```

Error Codes

If an error occurs when processing a request, the response to this request contains the detail element. For more information, see *http://www.w3.org/TR/2000/NOTE-SOAP-20000508/#_Toc478383507*.

The detail element might contain the faultDetail element that describes the error code.

SOAP example:

```
<Fault>
    <faultcode>Client</faultcode>
    <faultstring>Transaction is not found.</faultstring>
    <detail>
        <faultDetail>500.1.5</faultDetail>
    </detail>
</Fault>
```

JSON example:

```
{"fault": {
        "detail": {
            "faultDetail": "500.1.5"
        },
        "faultcode": "Client",
        "faultstring": "Transaction is not found"
}}
```

The following table lists MONETA.RU error codes:

100	Service is temporarily unavailable.
100.1	Specified values already exist.
100.2	You are using expired information.
100.3	Errors in the following user profile: *unitId*.
100.4	Errors in the accounts of the following user: *unitId*.
100.5	Errors in the structure of the following user: *unitId*.
200	XSD validation error.
300.1	Wrong credentials.
300.1.1	Wrong onetime password.
300.1.2	Unknown device.
300.1.3	The account is locked.
300.2	Access is denied.
300.3	Access is denied.
300.4.1	You have no access to the specified template.
300.4.2	Access to the order is denied.
300.4.3	Access to the transaction is denied.
300.4.4	You cannot edit the specified account.
300.4.5	You cannot create a profile of the specified type.
300.5	User is not found.
300.6	Cannot modify approved property.
300.7	Login from current IP is denied.
400.1.1	Transaction is not allowed.
400.1.2	Transaction is pending confirmation.
400.1.3	Transaction is queued for processing.
400.1.4	Transaction is being processed.
400.1.5	Payment canceled.
400.1.6	This transaction is already paid.
400.1.7	Transaction is not allowed.
400.1.8.1	Cannot refund an incomplete transaction.
400.1.8.2	Cannot refund a transaction. Payee unavailable.
400.1.8.3	Cannot refund a transaction.
400.1.8.4	Cannot refund a transaction.
400.1.9	Error when communicating with the payment system of the payee.

400.1.10	Transferring funds to the specified account is not allowed or incorrect amount was specified.
400.1.10.1	Only identified MONETA.RU users can perform this type of transaction. To complete simplified identification, specify the following information in your MONETA.RU account: full name, passport details, tax identification number, insurance number of individual ledger account, and mobile phone number.
400.1.10.2	The account of the recipient is anonymous. Federal law 161-FZ "On the National Payment System" dated June 27, 2011, Article 7, Part 2.1. prohibits funding electronic means of payment in this case.
400.1.11	Account is blocked.
400.1.12	Insufficient funds for the transfer.
400.1.13	You cannot cancel this transaction.
400.1.14	The last payment password is reserved for obtaining a new set of payment passwords.
400.1.15	Protection code expired.
400.1.16	Amount in a request cannot exceed amount in a transaction.
400.1.17	Error when communicating with the payment system of the payer.
400.1.18	Wrong transaction amount. For example, the transaction amount which is sent to MONETA.RU does not match the amount that the merchant returns to the check request.
400.1.19	Transaction that has the specified external transaction ID is being processed or already completed. MONETA.RU Does not allow to process multiple transactions that have the same external transaction ID.
400.1.20	Insufficient processing rights.
400.1.21	Transaction is frozen.
400.1.22	Payment invoice is canceled.
400.1.23	Payment password is expired.
400.1.24	Recurrent payment not allowed.
400.1.25	Retry with a different card number is impossible, this transaction is now canceled, you can try again, another transaction will be created.
400.1.26	Transactions with this account are not allowed. Please, contact Commercial department.
400.1.26.1	Withdrawal is not allowed. Please, contact Commercial department.
400.1.26.2	Deposit is not allowed. Please, contact Commercial department.
400.1.27	Inventory was not provided by merchant.
400.2	Transaction is queued.
400.2.1	Transaction is queued.
400.2.1.1	Transaction is not allowed.

400.2.2	Transaction is queued.
400.2.2.1	Transaction is not allowed.
400.2.3	Transaction is queued.
400.2.3.1	Transaction is not allowed.
400.2.4	Amount limits are exceeded.
400.2.4.1	Amount limits are exceeded.
400.2.4.2	You exceeded the limit on transactions execution.
400.2.5	Transaction is queued for processing.
400.2.5.1	Transaction is not allowed.
400.2.5.2	Transfer is possible only to the card which was used for deposit.
400.2.6	Amount limits are exceeded.
400.2.6.1	Amount limits are exceeded.
400.2.7	Transaction is queued for processing.
400.2.7.1	Transaction is not allowed.
400.2.7.2	Invalid payment purpose.
400.2.7.3	Invalid payment description.
400.2.7.4	Payments are accepted only from russian 3DSecure cards.
400.2.7.5	Payments are accepted only from russian cards or 3DSecure cards.
400.2.7.6	Payments are accepted only from 3DSecure cards.
400.2.7.7	Only identified MONETA.RU users can perform this type of transaction. To complete simplified identification, specify the following information in your MONETA.RU account: full name, passport details, tax identification number, insurance number of individual ledger account, and mobile phone number.
400.2.7.8	Card country is not allowed.
400.2.7.9	You set the limit on transactions execution.
400.2.8.1	Encryption error.
400.2.8.2	Decryption error.
500	Data validation error.
500.1	Cannot process the request because MONETA.RU cannot find the specified object.
500.1.1	Specified account does not exist.
500.1.2	Account with alias *alias* is not found.
500.1.3	Account *accountId* is not found.
500.1.4	Account is not found.
500.1.5	Transaction is not found.
500.1.6	Payer's account number is not found.
500.1.7	Payee's account number is not found.
500.1.8	Document is not found.

500.1.8.1	Wrong filename. Supported formats: jpg,png,bmp,pdf,doc,odt.
500.1.9	Profile information is not found.
500.1.10	Wrong order ID.
500.1.11	Wrong transaction ID.
500.1.12	The specified template ID is not found.
500.1.13	The specified template does not contain the specified parameters.
500.1.14	User is not found.
500.1.15	Wrong structure ID.
500.1.16	The SOURCEACCOUNTID parameter must point to the account to which you have access.
500.1.17	The SOURCEACCOUNTID parameter must be {TARGETACCOUNTID} or point to the account to which you have access.
500.1.18	Cannot fetch archived data.
500.1.18.1	Cannot fetch archived data. Specify a date later than *dd.mm.yyyy*.
500.1.19	Bank account is not found.
500.1.20	Bank account ID is empty.
500.1.21	User doesn't have Public ID.
500.1.22	Secure token data not found.
500.1.23	Cannot process the request because MONETA.RU cannot find the specified object property.
500.1.24	Civil passport with specified series and number already exists.
500.1.25	Juridical account is not found.
500.1.26	Juridical account ID is empty.
500.1.27	Juridical account already exists.
500.2	Wrong parameter value.
500.2.1	The *element_name* element is required.
500.2.1.1	The * element is wrong.
500.2.2	Required field.
500.2.3	The *element_name* element is required.
500.2.4	The *element_name* element must be null.
500.2.5	The *element_name* element must be null or empty.
500.2.6	The *element_name* and *element_name* values do not match in the request and response.
500.2.7	Wrong format.
500.2.7.1	Wrong boolean format: *.
500.2.7.2	Wrong number format: *.
500.2.8	Some required parameters are missing.
500.3.1.1	Wrong payment password.

500.3.1.2	The specified amount is not correct.
500.3.1.3	Invalid payer's account number.
500.3.1.4	Invalid payee's account number.
500.3.1.5	Invalid payer's amount.
500.3.1.6	Invalid payee's amount.
500.3.1.7	Amount cannot exceed *amountcurrency*.
500.3.1.8	Amount must be greater than *amountcurrency*.
500.3.1.9	Cannot determine the currency of the transaction amount. Use the isPayerAmount element to specify the currency in your requests.
500.3.1.10	You cannot use the transactionId element if the transactional element is set to true.
500.3.1.11	Wrong protection code.
500.3.1.12	Could not confirm the transaction because the transactionId and "ID" attribute's value of operationInfo object in your request do not match.
500.3.1.13	Amount must be a positive number.
500.3.1.17	The amount element is required. Specify a zero or a positive number.
500.3.1.18	The specified transaction ID already exists.
500.3.1.19	Too many transactions.
500.3.1.20	Transfers to external payment systems are not allowed in batch processing mode.
500.3.1.21	The amount element is missing.
500.3.1.22	Merchant transaction ID is missing.
500.3.1.23	The transaction has a different External transaction ID of the payer.
500.3.1.24	The transaction has a different External transaction ID of the payee.
500.3.1.25	Card number is required.
500.3.1.26	Expiration date is required.
500.3.1.27	CVV2/CVC2 is required.
500.3.1.28	Wrong card number format.
500.3.1.29	Wrong card expiration date (MM/YYYY).
500.3.1.30	Wrong CVV2/CVC2 format. It must contain 3 digits.
500.3.1.31	You cannot add a payment request and a confirmation request for the payment to the same batch request.
500.3.1.32	Unable to set CVV2/CVC2 code on invoice creation.
500.3.1.33	Wrong card holder name format (latin letters, first name and last name with the space between).
500.3.1.34	You cannot use transactional = true in this request.
500.3.1.35	Card holder name is required.
500.3.1.36	You must have PCI DSS certification passed.
500.3.1.37	Expired card.

500.3.1.38	Too long card holder name (max. * characters).
500.3.1.39.1	Payment session is complete or client IP address has changed.
500.3.1.39.2	Wrong merchant ID.
500.3.1.39.3	Merchant temporary unavailable.
500.3.1.39.4	Merchant transaction ID already exists.
500.3.1.39.5	Amount must be a positive number.
500.3.1.39.6	Mismatched amount. Use original value.
500.3.1.39.7	Currency mismatch.
500.3.1.39.8	Unknown currency.
500.3.1.39.9	Missing SUCCESS_URL.
500.3.1.39.10	Missing FAIL_URL.
500.3.1.39.11	Custom parameters size limit exceeded (1.5KB).
500.3.1.39.12	Wrong processing parameters.
500.3.1.39.13	Test mode on the account is not set.
500.3.1.39.14	Test mode on the account is on.
500.3.1.39.15	Missing payment system.
500.3.1.39.16	Missing payment system account.
500.3.1.39.17	No available payment systems.
500.3.1.39.18	Canceled merchant transaction ID.
500.3.1.39.19	Transaction is in progress or paid. You should go back to the shop and create a new payment.
500.3.1.39.20	Wrong transaction.
500.3.1.39.21	Processing of this operation is impossible. You should go back to the shop and create a new payment.
500.3.1.39.22	Wrong subscriber ID.
500.3.1.39.23	Wrong provider ID.
500.3.1.39.24	Refund request received too late.
500.3.1.39.25	Browser fingerprint required.
500.3.1.39.26	Merchant transaction ID is too long.
500.3.1.39.27	Card not allowed for this payment
500.3.1.40	Phone number for SMS payment password is not found.
500.3.2.1	You cannot save the new password because the old password was incorrect.
500.3.2.2	Account alias is missing.
500.3.2.3	Wrong document type.
500.3.2.4	Unknown profile type.
500.3.2.5	Wrong date of birth.
500.3.2.6	Wrong sex value.
500.3.2.7	Wrong phone number format.
500.3.2.8	Wrong e-mail format.

500.3.2.9	Payment password must contain minimum five characters.
500.3.2.10	Payment password length must not exceed 32 characters.
500.3.2.11	Payment password must contain numbers only.
500.3.2.12	Wrong URL format.
500.3.2.13	A URL must start with http[s]:// or template://.
500.3.2.14	Account alias must be unique. You already have an account with the following alias: *.
500.3.2.15	An error occurred when validating document elements.
500.3.2.15.1	Indicated date has not come yet. Insert the correct date. "*" field.
500.3.2.15.2.1	Document series is a required field.
500.3.2.15.3.1	Document number is a required field.
500.3.2.16	Last name is a required field.
500.3.2.17	Name is a required field.
500.3.2.18	Alias length must not exceed 64 characters.
500.3.2.19	You must be at least 14 years old to use MONETA.RU.
500.3.2.20	Date of birth cannot be earlier than 1900.
500.3.2.21	Specify a correct 10-digit or 12-digit TIN.
500.3.2.21.1	Specify a correct 10-digit TIN.
500.3.2.21.2	Specify a correct 12-digit TIN.
500.3.2.21.3	One of fields is required: INN or SNILS.
500.3.2.22	Wrong country for this region.
500.3.2.23	Wrong region for this city.
500.3.2.24	Specify a phone number in the international format. For example, 71234567890.
500.3.2.25	Wrong URL. Use the following format: http:// www.site.com or http://site.com.
500.3.2.26	Invalid e-mail address. Use the following format: user@site.com.
500.3.2.27	Company name is a required field.
500.3.2.28	Director's full name is a required field.
500.3.2.29	You cannot deny access for this user.
500.3.2.30	Wrong status ID.
500.3.2.31	Alias already exists.
500.3.2.32	Wrong value for the low threshold.
500.3.2.33	Access is empty.
500.3.2.34	Specify a different prototype account.
500.3.2.35	The specified account refers to another prototype account. Specify a different account.

500.3.2.36	You cannot create an account with the specified currency or the specified type.
500.3.2.36.1	Your request for opening an account with NCO MONETA (LLC) has been accepted.
500.3.2.37	Available only for accounts that use static payment passwords.
500.3.2.38	Wrong template format. Use the following format: template://template_id?param_name=param_value.
500.3.2.39	The specified account has a different type. Specify a different account.
500.3.2.40	UnitId must not match ParentId.
500.3.2.41	Could not identify the user.
500.3.2.42	Wrong identification code.
500.3.2.43	Registration address is a required field.
500.3.2.44	Date of birth is a required field.
500.3.2.45	You cannot change an approved property.
500.3.2.46	Wrong SNILS.
500.3.2.47	Mobile phone number is not specified.
500.3.2.48	Phone number is already confirmed.
500.3.2.48.1	Phone number is already disapproved.
500.3.2.49	Message does not contain the {CODE} placeholder.
500.3.2.50	SMS message is too long.
500.3.2.51	Wrong confirmation code.
500.3.2.51.1	Wrong disapproval code.
500.3.2.52.1	Valid 9-digit BIK is required in "*" field.
500.3.2.52.2	Settlement account number is not correct in "*" field. Specify the correct account number.
500.3.2.52.3	You cannot use the settlement account in "*" field because it is intended for individuals.
500.3.2.52.4	You specified the BIK number for the payment processing center of the Bank of Russia. In this case, you must leave the Correspondent account field empty.
500.3.2.52.5	Correspondent account number is not correct. Specify a correct correspondent account number.
500.3.2.52.6	Wrong Budgetary Classification Code format.
500.3.2.52.7	Wrong format of Russian National Classifier of Municipalities Territories in "*" field.
500.3.2.52.8	Wrong format of Tax Registration Reason Code in "*" field. Specify a 9-digit code.
500.3.2.52.9	Wrong SWIFT format. Specify an 8 or 11-symbol code that consists of upper-case Latin characters and digits.
500.3.2.52.10	Wrong IBAN format in "*" field. Use a string of upper-case Latin characters and digits that contains 5-40 symbols.
500.3.2.52.11	Wrong currency.

500.3.2.53	This type of payment password is unavailable.
500.3.2.54	You cannot use square brackets in alias.
500.3.2.55	The *element_name* length must not exceed *length* characters.
500.3.2.57	Wrong Secure token format.
500.3.2.58	Secure Token is expired.
500.3.2.59.1	Wrong code of data integrity verification.
500.3.2.59.2	Payment form signature is mandatory.
500.3.2.60	You cannot change the type of units of the entries of this kind.
500.3.2.61	Wrong account type.
500.3.2.61.1	Wrong account subtype.
500.3.2.62.1	Can't block this account, because it's not active.
500.3.2.63.1	Can't unblock this account, because it's not blocked.
500.3.2.63.2	Unable to unblock account because the secret question is not specified. Apply to customer support.
500.3.2.63.3	Unable to unblock account because you did not specify an answer to secret question. Apply to customer support.
500.3.2.63.4	This account is not blocked.
500.3.2.63.5	This account cannot be automatically unblocked. Apply to customer support.
500.3.2.63.6	The usage of your answer for secret question is blocked for one hour. Try later or apply to customer support.
500.3.2.63.7	The usage of your payment password for account activation is blocked for one hour. Try later or apply to customer support.
500.3.2.63.8	Wrong answer to secret question.
500.3.2.63.9	The profile is not blocked.
500.3.2.63.10	Unable to unblock profile because the secret question is not specified for the user with given e-mail or phone address. Apply to customer support.
500.3.2.63.11	Unable to unblock profile because you did not specify an answer to secret question. Apply to customer support.
500.3.2.63.12	This profile cannot be automatically unblocked. Apply to customer support.
500.3.2.63.13	Secret answer required.
500.3.2.64.1	Wrong Interface type.
500.3.2.64.2	Wrong payment method IDs.
500.3.2.64.3	Wrong default payment method.
500.3.2.64.4	Wrong HTTP method.
500.3.2.64.5	Wrong target.
500.3.2.64.6	Exceeded maximum number of characters.

500.3.2.64.7	You cannot use square brackets in alias.
500.3.2.64.8	Wrong MONETA.Assistant version.
500.3.2.65.1	Malformed URL.
500.3.2.65.2	Unable to establish connection.
500.3.2.65.3	Server certificate expired.
500.3.2.65.4	Unable to verify server certificate certification authority.
500.3.2.65.5	Server certificate not yet valid.
500.3.2.66.1	Can't change payment password.
500.3.2.66.2	New type of payment password is required.
500.3.2.66.3	Accounts belong to different users.
500.3.2.66.4	Accounts with different payment passwords are selected.
500.3.2.66.5	Too many invalid codes. Try to fill the form once again and receive the code in new SMS.
500.3.2.66.6	Select one account.
500.3.2.66.7	Wrong one-time password. Check secret key (encoded in Base32), time interval - 30 sec, length - 6 digits, algorithm - SHA-1. Be sure your system clock is synchronized with global NTP services.
500.3.2.66.8	Select one account.
500.3.2.67	No account selected.
500.3.2.68.1	Wrong token.
500.3.2.69	Incorrect CSS. Consult customer support.
500.3.2.70.1	Password length between 8 and 32 characters.
500.3.2.70.2	Password must contain at least 1 class of symbols.
500.3.2.70.3	Mentioning a symbol in password any number of times.
500.3.2.70.4	Number of at least 1 symbol of each class in password.
500.3.2.70.5	A symbol can be repeated in series not more than 3 times in password.
500.3.2.71	Wrong code of organization legal form in "*" field. See documentation.
500.3.2.71.2	E-mail address is already registered, enter other address.
500.3.2.71.3	Phone number is already registered, enter other number.
500.3.2.72	Incorrect classifier of organization legal form. Use classifier 5 for private entrepreneur. Use other classifiers for organization.
500.3.2.72.1	Activation request is expired or not found.
500.3.2.72.2	Can't use this activation request.
500.3.2.72.3	User was activated already.
500.3.2.73.1	9 digit CPR is required.

500.3.2.73.2	Requires a valid 13-digit or 15-digit BIN.
500.3.2.73.3	Requires a valid 13-digit BIN.
500.3.2.73.4	Requires a valid 15-digit BIN.
500.3.2.73.5	Requires the correct 8-digit or 10-digit OKPO.
500.3.2.73.6	Requires the correct 8-digit OKPO.
500.3.2.73.7	Requires the correct 10-digit OKPO.
500.3.2.73.8	Requires a valid RNCATO (from 2 to 11 digits).
500.3.3.1	Amount 'from' is greater than amount 'to'.
500.3.3.2	The start date is later than the end date.
500.3.3.3	Transaction history period must not exceed one month.
500.3.3.4	Amount must not be negative.
500.3.3.5	If you specify amount, you must also specify currency.
500.3.3.6	Wrong transaction amount type.
500.3.3.7	Wrong currency code.
500.3.3.8	Period is not specified.
500.3.3.9	Wrong period is specified.
500.3.3.10	Wrong operation type.
500.3.3.11	You cannot view financial flows for more than three months.
500.3.3.12	The start date is later than the end date.
500.3.3.13	You cannot view turnovers grouped by day for more than one month.
500.3.3.14	You cannot view financial flows grouped by day for more than one month.
500.3.3.15	You cannot view the history of transactions for more than one year.
500.3.3.16	Payments from the same account are not allowed.
500.4.1.1	Wrong personal account number. Account not found.
500.4.1.2	Wrong format "*".
500.4.1.7	Service provider was not found.
500.4.2.1	Amount is required.
500.4.2.2	The amount of payment should be in the range of * to *.
500.4.2.3	Protocol already paid.
500.4.2.4	Penalties not found.
500.4.2.6	Search error for protocol.
500.4.2.7	Payment purpose is empty.
500.4.2.10	Bill not found.
500.4.2.12	External service is not responding. Please repeat request.
500.4.4.2	Service provider is disabled.

500.5.1	Account to withdraw and account to deposit must be different.
500.5.2	You has access to no one account.
500.5.3	Cannot create operation template.
500.5.4	Name must not be greater than * symbols.
500.5.5	Labels can contain words, digits, symbols "_" and "-", whitespaces.
500.5.6	Each label length must not be greater than 128 symbols.
500.5.7	Amount must be greater than zero.
500.5.8	Template can not be a regular payment.
500.5.9	Execution start date and time is before current date and time.
500.5.10	Execution start date is not last day of month.
500.5.11	Execution end date is before execution start date and time.
500.5.12	Execution end date is before current date and time.
500.5.13	Hours in payment status notification must be greater than zero.
500.5.14	Wrong amount type.
500.5.15	Minimal amount must be positive number.
500.5.16	Maximal amount must be greater than zero.
500.5.17	Maximal amount must be greater than minimal amount.
500.5.18	Remaining balance value must be greater than zero.
500.5.19	Define, at least, one of range limits.
500.5.20	Wrong template ID.
500.6.1.1	No bank details.
500.6.1.2	No approved bank details.
500.6.1.3	Wrong bank details.
500.6.1.4	Missing payee bank account number.
500.6.1.5	Missing payee bank name.
500.6.1.6	Missing payee bank BIN.
500.6.1.7	Missing payee bank corr. account number.
500.6.1.8	Missing payee name.
500.6.1.9	Wrong document index format.
500.6.1.10	Wrong payer identifier format.
500.6.1.11	Missing document index or payer identifier.
500.6.1.12	Missing payee or payee's bank Tax ID.
500.6.1.13	Missing contract number.
500.6.1.14	Wrong bank account number format (20 digits).
500.6.1.15	Transfer to the given account is not possible. Please specify different payee's account.

500.6.1.16	Wrong BIN format (9 digits).
500.6.1.17	Unknown bank BIN.
500.6.1.17.1	Fill in the Correspondent account field.
500.6.1.18	Bank BIN and account number do not match.
500.6.1.19	Wrong bank corr. account number format (20 digits).
500.6.1.20	Wrong KPP format (9 digits).
500.6.1.21	Wrong KBK format.
500.6.1.22	Wrong OKTMO format.
500.6.1.23	KPP, KBK and OKTMO parameters must be set all at budget payments.
500.6.1.24	Wrong payee name.
500.6.1.25	Wrong payee Tax ID format (10 or 12 digits).
500.6.1.26	Payee name length limit exceeded (* > *).
500.6.1.27	Field value length limit exceeded (* > *).
500.6.1.28	Purpose length limit exceeded (* > *) for specified currency.
500.6.1.29	Wrong payment purpose format.
500.6.1.30	Missing card number.
500.6.1.31	Latin letters are not allowed.
500.6.1.32	Field "Payee name" or "Payment purpose" must be "*".
500.6.1.33	Withdraw to this account is not allowed for anonymouse.
500.6.1.34	Wrong bank SWIFT (8 digits/letters).
500.6.1.35	Wrong bank IBAN (5..40 digits and/or latinic letters in upper case).
500.6.1.36	Unknown bank SWIFT.
500.6.1.37	Unknown intermediary russian bank BIN.
500.6.1.38	Wrong international intermediary bank SWIFT (8 digits/letters).
500.6.1.39	Unknown international intermediary bank SWIFT.
500.6.1.40	Wrong transfer order number (numbers from 1 to 999 999 expected).
500.6.1.41	Missing transfer order date.
500.6.1.42	Transfer order number must not end with three zeros.
500.6.1.43	This BIC is changed. Specify the bank details.
500.6.1.44	For the indicated BIC of the bank, settlements are terminated.
500.6.1.45	Missing payment purpose.
500.6.1.46	Wrong payer name.
500.6.1.47	Payer name length limit exceeded (* > *).
500.6.1.50	Missing document index or payer identifier inn.
500.6.1.52	When producing the Payer Status field, the Docindex must start with 322.

500.6.1.131	Wrong contract number.
500.6.2.1	Specify Qiwi account number.
500.6.2.2	Wrong account number. 11 or 12 digits are required.
500.6.3.1.1	Error communicating with SBP: *.
500.6.3.1.2	The total refunds are greater than the original transaction.
500.6.3.1.3	Limits on the amount of output for this user type have been violated.
500.6.3.1.4	Start over.
500.6.3.1.5	Timeout exceeded, please try again later.
500.6.3.1.6	A valid account number is required.
500.6.3.1.7	The account to be debited must be registered with the SBP.
500.6.3.1.8	Field * not filled.
500.6.3.1.9	Error getting data from SBP.
500.6.3.1.10	No data for request A02.
500.6.3.1.11	No data requested for C02/D02.
500.6.3.1.12	No data requested C05.
500.6.3.1.13	No data requested C06.
500.6.3.1.14	No data requested for C11.
500.6.3.1.15	No data requested for C12.
500.6.3.1.16	No data requested for C21.
500.6.3.1.17	No data for request B05.
500.6.3.1.18	No data for request B06.
500.6.3.1.19	No data requested for B11.
500.6.3.1.20	No data requested for B12.
500.6.3.1.21	No data for request B21.
500.6.3.1.22	Failed to receive payment data.
500.6.3.1.23	The field value is missing, should be obtained in step C02.
500.6.3.1.24	Invalid operation status.
500.6.3.1.25	Please enter the correct beneficiary bank.
500.6.3.1.25.1	Recipient Id is required.
500.6.3.1.26	Invalid QR payload.
500.6.3.1.27	Invalid QR payload (*).
500.6.3.1.28	Required TIN in profile.
500.6.3.1.29	Required username in profile.
500.6.3.1.30	Wrong unit type.
500.6.3.1.31	The sender's phone number is required (specified in the PC profile).
500.6.3.1.32	The correct name of the sender is required (specified in the PC profile).

500.6.3.1.33	Payment destination exceeded 140 characters.
500.6.3.1.34	Invalid payment purpose.
500.6.3.1.35	Purpose of payment required.
500.6.3.1.36	Required recipient amount in sender operation.
500.6.3.1.37	The recipient and sender PAMs for Simplified Users did not match.
500.6.3.1.38	Invalid QR code parameter value.
500.6.3.1.39	Please enter the phone number to which you would like to transfer funds.
500.6.3.1.40	Invalid russian phone number (country code is 7, contains 11 or 12 digits).
500.6.3.1.40.1	The recipient's phone number must be a valid international number.
500.6.3.1.41	Invalid or paid QR code.
500.6.3.1.42	Payment amount does not match QR code.
500.6.3.1.43	Payment currency does not match QR code.
500.6.3.1.44	Users of this type are not allowed to output to the SBP.
500.6.3.1.44.1	Sender phone from profile must be russian.
500.6.3.1.44.2	Transfer is only possible to your phone number.
500.6.3.1.44.3	Last name, first name, middle name of sender and recipient must match.
500.6.3.1.44.4	Translation could not be completed.
500.6.3.1.44.4.1	Transfer can only be made to a bank account or personalized ESP.
500.6.3.1.44.4.2	Transfer can only be made to a bank account.
500.6.3.1.44.4.3	Transfer can only be made to a personalized ESP.
500.6.3.1.44.5	Quantity of phone number requests per day exceeded.
500.6.3.1.44.6	Exceeded default bank requests without payment request per day.
500.6.3.1.44.7	The phone number in the user profile must be a correct international number.
500.6.3.1.45	An error in the payment process. Check the sequence and timeliness of the payment steps, check correctness of the data for the transfer.
500.6.3.1.46	An error occurred while receiving response data from SBP (A02). Follow step 1.
500.6.3.1.46.1	An error occurred while getting the request data for A01. Ensure that step 1 is completed correctly.
500.6.3.1.47	Unexpected Partner request. Follow step 4.
500.6.3.1.48	Request too late after receiving *. Expiration time: *.
500.6.3.1.48.1	There is no previous request *. Check if your request is in the time frame (parameters 996, 997).
500.6.3.1.48.2	Request too early, expected time: *.
500.6.3.1.49	There must be an operation to refund or a debited account.

500.6.3.1.50	Payment amount does not match QR code.
500.6.3.1.51	Invalid refund amount.
500.6.3.1.52	No operation found to return.
500.6.3.1.53	Not enough parameters.
500.6.3.1.54	22 The Sender Bank ID is invalid.
500.6.3.1.55	22 The Sender Bank ID is invalid (Rltd block).
500.6.3.1.56	23 The identifier of the UPCS SBP is invalid.
500.6.3.1.57	23 The identifier of the OPKTS SBP is invalid (Rltd block).
500.6.3.1.58	27 The SBP operation number is invalid.
500.6.3.1.59	79 Unique Message Number from the Sender Bank is incorrect.
500.6.3.1.60	81 Unique Message Number from OPCS UBP is invalid.
500.6.3.1.61	Format of field * is invalid, pattern: *, value: *.
500.6.3.1.63	field * in the request must have the same value *.
500.6.3.1.64	Error parsing SBP message.
500.6.3.1.65	Invalid request type: *.
500.6.3.1.66	*.
500.6.3.1.67	Operation on this QR has already been processed, current status: "*".
500.6.3.1.68	Failed, operation completed.
500.6.3.1.69	Payment failed. Check the account settings to work with the SBP.
500.6.3.1.70	Operation rejected by SBP.
500.6.3.1.71	Error processing request on the side of SBP *.
500.6.3.1.72	Error parsing message.
500.6.3.1.73	Authentication Failed.
500.6.3.1.74	Cryptography Error.
500.6.3.1.75	Invalid data in message URL.
500.6.3.1.76	Invalid data type in Accept header.
500.6.3.1.77	Invalid Request Type.
500.6.3.1.78	Element Required (*).
500.6.3.1.79	Invalid field value (*).
500.6.3.1.80	Service is temporarily unavailable.
500.6.3.1.81	Payment via SBP is temporarily unavailable. Please try again later.
500.6.3.1.90	The operation cannot be created. Operations found *,with the cash link already activated *.
500.6.3.1.91	The settings for working with the SBP for the account * are missing or inactive.
500.6.3.1.92	The cash link * does not belong to the account *.
500.6.3.1.93	Wrong format QrTtl.

500.6.3.1.94	The QrTtl value must be in the range 1 to *.
500.6.3.1.95	The DESCRIPTION field is required when registering a QR code in the "Account Binding" scenario.
500.6.3.1.96	Invalid request parameters: Invalid recurrent token.
500.6.3.1.97	Invalid request parameters: An attempt to make a recurring payment based on data for which the original payment was not successfully completed.
500.6.3.1.98	Error interacting with the SBP agent.
500.6.3.1.99	Recurrent SBP payments are not available for the specified account.
500.6.3.1.100	Make a request for qr information with the ticket=* parameter to get information about the previous request for package registration.
500.6.3.1.101	Error: *.
500.6.3.1.102	In a request for batch registration of static QR, you can specify no more than * applications.
500.6.3.2.1	SBP message processed successfully.
500.6.3.2.1.18	Incorrect subscription Purpose format.
500.6.3.2.1.19	There is no amount when activating the SBP Cash Link.
500.6.3.2.1.21	Invalid subscriptionToken format.
500.6.3.2.1.22	It is not possible to deactivate the SBP Cash Link.
500.6.3.2.1.23	The qrtl for a Cash link With a PS is out of the acceptable range.
500.6.3.2.1.24	Incorrect redirectUrl format.
500.6.3.2.1.25	Refusal to change the account of a legal entity, sole proprietor or self-employed. The cash link From the BP has not been deactivated.
500.6.3.2.1.30	Refusal of the Payer's Bank. The payer refused to link the account.
500.6.3.2.1.31	Refusal of the Payer's Bank. Account binding not found.
500.6.3.2.1.32	Refusal of the Payer's Bank to perform the payment.
500.6.3.2.1.33	Refusal to re-pay.
500.6.3.2.1.34	The scenario is not supported by the Payer's Bank.
500.6.3.2.1.35	The waiting time for the notification from the Payer's Bank has expired.
500.6.3.2.1.36	Duplication of the request identifier assigned by the TSP or the TSP Agent - agentRefundRequestId.
500.6.3.2.1.37	The previous request for a refund for the C2B SBP Operation has not yet been processed.
500.6.3.2.1.38	The request parameters differ from the parameters of the original C2B SBP Operation.
500.6.3.2.1.39	The original SBP C2B Operation was not found.

500.6.3.2.1.40	Refusal from the Payer's Bank. The refund amount exceeds the amount of the original SBP C2B Transaction.
500.6.3.2.1.41	Refund request for C2B operation not found.
500.6.3.2.1.42	Incorrect format or missing amount.
500.6.3.2.1.43	Incorrect format or missing takeTax.
500.6.3.2.1.44	Incorrect format or missing total Tax Amount.
500.6.3.2.1.45	Incorrect uip format.
500.6.3.2.1.46	Invalid payment link type.
500.6.3.2.1.47	A refund request with the specified ID was not found.
500.6.3.2.1.51	The decision to pay from the linked account has not yet been received from the Payer's Bank.
500.6.3.2.1.52	The Order to perform the SBP Transaction was not received from the Payer's Bank. Timeout expired.
500.6.3.2.1.53	Received consent to make a payment from the Payer's Bank, waiting for the Order.
500.6.3.2.1.144	The period of use of a dynamic payment link is beyond the acceptable range from 1 to 129,600 minutes (90 days).
500.6.3.2.2	SBP message processed successfully.
500.6.3.2.3	SBP message processed successfully.
500.6.3.2.4	There is no response from the Beneficiary bank. Repeat the operation.
500.6.3.2.5	Timeout exceeded, retry the operation.
500.6.3.2.6	SBP: Date / Time Mismatch (one of: [13], [14], [15]) in <AppHdr> and <Document>.
500.6.3.2.7	SBP: Mismatch [27] Identifier of Operation OPCC SBP (IO OPCC SBP) in <AppHdr> and in <Document>.
500.6.3.2.8	SBP: Message sender id mismatch (one of: [22], [23], [24]) date-time in <AppHdr> and in <Document>.
500.6.3.2.9	SBP: Message recipient ID mismatch (one of: [22], [23], [24]) date-time in <AppHdr> and in <Document>.
500.6.3.2.10	Transfer to the same bank is not allowed.
500.6.3.2.11	SBP: Validation Error (Semantic).
500.6.3.2.12	SBP: Validation error (format).
500.6.3.2.13	SBP: BIK of the Sender's Bank is not in the Directory.
500.6.3.2.14	SBP: BIK of the Beneficiary's Bank is not in the Directory.
500.6.3.2.15	SBP: Correspondent account of the Sender's Bank does not match the BIC.
500.6.3.2.16	SBP: The correspondent account of the Beneficiary's Bank does not correspond to the BIC.
500.6.3.2.17	SBP: Not enough funds in the Sender's Bank account.
500.6.3.2.18	SBP is not available to the Sender's bank.

500.6.3.2.19	SBP is not available to the Beneficiary's bank.
500.6.3.2.20	SBP: A restriction on the provision of SBP is on the Sender's Bank account.
500.6.3.2.21	SBP: A restriction on the provision of SBP is on the Beneficiary's Bank account.
500.6.3.2.22	SBP: BR MS failure. Operation completed.
500.6.3.2.23	SBP: Insufficient Sender or Recipient Data.
500.6.3.2.24	Unable to credit the transfer amount to the Beneficiary's account.
500.6.3.2.25	SBP: Unable to credit the transfer amount to the Beneficiary's account.
500.6.3.2.26	SBP: No such value EDB [99] Time stamp of the operation of the SBP (MVO SBP) in ED <AccptncDtTm>.
500.6.3.2.27	SBP: No such value EDS [27] Operation identifier OPKTS SBP (IO OPKTS SBP).
500.6.3.2.28	SBP: No such value EDB [22] Sender Bank Identifier (BO ID).
500.6.3.2.29	SBP: No such value EDB [24] Beneficiary Bank Identifier (BP ID).
500.6.3.2.30	SBP: No such ESD value [9] SBP transaction currency (VLT OP SBP).
500.6.3.2.31	SBP: No such value EDS [48] SBP Operation Type (SBP Maintenance).
500.6.3.2.32	SBP: No such value EDB [90] Sender Document Type (PKD OT).
500.6.3.2.33	SBP: No such value EDB [92] Recipient document type (PKD TP PO).
500.6.3.2.34	SBP: No such ESD value [46] Sender ID Type (TID FROM).
500.6.3.2.35	SBP: No such ESD value [47] Recipient Identifier Type (TID PO).
500.6.3.2.36	SBP: Invalid value of ESD [27] Operation identifier OPKTS SBP (IO OPKTS SBP).
500.6.3.2.37	SBP: Message not valid for this type of operation.
500.6.3.2.38	SBP: Error.
500.6.3.2.39	Error.
500.6.3.2.40	SBP: Error.
500.6.3.2.41	SBP: The short circuit does not match the SBP operation data.
500.6.3.2.42	Error.
500.6.3.2.43	SBP: Error.
500.6.3.2.44	Error.
500.6.3.2.45	Error processing C23.
500.6.3.2.46	Error processing C21.
500.6.3.2.47	SBP: Error processing message.

500.6.3.2.48	SBP: Duplicate previously received message.
500.6.3.2.49	Duplicate previously received message.
500.6.3.2.50	Duplicate previously received message.
500.6.3.2.51	SBP: Receive Late Message.
500.6.3.2.52	SBP: Receive unexpected message.
500.6.3.2.53	SBP: OTP entered incorrectly. Operation completed.
500.6.3.2.54	SBP: OTP entered incorrectly. Re-enter.
500.6.3.2.55	SBP: PAM Sender and Receiver did not match (only for Me2Me).
500.6.3.2.56	SBP: Sender Bank Blocked.
500.6.3.2.57	SBP: Sender Bank and Recipient Bank matched for SBP operation.
500.6.3.2.58	SBP: Recipient's bank blocked.
500.6.3.2.59	SBP: The bank is already registered as the default bank for this client.
500.6.3.2.60	SBP: BIK of the Sender's bank does not match the bank identifier in SBP.
500.6.3.2.61	SBP: BIK of the Beneficiary's bank does not match the bank identifier in SBP.
500.6.3.2.62	SBP: Operation parameters changed.
500.6.3.2.63	SBP: Fraud Suspected.
500.6.3.2.64	SBP: Suspected Fraud from BP.
500.6.3.2.65	SBP: Legal Restrictions.
500.6.3.2.66	SBP: Legislative restrictions - the level of funds identification is insufficient (only for C2C Push).
500.6.3.2.67	Contact the Beneficiary of funds and specify the details for crediting funds.
500.6.3.2.68	SBP: BR MS Failure.
500.6.3.2.69	SBP: Number of attempts to set default bank exceeded (per day).
500.6.3.2.70	SBP: Transaction amount exceeds the allowed limit per transaction.
500.6.3.2.71	Prohibited from crediting the Beneficiary's account.
500.6.3.2.72	More than one Recipient found.
500.6.3.2.73	Recipient not found.
500.6.3.2.74	Legislative restrictions on crediting (for example, the amount exceeded the allowed amount for this payment method or the level of identification is insufficient).
500.6.3.2.75	The recipient has not agreed to receive funds through the SBP.
500.6.3.2.76	Recipient refused to receive funds via SBP.
500.6.3.2.77	Recipient's account blocked or closed.
500.6.3.2.78	Recipient account not found.
500.6.3.2.79	Service not available for Recipient.

500.6.3.2.81	SBP: Success.
500.6.3.2.82	SBP: Refuse to resolve: error.
500.6.3.2.83	SBP: Receiving Party Refused Settlement (money not transferred).
500.6.3.2.84	SBP: Settlement Denied: Initial Payment Not Found.
500.6.3.2.85	SBP: Refusal to resolve: message submission deadline exceeded.
500.6.3.2.86	SBP: Refuse to resolve: duplicates a previously successfully processed message.
500.6.3.2.87	SBP: Refused Settlement: Message Response Timeout exceeded per Settlement Rules [2].
500.6.3.2.88	Settlement Denied: Could not contact customer.
500.6.3.2.89	Settlement Rejected: Customer Disagree.
500.6.3.2.90	Settlement Denied: Account Closed.
500.6.3.2.91	Settlement Denied: Could not contact customer.
500.6.3.2.92	Settlement Rejected: Customer Disagree.
500.6.3.2.93	Settlement Denied: Account Closed.
500.6.3.2.94	Error.
500.6.3.2.95	Error.
500.6.3.2.96	Error.
500.6.3.2.97	Request processed successfully.
500.6.3.2.98	Validation failed. Invalid request format.
500.6.3.2.99	Legal entity or individual entrepreneur blocked by the SBP administrator.
500.6.3.2.101	Validation failed. Invalid check digit OGRN.
500.6.3.2.102	Validation failed. The length of the PSRN does not match the length of the INN.
500.6.3.2.106	Validation error. Invalid merchant phone number format.
500.6.3.2.107	The agent with the specified ID could not be found.
500.6.3.2.108	No member with the specified ID found.
500.6.3.2.109	No legal entity with the specified ID found.
500.6.3.2.110	Validation failed. Incorrect format of the country of registration of the legal entity.
500.6.3.2.111	Validation failed. Incorrect format of the region of the legal entity.
500.6.3.2.112	Validation failed. Invalid merchant country code format.
500.6.3.2.113	Validation failed. Invalid format of the merchant region code.
500.6.3.2.114	Invalid Customer Experience Point Capability.
500.6.3.2.115	Invalid MCC value specified.
500.6.3.2.116	Validation failed. Invalid merchant name format.
500.6.3.2.117	Validation failed. Invalid legal entity name format.

500.6.3.2.118	Temporary registration number not found.
500.6.3.2.119	Duplicate. Legal entity is already registered.
500.6.3.2.120	Duplicate. Legal entity account is already registered.
500.6.3.2.121	Duplicate. Merchant already registered.
500.6.3.2.122	Technical error.
500.6.3.2.124	Request processed successfully.
500.6.3.2.125	Validation failed. Invalid request format.
500.6.3.2.126	No agent found with the specified ID.
500.6.3.2.127	No member with the specified ID found.
500.6.3.2.128	No legal entity with the specified ID found.
500.6.3.2.129	Legal entity account not found.
500.6.3.2.130	The merchant with the specified ID is not registered in the SBP.
500.6.3.2.131	Invalid Currency Code.
500.6.3.2.132	Invalid QRC Template Version.
500.6.3.2.133	Invalid QRC Type.
500.6.3.2.134	Transaction amount is less than the minimum allowed.
500.6.3.2.135	Transaction amount is greater than the maximum allowed.
500.6.3.2.136	Amount and currency code must be present or absent at the same time in the request for QRC Static.
500.6.3.2.137	Missing currency code in request for QRC Dynamic.
500.6.3.2.138	Amount missing in request for QRC Dynamic.
500.6.3.2.139	QR with the specified ID could not be found.
500.6.3.2.140	Exceeded% value% value of items in request.
500.6.3.2.141	Invalid QRC ID format.
500.6.3.2.142	Incorrect format for the payment purpose field.
500.6.3.2.143	Technical error.
500.6.3.2.145	Invalid EDB Data Unique Identifier [159].
500.6.3.2.146	Error on the side of the Beneficiary Bank. Operation completed.
500.6.3.2.147	Insufficient Sender Information.
500.6.3.2.148	Maximum Payment Amount Exceeded.
500.6.3.2.149	Invalid EDB Data Unique Identifier [159].
500.6.3.2.150.1	Technological work at the recipient's bank Try to repeat the translation after a while.
500.6.3.2.150.2	Contact the Beneficiary of funds and specify the details for crediting funds.
500.6.3.2.151	SBP: Suspected Fraud.
500.6.3.2.152	SBP: EDB format error [2].
500.6.3.2.153	SBP: EDB format error [9].
500.6.3.2.154	SBP: EDB format error [13].

500.6.3.2.155	SBP: EDB format error [14].
500.6.3.2.156	SBP: EDB format error [15].
500.6.3.2.157	SBP: EDB format error [20].
500.6.3.2.158	SBP: EDB format error [22].
500.6.3.2.159	SBP: EBD format error [23].
500.6.3.2.160	SBP: EDB format error [24].
500.6.3.2.161	SBP: EDB format error [27].
500.6.3.2.162	SBP: EDB format error [29].
500.6.3.2.163	SBP: EBD format error [30].
500.6.3.2.164	SBP: EDB format error [31].
500.6.3.2.165	SBP: EBD format error [32].
500.6.3.2.166	SBP: EBD format error [33].
500.6.3.2.167	SBP: EDB format error [43].
500.6.3.2.168	SBP: EDB format error [44].
500.6.3.2.169	SBP: EDB format error [47].
500.6.3.2.170	SBP: EBD format error [48].
500.6.3.2.171	SBP: EBD format error [51].
500.6.3.2.172	SBP: EDB format error [74].
500.6.3.2.173	SBP: EDB format error [79].
500.6.3.2.174	SBP: EDB format error [80].
500.6.3.2.175	SBP: EBD format error [81].
500.6.3.2.176	SBP: EBD format error [92].
500.6.3.2.177	SBP: EBD format error [93].
500.6.3.2.178	SBP: EBD format error [94].
500.6.3.2.179	SBP: EDB format error [95].
500.6.3.2.180	SBP: EBD format error [96].
500.6.3.2.181	SBP: EDB format error [98].
500.6.3.2.182	SBP: EBD format error [99].
500.6.3.2.183	SBP: EDB format error [103].
500.6.3.2.184	SBP: EDB format error [104].
500.6.3.2.185	SBP: EDB format error [107].
500.6.3.2.186	SBP: EBD format error [108].
500.6.3.2.187	SBP: EDB format error [125].
500.6.3.2.188	SBP: EBD format error [126].
500.6.3.2.189	SBP: EDB format error [130].
500.6.3.2.190	SBP: EDB format error [131].
500.6.3.2.191	SBP: EDB format error [139].
500.6.3.2.192	SBP: EBD format error [140].
500.6.3.2.193	SBP: Refusal to process a message from the Recipient's Bank.

500.6.3.2.194	SBP: EBD [13] is missing in HEADER or DOCUMENT or the values of EBD [13] in HEADER and DOCUMENT do not match.
500.6.3.2.195	SBP: EBD [14] is missing in HEADER or DOCUMENT, or the values of EBD [14] in HEADER and DOCUMENT do not match.
500.6.3.2.196	SBP: EBD [22] is missing in HEADER or DOCUMENT or the values of EBD [22] in HEADER and DOCUMENT do not match.
500.6.3.2.197	SBP: EBD [23] is missing in HEADER or DOCUMENT or the values of EBD [23] in HEADER and DOCUMENT do not match.
500.6.3.2.198	SBP: EBD [24] is missing in HEADER or DOCUMENT or the values of EBD [24] in HEADER and DOCUMENT do not match.
500.6.3.2.199	SBP: EDB [27] is missing in DOCUMENT or the values of EDB [27] in HEADER and DOCUMENT do not match.
500.6.3.2.200	SBP: EBD [20] not present in DOCUMENT.
500.6.3.2.201	SBP: EBD [29] not present in DOCUMENT.
500.6.3.2.202	SBP: EBD [30] not present in DOCUMENT.
500.6.3.2.203	SBP: EBD [31] not present in DOCUMENT.
500.6.3.2.204	SBP: EBD [44] not present in DOCUMENT.
500.6.3.2.205	SBP: EBD [50] not present in DOCUMENT.
500.6.3.2.206	SBP: EBD [79] not present in DOCUMENT.
500.6.3.2.207	SBP: EBD [80] not present in DOCUMENT.
500.6.3.2.208	SBP: EBD [81] not present in DOCUMENT.
500.6.3.2.209	SBP: EBD [95] not present in DOCUMENT.
500.6.3.2.210	SBP: EBD [96] not present in DOCUMENT.
500.6.3.2.211	SBP: EBD [98] not present in DOCUMENT.
500.6.3.2.212	SBP: EBD [99] not present in DOCUMENT.
500.6.3.2.213	SBP: EBD [125] not present in DOCUMENT.
500.6.3.2.214	SBP: EBD [126] not present in DOCUMENT.
500.6.3.2.215	SBP: EBD [130] not present in DOCUMENT.
500.6.3.2.216	SBP: ESD [2] value differs from ESD [2] value in previous messages.
500.6.3.2.217	SBP: ESD value [9] is different from ESD value [9] in previous messages.
500.6.3.2.218	SBP: ESD value [20] is different from ESD value [20] in previous messages.
500.6.3.2.219	SBP: ESD value [22] is different from ESD value [22] in previous messages.
500.6.3.2.220	SBP: ESD value [24] is different from ESD value [24] in previous messages.
500.6.3.2.221	SBP: ESD value [30] differs from ESD value [30] in previous messages.

500.6.3.2.222	SBP: ESD value [32] differs from ESD value [32] in previous messages.
500.6.3.2.223	SBP: ESD value [44] differs from ESD value [44] in previous messages.
500.6.3.2.224	SBP: ESD value [47] differs from ESD value [47] in previous messages.
500.6.3.2.225	SBP: ESD value [48] differs from ESD value [48] in previous messages.
500.6.3.2.226	SBP: ESD value [51] differs from ESD value [51] in previous messages.
500.6.3.2.227	SBP: ESD value [74] differs from ESD value [74] in previous messages.
500.6.3.2.228	SBP: ESD value [81] differs from ESD value [81] in previous messages.
500.6.3.2.229	SBP: ESD value [94] differs from ESD value [94] in previous messages.
500.6.3.2.230	SBP: ESD value [95] differs from ESD value [95] in previous messages.
500.6.3.2.231	SBP: ESD value [96] differs from ESD value [96] in previous messages.
500.6.3.2.232	SBP: ESD value [97] differs from ESD value [97] in previous messages.
500.6.3.2.233	SBP: ESD value [99] differs from ESD value [99] in previous messages.
500.6.3.2.234	SBP: ESD value [104] is different from ESD value [104] in previous messages.
500.6.3.2.235	SBP: ESD value [125] is different from ESD value [125] in previous messages.
500.6.3.2.236	SBP: ESD value [130] differs from ESD value [130] in previous messages.
500.6.3.2.237	SBP: Invalid ESD value [9].
500.6.3.2.238	SBP: Invalid ESD value [22].
500.6.3.2.239	SBP: Invalid ESD value [23].
500.6.3.2.240	SBP: Invalid ESD value [27].
500.6.3.2.241	SBP: Invalid ESD value [43].
500.6.3.2.242	SBP: Invalid ESD value [47].
500.6.3.2.243	SBP: ESD value [94] does not match ESD value [22].
500.6.3.2.244	SBP: ESD value [95] does not match ESD value [24].
500.6.3.2.245	SBP: Invalid ESD value [125].
500.6.3.2.246	SBP: EBD [130] is not registered in the OPKTS SBP.
500.6.3.2.247	SBP: EBD [139] does not match EBD [130].
500.6.3.2.248	SBP: ESD value [44] exceeds the allowed amount of SBP Operations.
500.6.3.2.250	SBP: EDB format error [21].
500.6.3.2.251	SBP: EDB format error [50].

500.6.3.2.252	SBP: EBD [21] not present in DOCUMENT.
500.6.3.2.253	SBP: EBD [46] not present in DOCUMENT.
500.6.3.2.254	SBP: EBD [47] not present in DOCUMENT.
500.6.3.2.255	SBP: EBD [48] not present in DOCUMENT.
500.6.3.2.256	SBP: EBD [50] not present in DOCUMENT.
500.6.3.2.257	SBP: EBD [94] not present in DOCUMENT.
500.6.3.2.258	SBP: EBD [155] not present in DOCUMENT.
500.6.3.2.259	SBP: EBD [159] not present in DOCUMENT.
500.6.3.2.260	SBP: ESD [1] value is different from ESD [1] value in previous messages.
500.6.3.2.261	SBP: ESD value [21] differs from ESD value [21] in previous messages.
500.6.3.2.262	SBP: ESD value [31] is different from ESD value [31] in previous messages.
500.6.3.2.263	SBP: ESD value [46] differs from ESD value [46] in previous messages.
500.6.3.2.264	SBP: ESD value [50] differs from ESD value [50] in previous messages.
500.6.3.2.265	SBP: ESD value [126] differs from ESD value [126] in previous messages.
500.6.3.2.266	SBP: ESD value [131] differs from ESD value [131] in previous messages.
500.6.3.2.267	SBP: ESD value [139] differs from ESD value [139] in previous messages.
500.6.3.2.268	SBP: ESD value [140] differs from ESD value [140] in previous messages.
500.6.3.2.269	SBP: ESD value [159] differs from ESD value [159] in previous messages.
500.6.3.2.270	SBP: Invalid ESD value [46].
500.6.3.2.271	SBP: EBD value [24] does not match the Recipient's Bank Identifier in the registered QR.
500.6.3.2.272	SBP: ESD value [44] does not match the amount in the registered QR.
500.6.3.2.273	SBP: ESD value [155] does not match ESD value [159] for QR.
500.6.3.2.274	SBP: QR with EDB [159] not registered in SBP.
500.6.3.2.275	SBP: Missing EDB [1] and EDB [96].
500.6.3.2.276	Exceeded the allowed number of attempts to install the default Bank per day.
500.6.3.2.277	This payment link was previously denied.
500.6.4.1	The operation was paid for or in the process of payment.
500.6.4.2	Unsuccessful payment.
500.6.4.3	User redirection for payment is not required.
500.6.4.4	Error processing request result.

500.6.4.5	Unknown status.
500.6.4.6	Waiting for status.
500.6.5.1.1	Recurrent sberpay payments are not available for the specified account.
500.6.5.1.2	The main operation for subscription payment was not found.
500.6.5.1.3	Subscription payment is not possible. Subscription revoked.
500.6.5.1.4	There is no sberpay subscription token in the main operation.
500.6.5.1.5	Recurring token payment error *.
500.6.5.1.6	Error interacting with the Sber Pay service. On the service side, routine maintenance is underway.
500.6.5.1.7	The invoice has not been created. Perhaps the phone number is not correct or is not associated with the SBOL application.
500.6.5.1.8	Phone number in the format: 10 or 11 digits, possible without +7 or 8.
500.6.5.1.9	SberPay service is temporarily unavailable.
500.6.5.1.10	The time limit for waiting for a response from the SberPay service has been exceeded.
500.6.5.1.11	There is an uncompleted return operation *.
500.6.5.1.12	Error: *.
500.6.5.1.13	There are no necessary attributes for making a payment.
500.6.6.1.1	An error occurred during the operation.
500.6.6.1.2	Operation declined.
500.6.6.1.3	Operation declined.
500.6.6.1.4	Payment by foreign card not available.
500.6.6.1.5	Insufficient Funds.
500.6.6.1.6	Invalid Card Number.
500.6.6.1.7	Card is expired.
500.6.6.1.8	The limit on the count of transactions on the card has been exceeded.
500.6.6.1.9	The limit on the amount of transactions on the card has been exceeded.
500.6.6.1.10	Operation not permitted to card.
500.6.6.1.11	Transactions on this card are not available.
500.6.6.1.12	Incorrect CVV.
500.6.6.1.13	Too many wrong PIN tries.
500.6.6.1.14	Invalid card data.
500.6.6.1.15	Card with this payment system are not supported.
500.6.6.1.16	Issuer or payment system are not available.
500.6.6.1.17	Reconcile error on the payment system.

500.6.6.1.18	Error on the payment system.
500.6.6.1.19	Rejected by acquirer. Authorization requests limit exceeded.
500.6.6.1.20	Operations of this type are prohibited by the Rules and Standards of the payment system.
500.6.6.1.21	Not permitted to acquirer.
500.6.6.1.22	Authorization completion time expired.
500.6.6.1.23	The amount exceeds the limits established by the issuer for this type of transaction.
500.6.6.1.24	Duplicate operation.
500.6.6.1.25	Incorrect CVV or 3DS code.
500.6.6.1.26	3DS authentication failed.
500.6.6.1.27	The issuer rejected the 3DS authentication.
500.6.6.1.28	The issuer rejected the 3DS authentication.
500.6.6.1.29	Operation amount limit exceeded.
500.6.6.1.30	Daily amount/count limit exceeded.
500.6.6.1.31	Not permitted to this merchant type.
500.6.6.1.32	Acquirer error.
500.6.6.1.33	Acquirer error.
500.6.6.1.34	The parent payment was created incorrectly.
500.6.6.1.35	Invalid request parameters.
500.6.6.2.1	An error occurred during the operation.
500.6.6.2.2	Operation declined.
500.6.6.2.3	Operation declined.
500.6.6.2.4	The limit on the amount of refunds on the card has been exceeded.
500.6.6.2.5	Reconcile error on the payment system.
500.6.6.2.6	Not permitted to acquirer.
500.6.6.2.7	Issuer or payment system are not available.
500.6.6.2.8	Operation declined.
500.6.6.2.10	Unexpected error.
500.7.1	There was a problem during simplified identification. Please contact MONETA.RU customer service.
500.7.2	Simplified identification was not completed. There was a time-out in waiting for result of checking. To complete the simplified identification process, please make another request.
500.7.3	Simplified identification failed.
500.7.4	To complete simplified identification, please enter your passport details. The document you provided is not an active passport.
500.7.5	Mobile phone number was not confirmed.
500.7.6	You did not provide full information about yourself. To complete simplified identification, please provide

	the following details in your account: your full name, passport details, INN, SNILS, mobile phone number.
500.7.7	Automatic simplified identification is available to citizens of Russia only.
500.7.8	Similar accounts with the status "Simplified identification" were found. Please contact MONETA.RU customer service.
500.7.9	Some information is missing or your personal data required for simplified identification was not confirmed.
500.7.10	Your identification document did not pass the authenticity check. Please contact MONETA.RU customer service.
500.7.11	The customer has already passed simplified identification.
500.7.12	Could not identify the user.
500.8	There was an external error: *.
500.8.3.1.1	There was an external error.
500.8.4769.1	Electronic system of identification and authentication (ESIA) does not respond to the request. Please try later.
500.8.4769.2	There was an error during making a request to the electronic system of identification and authentication (ESIA).
500.9.1	Asynchronous task is not found.
500.9.2	Task is not supported asynchronous work.
500.9.3	The mismatch between the namespace of the asynchronous request and the result.
500.9.4	You can use this request in asynchronous mode only. Use AsyncRequest for this method.
500.10.1	Subscriber not found.
500.10.2	CallbackUrl is not a valid URI.
500.10.4	E-mail is empty or wrong format.
500.10.5	Please set payerIdentifiers or supplierBillIds.
500.10.7	Subscription limits are exceeded.
500.11.1.1	Invalid request parameters.
500.11.2.1	The necessary data for interaction with the SBP service is missing.
500.11.2.2	The ID of the cash link does not belong to this account.
500.11.2.3	The allowed limit for registering new cash links has been exceeded.
500.11.2.4	A cash link with such a customer ID already exists.
500.11.2.5	The limit for registering static QR codes has been exceeded.
500.11.2.6	Banning QRS registration via api.

Chapter 4

Quick Start

Topics:

Sandbox Overview

The MONETA.RU sandbox is a testing environment that copies the real MONETA.RU production environment.

You can use the MONETA.RU sandbox server at *https://demo.moneta.ru* to test all of the requests to the MONETA.RU APIs.

Registering Demo Accounts

Before you can start, you must register two demo accounts: a merchant account to accept payments and a regular MONETA.RU account to pay with. Use different email addresses to register each account.

1. Create a demo merchant account:

   https://demo.moneta.ru/backoffice/auth/register
2. Create a regular MONETA.RU account. Use the following link:

   https://demo.moneta.ru/locale.htm?moneta.locale=en&redirect=/register.htm
3. Contact PayAnyWay (business@support.payanyway.ru) support and request to complete your registration and to add virtual money to your regular demo account.

   Note: To identify your accounts, include the email addresses that you used to register your demo accounts.
4. After the support team completes your registration, add an advanced account for your merchant account. Perform the following substeps:

   a) Open *https://demo.moneta.ru* and log in to your merchant account.

   b) Go to **My account** > **Account management**.

   c) Under **Advanced accounts**, click **Add account**.

   d) Complete the form and click **Save**.

Merchant Account Settings

In account management section («Management» link in the «Accounts» block on the left part of the review page) the authorized personnel of the merchant can set the parameters of interaction between the merchant registration system and MONETA.RU system.

Setting	Required?	Description
Test mode		Indicates whether your merchant account works in test mode. In test mode, Moneta.Assistant does not create transactions to transfer payments to your merchant account. Use this mode to test status requests and processed payment reports.
Payment method IDs		Specifies a comma-separated list of payment method IDs that a customer can use for payment. For example, "1015,1017" – resulting payment options list will be limited to "MONETA.RU" and "WebMoney".
Default payment method		Specifies the default payment method for the payment. For example, 1015 – MONETA.RU, 1020 – Yandex.Money, 1017 – WebMoney, etc. The list of available payment methods for account number you can find in section "Merchant back office / Payment methods".
Check URL	no	Specifies the address of the script in your online store that processes status requests from MONETA.Assistant. If you specify the **Check URL** field, MONETA.Assistant sends status requests to the specified address. For more information about status requests, see *Status Requests*. **Important:** The web address must include the URI scheme. For example, `http://www.yourstore.com/check_url.php`
Pay URL	yes	Specifies the address of the script in your online store that processed payment reports from MONETA.Assistant. For more information about processed payment reports, see *Processed Payment Reports*" **Important:** The web address must include the URI scheme. For example, `http://www.yourstore.com/pay_url.php`
HTTP Method		Specifies the HTTP request method that MONETA.Assistant uses to pass parameters to the pages which addresses you specify in **Check URL** and **Pay URL** fields. Options are: • GET • POST
Code of data integrity verification	yes	Specifies a character string that MONETA.Assistant and your online store must use to generate a unique signature for each request and response. You must keep this code secret.
Mandatory payment form signature		Indicates whether MONETA.Assistant requires the MNT_SIGNATURE parameter in payment requests from the payment form. Options are: • **Yes**. MONETA.Assistant requires the MNT_SIGNATURE parameter in payment requests. If the payment form sends a request to MONETA.Assistant without the MNT_SIGNATURE parameter or this parameter has a wrong value, MONETA.Assistant returns an error. • **No**. MONETA.Assistant does not require the MNT_SIGNATURE parameter in payment requests. **Note:** This option is not secure.

Setting	Required?	Description
URL settings can be reset		Indicates whether you can redefine the **Success URL**, **InProgress URL**, **Return URL**, and **Fail URL** addresses by using the MNT_SUCCESS_URL, MNT_INPROGRESS_URL, MNT_RETURN_URL, and MNT_FAIL_URL parameters in a payment request. Options are: • **Yes**. You can redefine the Success and Fail URLs in payment requests. • **No**. MONETA.Assistant always uses the values that you specify in your merchant account settings.
Success URL		Specifies the address of a web page to which MONETA.Assistant redirects a customer after completing the checkout successfully. The customer is redirected to this web page even if the merchant does not receive a processed payment report. **Important:** The web address must include the URI scheme. For example, `http://www.yourstore.com/success_url.php`
Fail URL		Specifies the address of a web page to which MONETA.Assistant redirects a customer if the external payment system cannot process the payment. In this case, MONETA.Assistant does not send a processed payment report to the merchant. **Important:** The web address must include the URI scheme. For example, `http://www.yourstore.com/fail_url.php`
InProgress URL	no	Specifies the address of a web page to which MONETA.Assistant might redirect a customer after a successful funds authorization request, before the payment is confirmed and transferred to the merchant account. This parameter affects only some of the payment methods. MONETA.Assistant redirects a customer to this page even if the merchant does not receive the processed payment report. If you do not specify the **InProgress URL**, MONETA.Assistant uses the default page to show the payment progress. **Important:** The web address must include the URI scheme. For example, `http://www.yourstore.com/inprogress_url.php`
Return URL		Specifies the address of a web page to which MONETA.Assistant might redirect customers after they decide to cancel a payment and return to the web store. In this case, MONETA.Assistant does not send a processed payment report to the merchant. **Important:** The web address must include the URI scheme. For example, `http://www.yourstore.com/return_url.php`

Using CURL to Send SOAP Requests and Get Responses

You can use the cURL utility to test the Merchant API. Use this command-line tool to send SOAP requests and receive responses.

1. Create an XML request to the Merchant API and save it to the request.xml file.
   - If you plan to use UsernameToken authentication, the XML request must include a security header.

     For example, use the following XML request to get information about your account:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
    xmlns="http://www.moneta.ru/schemas/messages.xsd"
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <wsse:Security soap:mustUnderstand="1"
```

```
     xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
     wss-wssecurity-secext-1.0.xsd">
      <wsse:UsernameToken wsu:Id="UsernameToken-31877484"
     xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
     wss-wssecurity-utility-1.0.xsd">
 <wsse:Username>user_name</wsse:Username>
 <wsse:Password
     Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-
     token-profile-1.0#PasswordText">password</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    <FindProfileInfoByAccountIdRequest>12345678</
FindProfileInfoByAccountIdRequest>
  </soap:Body>
</soap:Envelope>
```

- If you plan to use SSL certificate authentication, do not include a security header.

  For example, the same XML request without security headers:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
    xmlns="http://www.moneta.ru/schemas/messages.xsd"
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <FindProfileInfoByAccountIdRequest>12345678</
FindProfileInfoByAccountIdRequest>
  </soap:Body>
</soap:Envelope>
```

2. Send the request to the Merchant API. Use one of the following CURL commands depending on your preferred authentication method:

   - To send the SOAP request that includes a security header, execute the following command:

```
curl --verbose --http1.0 --show-error --data-binary "@request.xml" --output
 "response.xml" --header "Content-type: text/xml;charset=UTF-8" --header
 "SOAPAction:\"\"" --insecure https://www.moneta.ru/services
```

   - To send the SOAP request without a security header by using SSL certificate authentication, execute the following command:

```
curl --verbose --http1.0 --show-error --data-binary "@request.xml" --output
 "response.xml" --cert "certificate.pem" --key "private.key" --header "Content-
type: text/xml;charset=UTF-8" --header "SOAPAction: \"\"" --insecure https://
www.moneta.ru:8443/services/x509
```

   These CURL commands save the response to your requests to the response.xml file.

   The following listing shows a sample XML response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:FindProfileInfoByAccountIdResponse
          xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd">
      <ns2:accountId>12345678</ns2:accountId>
      <ns2:currency>RUB</ns2:currency>
      <ns2:profile>
        <ns2:attribute>
          <ns2:key>last_name</ns2:key>
          <ns2:value>moneta.ru</ns2:value>
          <ns2:approved>true</ns2:approved>
        </ns2:attribute>
        <ns2:attribute>
          <ns2:key>first_name</ns2:key>
          <ns2:value>support</ns2:value>
          <ns2:approved>true</ns2:approved>
```

```
        </ns2:attribute>
        <ns2:attribute>
          <ns2:key>unitid</ns2:key>
          <ns2:value>1256</ns2:value>
        </ns2:attribute>
      </ns2:profile>
    </ns2:FindProfileInfoByAccountIdResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Using CURL to Send JSON Requests and Get Responses

You can use the cURL utility to test the Merchant API. Use this command-line tool to send JSON requests and receive responses.

1. Create an JSON request to the Merchant API and save it to the request.json file.

   If you plan to use username and password authentication, the JSON request must include a security header.

   For example, use the following JSON request to get information about your account:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "FindProfileInfoByAccountIdRequest": {
            "value": 12345678,
            "version": "VERSION_2"
        }
    }
}}
```

2. Send the request to the Merchant API.

   To send the SOAP request that includes a security header, execute the following command:

```
curl --verbose --http1.0 --show-error --data-binary "@request.json" --output
 "response.json" --header "Content-type: application/json;charset=UTF-8" --insecure
 https://www.moneta.ru/services
```

   These CURL commands save the response to your requests to the response.json file.

   The following listing shows a sample JSON response:

```
{"Envelope": {
    "Body": {
        "FindProfileInfoByAccountIdResponse": {
            "accountId": 12345678,
            "profile": {
                "attribute": [
                    {
                        "approved": true,
                        "value": "Last name",
                        "published": true,
                        "key": "last_name"
                    },
                    {
                        "approved": true,
                        "value": "First name",
                        "published": true,
                        "key": "first_name"
                    },
```

```
                    {
                        "value": "1000",
                        "key": "unitid"
                    },
                    {
                        "value": "1008",
                        "key": "profileid"
                    }
                ]
            },
            "currency": "RUB"
        }
    }
}}
```

Chapter 5

Merchant API Definition

Topics:

Base Data Types and Formats

Merchant API uses the following base data types and formats:

- **string**. Data type represents character strings
- **normalizedString**. Represents white space normalized strings, the value space is the set of strings that do not contain the carriage return (#xD), line feed (#xA) nor tab (#x9) characters
- **decimal**. represents a subset of the real numbers, which can be represented by decimal numerals, the value space is the set of numbers that can be obtained by multiplying an integer by a non-positive power of ten
- **integer**. Derived from decimal by fixing the value of fractionDigits to be 0 and disallowing the trailing decimal point
- **long**. Derived from integer by setting the value of maxInclusive to be 9223372036854775807 and minInclusive to be -9223372036854775808
- **int**. Derived from long by setting the value of maxInclusive to be 2147483647 and minInclusive to be -2147483648
- **dateTime**. Represents date and time, the value space is closely related to the dates and times described in ISO 8601, the lexical space is consists of finite-length sequences of characters of the form: `'-'? yyyy '-' mm '-' dd 'T' hh ':' mm ':' ss ('.' s+)? (zzzzzz)?`, server time is set to MSD which is Moscow time (GMT+3 with daylight saving)

Important: Requests to the Merchant API must use the UTF-8 encoding.

Exceptions that might be thrown are:

- Service is temporarily unavailable

- Wrong payment password
- Specified account does not exist
- Specified amount is incorrect
- Wrong value of argument in transfer
- Not enough money for transfer
- Processing is impossible. No such objects have been found.
- Payer account has been set incorrectly
- Payee account has been set incorrectly
- Account is inactive
- Payer amount has been set incorrectly
- Payee amount has been set incorrectly
- Specified values already exist
- Transaction is queued
- Access is denied
- Transaction is not allowed
- You are using expired information
- Transaction has been put on acknowledgment

Global Type Definition

AbstractAttributeObject Complex Type

Specifies additional attributes.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema" abstract="true"
                 name="AbstractAttributeObject">
   <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="attribute"
                   type="tns:KeyValueApprovedAttribute"/>
   </xsd:sequence>
</xsd:complexType>
```

AccountAccessInfo Complex Type

Provides information about delegated access to a merchant account.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="AccountAccessInfo">
   <xsd:sequence>
      <xsd:element name="accessToWrite" type="xsd:boolean"/>
      <xsd:element name="accessToTakenOut" type="xsd:boolean"/>
      <xsd:element name="accessToTakenIn" type="xsd:boolean"/>
   </xsd:sequence>
</xsd:complexType>
```

Description

accessToWrite

Access for changing account settings.

Type: boolean

accessToTakenOut

Access for transferring funds from the account.

Type: boolean

accessToTakenIn

Access for transferring funds to the account.

Type: boolean

AccountId Simple Type

Specifies a MONETA.RU account number. An account number in MONETA.RU must contain 1-10 digits.

Type: long

XML Schema

```
<xsd:simpleType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="AccountId">
   <xsd:restriction base="xsd:long">
      <xsd:minInclusive value="1"/>
      <xsd:totalDigits value="10"/>
   </xsd:restriction>
</xsd:simpleType>
```

AccountInfo Complex Type

Information about a MONETA.RU account.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="AccountInfo">
   <xsd:sequence>
      <xsd:element name="id" type="tns:AccountId"/>
      <xsd:element minOccurs="0" name="currency" type="tns:Currency"/>
      <xsd:element minOccurs="0" name="balance" type="tns:Money"/>
      <xsd:element minOccurs="0" name="availableBalance" type="tns:Money"/>
      <xsd:element name="type" type="tns:AccountType"/>
      <xsd:element name="status" type="tns:AccountStatus"/>
      <xsd:element minOccurs="0" name="alias" type="xsd:string"/>
      <xsd:element minOccurs="0" name="onSuccessfulDebitUrl" type="xsd:string"/>
      <xsd:element minOccurs="0" name="onSuccessfulCreditUrl" type="xsd:string"/>
      <xsd:element minOccurs="0" name="signature" type="xsd:string"/>
      <xsd:element minOccurs="0" name="lowBalanceThreshold" type="tns:Money"/>
      <xsd:element minOccurs="0" name="highBalanceThreshold" type="tns:Money"/>
      <xsd:element minOccurs="0" name="accountAccess" type="tns:AccountAccessInfo"/>
      <xsd:element minOccurs="0" name="prototypeAccountId" type="xsd:long"/>
      <xsd:element minOccurs="0" name="onCancelledDebitUrl" type="xsd:string"/>
      <xsd:element minOccurs="0" name="onCancelledCreditUrl" type="xsd:string"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="attribute"
                   type="tns:KeyValueApprovedAttribute"/>
   </xsd:sequence>
</xsd:complexType>
```

Description

id

MONETA.RU account number.

Type: *AccountId Simple Type* on page 51

currency

Currency of the specified account.

Optional element.

Type: *Currency Simple Type* on page 66

balance

Current balance of account

Optional element.

Type: *Money Simple Type* on page 77

availableBalance

Account balance.

Optional element.

Type: *Money Simple Type* on page 77

type

MONETA.RU account type.

Type: *AccountType Simple Type* on page 58

status

Status of the MONETA.RU account.

Type: *AccountStatus Simple Type* on page 57

alias

Account alias.

Optional element.

Type: string

onSuccessfulDebitUrl

Specifies the URL of the script that MONETA.RU calls after debiting the payer's account.

Optional element.

Type: string

onSuccessfulCreditUrl

Specifies the URL of the script that MONETA.RU calls after crediting the payee's account.

Optional element.

Type: string

signature

Signature that MONETA.RU uses to verify data integrity submitted from a payment form.

Optional element.

Type: string

lowBalanceThreshold

Specifies the minimum balance threshold for an account. If the balance drops below the threshold, MONETA.RU sends daily notifications to the account owner.

Optional element.

Type: *Money Simple Type* on page 77

highBalanceThreshold

Specifies the maximum balance threshold for an account. If the balance exceeds the threshold, MONETA.RU sends daily notifications to the account owner.

Optional element.

Type: *Money Simple Type* on page 77

accountAccess

Account access information. MONETA.RU provides this information if the account is delegated. If the account is not delegated, this element is omitted.

Optional element.

Type: *AccountAccessInfo Complex Type* on page 50

prototypeAccountId

Prototype account number. Settings from this account are used as default values.

Optional element.

Type: long

onCancelledDebitUrl

Specifies the URL of the script that MONETA.RU calls if a debit transaction is canceled.

Optional element.

Type: string

onCancelledCreditUrl

Specifies the URL of the script that MONETA.RU calls if a deposit transaction is canceled.

Optional element.

Type: string

attribute

Additional account attributes.

> **Important:** MONETA.RU returns this element only if you set the version attribute of your request to**VERSION_2**.

Valid values for the key and value elements:

- **subtypeid**. Account subtype. Can see available values in "CreateAccountRequest": element "subTypeId".
- **paymentPasswordType**
  - **STATIC**. Static payment password.
  - **SEQUENCE_BY_ORDER**. Ordered sequence.
  - **SMS_SIMPLE**. SMS password.
  - **SMS_SESSION**. SMS password (sessional).
  - **TOTP_RFC6238**. Time-based one-time password algorithm.
- **paymentPasswordChallengeRequired**. If the response contains an attribute with the key="paymentPasswordChallengeRequired" and value="true", you must call GetAccountPaymentPasswordChallengeRequest for the payer's account to get the paymentPasswordChallenge element. This element is used in the following requests: PaymentRequest, TransferRequest, AuthoriseTransactionRequest.
- **paymentPasswordExpirationDate**. Payment password expiration date. Example: 2017-01-09T14:11:24.000+03:00
- **alias**. Account alias.
- **primary**. Value is **true**, if the account is primary.

  Value is used in FindAccountsListRequest method only.
- **delegated**. Value is **true**, if the account is delegated.
- **balanceChangesDate**. Balance changes. Example: 2017-01-09T14:11:24.000+03:00
- **bankAccountForCredits**. Bank account for credits.
- **bankAccountForDebits**. Bank account for debits.
- **assistantVersion**. MONETA.Assistant version. Available values:

  v1

  v2

  v3
- **interfacetype**. Interface type.

  1 - MONETA.Assistant.
- **testmode**. Test mode (true|false).
- **paymentsystem_limitids**. Payment method IDs.
- **paymentsystem_unitid**. Default payment method.

- **checkurl**. Check URL.
- **payurl**. Pay URL.
- **httpmethod**. HTTP method (PayUrl, CheckUrl).
- **signature**. Code of data integrity verification.
- **issignaturemandatory**. Mandatory payment form signature (true|false).
- **redefinesettingsinurl**. Settings can be redefined in URL (true|false).
- **successurl**. Success URL.
- **failurl**. Fail URL.
- **inprogressurl**. InProgress URL.
- **returnurl**. Return URL.
- **assistantformtarget**. Target (redirect in iframe).
- **onsuccessdebiturl**. URL on debiting.
- **onsuccesscrediturl**. URL on crediting.
- **oncancelleddebiturl**. URL on cancelled debit.
- **oncancelledcrediturl**. URL on cancelled credit.
- **onauthoriseurl**. URL on authorising.

Optional element.

Type: *KeyValueApprovedAttribute Complex Type* on page 76

AccountPaymentPasswordType Simple Type

Payment password type.

Type: string

Valid values:

- **STATIC**.Static payment password.
- **SMS_SIMPLE**.SMS password.

XML Schema

```
<xsd:simpleType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="AccountPaymentPasswordType">
   <xsd:restriction base="xsd:string">
      <xsd:enumeration value="STATIC"/>
      <xsd:enumeration value="SMS_SIMPLE"/>
   </xsd:restriction>
</xsd:simpleType>
```

AccountRelation Complex Type

Specifies delegated access to a MONETA.RU account.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="AccountRelation">
   <xsd:sequence>
      <xsd:element name="accountId" type="tns:AccountId"/>
      <xsd:element name="principalEmail" type="tns:Email"/>
      <xsd:element minOccurs="0" name="canViewAccount" type="xsd:boolean"/>
      <xsd:element minOccurs="0" name="canEditAccount" type="xsd:boolean"/>
      <xsd:element minOccurs="0" name="canProcessOperation" type="xsd:boolean"/>
   </xsd:sequence>
</xsd:complexType>
```

Description

accountId

MONETA.RU account number.

Type: *AccountId Simple Type* on page 51

principalEmail

Email of the user who has delegated access to the specified account.

Type: *Email Simple Type* on page 68

canViewAccount

Indicates whether the specified user can view the account information. Valid valus are:

- **true**. The specified user can view the account information.
- **false**. The specified user cannot view the account information.

Optional element.

Type: boolean

canEditAccount

Indicates whether the specified user can change the account settings. Valid values are:

- **true**. The specified user can change the account settings.
- **false**. The specified user cannot change the account settings.

Optional element.

Type: boolean

canProcessOperation

Indicates whether the specified user can process transactions. Valid values are:

- **true**. The specified user can process transactions.
- **false**. The specified user cannot process transactions.

Optional element.

Type: boolean

AccountStatementPaging Complex Type

Type for Account statement paging.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="AccountStatementPaging">
   <xsd:sequence>
      <xsd:element minOccurs="0" name="date" type="xsd:dateTime"/>
      <xsd:element minOccurs="0" name="id" type="xsd:long"/>
   </xsd:sequence>
</xsd:complexType>
```

AccountStatementRecordType Complex Type

Operation details.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="AccountStatementRecordType">
   <xsd:sequence>
      <xsd:element name="date" type="xsd:dateTime"/>
      <xsd:element minOccurs="0" name="operationId" type="xsd:long"/>
      <xsd:element minOccurs="0" name="operationView" type="xsd:int"/>
      <xsd:element minOccurs="0" name="correspondentName" type="xsd:string"/>
      <xsd:element name="correspondentAccountId" type="tns:AccountId"/>
      <xsd:element minOccurs="0" name="correspondentAccount" type="xsd:string"/>
      <xsd:element minOccurs="0" name="debitAmount" type="tns:Money"/>
```

```
      <xsd:element minOccurs="0" name="creditAmount" type="tns:Money"/>
      <xsd:element minOccurs="0" name="payerINN" type="xsd:string"/>
      <xsd:element minOccurs="0" name="payeeINN" type="xsd:string"/>
      <xsd:element minOccurs="0" name="documentNumber" type="xsd:string"/>
      <xsd:element minOccurs="0" name="BIK" type="xsd:string"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="operationAttributes"
                   type="tns:KeyValueAttribute"/>
      <xsd:element minOccurs="0" name="description" type="xsd:string"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="actions"
 type="xsd:string"/>
   </xsd:sequence>
</xsd:complexType>
```

Description

date

Operation date.

Type: dateTime

operationId

MONETA.RU transaction ID.

Optional element.

Type: long

operationView

Operation type.

Optional element.

Type: int

correspondentName

Correspondent.

Optional element.

Type: string

correspondentAccountId

Correspondent account number in MONETA.RU.

Type: *AccountId Simple Type* on page 51

correspondentAccount

Corresponding account.

Optional element.

Type: string

debitAmount

Debit amount.

Optional element.

Type: *Money Simple Type* on page 77

creditAmount

Credit amount.

Optional element.

Type: *Money Simple Type* on page 77

payerINN

Payer INN.

Optional element.

Type: string

payeeINN

Payee INN.

Optional element.

Type: string

documentNumber

The number of payment document.

Optional element.

Type: string

BIK

BIK.

Optional element.

Type: string

operationAttributes

Operation attributes.

The list of returned attributes is setting in AccountStatementRequest operationPropertyNames

.

Optional element.

Type: *KeyValueAttribute Complex Type* on page 76

description

Operation purpose.

Optional element.

Type: string

actions

Actions, you can do with this record. You can receive this field if set in request: **showActions=true**.

- **receipt** - can get "Receipt" (/report/receipt.htm?operationId={operationId}&format=pdf).
- **paymentOrder** - can get "Payment order" (/report/paymentOrder.htm?operationId={operationId}&format=pdf).
- **form0401067** - can get "Bank order" (/report/form0401067.htm?operationId={operationId}&format=pdf).

Optional element.

Type: string

AccountStatus Simple Type

Specifies a status of a MONETA.RU account.

Type: int

Valid values:

- **1**.Active account.
- **2**.Blocked account.
- **3**.Closed account.

XML Schema

```
<xsd:simpleType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="AccountStatus">
```

```
   <xsd:restriction base="xsd:int">
      <xsd:enumeration value="1"/>
      <xsd:enumeration value="2"/>
      <xsd:enumeration value="3"/>
   </xsd:restriction>
</xsd:simpleType>
```

AccountType Simple Type

Specifies an account type in MONETA.RU.

Type: int

Valid values:

- **1**.Standard account.
- **2**.Advanced account.
- **3**.Bonus account.
- **4**.Loan account.
- **5**.Special-purpose account.
- **6**.Commission account.
- **7**.Bank account.

XML Schema

```
<xsd:simpleType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="AccountType">
   <xsd:restriction base="xsd:int">
      <xsd:enumeration value="1"/>
      <xsd:enumeration value="2"/>
      <xsd:enumeration value="3"/>
      <xsd:enumeration value="4"/>
      <xsd:enumeration value="5"/>
      <xsd:enumeration value="6"/>
      <xsd:enumeration value="7"/>
   </xsd:restriction>
</xsd:simpleType>
```

AsyncStatus Simple Type

The status of a asynchronous task in MONETA.RU.

Type: string

Valid values:

- **INPROGRESS**.MONETA.RU is processing the asynchronous task.
- **CREATED**.Asynchronous task is registered in MONETA.RU.

XML Schema

```
<xsd:simpleType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="AsyncStatus">
   <xsd:restriction base="xsd:string">
      <xsd:enumeration value="INPROGRESS"/>
      <xsd:enumeration value="CREATED"/>
   </xsd:restriction>
</xsd:simpleType>
```

AuthoriseTransactionBatchRequestType Complex Type

Transaction parameters for requests in batch mode.

Type: *EntityBatchRequestType Complex Type* on page 69.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                 name="AuthoriseTransactionBatchRequestType">
   <xsd:complexContent>
      <xsd:extension base="tns:EntityBatchRequestType">
         <xsd:sequence>
            <xsd:element maxOccurs="unbounded" minOccurs="1" name="transaction"
                         type="tns:AuthoriseTransactionRequestType"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

Description

transaction

The set of transactions that must be processed in a single batch. MONETA.RU processes transactions in the order that they are defined in the request.

Required element.

Type: *AuthoriseTransactionRequestType Complex Type* on page 59

AuthoriseTransactionRequestType Complex Type

Request parameters for an authorization hold on the payment amount in a customer's account.

Type: *TransactionRequestType Complex Type* on page 103.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                 name="AuthoriseTransactionRequestType">
   <xsd:complexContent>
      <xsd:extension base="tns:TransactionRequestType">
         <xsd:sequence>
            <xsd:element minOccurs="0" name="transactionId" type="xsd:long"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

Description

transactionId

MONETA.RU transaction ID.

Optional element.

Type: long

BankAccount Complex Type

Bank account details. The information is returned as a list of key-value pairs.

Valid keys are:

- **is_international**. Indicates whether the bank uses international account details. Valid values are:
  - **true**. International bank account details.
  - **false**. Russian bank account details.
- **bank**. Bank name.
- **bik**. SWIFT/BIC number.
- **account**. IBAN/settlement account.
- **corr_bank**. Correspondent bank.
- **corr_account**. Correspondent account.

- **country**. Country.
- **state**. Region/state.
- **city**. City.
- **kbk**. Budget classification code.
- **oktmo**. Russian Classifier of Municipal Unit Territories.
- **kpp**. Code of reason for tax registration.
- **username**. Payee's name.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="BankAccount">
   <xsd:sequence>
      <xsd:element minOccurs="0" name="id" type="xsd:long"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="attribute"
                   type="tns:KeyValueApprovedAttribute"/>
   </xsd:sequence>
</xsd:complexType>
```

Description

id

Bank account ID.

Optional element.

Type: long

attribute

Optional element.

Type: *KeyValueApprovedAttribute Complex Type* on page 76

CTID Simple Type

Specifies a unique identifier of a transaction in the merchant system.

Type: string

Maximum length: 255

XML Schema

```
<xsd:simpleType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="CTID">
   <xsd:restriction base="xsd:string">
      <xsd:maxLength value="255"/>
   </xsd:restriction>
</xsd:simpleType>
```

CancelTransactionBatchRequestType Complex Type

Request parameters for canceling transactions in batch mode.

Type: *EntityBatchRequestType Complex Type* on page 69.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                 name="CancelTransactionBatchRequestType">
   <xsd:complexContent>
      <xsd:extension base="tns:EntityBatchRequestType">
         <xsd:sequence>
            <xsd:element maxOccurs="unbounded" minOccurs="1" name="transaction"
                         type="tns:CancelTransactionRequestType"/>
         </xsd:sequence>
      </xsd:extension>
```

```
   </xsd:complexContent>
</xsd:complexType>
```

Description

transaction

The set of transactions that must be processed in a single batch. MONETA.RU processes transactions in the order that they are defined in the request.

Required element.

Type: *CancelTransactionRequestType Complex Type* on page 62

CancelTransactionBatchResponseType Complex Type

Response parameters for canceling transactions in batch mode.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                 name="CancelTransactionBatchResponseType">
   <xsd:choice>
      <xsd:element name="transaction" type="tns:CancelTransactionResponseType"/>
      <xsd:sequence>
         <xsd:element minOccurs="0" name="error" type="xsd:string"/>
         <xsd:element minOccurs="0" name="errorCode" type="xsd:string"/>
         <xsd:element minOccurs="0" name="transactionId" type="xsd:long"/>
      </xsd:sequence>
   </xsd:choice>
</xsd:complexType>
```

Description

error

If an error occurs when processing a transaction, the error element contains an error description.

If the transaction completes successfully, the error element is empty and the transaction element contains transaction details.

Optional element.

Type: string

errorCode

If an error occurs when processing a transaction, the errorCode element contains an error code.

MONETA.RU returns this element only if you set the version attribute of your request to **VERSION_2**.

See the faultDetail element for the error code descriptions.

Optional element.

Type: string

transactionId

If an error occurs when processing a transaction, the transactionId element contains an transaction ID.

MONETA.RU returns this element only if you set the version attribute of your request to **VERSION_5**.

Optional element.

Type: long

CancelTransactionRequestType Complex Type

Request parameters for canceling transactions.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="CancelTransactionRequestType">
   <xsd:sequence>
      <xsd:element name="transactionId" type="xsd:long"/>
      <xsd:element minOccurs="0" name="protectionCode" type="xsd:string"/>
      <xsd:element minOccurs="0" name="description" type="tns:Description"/>
   </xsd:sequence>
</xsd:complexType>
```

Description

transactionId

Transaction ID.

Type: long

protectionCode

Protection code.

Optional element.

Type: string

description

Transaction description or comments.

Optional element.

Type: *Description Simple Type* on page 66

CancelTransactionResponseType Complex Type

Response parameters for cancelling transactions.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="CancelTransactionResponseType">
   <xsd:sequence>
      <xsd:element name="transactionId" type="xsd:long"/>
      <xsd:element minOccurs="0" name="operationStatus" type="tns:OperationStatus"/>
   </xsd:sequence>
</xsd:complexType>
```

Description

transactionId

Transaction ID.

Type: long

operationStatus

Transaction status.

Optional element.

Type: *OperationStatus Simple Type* on page 81

CommonOperationTemplateParameters Complex Type

Type: *OperationTemplateAmount Complex Type* on page 83.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                 name="CommonOperationTemplateParameters">
   <xsd:complexContent>
      <xsd:extension base="tns:OperationTemplateAmount"/>
   </xsd:complexContent>
</xsd:complexType>
```

ConfirmTransactionBatchRequestType Complex Type

Request parameters for confirming transactions in batch mode.

Type: *EntityBatchRequestType Complex Type* on page 69.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                 name="ConfirmTransactionBatchRequestType">
   <xsd:complexContent>
      <xsd:extension base="tns:EntityBatchRequestType">
         <xsd:sequence>
            <xsd:element maxOccurs="unbounded" minOccurs="1" name="transaction"
                         type="tns:ConfirmTransactionRequestType"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

Description

transaction

The set of transactions that must be processed in a single batch. MONETA.RU processes transactions in the order that they are defined in the request.

Required element.

Type: *ConfirmTransactionRequestType Complex Type* on page 63

ConfirmTransactionRequestType Complex Type

Parameters for requests that confirm transactions.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="ConfirmTransactionRequestType">
   <xsd:sequence>
      <xsd:element name="transactionId" type="xsd:long"/>
      <xsd:element minOccurs="0" name="paymentPassword" type="tns:Password"/>
      <xsd:element minOccurs="0" name="protectionCode" type="xsd:string"/>
      <xsd:element minOccurs="0" name="operationInfo" type="tns:OperationInfo"/>
      <xsd:element minOccurs="0" name="amount" type="tns:Money"/>
      <xsd:element minOccurs="0" name="isPayerAmount" type="xsd:boolean"/>
      <xsd:element minOccurs="0" name="paymentPasswordChallenge" type="xsd:string"/>
      <xsd:element minOccurs="0" name="clientTransaction" type="tns:CTID"/>
   </xsd:sequence>
</xsd:complexType>
```

Description

transactionId

Transaction ID.

Type: long

paymentPassword

Payment password for the payer's account.

Optional element.

Type: *Password Simple Type* on page 89

protectionCode

Protection code.

Optional element.

Type: string

operationInfo

Key-value pairs that will be saved as transaction attributes. For dates, use the following format: dd.MM.yyyy HH:mm:ss

Optional element.

Type: *OperationInfo Complex Type* on page 78

amount

Transaction amount. The currency of the transaction is specified by the isPayerAmount element. If a request does not include the isPayerAmount element, MONETA.RU uses the following rules to determine the currency of the transaction:

- If a user has access only to the payer's account, MONETA.RU uses the currency of the payer's account for the transaction.
- If a user has access only to the payee's account, MONETA.RU uses the currency of the payee's account for the transaction.

Optional element.

Type: *Money Simple Type* on page 77

isPayerAmount

This element is required if a user has access to the payer's and to the payee's accounts. Valid values:

- **true**. MONETA.RU uses the currency of the payer's account for the transaction amount.
- **false**. MONETA.RU uses the currency of the payee's account for the transaction amount.

Optional element.

Type: boolean

paymentPasswordChallenge

Challenge passcode that you received in the GetAccountPaymentPasswordChallenge response in the paymentPasswordChallenge element. Specify this element in the following cases:

- If a user gets payment passwords by SMS, set paymentPasswordChallenge to SMS. Set paymentPassword to the value that the user receives in the SMS from MONETA.RU.
- If a user gets a sequence number (index) for a password from a list of transaction authentication numbers (TANs), set paymentPasswordChallenge to the TAN index. Set paymentPassword to the TAN that has the specified index.

Optional element.

Type: string

clientTransaction

Merchant transaction ID.

Optional element.

Type: *CTID Simple Type* on page 60

Contract Complex Type

Contract information in the list of key-value pairs.

Valid contract keys are:

- **num**. Contract number.
- **unitid**. Unique identifier of the contract.
- **own**. Indicates whether the contract is created for the specified user account. Valid values are:
  - **true**. The contract is created for the account which ID is specified in the unitid element.
  - **false**. The contract is created for the parent account.
- **unitname**. The name of the parent account, if the own element is set to false.
- **datestart**. The start date of the contract.
- **datesigned**. Contract signing date.
- **datestop**. The end date of the contract.
- **status**. Contract status. Valid values are:
  - ACTIVE
  - RESTRICTED
  - INACTIVE
- **type**. Inheritance type. Valid values are:
  - **PARTNER**. Non-inherited.
  - **VIRTUAL**. Inherited.
- **usage**. Contract usage:
  - **DIRECT**. Direct contact.
  - **TECH**. Technical contact.
  - **AGENT**. Agent contact.
- **baseid**. Unique identifier of the base contract for agent contract.
- **category**. Contract subject. Valid values are:
  - **INDIVIDUALS_ANONYMOUS**. Anonymous individuals.
  - **INDIVIDUALS_SIMPLIFIED**. Simplified identified individuals.
  - **INDIVIDUALS_SIMPLIFIED_SELF_EMPLOYED**. Self employed simplified identified individuals.
  - **INDIVIDUALS_PERSON**. Identified individuals.
  - **INDIVIDUALS_PERSON_SELF_EMPLOYED**. Self employed identified individuals.
  - **INDIVIDUALS_ENTREPRENEUR**. Entrepreneur individuals.
  - **ORGANIZATION_LEGAL_PERSON**. Legal person organization.
  - **SERVICE_ORGANIZATION**. Service / Service organization.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="Contract">
   <xsd:sequence>
      <xsd:element minOccurs="0" name="id" type="xsd:long"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="attribute"
                   type="tns:KeyValueApprovedAttribute"/>
   </xsd:sequence>
</xsd:complexType>
```

Description

id

Contract ID.

Optional element.

Type: long

attribute

Optional element.

Type: *KeyValueApprovedAttribute Complex Type* on page 76

Currency Simple Type

Specifies the currency that is used in a transaction.

Type: string

Valid values:

- **RUB**
- **USD**
- **EUR**
- **GBP**
- **CNY**

XML Schema

```
<xsd:simpleType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="Currency">
   <xsd:restriction base="xsd:string">
      <xsd:enumeration value="RUB"/>
      <xsd:enumeration value="USD"/>
      <xsd:enumeration value="EUR"/>
      <xsd:enumeration value="GBP"/>
      <xsd:enumeration value="CNY"/>
   </xsd:restriction>
</xsd:simpleType>
```

Description Simple Type

Description or comments that are used in complex types.

Type: normalizedString

Maximum length: 255

XML Schema

```
<xsd:simpleType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="Description">
   <xsd:restriction base="xsd:normalizedString">
      <xsd:maxLength value="255"/>
   </xsd:restriction>
</xsd:simpleType>
```

DirectDebitOperationTemplateParameters Complex Type

Type: *OperationTemplateAmountRange Complex Type* on page 85.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                 name="DirectDebitOperationTemplateParameters">
   <xsd:complexContent>
      <xsd:extension base="tns:OperationTemplateAmountRange">
         <xsd:sequence>
            <xsd:element name="active" type="xsd:boolean"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

Description

active

Type: boolean

Document Complex Type

Specifies information about a document in MONETA.RU.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="Document">
   <xsd:sequence>
      <xsd:element minOccurs="0" name="id" type="xsd:long"/>
      <xsd:element minOccurs="0" name="type" type="tns:DocumentType"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="attribute"
                   type="tns:KeyValueApprovedAttribute"/>
      <xsd:element minOccurs="0" name="hasAttachedFiles" type="xsd:boolean"/>
   </xsd:sequence>
</xsd:complexType>
```

Description

id

Unique identifier of a document in MONETA.RU.

Optional element.

Type: long

type

Document type.

Optional element.

Type: *DocumentType Simple Type* on page 68

attribute

Document attributes in MONETA.RU.

Document information is specified in a list of key-value pairs

Valid keys for the PASSPORT and MILITARY_ID document types are:

- **SERIES**. Document series.
- **NUMBER**. Document number.
- **ISSUER**. Issuing authority.
- **ISSUED**. Issue date.
- **COMMENTS**. Optional description.
- **MODIFICATIONDATE**. The date of the last change in the document.
- **customfield:*custom_attribute_name***. Custom attribute.

  A document might include multiple custom attributes.

  > **Note:** The *custom_attribute_name* part of the custom attribute key might include up to 32 characters.

Valid keys for the DRIVING_LICENCE document type:

- **SERIES**. Document series.
- **NUMBER**. Document number.
- **ISSUER**. Issuing authority.
- **ISSUED**. Issue date.
- **EXPIRATIONDATE**. Expiration date.
- **COMMENTS**. Optional description.
- **MODIFICATIONDATE**. The date of the last change in the document.
- **customfield:*custom_attribute_name***. Custom attribute.

  A document might include multiple custom attributes.

  > **Note:** The *custom_attribute_name* part of the custom attribute key might include up to 32 characters.

Valid keys for the OTHER document type:

- **COMMENTS**. Document name, description, comments, or explanation.
- **MODIFICATIONDATE**. The date of the last change in the document.
- **customfield:*custom_attribute_name***. Custom attribute.

  A document might include multiple custom attributes.

  > **Note:** The *custom_attribute_name* part of the custom attribute key might include up to 32 characters.

Optional element.

Type: *KeyValueApprovedAttribute Complex Type* on page 76

hasAttachedFiles

Indicates whether the document has attachments.

> **Tip:** To get an attachment, use FindProfileDocumentFilesRequest.

Optional element.

Type: boolean

DocumentType Simple Type

Document types that are supported in MONETA.RU.

Type: string

Valid values:

- **PASSPORT**.Civil passport.
- **DRIVING_LICENCE**.Driver's licence.
- **MILITARY_ID**.Military ID.
- **OTHER**.Other document.
- **PARTNER_LICENCE**.Licence for partner activity.
- **IDENTITY_DOCUMENT**.PASSPORT for non resident partner.
- **ADMINISTRATIVE**.Document available only for administrator.
- **CUSTOM**

XML Schema

```
<xsd:simpleType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="DocumentType">
   <xsd:restriction base="xsd:string">
      <xsd:enumeration value="PASSPORT"/>
      <xsd:enumeration value="DRIVING_LICENCE"/>
      <xsd:enumeration value="MILITARY_ID"/>
      <xsd:enumeration value="OTHER"/>
      <xsd:enumeration value="PARTNER_LICENCE"/>
      <xsd:enumeration value="IDENTITY_DOCUMENT"/>
      <xsd:enumeration value="ADMINISTRATIVE"/>
      <xsd:enumeration value="CUSTOM"/>
   </xsd:restriction>
</xsd:simpleType>
```

Email Simple Type

Email.

Type: string

XML Schema

```
<xsd:simpleType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="Email">
   <xsd:restriction base="xsd:string">
      <xsd:minLength value="5"/>
```

```
      <xsd:whiteSpace value="collapse"/>
   </xsd:restriction>
</xsd:simpleType>
```

Entity Complex Type

A base type that contains the **version** attribute.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="Entity">
   <xsd:attribute default="VERSION_1" name="version" type="tns:Version" use="optional"/
>
</xsd:complexType>
```

EntityBatchRequestType Complex Type

Parameters for requests in batch mode.

Type: *Entity Complex Type* on page 69.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="EntityBatchRequestType">
   <xsd:complexContent>
      <xsd:extension base="tns:Entity">
         <xsd:sequence>
            <xsd:element name="transactional" type="xsd:boolean"/>
            <xsd:element name="exitOnFailure" type="xsd:boolean"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

Description

transactional

Indicates whether to merge all transfers during batch processing into a single transaction. Valid values are:

- **True**. If an error occurs, all of the transactions are canceled. Only transactions with MONETA.RU accounts are supported. Withdrawals to third-party payment systems are not allowed.
- **False**. If an error occurs, MONETA.RU cancels only the current transaction. Processed transactions are not rolled back. Any transactions that are supported by the Transfer request are allowed. If you set the exitOnFailure element to false, the Merchant API skips the current transaction and continues processing other transactions.

Type: boolean

exitOnFailure

If the transactional element is set to false, indicates whether to stop batch processing if an error occurs in one of the transactions.

Type: boolean

Fee Simple Type

The transaction fee amount.

Type: decimal

XML Schema

```
<xsd:simpleType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="Fee">
   <xsd:restriction base="xsd:decimal">
      <xsd:fractionDigits value="4"/>
   </xsd:restriction>
</xsd:simpleType>
```

File Complex Type

Binary data.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="File">
   <xsd:sequence>
      <xsd:element name="documentId" type="xsd:long"/>
      <xsd:element xmlns:xmime="http://www.w3.org/2005/05/xmlmime" minOccurs="0"
 name="blob"
                   type="xsd:base64Binary"
                   xmime:expectedContentTypes="application/octet-stream"/>
      <xsd:element minOccurs="0" name="approved" type="xsd:boolean"/>
      <xsd:element minOccurs="0" name="fileId" type="xsd:long"/>
      <xsd:element minOccurs="0" name="mimeType" type="xsd:string"/>
      <xsd:element minOccurs="0" name="title" type="xsd:string"/>
      <xsd:element minOccurs="0" name="requestUri" type="xsd:string"/>
   </xsd:sequence>
</xsd:complexType>
```

Description

documentId

Unique identifier of the document to which the file belongs.

Type: long

blob

File contents. Use MTOM (Message Transmission Optimization Mechanism) to send and fetch the data.

Optional element.

Type: base64Binary

approved

Indicates whether the file is verified or not.

Optional element.

Type: boolean

fileId

File ID.

Optional element.

Type: long

mimeType

MIME type of the file.

For example, image/jpeg for an image in jpeg format.

Optional element.

Type: string

title

File name or description.

Optional element.

Type: string

requestUri

Direct link for binary data

Optional element.

Type: string

ForecastTransactionResponseType Complex Type

Transaction amount and estimated fees for a transaction.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                 name="ForecastTransactionResponseType">
   <xsd:sequence>
      <xsd:element name="payer" type="xsd:long"/>
      <xsd:element name="payerCurrency" type="tns:Currency"/>
      <xsd:element name="payerAmount" type="tns:Money"/>
      <xsd:element minOccurs="0" name="payerFee" type="tns:Money"/>
      <xsd:element minOccurs="0" name="payerAccountTotal" type="tns:Money"/>
      <xsd:element name="payee" type="xsd:long"/>
      <xsd:element name="payeeCurrency" type="tns:Currency"/>
      <xsd:element name="payeeAmount" type="tns:Money"/>
      <xsd:element minOccurs="0" name="payeeFee" type="tns:Money"/>
      <xsd:element minOccurs="0" name="payeeAccountTotal" type="tns:Money"/>
      <xsd:element minOccurs="0" name="payerAlias" type="xsd:string"/>
      <xsd:element minOccurs="0" name="payeeAlias" type="xsd:string"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="attribute"
                   type="tns:KeyValueApprovedAttribute"/>
   </xsd:sequence>
</xsd:complexType>
```

Description

payer

Account number of the payer.

Type: long

payerCurrency

Currency of the payer's account.

Type: *Currency Simple Type* on page 66

payerAmount

The amount to be transferred from the payer's account.

Type: *Money Simple Type* on page 77

payerFee

Transaction fee to the payer.

Optional element.

Type: *Money Simple Type* on page 77

payerAccountTotal

Payer's account balance changes.

> **Important:** MONETA.RU returns this element only if you set the version attribute of your request to **VERSION_4**.

Optional element.

Type: *Money Simple Type* on page 77

payee

Payee's account number.

Type: long

payeeCurrency

Currency of the payee's account.

Type: *Currency Simple Type* on page 66

payeeAmount

The amount to be transferred to the payee's account.

Type: *Money Simple Type* on page 77

payeeFee

Transaction fee to the payee.

Optional element.

Type: *Money Simple Type* on page 77

payeeAccountTotal

Payee's account balance changes.

> **Important:** MONETA.RU returns this element only if you set the version attribute of your request to **VERSION_4**.

Optional element.

Type: *Money Simple Type* on page 77

payerAlias

Payer's account alias.

Optional element.

Type: string

payeeAlias

Payee's account alias.

Optional element.

Type: string

attribute

Additional parameters.

> **Important:** MONETA.RU returns this element only if you set the version attribute of your request to **VERSION_4**.

Optional element.

Type: *KeyValueApprovedAttribute Complex Type* on page 76

InfoTariff Complex Type

Information on tariff

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="InfoTariff">
   <xsd:sequence>
      <xsd:element minOccurs="0" name="sourceRangeCurrency" type="tns:Currency"/>
      <xsd:element minOccurs="0" name="sourceAmountMin" type="tns:Money"/>
      <xsd:element minOccurs="0" name="sourceAmountMax" type="tns:Money"/>
```

```
      <xsd:element minOccurs="0" name="targetRangeCurrency" type="tns:Currency"/>
      <xsd:element minOccurs="0" name="targetAmountMin" type="tns:Money"/>
      <xsd:element minOccurs="0" name="targetAmountMax" type="tns:Money"/>
   </xsd:sequence>
</xsd:complexType>
```

Description

sourceRangeCurrency

The payer's account currency for which the calculated limit of the transaction amount.

Optional element.

Type: *Currency Simple Type* on page 66

sourceAmountMin

The minimum amount to the payer that is permitted in operation.

Optional element.

Type: *Money Simple Type* on page 77

sourceAmountMax

The maximum amount to the payer that is permitted in operation.

Optional element.

Type: *Money Simple Type* on page 77

targetRangeCurrency

Payee account currency for which the calculated limit of the transaction amount.

Optional element.

Type: *Currency Simple Type* on page 66

targetAmountMin

The minimum amount to the payee that is permitted in operation.

Optional element.

Type: *Money Simple Type* on page 77

targetAmountMax

The maximum amount to the payee that is permitted in operation.

Optional element.

Type: *Money Simple Type* on page 77

InfoUrl Complex Type

Fully information about URL of the payment system.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="InfoUrl">
   <xsd:sequence>
      <xsd:element minOccurs="0" name="href" type="xsd:string"/>
      <xsd:element minOccurs="0" name="text" type="xsd:string"/>
   </xsd:sequence>
</xsd:complexType>
```

Description

href

URL of the payment system.

Optional element.

Type: string

text

URL name of the payment system.

Optional element.

Type: string

InvoiceBatchRequestType Complex Type

Request parameters for creating invoices in batch mode.

Type: *EntityBatchRequestType Complex Type* on page 69.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="InvoiceBatchRequestType">
   <xsd:complexContent>
      <xsd:extension base="tns:EntityBatchRequestType">
         <xsd:sequence>
            <xsd:element maxOccurs="unbounded" minOccurs="1" name="transaction"
                         type="tns:InvoiceRequestType"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

Description

transaction

The set of transactions that must be processed in a single batch. MONETA.RU processes transactions in the order that they are defined in the request.

Required element.

Type: *InvoiceRequestType Complex Type* on page 74

InvoiceRequestType Complex Type

Transaction parameters for creating an invoice.

Type: *Entity Complex Type* on page 69.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="InvoiceRequestType">
   <xsd:complexContent>
      <xsd:extension base="tns:Entity">
         <xsd:sequence>
            <xsd:element minOccurs="0" name="payer" type="xsd:string"/>
            <xsd:element name="payee" type="xsd:long"/>
            <xsd:element minOccurs="0" name="amount" type="tns:Money"/>
            <xsd:element name="clientTransaction" type="tns:CTID"/>
            <xsd:element minOccurs="0" name="description" type="tns:Description"/>
            <xsd:element minOccurs="0" name="mnt_custom1" type="xsd:string"/>
            <xsd:element minOccurs="0" name="mnt_custom2" type="xsd:string"/>
            <xsd:element minOccurs="0" name="mnt_custom3" type="xsd:string"/>
            <xsd:element minOccurs="0" name="operationInfo" type="tns:OperationInfo"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

Description

payer

MONETA.RU account number of the payer.

Optional element.

Type: string

payee

Account number of the payee.

Type: long

amount

Invoice amount.

Optional element.

Type: *Money Simple Type* on page 77

clientTransaction

Merchant transaction ID.

Type: *CTID Simple Type* on page 60

description

Transaction description or comments.

Optional element.

Type: *Description Simple Type* on page 66

mnt_custom1

Custom parameter.

Optional element.

Type: string

mnt_custom2

Custom parameter.

Optional element.

Type: string

mnt_custom3

Custom parameter.

Optional element.

Type: string

operationInfo

Key-value pairs that will be saved as transaction attributes. For dates, use the following format: dd.MM.yyyy HH:mm:ss. To initiate direct debit, pass the "SUBSCRIBERID" attribute with corresponding subscriber id in the value. The direct debit will be processed if subscriber id is specified, the direct debit processing is permitted by user and user has enough funds on his account. If the direct debit is processed successfully, user's account is debited, the transaction status reported in the response is INPROGRESS, crediting the payee's account starts asynchronously. If the direct debit can't be processed or an error occures during the processing, the invoice is left registered and the transaction status is CREATED. USERCONTACT - payment link will be sent via e-mail or SMS (comma separated recepients).

Optional element.

Type: *OperationInfo Complex Type* on page 78

KeyValueApprovedAttribute Complex Type

Specifies extended attributes.

Type: *KeyValueAttribute Complex Type* on page 76.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="KeyValueApprovedAttribute">
   <xsd:complexContent>
      <xsd:extension base="tns:KeyValueAttribute">
         <xsd:sequence>
            <xsd:element maxOccurs="1" minOccurs="0" name="approved"
 type="xsd:boolean"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="published"
 type="xsd:boolean"/>
            <xsd:element maxOccurs="1" minOccurs="0" name="fromPrototype"
 type="xsd:boolean"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

Description

approved

Indicates whether the attribute is confirmed.

For example, users can confirm their email addresses or mobile phone numbers on their own. To confirm their passports, users must contact the MONETA.RU support team.

Optional element.

Type: boolean

published

Indicates whether all users can read this attribute.

> **Important:** This element is optional. MONETA.RU returns and accepts this element only if you set the version attribute of your request to**VERSION_2**. Use it if you need to specify attributes in a request that creates or updates MONETA.RU objects.

Optional element.

Type: boolean

fromPrototype

Indicates that property was read form "prototype object".

> **Important:** MONETA.RU returns this element only if you set the version attribute of your request to **VERSION_3**.

Optional element.

Type: boolean

KeyValueAttribute Complex Type

Specifies additional attributes. Each attribute contains the key element, which specifies the name of the attribute, and the value element that specifies the value of the attribute.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="KeyValueAttribute">
   <xsd:sequence>
      <xsd:element name="key" type="xsd:string"/>
      <xsd:element name="value" nillable="true" type="xsd:string"/>
```

```
   </xsd:sequence>
</xsd:complexType>
```

LegalInformation Complex Type

Legal information in the list of key-value pairs.

Valid keys are:

- **inn**. Tax Reference Number.
- **kpp**. Industrial Enterprises Classifier.
- **ogrn**. Primary State Registration Number (PSRN).
- **ogrnip**. Primary State Registration Number of the Sole Proprietor.
- **okpo**. Russian National Nomenclature of Businesses and Organizations.
- **okved**. Russian National Classifier of Economic Activities.
- **okato**. Russian National Classifier of Administrative and Territorial Objects.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="LegalInformation">
   <xsd:sequence>
      <xsd:element minOccurs="0" name="id" type="xsd:long"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="attribute"
                   type="tns:KeyValueApprovedAttribute"/>
   </xsd:sequence>
</xsd:complexType>
```

Description

id

Legal information ID.

Optional element.

Type: long

attribute

Optional element.

Type: *KeyValueApprovedAttribute Complex Type* on page 76

Money Simple Type

The transaction amount.

Type: decimal

XML Schema

```
<xsd:simpleType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="Money">
   <xsd:restriction base="xsd:decimal">
      <xsd:maxExclusive value="10000000000000"/>
      <xsd:fractionDigits value="2"/>
   </xsd:restriction>
</xsd:simpleType>
```

OperationAmountType Simple Type

Amount type.

Type: string

Valid values:

- **INCOME**.Income
- **EXPENSE**.Expense

XML Schema

```
<xsd:simpleType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="OperationAmountType">
   <xsd:restriction base="xsd:string">
      <xsd:enumeration value="INCOME"/>
      <xsd:enumeration value="EXPENSE"/>
   </xsd:restriction>
</xsd:simpleType>
```

OperationInfo Complex Type

Provides information about a MONETA.RU transaction. The information includes a transaction ID and transaction attributes in the form of key-value pairs.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="OperationInfo">
   <xsd:sequence>
      <xsd:element minOccurs="0" name="id" type="xsd:long"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="attribute"
                   type="tns:KeyValueAttribute"/>
   </xsd:sequence>
</xsd:complexType>
```

Description

id

Transaction ID.

Optional element.

Type: long

attribute

Transaction attributes. Valid attributes are:

- **clienttransaction**. Merchant transaction ID.
- **statusid**. Transaction status.
- **typeid**. Transaction type ID. Valid transaction type IDs are:
  - **2,10**. Payment from a MONETA.RU account.
  - **3,19**. Payment from an external payment system.
  - **7,14**. Deposit.
  - **4,13**. Withdrawal to an external payment system.
  - **11**. Withdrawal to another MONETA.RU account.
  - **12**. Transfer to a MONETA.RU account that belongs to you or to another user.
  - **17,18**. Refund to an external payment system.
- **category**. Transaction category. Valid values are:
  - DEPOSIT
  - WITHDRAWAL
  - TRANSFER
  - BUSINESS
- **modified**. Transaction modification date and time.
- **sourceaccountid**. Account number.
- **sourcecurrencycode**. Account currency.
- **sourceamount**. Transaction amount.
- **sourceamountfee**. Transaction fee.
- **sourceamounttotal**. Total transaction amount including the transaction fee.
- **sourceaccounttotal**. Account balance changes.
- **targetaccountid**. Correspondent account number.

- **targetalias**. Alias of the correspondent account.
- **isreversed**. Indicates whether sourceaccountid and targetaccountid are associated with the payer and payee or vice versa.
  - **true**. sourceaccountid=payee and targetaccountid=payer.
  - **false**. sourceaccountid=payer and targetaccountid=payee.
- **customfield: *custom_attribute_name***. Custom attribute. Transaction information might include multiple custom attributes.

  > **Note:** The *custom_attribute_name* part of the custom attribute key might include up to 32 characters.
- **subscriberid**. A unique identifier of a customer in the merchant system.
- **purpose: *purpose_tag***. Transaction purpose. Transaction information might include multiple purpose attributes.

  > **Note:** The *purpose_tag* part of the purpose attribute key might include up to 32 characters. Example "purpose:cashback".

  Available tag values:

  - **cashback**. Specifies transaction as cashback. Available value is **1**.
- **receiptstatus**. Receipt status. Valid values are:
  - NEW
  - INPROGRESS
  - PROCESSED
  - FAILED
  - DRAFT
  - UNKNOWN

  To get receipt status, operation must have property "receiptid" and set showReceiptInfo=true in requests:

  - FindOperationsListRequest - element filter.showReceiptInfo
  - FindOperationsListByCTIDRequest - element showReceiptInfo
  - FindLastOperationsListRequest - element showReceiptInfo

Optional element.

Type: *KeyValueAttribute Complex Type* on page 76

OperationInfoBatchResponseType Complex Type

Transaction attributes for responses in batch mode.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                 name="OperationInfoBatchResponseType">
   <xsd:choice>
      <xsd:element name="transaction" type="tns:OperationInfo"/>
      <xsd:sequence>
         <xsd:element name="error" type="xsd:string"/>
         <xsd:element minOccurs="0" name="errorCode" type="xsd:string"/>
         <xsd:element minOccurs="0" name="transactionId" type="xsd:long"/>
      </xsd:sequence>
   </xsd:choice>
</xsd:complexType>
```

Description

error

If an error occurs when processing a transaction, the error element contains an error description.

If the transaction completes successfully, the error element is empty and the transaction element contains transaction details.

Type: string

errorCode

If an error occurs when processing a transaction, the errorCode element contains an error code.

MONETA.RU returns this element only if you set the version attribute of your request to **VERSION_2**.

See the faultDetail element for the error code descriptions.

Optional element.

Type: string

transactionId

If an error occurs when processing a transaction, the transactionId element contains an transaction ID.

MONETA.RU returns this element only if you set the version attribute of your request to **VERSION_5**.

Optional element.

Type: long

OperationInfoList Complex Type

A list of transactions. Long transaction lists are split into multiple pages.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="OperationInfoList">
   <xsd:sequence>
      <xsd:element name="pageSize" type="xsd:long"/>
      <xsd:element name="pageNumber" type="xsd:long"/>
      <xsd:element name="pagesCount" type="xsd:long"/>
      <xsd:element name="size" type="xsd:long"/>
      <xsd:element name="totalSize" type="xsd:long"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="operation"
 type="tns:OperationInfo"/>
   </xsd:sequence>
</xsd:complexType>
```

Description

pageSize

The maximum number of transactions that the response must return on a single page.

Type: long

pageNumber

The current page number. Page numbering starts from 1.

Type: long

pagesCount

The total number of pages for a given request.

Type: long

size

The number of transactions on the current page. This number is less than or equal to the pageSize number.

Type: long

totalSize

The total number of transactions for a given request.

Type: long

operation

The list of transactions.

Optional element.

Type: *OperationInfo Complex Type* on page 78

OperationStatus Simple Type

The status of a transaction in MONETA.RU.

Type: string

Valid values:

- **INPROGRESS**.MONETA.RU is processing the transaction.
- **SUCCEED**.Transaction processing completed successfully.
- **CANCELED**.Transaction has been canceled.
- **TAKENIN_NOTSENT**.The funds are credited to the payee's account. MONETA.RU did not send the processed payment report or you did not return a plain text response to the report.
- **CREATED**.Transaction is registered in MONETA.RU.
- **FROZEN**.Transaction is frozen.
- **TAKENOUT**.Debited.

XML Schema

```
<xsd:simpleType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="OperationStatus">
   <xsd:restriction base="xsd:string">
      <xsd:enumeration value="INPROGRESS"/>
      <xsd:enumeration value="SUCCEED"/>
      <xsd:enumeration value="CANCELED"/>
      <xsd:enumeration value="TAKENIN_NOTSENT"/>
      <xsd:enumeration value="CREATED"/>
      <xsd:enumeration value="FROZEN"/>
      <xsd:enumeration value="TAKENOUT"/>
   </xsd:restriction>
</xsd:simpleType>
```

OperationStatusState Simple Type

Type: string

Valid values:

- **DEBITED**.The funds were debited from the payer's account.
- **CREDITED**.The funds were credited to the payee's account
- **COMPLETED**.The transaction is completed.
- **FROZEN**.The transaction is frozen.
- **CANCELED**.The transaction is cancelled.

XML Schema

```
<xsd:simpleType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="OperationStatusState">
   <xsd:restriction base="xsd:string">
      <xsd:enumeration value="DEBITED"/>
      <xsd:enumeration value="CREDITED"/>
      <xsd:enumeration value="COMPLETED"/>
      <xsd:enumeration value="FROZEN"/>
      <xsd:enumeration value="CANCELED"/>
   </xsd:restriction>
</xsd:simpleType>
```

OperationTemplate Complex Type

Operation template.

Type: *AbstractAttributeObject Complex Type* on page 50.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="OperationTemplate">
   <xsd:complexContent>
      <xsd:extension base="tns:AbstractAttributeObject">
         <xsd:sequence>
            <xsd:element minOccurs="0" name="id" type="xsd:long"/>
            <xsd:element minOccurs="0" name="unitId" type="xsd:long"/>
            <xsd:element minOccurs="0" name="type" type="tns:OperationTemplateType"/>
            <xsd:element minOccurs="0" name="name" type="xsd:string"/>
            <xsd:element minOccurs="0" name="payer" type="xsd:long"/>
            <xsd:element minOccurs="0" name="payee" type="xsd:long"/>
            <xsd:element minOccurs="0" name="operationTypeCategory"
 type="tns:OperationTypeCategory"/>
            <xsd:element minOccurs="0" name="description" type="tns:Description"/>
            <xsd:element minOccurs="0" name="prototypeOperationId" type="xsd:long"/>
            <xsd:element minOccurs="0" name="tags" type="xsd:string"/>
            <xsd:element minOccurs="0" name="favorite" type="xsd:boolean"/>
            <xsd:choice minOccurs="0">
               <xsd:element name="commonParameters"
 type="tns:CommonOperationTemplateParameters"/>
               <xsd:element name="regularParameters"
 type="tns:RegularOperationTemplateParameters"/>
               <xsd:element name="directDebitParameters"
 type="tns:DirectDebitOperationTemplateParameters"/>
            </xsd:choice>
            <xsd:element maxOccurs="unbounded" minOccurs="0" name="operationInfo"
                         type="tns:KeyValueApprovedAttribute"/>
            <xsd:element maxOccurs="unbounded" minOccurs="0" name="additionalInfo"
                         type="tns:KeyValueApprovedAttribute"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

Description

id

Optional element.

Type: long

unitId

The owner of the operation template. If you omit this element, the request uses the name of the user who sends the request.

Optional element.

Type: long

type

Operation template type.

Optional element.

Type: *OperationTemplateType Simple Type* on page 87

name

Operation template name.

Optional element.

Type: string

payer

Account number of the payer.

Optional element.

Type: long

payee

Account number of the payee.

Optional element.

Type: long

operationTypeCategory

Transaction category.

Optional element.

Type: *OperationTypeCategory Simple Type* on page 88

description

Operation template description.

Optional element.

Type: *Description Simple Type* on page 66

prototypeOperationId

Optional element.

Type: long

tags

Optional element.

Type: string

favorite

Optional element.

Type: boolean

commonParameters

Type: *CommonOperationTemplateParameters Complex Type* on page 62

regularParameters

Type: *RegularOperationTemplateParameters Complex Type* on page 99

directDebitParameters

Type: *DirectDebitOperationTemplateParameters Complex Type* on page 66

operationInfo

Key-value pairs that will be saved as transaction attributes.

Optional element.

Type: *KeyValueApprovedAttribute Complex Type* on page 76

additionalInfo

Optional element.

Type: *KeyValueApprovedAttribute Complex Type* on page 76

OperationTemplateAmount Complex Type

Type: *AbstractAttributeObject Complex Type* on page 50.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="OperationTemplateAmount">
   <xsd:complexContent>
      <xsd:extension base="tns:AbstractAttributeObject">
         <xsd:sequence>
            <xsd:element minOccurs="0" name="amount" type="tns:Money"/>
            <xsd:element minOccurs="0" name="isPayerAmount" type="xsd:boolean"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

Description

amount

Transaction amount.

Optional element.

Type: *Money Simple Type* on page 77

isPayerAmount

- **true**. MONETA.RU uses the currency of the payer's account for the transaction amount.
- **false**. MONETA.RU uses the currency of the payee's account for the transaction amount.

Optional element.

Type: boolean

OperationTemplateAmountInfo Complex Type

Type: *AbstractAttributeObject Complex Type* on page 50.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="OperationTemplateAmountInfo">
   <xsd:complexContent>
      <xsd:extension base="tns:AbstractAttributeObject">
         <xsd:sequence>
            <xsd:element name="type" type="tns:OperationTemplateAmountInfoType"/>
            <xsd:choice minOccurs="0">
               <xsd:element name="amount" type="tns:OperationTemplateAmount"/>
               <xsd:element name="range" type="tns:OperationTemplateAmountRange"/>
               <xsd:element name="rest" type="tns:OperationTemplateAmountRest"/>
            </xsd:choice>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

Description

type

Type: *OperationTemplateAmountInfoType Simple Type* on page 85

amount

Type: *OperationTemplateAmount Complex Type* on page 83

range

Type: *OperationTemplateAmountRange Complex Type* on page 85

rest

Type: *OperationTemplateAmountRest Complex Type* on page 86

OperationTemplateAmountInfoType Simple Type

Type: string

Valid values:

- **AMOUNT**.
- **BALANCE**.

  The amount to be debited will equal all available funds on your account.
- **PAYMENTS**.

  The amount to be debited will equal the merchant balance, depending on the payment period.
- **RANGE**.

  The amount to be debited will equal all available funds on your account if this amount is in the set range as of the moment of transaction.
- **REST**.

  The amount to be debited will equal all available funds on your account less the specified balance.
- **CREDIT**.

  The amount to be debited will equal the inflow amount, depending on the payment period.
- **OPENING**.

  The amount to be debited will equal the opening balance on your account.

XML Schema

```
<xsd:simpleType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                name="OperationTemplateAmountInfoType">
   <xsd:restriction base="xsd:string">
      <xsd:enumeration value="AMOUNT"/>
      <xsd:enumeration value="BALANCE"/>
      <xsd:enumeration value="PAYMENTS"/>
      <xsd:enumeration value="RANGE"/>
      <xsd:enumeration value="REST"/>
      <xsd:enumeration value="CREDIT"/>
      <xsd:enumeration value="OPENING"/>
   </xsd:restriction>
</xsd:simpleType>
```

OperationTemplateAmountRange Complex Type

Type: *AbstractAttributeObject Complex Type* on page 50.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="OperationTemplateAmountRange">
   <xsd:complexContent>
      <xsd:extension base="tns:AbstractAttributeObject">
         <xsd:sequence>
            <xsd:element minOccurs="0" name="amountMinValue" type="tns:Money"/>
            <xsd:element minOccurs="0" name="amountMaxValue" type="tns:Money"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

Description

amountMinValue

Optional element.

Type: *Money Simple Type* on page 77

amountMaxValue

Optional element.

Type: *Money Simple Type* on page 77

OperationTemplateAmountRest Complex Type

Type: *AbstractAttributeObject Complex Type* on page 50.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="OperationTemplateAmountRest">
   <xsd:complexContent>
      <xsd:extension base="tns:AbstractAttributeObject">
         <xsd:sequence>
            <xsd:element name="amount" type="tns:Money"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

Description

amount

Type: *Money Simple Type* on page 77

OperationTemplateReminderInfo Complex Type

Type: *AbstractAttributeObject Complex Type* on page 50.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="OperationTemplateReminderInfo">
   <xsd:complexContent>
      <xsd:extension base="tns:AbstractAttributeObject">
         <xsd:sequence>
            <xsd:element name="remind" type="xsd:boolean"/>
            <xsd:element minOccurs="0" name="hoursBeforeExecution" type="xsd:int"/>
            <xsd:element maxOccurs="10" minOccurs="0" name="notification"
                         type="tns:ProfileNotificationSelection"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

Description

remind

Type: boolean

hoursBeforeExecution

Optional element.

Type: int

notification

Optional element.

Type: *ProfileNotificationSelection Complex Type* on page 97

OperationTemplateTimeInfo Complex Type

Type: *AbstractAttributeObject Complex Type* on page 50.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="OperationTemplateTimeInfo">
   <xsd:complexContent>
      <xsd:extension base="tns:AbstractAttributeObject">
         <xsd:sequence>
            <xsd:element name="type" type="tns:OperationTemplateTimeInfoType"/>
            <xsd:element name="startDateTime" type="xsd:dateTime"/>
            <xsd:element minOccurs="0" name="endDateTime" type="xsd:dateTime"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

Description

type

Type: *OperationTemplateTimeInfoType Simple Type* on page 87

startDateTime

Type: dateTime

endDateTime

Optional element.

Type: dateTime

OperationTemplateTimeInfoType Simple Type

Type: string

Valid values:

- **ONCE**.
- **EVERY_DAY**.
- **EVERY_WORKDAY**.
- **EVERY_WEEK**.
- **EVERY_MONTH**.
- **EVERY_LAST_DAY_OF_MONTH**.
- **EVERY_YEAR**.

XML Schema

```
<xsd:simpleType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="OperationTemplateTimeInfoType">
   <xsd:restriction base="xsd:string">
      <xsd:enumeration value="ONCE"/>
      <xsd:enumeration value="EVERY_DAY"/>
      <xsd:enumeration value="EVERY_WORKDAY"/>
      <xsd:enumeration value="EVERY_WEEK"/>
      <xsd:enumeration value="EVERY_MONTH"/>
      <xsd:enumeration value="EVERY_LAST_DAY_OF_MONTH"/>
      <xsd:enumeration value="EVERY_YEAR"/>
   </xsd:restriction>
</xsd:simpleType>
```

OperationTemplateType Simple Type

Operation template type.

Type: string

Valid values:

- **COMMON**.

- **REGULAR**.
- **DIRECT_DEBIT**.

XML Schema

```
<xsd:simpleType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="OperationTemplateType">
   <xsd:restriction base="xsd:string">
      <xsd:enumeration value="COMMON"/>
      <xsd:enumeration value="REGULAR"/>
      <xsd:enumeration value="DIRECT_DEBIT"/>
   </xsd:restriction>
</xsd:simpleType>
```

OperationTypeCategory Simple Type

Transaction category.

Type: string

Valid values:

- **DEPOSIT**.Operations that deposit funds to your MONETA.RU account.
- **WITHDRAWAL**.Operations that withdraw funds from your MONETA.RU account to an external payment system.
- **TRANSFER**.Operations that transfer funds from your MONETA.RU account to another MONETA.RU account that belongs to you or to a different user. This operation type also includes currency exchange operations.
- **BUSINESS**.Merchant operations that include payments to a store, refunds, all MONETA.Assistant operations.

XML Schema

```
<xsd:simpleType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="OperationTypeCategory">
   <xsd:restriction base="xsd:string">
      <xsd:enumeration value="DEPOSIT"/>
      <xsd:enumeration value="WITHDRAWAL"/>
      <xsd:enumeration value="TRANSFER"/>
      <xsd:enumeration value="BUSINESS"/>
   </xsd:restriction>
</xsd:simpleType>
```

Pager Complex Type

Specifies paginating parameters for a long list of MONETA.RU objects, such as transactions or users.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="Pager">
   <xsd:sequence>
      <xsd:element default="1" name="pageNumber">
         <xsd:simpleType>
            <xsd:restriction base="xsd:int">
               <xsd:minInclusive value="1"/>
            </xsd:restriction>
         </xsd:simpleType>
      </xsd:element>
      <xsd:element default="25" name="pageSize">
         <xsd:simpleType>
            <xsd:restriction base="xsd:int">
               <xsd:minInclusive value="1"/>
               <xsd:maxInclusive value="10000"/>
            </xsd:restriction>
         </xsd:simpleType>
```

```
      </xsd:element>
   </xsd:sequence>
</xsd:complexType>
```

Description

pageNumber

Specifies the number of the page that you want to get.

Minimum value: 1.

Default value: 1.

Type: int

Default value: 1

pageSize

Specifies the maximum number of transactions per page.

Valid range: 1-10000.

Default value: 25.

Type: int

Default value: 25

Password Simple Type

A payment password that is used in complex types.

Type: normalizedString

Maximum length: 255

XML Schema

```
<xsd:simpleType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="Password">
   <xsd:restriction base="xsd:normalizedString">
      <xsd:maxLength value="255"/>
   </xsd:restriction>
</xsd:simpleType>
```

PaymentBatchRequestType Complex Type

Transaction parameters for requests in batch mode.

Type: *EntityBatchRequestType Complex Type* on page 69.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="PaymentBatchRequestType">
   <xsd:complexContent>
      <xsd:extension base="tns:EntityBatchRequestType">
         <xsd:sequence>
            <xsd:element maxOccurs="unbounded" minOccurs="1" name="transaction"
                         type="tns:PaymentRequestType"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

Description

transaction

The set of transactions that must be processed in a single batch. MONETA.RU processes transactions in the order that they are defined in the request.

Required element.

Type: *PaymentRequestType Complex Type* on page 90

PaymentPassword Complex Type

Payment password for the payer's account.

Type: *AbstractAttributeObject Complex Type* on page 50.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="PaymentPassword">
   <xsd:complexContent>
      <xsd:extension base="tns:AbstractAttributeObject">
         <xsd:sequence>
            <xsd:element name="paymentPassword" type="tns:Password"/>
            <xsd:element minOccurs="0" name="paymentPasswordChallenge"
 type="xsd:string"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

Description

paymentPassword

Payment password.

Type: *Password Simple Type* on page 89

paymentPasswordChallenge

Challenge passcode that you received in the GetAccountPaymentPasswordChallenge response in the paymentPasswordChallenge element. Specify this element in the following cases:

- If a user gets payment passwords by SMS, set paymentPasswordChallenge to SMS. Set paymentPassword to the value that the user receives in the SMS from MONETA.RU.
- If a user gets a sequence number (index) for a password from a list of transaction authentication numbers (TANs), set paymentPasswordChallenge to the TAN index. Set paymentPassword to the TAN that has the specified index.

Optional element.

Type: string

PaymentRequestType Complex Type

Transaction parameters for requests. Unlike the TransactionRequestType element, the payee element in PaymentRequestType can specify not only the account number, but also a transaction ID, e-mail, or phone number.

Type: *Entity Complex Type* on page 69.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="PaymentRequestType">
   <xsd:complexContent>
      <xsd:extension base="tns:Entity">
         <xsd:sequence>
            <xsd:element name="payer" type="xsd:string"/>
            <xsd:element name="payee" type="xsd:string"/>
            <xsd:element minOccurs="0" name="amount" type="tns:Money"/>
            <xsd:element minOccurs="0" name="isPayerAmount" type="xsd:boolean"/>
```

```
            <xsd:element minOccurs="0" name="paymentPassword" type="tns:Password"/>
            <xsd:element minOccurs="0" name="clientTransaction" type="tns:CTID"/>
            <xsd:element minOccurs="0" name="description" type="tns:Description"/>
            <xsd:element minOccurs="0" name="operationInfo" type="tns:OperationInfo"/>
            <xsd:element minOccurs="0" name="paymentPasswordChallenge"
 type="xsd:string"/>
        </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

Description

payer

The account number of the payer.

Type: string

payee

The unique identifier of the payee. This element can take one of the following values:

- E-mail address of the payee. The funds are transferred to the primary MONETA.RU account that is registered to the specified e-mail address.
- Transaction ID that is prefixed with a zero. For example, to indicate that you specify a transaction ID 12345678, use 012345678.
- Phone number of that payee that is prefixed with the plus sign (+). The funds are transferred to the primary MONETA.RU account that is registered to the specified phone number.
- Account number of the payee. This number must contain only digits.

Type: string

amount

Transaction amount. The currency of the transaction is specified by the isPayerAmount element. If a request does not include the isPayerAmount element, MONETA.RU uses the following rules to determine the currency of the transaction:

- If a user has access only to the payer's account, MONETA.RU uses the currency of the payer's account for the transaction.
- If a user has access only to the payee's account, MONETA.RU uses the currency of the payee's account for the transaction.

Optional element.

Type: *Money Simple Type* on page 77

isPayerAmount

This element is required if a user has access to the payer's and to the payee's accounts. Valid values:

- **true**. MONETA.RU uses the currency of the payer's account for the transaction amount.
- **false**. MONETA.RU uses the currency of the payee's account for the transaction amount.

Optional element.

Type: boolean

paymentPassword

Payment password for the payer's MONETA.RU account.

Optional element.

Type: *Password Simple Type* on page 89

clientTransaction

Merchant transaction ID.

Optional element.

Type: *CTID Simple Type* on page 60

description

Transaction description or comments.

Optional element.

Type: *Description Simple Type* on page 66

operationInfo

Key-value pairs that will be saved as transaction attributes. For dates, use the following format: dd.MM.yyyy HH:mm:ss

Optional element.

Type: *OperationInfo Complex Type* on page 78

paymentPasswordChallenge

Challenge passcode that you received in the GetAccountPaymentPasswordChallenge response in the paymentPasswordChallenge element. Specify this element in the following cases:

- If a user gets payment passwords by SMS, set paymentPasswordChallenge to SMS. Set paymentPassword to the value that the user receives in the SMS from MONETA.RU.
- If a user gets a sequence number (index) for a password from a list of transaction authentication numbers (TANs), set paymentPasswordChallenge to the TAN index. Set paymentPassword to the TAN that has the specified index.

Optional element.

Type: string

PaymentSystemInfoComplexType Complex Type

Type of present information of payment system

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="PaymentSystemInfoComplexType">
   <xsd:sequence>
      <xsd:element minOccurs="0" name="unitId" type="xsd:long"/>
      <xsd:element minOccurs="0" name="name" type="xsd:string"/>
      <xsd:element minOccurs="0" name="icon" type="xsd:string"/>
      <xsd:element minOccurs="0" name="logicalGroup" type="xsd:string"/>
      <xsd:element minOccurs="0" name="infoUrl" type="tns:InfoUrl"/>
      <xsd:element minOccurs="0" name="referenceData" type="tns:ReferenceData"/>
      <xsd:element minOccurs="0" name="infoTariff" type="tns:InfoTariff"/>
      <xsd:element minOccurs="0" name="currencies" type="xsd:string"/>
      <xsd:element minOccurs="0" name="psAccountIds" type="xsd:string"/>
   </xsd:sequence>
</xsd:complexType>
```

Description

unitId

Unique identifier of the user in MONETA.RU.

Optional element.

Type: long

name

The name of payment system.

Optional element.

Type: string

icon

The URL of icon of the payment system.

Optional element.

Type: string

logicalGroup

The name of logical group of the payment system.

Optional element.

Type: string

infoUrl

Fully information about URL of the payment system.

Optional element.

Type: *InfoUrl Complex Type* on page 73

referenceData

Reference information.

Optional element.

Type: *ReferenceData Complex Type* on page 98

infoTariff

Short information on tariff.

Optional element.

Type: *InfoTariff Complex Type* on page 72

currencies

List of currency payment system accounts. Are located in the same order, as well as accounts in psAccountIds are.

Optional element.

Type: string

psAccountIds

List of payment system accounts. Are located in the same order, as well as accounts currencies in currencies are.

Optional element.

Type: string

PersonalInformation Complex Type

Personal information.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="PersonalInformation">
   <xsd:sequence>
      <xsd:element name="profile" type="tns:Profile"/>
      <xsd:element minOccurs="0" name="document" type="tns:Document"/>
   </xsd:sequence>
</xsd:complexType>
```

Profile Complex Type

User profile information.

Information about a user profile contains a list of key-value pairs.

MONETA.RU returns only information about a user profile that you own or publicly available information.

Valid keys for client profiles are:

- **unitid**. The unique identifier of a MONETA.RU user.
- **last_name**. The last name of the user.
- **first_name**. The first name of the user.
- **middle_initial_name**. The middle name of the user.
- **alias**. User alias.
- **country**. Country of residence.
- **state**. State.
- **city**. City.
- **zip**. ZIP code.
- **address**. Address of the user.
- **email_for_notifications**. Email address of the user.
- **phone**. Phone number of the user.
- **cell_phone**. Mobile phone number.
- **url**. URL of the user's web site.
- **sex**. Sex of the user. Valid values are:
  - MALE
  - FEMALE
- **date_of_birth**. Date of birth of the user in the following format: `yyyy-mm-dd`.
- **inn**. Tax identification number (TIN).
- **snils**. Personal insurance policy number (SNILS).
- **timezone**. The time zone of the user.

  Note: For the list of time zone names (TZ names) that MONETA.ru supports, see *wikipedia.org*.
- **ui_language**. Valid values are:
  - EN
  - RU
- **customfield:*field_name***. Custom attribute. The *field_name* part of the custom key might include up to 32 characters. User profiles may contain multiple custom attributes.

Mandatory keys for client profile creation are **last_name** and **first_name** or **alias**.

Valid keys for organization profiles:

- **unitid**. The unique identifier of an organization in MONETA.RU.
- **rf_resident**. Partner type [resident/not resident].
- **goal_and_business_relationships**. Purpose and suggested character of establishing business relations with NCO.
- **planned_turnovers**. Planned monthly cash flow.
- **planned_payers**. planned payers.
- **tariff**. Tariff, used in partner registration, only for oferta partners [PAY_MYSELF, ZKH].
- **incorporation_form_ru**. Entity kind [organization or private entrepreneur], explicitly specified when registering a non-resident partner, for a resident, it is established by INN.
- **international_name**. Merchant name in the native language.
- **alias**. Partner alias.
- **organization_name**. Merchant full name or private entrepreneur full name.
- **organization_name_short**. Merchant contracted name.
- **position_director**. Managing director title (for organization) [DIRECTOR, GENERAL_DIRECTOR, EXECUTIVE_DIRECTOR, OTHER].
- **position_director_details**. Details for managing director title (for organization), only for OTHER.
- **acting_document**. Document on basis of which Managing director acts (for organization) [POWER_OF_ATTORNEY, ARTICLES_OF_ASSOCIATION, OTHER].
- **acting_document_details**. Details for document on basis of which Managing director acts (for organization), only for OTHER.
- **attorney_number_ru**. Number (attorney, for organization), only for POWER_OF_ATTORNEY.
- **attorney_date_from_ru**. Date (from) (attorney, for organization), only for POWER_OF_ATTORNEY.

- **attorney_date_to_ru**. Date (validity period - till) (attorney, for organization), only for POWER_OF_ATTORNEY.
- **agreement_signer_fio**. Signatory's full name (for organization).
- **registration_date_ru**. Date (certificate).
- **registration_series_ru**. Series (certificate).
- **registration_number_ru**. Number (certificate).
- **registration_authority_ru**. Registering authority name (certificate).
- **registration_state_place_ru**. State registration place (certificate).
- **agreement_signer_fio_ie**. Signator's full name (for private entrepreneur).
- **fio_accountant**. Financial support contact full name.
- **position_accountant**. Financial support contact title.
- **fio_contact**. Technical support contact full name.
- **position_contact**. Technical support contact title.
- **joint_governing_body**. Joint governing body (executive body).
- **joint_body_members**. Full name and title of collective body members, only if there is a value of joint_governing_body.
- **business_activity**. Business activity for "Other" category.
- **url**. Website (URL), used in partner registration.
- **where_from_do_you_know**. How you learnt about us.
- **where_from_do_you_know_details**. Details for how you learnt about us.
- **promocode**. Promo code.
- **inn**. Tax identification number (TIN).
- **fio_director**. Managing director full name.
- **phone_director**. Managing director office phone.
- **country**Country. Use isocode [RUS], isocode2L [RU] or name[Russia]
- **legal_address**. Legal address.
- **management_address**. Address of the permanent management body.
- **post_address**. Postal address.
- **phone_contact**. Contact phone number.
- **phone_accountant**. Financial support phone number.
- **phone_support**. Technical support phone number.
- **fax**. Fax.
- **cell_phone**. Cell phone.
- **contact_email**. Contact E-mail address, used in partner registration.
- **finance_email**. Financial support E-mail address.
- **technical_email**. Technical support E-mail address.
- **contact_info**. Contact info.
- **timezone**. Time zone of the organization.

  | **Note:** For the list of time zone names (TZ names) that MONETA.ru supports, see *wikipedia.org*.
- **ui_language**. User interface langugage. Valid values are:
  - EN
  - RU
- **email_for_notifications**. Email address of the organization that is used for notifications.
- **capital_type**. Capital type.
- **registered_capital_size**. Register capital ammount.
- **paid_capital_size**. Paid-in capital ammount.
- **budget_arrears_absence**. Information on taxes and dues tax liabilities.
- **founder_share_percentage**. Founder share in percentage. Value from 0 to 100. If percentage is more then 0, then numerator and denominator of fraction should be equal 0.
- **founder_share_numerator**. Fraction numerator for founder share. Value from 0. If numerator is more then 0, then denominator of fraction should be more then 0 and percentage should be equal 0.
- **founder_share_denominator**. Fraction denominator for founder share. Value from 0. If denominator is more then 0, then numerator of fraction should be more then 0 and percentage should be equal 0.

- **conditions_payee**. State for "There is information on beneficiary of payment available for unauthorized payers on the website".
- **conditions_payer**. State for "There's contact information available for unauthorized payers on the website".
- **conditions_site**. State for "Website is fully functional and filled in with content".
- **conditions_payment_info**. State for "Payment order and process description, as well as information on temporary interval between payment and dispatch (delivery) of wares or provision of service are available for unauthorized users.".
- **conditions_correct_data**. State for "All essential data are inserted and are supposed to be up-to-date".
- **agreement_sent_date**. Date of sending the document.
- **agreement_sent_method**. Method of dispatch the document.
- **customfield:*field_name***. Custom attribute. The *field_name* part of the custom key might include up to 32 characters. Organization profiles may contain multiple custom attributes.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="Profile">
   <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="attribute"
                   type="tns:KeyValueApprovedAttribute"/>
   </xsd:sequence>
</xsd:complexType>
```

ProfileNotification Complex Type

Type: *AbstractAttributeObject Complex Type* on page 50.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="ProfileNotification">
   <xsd:complexContent>
      <xsd:extension base="tns:AbstractAttributeObject">
         <xsd:sequence>
            <xsd:element minOccurs="0" name="id" type="xsd:long"/>
            <xsd:element minOccurs="0" name="unitId" type="xsd:long"/>
            <xsd:element minOccurs="0" name="type" type="tns:ProfileNotificationType"/>
            <xsd:element maxOccurs="10" minOccurs="0" name="recipient"
 type="xsd:string"/>
            <xsd:element maxOccurs="unbounded" minOccurs="0" name="flag"
                         type="tns:ProfileNotificationFlag"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

Description

id

Optional element.

Type: long

unitId

The owner of the notifications.

Optional element.

Type: long

type

Optional element.

Type: *ProfileNotificationType Simple Type* on page 98

recipient

Optional element.

Type: string

flag

Optional element.

Type: *ProfileNotificationFlag Complex Type* on page 97

ProfileNotificationFlag Complex Type

Type: *AbstractAttributeObject Complex Type* on page 50.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="ProfileNotificationFlag">
   <xsd:complexContent>
      <xsd:extension base="tns:AbstractAttributeObject">
         <xsd:sequence>
            <xsd:element name="flag" type="tns:ProfileNotificationFlagType"/>
            <xsd:element name="enabled" type="xsd:boolean"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

Description

flag

Type: *ProfileNotificationFlagType Simple Type* on page 97

enabled

Type: boolean

ProfileNotificationFlagType Simple Type

Type: string

XML Schema

```
<xsd:simpleType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="ProfileNotificationFlagType">
   <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
```

ProfileNotificationSelection Complex Type

Type: *ProfileNotification Complex Type* on page 96.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="ProfileNotificationSelection">
   <xsd:complexContent>
      <xsd:extension base="tns:ProfileNotification">
         <xsd:sequence>
            <xsd:element name="selected" type="xsd:boolean"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

Description

selected

Type: boolean

ProfileNotificationType Simple Type

Type: string

Valid values:

- **EMAIL**

XML Schema

```
<xsd:simpleType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="ProfileNotificationType">
   <xsd:restriction base="xsd:string">
      <xsd:enumeration value="EMAIL"/>
   </xsd:restriction>
</xsd:simpleType>
```

ProfileType Simple Type

Specifies user profile information.

Type: string

Valid values:

- **organization**. Organization profile. Profiles of this type contain such organization-specific elements as the name of the director and accountant, legal address, etc.
- **client**. Client profile. Profiles of this type contain such user-specific elements as first name, last name, etc.

XML Schema

```
<xsd:simpleType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="ProfileType">
   <xsd:restriction base="xsd:string">
      <xsd:enumeration value="organization"/>
      <xsd:enumeration value="client"/>
   </xsd:restriction>
</xsd:simpleType>
```

ReferenceData Complex Type

Reference information.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="ReferenceData">
   <xsd:sequence>
      <xsd:element minOccurs="0" name="period" type="xsd:string"/>
      <xsd:element minOccurs="0" name="percentage" type="xsd:string"/>
      <xsd:element minOccurs="0" name="percentageExt" type="xsd:string"/>
      <xsd:element minOccurs="0" name="sourceFee" type="tns:Fee"/>
      <xsd:element minOccurs="0" name="sourceFeeExt" type="tns:Fee"/>
      <xsd:element minOccurs="0" name="targetFee" type="tns:Fee"/>
      <xsd:element minOccurs="0" name="targetFeeExt" type="tns:Fee"/>
   </xsd:sequence>
</xsd:complexType>
```

Description

period

The term of carrying out banking transactions.

Optional element.

Type: string

percentage

The commission, in percent.

Optional element.

Type: string

percentageExt

The external commission, in percent.

Optional element.

Type: string

sourceFee

The commission payeer (amount).

Optional element.

Type: *Fee Simple Type* on page 69

sourceFeeExt

External commission from the payer (amount).

Optional element.

Type: *Fee Simple Type* on page 69

targetFee

Commission to the payee (amount).

Optional element.

Type: *Fee Simple Type* on page 69

targetFeeExt

External commission to the payee (amount).

Optional element.

Type: *Fee Simple Type* on page 69

RegularOperationTemplateParameters Complex Type

Type: *AbstractAttributeObject Complex Type* on page 50.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                 name="RegularOperationTemplateParameters">
   <xsd:complexContent>
      <xsd:extension base="tns:AbstractAttributeObject">
         <xsd:sequence>
            <xsd:element minOccurs="0" name="amountInfo"
 type="tns:OperationTemplateAmountInfo"/>
            <xsd:element minOccurs="0" name="timeInfo"
 type="tns:OperationTemplateTimeInfo"/>
            <xsd:element minOccurs="0" name="reminderInfo"
 type="tns:OperationTemplateReminderInfo"/>
            <xsd:element maxOccurs="10" minOccurs="0"
 name="operationsReportNotifications"
                         type="tns:ProfileNotificationSelection"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

Description

amountInfo

Optional element.

Type: *OperationTemplateAmountInfo Complex Type* on page 84

timeInfo

Optional element.

Type: *OperationTemplateTimeInfo Complex Type* on page 86

reminderInfo

Optional element.

Type: *OperationTemplateReminderInfo Complex Type* on page 86

operationsReportNotifications

Optional element.

Type: *ProfileNotificationSelection Complex Type* on page 97

Report Complex Type

Report.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="Report">
   <xsd:sequence>
      <xsd:element minOccurs="0" name="id" type="xsd:long"/>
      <xsd:element minOccurs="0" name="name" type="xsd:string"/>
      <xsd:element name="unitId" type="xsd:long"/>
      <xsd:element name="typeId" type="xsd:long"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="reportInstance"
                   type="tns:ReportInstance"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="attribute"
                   type="tns:KeyValueApprovedAttribute"/>
   </xsd:sequence>
</xsd:complexType>
```

Description

id

Report ID.

Optional element.

Type: long

name

Report name.

Optional element.

Type: string

unitId

User ID.

Type: long

typeId

Report type ID.

Type: long

reportInstance

Report instances list.

Optional element.

Type: *ReportInstance Complex Type* on page 101

attribute

Optional element.

Type: *KeyValueApprovedAttribute Complex Type* on page 76

ReportInstance Complex Type

Report instance.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="ReportInstance">
   <xsd:sequence>
      <xsd:element minOccurs="0" name="id" type="xsd:long"/>
      <xsd:element name="reportId" type="xsd:long"/>
      <xsd:element name="year" type="xsd:int"/>
      <xsd:element name="month" type="xsd:int"/>
      <xsd:element minOccurs="0" name="url" type="xsd:string"/>
      <xsd:element minOccurs="0" name="urlExpirationDate" type="xsd:dateTime"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="attribute"
                   type="tns:KeyValueApprovedAttribute"/>
   </xsd:sequence>
</xsd:complexType>
```

Description

id

Report instance ID.

Optional element.

Type: long

reportId

Report ID.

Type: long

year

Report instance year.

Type: int

month

Report instance month (1-12).

Type: int

url

Download URL.

Optional element.

Type: string

urlExpirationDate

URL expiration date.

Optional element.

Type: dateTime

attribute

Optional element.

Type: *KeyValueApprovedAttribute Complex Type* on page 76

TransactionBatchRequestType Complex Type

Transaction parameters for requests in batch mode.

Type: *EntityBatchRequestType Complex Type* on page 69.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="TransactionBatchRequestType">
   <xsd:complexContent>
      <xsd:extension base="tns:EntityBatchRequestType">
         <xsd:sequence>
            <xsd:element maxOccurs="unbounded" minOccurs="1" name="transaction"
                         type="tns:TransactionRequestType"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

Description

transaction

The set of transactions that must be processed in a single batch. MONETA.RU processes transactions in the order that they are defined in the request.

Required element.

Type: *TransactionRequestType Complex Type* on page 103

TransactionBatchResponseType Complex Type

Transaction attributes for responses in batch processing mode.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="TransactionBatchResponseType">
   <xsd:sequence>
      <xsd:element minOccurs="0" name="error" type="xsd:string"/>
      <xsd:element minOccurs="0" name="transaction" type="tns:TransactionResponseType"/
>
      <xsd:element minOccurs="0" name="errorCode" type="xsd:string"/>
   </xsd:sequence>
</xsd:complexType>
```

Description

error

If an error occurs when processing a transaction, the error element contains an error description. If the transaction completes successfully, the error element is empty and the transaction element contains transaction details.

Optional element.

Type: string

transaction

Optional element.

Type: *TransactionResponseType Complex Type* on page 104

errorCode

If an error occurs when processing a transaction, the errorCode element contains an error code.

MONETA.RU returns this element only if you set the version attribute of your request to **VERSION_2** or higher.

See the faultDetail element for the error code descriptions.

Optional element.

Type: string

TransactionRequestType Complex Type

Transaction parameters for SOAP requests.

Type: *Entity Complex Type* on page 69.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="TransactionRequestType">
   <xsd:complexContent>
      <xsd:extension base="tns:Entity">
         <xsd:sequence>
            <xsd:element name="payer" type="xsd:string"/>
            <xsd:element name="payee" type="xsd:string"/>
            <xsd:element minOccurs="0" name="amount" type="tns:Money"/>
            <xsd:element minOccurs="0" name="isPayerAmount" type="xsd:boolean"/>
            <xsd:element minOccurs="0" name="paymentPassword" type="tns:Password"/>
            <xsd:element minOccurs="0" name="clientTransaction" type="tns:CTID"/>
            <xsd:element minOccurs="0" name="description" type="tns:Description"/>
            <xsd:element minOccurs="0" name="operationInfo" type="tns:OperationInfo"/>
            <xsd:element minOccurs="0" name="paymentPasswordChallenge"
 type="xsd:string"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

Description

payer

Account number of the payer.

Type: string

payee

Account number of the payee.

Type: string

amount

Transaction amount. The currency of the transaction is specified by the isPayerAmount element. If a request does not include the isPayerAmount element, MONETA.RU uses the following rules to determine the currency of the transaction:

- If a user has access only to the payer's account, MONETA.RU uses the currency of the payer's account for the transaction.
- If a user has access only to the payee's account, MONETA.RU uses the currency of the payee's account for the transaction.

Tip: Always use the isPayerAmount element to avoid ambiguity.

Optional element.

Type: *Money Simple Type* on page 77

isPayerAmount

This element is required if a user has access to the payer's and to the payee's accounts. Valid values:

- **true**. MONETA.RU uses the currency of the payer's account for the transaction amount.
- **false**. MONETA.RU uses the currency of the payee's account for the transaction amount.

Optional element.

Type: boolean

paymentPassword

Payment password for the payer's MONETA.RU account.

Optional element.

Type: *Password Simple Type* on page 89

clientTransaction

Merchant transaction ID.

Optional element.

Type: *CTID Simple Type* on page 60

description

Transaction description or comments.

Optional element.

Type: *Description Simple Type* on page 66

operationInfo

Key-value pairs that will be saved as transaction attributes. For dates, use the following format: dd.MM.yyyy HH:mm:ss

Optional element.

Type: *OperationInfo Complex Type* on page 78

paymentPasswordChallenge

Challenge passcode that you received in the GetAccountPaymentPasswordChallenge response in the paymentPasswordChallenge element. Specify this element in the following cases:

- If a user gets payment passwords by SMS, set paymentPasswordChallenge to SMS. Set paymentPassword to the value that the user receives in the SMS from MONETA.RU.
- If a user gets a sequence number (index) for a password from a list of transaction authentication numbers (TANs), set paymentPasswordChallenge to the TAN index. Set paymentPassword to the TAN that has the specified index.

Optional element.

Type: string

TransactionResponseType Complex Type

Transaction attributes in responses.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="TransactionResponseType">
   <xsd:sequence>
      <xsd:element name="status" type="xsd:string"/>
      <xsd:element name="dateTime" type="xsd:dateTime"/>
      <xsd:element name="transaction" type="xsd:long"/>
      <xsd:element minOccurs="0" name="clientTransaction" type="tns:CTID"/>
      <xsd:element minOccurs="0" name="operationInfo" type="tns:OperationInfo"/>
   </xsd:sequence>
</xsd:complexType>
```

Description

status

The current transaction status.

Type: string

dateTime

The date and time of the latest change in a transaction.

Type: dateTime

transaction

Unique identifier of a transaction in MONETA.RU.

Type: long

clientTransaction

Merchant transaction ID.

Optional element.

Type: *CTID Simple Type* on page 60

operationInfo

Additional transaction details. For retrieving operationInfo the request attribute "version" has to be set to "VERSION_2" or greater (only in InvoiceRequest).

Optional element.

Type: *OperationInfo Complex Type* on page 78

VerifyTransactionResponseType Complex Type

Information about the transaction verification.

Type: *VerifyTransferResponseType Complex Type* on page 106.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="VerifyTransactionResponseType">
   <xsd:complexContent>
      <xsd:extension base="tns:VerifyTransferResponseType">
         <xsd:sequence>
            <xsd:element minOccurs="0" name="transactionId" type="xsd:long"/>
            <xsd:element minOccurs="0" name="operationStatus"
 type="tns:OperationStatus"/>
         </xsd:sequence>
      </xsd:extension>
   </xsd:complexContent>
</xsd:complexType>
```

Description

transactionId

MONETA.RU transaction ID.

Optional element.

Type: long

operationStatus

Transaction status in MONETA.RU.

Optional element.

Type: *OperationStatus Simple Type* on page 81

VerifyTransferResponseType Complex Type

MONETA.RU verification status of a transaction.

XML Schema

```
<xsd:complexType xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="VerifyTransferResponseType">
   <xsd:sequence>
      <xsd:element name="isTransactionValid" type="xsd:boolean"/>
      <xsd:element minOccurs="0" name="description" type="xsd:string"/>
      <xsd:element minOccurs="0" name="forecast"
 type="tns:ForecastTransactionResponseType"/>
      <xsd:element minOccurs="0" name="errorCode" type="xsd:string"/>
      <xsd:element minOccurs="0" name="operationInfo" type="tns:OperationInfo"/>
   </xsd:sequence>
</xsd:complexType>
```

Description

isTransactionValid

If this element is set to **true**, MONETA.RU can process the specified transaction.

Type: boolean

description

Description of the current transaction status.

Optional element.

Type: string

forecast

If the transaction is valid, this element includes additional information about the transaction. If MONETA.RU cannot process the transaction, this element is empty.

Optional element.

Type: *ForecastTransactionResponseType Complex Type* on page 71

errorCode

If an error occurs when processing a transaction, the errorCode element contains an error code.

MONETA.RU returns this element only if you set the version attribute of your request to **VERSION_2** or higher.

See the faultDetail element for the error code descriptions.

Optional element.

Type: string

operationInfo

Additional transaction details.

> **Important:** MONETA.RU returns operationInfo only if you set the version attribute of your request to **VERSION_2** or higher.

If the response includes the **paymentPasswordChallengeRequired** key with the **true** value in the attribute element, you must call GetAccountPaymentPasswordChallengeRequest for the payer's account to retrieve the paymentPasswordChallenge element. This element is used in such requests as PaymentRequest, TransferRequest, AuthoriseTransactionRequest.

Optional element.

Type: *OperationInfo Complex Type* on page 78

Version Simple Type

Defines the MONETA.Merchant API version. Differences between versions, if any, are described in the documentation.

Default value: VERSION_1.

Type: string

Valid values:

- **VERSION_1**
- **VERSION_2**
- **VERSION_3**
- **VERSION_4**
- **VERSION_5**

XML Schema

```
<xsd:simpleType xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="Version">
   <xsd:restriction base="xsd:string">
      <xsd:enumeration value="VERSION_1"/>
      <xsd:enumeration value="VERSION_2"/>
      <xsd:enumeration value="VERSION_3"/>
      <xsd:enumeration value="VERSION_4"/>
      <xsd:enumeration value="VERSION_5"/>
   </xsd:restriction>
</xsd:simpleType>
```

Financial Endpoints

AuthoriseTransactionBatch Endpoint

Input message: AuthoriseTransactionBatchRequest

Request for putting an authorization hold on the payment amount in the payer's account in batch mode.

Type: *AuthoriseTransactionBatchRequestType Complex Type* on page 58

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             name="AuthoriseTransactionBatchRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:AuthoriseTransactionBatchRequestType"/>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Output message: AuthoriseTransactionBatchResponse

Response to the authorization hold request in batch mode.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             name="AuthoriseTransactionBatchResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element maxOccurs="unbounded" minOccurs="1" name="transaction"
                      type="tns:OperationInfoBatchResponseType"/>
      </xsd:sequence>
   </xsd:complexType>
```

```
</xsd:element>
```

Description

transaction

Either information about completed transactions or error descriptions in the same order as in AuthoriseTransactionBatchRequest.

Required element.

Type: *OperationInfoBatchResponseType Complex Type* on page 79

AuthoriseTransaction Endpoint

Input message: AuthoriseTransactionRequest

Request for putting an authorization hold on the payment amount in the payer's account.

To complete the transaction, use ConfirmTransactionRequest. If MONETA.RU does not receive ConfirmTransactionRequest until the expiration date and time, the transaction is canceled automatically.

Type: *AuthoriseTransactionRequestType Complex Type* on page 59

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="AuthoriseTransactionRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:AuthoriseTransactionRequestType"/>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Output message: AuthoriseTransactionResponse

Response to the authorization hold request.

Type: *OperationInfo Complex Type* on page 78

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="AuthoriseTransactionResponse">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:OperationInfo"/>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Description

Two additional elements are added to the list of transaction attributes:

- **protectioncode**. The protection code that must be specified in ConfirmTransactionRequest.
- **protectioncodeexpirationdate**. Authorization expiration date. If the transaction is not confirmed until this date, the transaction is canceled automatically.

CancelTransactionBatch Endpoint

Input message: CancelTransactionBatchRequest

Request for canceling transactions in batch mode.

Type: *CancelTransactionBatchRequestType Complex Type* on page 60

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="CancelTransactionBatchRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:CancelTransactionBatchRequestType"/>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Output message: CancelTransactionBatchResponse

Response to a transaction cancellation request in batch mode.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             name="CancelTransactionBatchResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element maxOccurs="unbounded" minOccurs="1" name="transaction"
                      type="tns:CancelTransactionBatchResponseType"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

transaction

Either information about completed transactions or error descriptions in the same order as in CancelTransactionBatchRequest.

Required element.

Type: *CancelTransactionBatchResponseType Complex Type* on page 61

CancelTransaction Endpoint

Input message: CancelTransactionRequest

Request for canceling a transaction. You can cancel a transaction only if both conditions are true:

- The transaction is in processing (incomplete).
- You have access to the accounts of the payer and payee.

Type: *CancelTransactionRequestType Complex Type* on page 62

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="CancelTransactionRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:CancelTransactionRequestType"/>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Output message: CancelTransactionResponse

Response to a transaction cancellation request.

Type: *CancelTransactionResponseType Complex Type* on page 62

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="CancelTransactionResponse">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:CancelTransactionResponseType"/>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

ConfirmTransactionBatch Endpoint

Input message: ConfirmTransactionBatchRequest

Request for confirming transactions in batch mode.

Type: *ConfirmTransactionBatchRequestType Complex Type* on page 63

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             name="ConfirmTransactionBatchRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:ConfirmTransactionBatchRequestType"/>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Output message: ConfirmTransactionBatchResponse

Response to a request for transaction confirmations in batch mode.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             name="ConfirmTransactionBatchResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element maxOccurs="unbounded" minOccurs="1" name="transaction"
                      type="tns:OperationInfoBatchResponseType"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

transaction

Either information about completed transactions or error descriptions in the same order as in ConfirmTransactionBatchRequest.

Required element.

Type: *OperationInfoBatchResponseType Complex Type* on page 79

ConfirmTransaction Endpoint

Input message: ConfirmTransactionRequest

Request for confirming the transaction.

Type: *ConfirmTransactionRequestType Complex Type* on page 63

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="ConfirmTransactionRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:ConfirmTransactionRequestType"/>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Output message: ConfirmTransactionResponse

Response to a transaction confirmation request.

Type: *OperationInfo Complex Type* on page 78

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="ConfirmTransactionResponse">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:OperationInfo"/>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Description

Transaction information.

Transaction is completed if statusid is set to **SUCCEED**.

FindLastOperationsList Endpoint

Input message: FindLastOperationsListRequest

Request for getting the list of the last transactions.

If MONETA.RU finds no transactions, the size element in the response is set to 0.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="FindLastOperationsListRequest">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element minOccurs="0" name="unitId" type="xsd:long"/>
         <xsd:element default="5" minOccurs="0" name="transactionsQuantity">
            <xsd:simpleType>
               <xsd:restriction base="xsd:int">
                  <xsd:minInclusive value="1"/>
                  <xsd:maxInclusive value="100"/>
               </xsd:restriction>
            </xsd:simpleType>
         </xsd:element>
         <xsd:element minOccurs="0" name="attributeNamesInResponse" type="xsd:string"/>
         <xsd:element minOccurs="0" name="showReceiptInfo" type="xsd:boolean"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

unitId

Unique identifier of a user in MONETA.RU.

If you omit this element, MONETA.RU sets the value of this element to the user who sends the request

Optional element.

Type: long

transactionsQuantity

Maximum number of transactions to include in the response.

Optional element.

Type: int

Default value: 5

attributeNamesInResponse

List of comma separated transaction properties which will be returned in a response.

It is not obligatory to provide standard transaction properties since they are always returned (if they exist in transaction). These properties are: modified, statusid, typeid, category, isinvoice, parentid, sourceaccountid, sourceamount, sourceamounttotal, sourceaccounttotal, sourcecurrencycode, sourcealias, targetaccountid, targetamount, targetamounttotal, targetaccounttotal, targetcurrencycode, targetalias, description, clienttransaction, targettransaction.

For example, in case with bank transfer when it is required to receive some properties in a response, you can use the following: wirebankbik, wirebankaccount, wireusername, customfield:email

Optional element.

Type: string

showReceiptInfo

Set true to get receipt status (if exists).

Optional element.

Type: boolean

Output message: FindLastOperationsListResponse

Response to FindLastOperationsListRequest that contains the list of last transactions.

Type: *OperationInfoList Complex Type* on page 80

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             name="FindLastOperationsListResponse">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:OperationInfoList"/>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

FindOperationsListByCTID Endpoint

Input message: FindOperationsListByCTIDRequest

Request for retrieving transaction details by a merchant transaction ID.

The response might include multiple transactions. A long list of transactions might be split into multiple pages.

If MONETA.RU finds no transactions, the size element in the response is set to 0.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             name="FindOperationsListByCTIDRequest">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element minOccurs="0" name="pager" type="tns:Pager"/>
         <xsd:element name="accountId" type="tns:AccountId"/>
         <xsd:element name="clientTransaction">
            <xsd:simpleType>
               <xsd:restriction base="tns:CTID">
                  <xsd:minLength value="1"/>
               </xsd:restriction>
            </xsd:simpleType>
         </xsd:element>
         <xsd:element minOccurs="0" name="showReceiptInfo" type="xsd:boolean"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

pager

Specifies the sequence number of the page that you want to retrieve if the list of transactions includes multiple pages.

Optional element.

Type: *Pager Complex Type* on page 88

accountId

MONETA.RU account number.

Type: *AccountId Simple Type* on page 51

clientTransaction

The unique identifier of the transaction in the merchant system.

Type: *CTID Simple Type* on page 60

Minimum length: 1

showReceiptInfo

Set true to get receipt status (if exists).

Optional element.

Type: boolean

Output message: FindOperationsListByCTIDResponse

Response to the request for searching transactions by a merchant transaction ID.

The response might include multiple transactions. If the list of transactions is long, the response is split into pages.

Type: *OperationInfoList Complex Type* on page 80

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             name="FindOperationsListByCTIDResponse">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:OperationInfoList"/>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

FindOperationsList Endpoint

Input message: FindOperationsListRequest

Request for getting a list of transactions by the specified filter.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="FindOperationsListRequest">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element minOccurs="0" name="pager" type="tns:Pager"/>
         <xsd:element name="filter">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element minOccurs="0" name="unitId" type="xsd:long"/>
                  <xsd:element maxOccurs="unbounded" minOccurs="0" name="accountType"
 type="tns:AccountType"/>
                  <xsd:element minOccurs="0" name="accountId" type="tns:AccountId"/>
                  <xsd:element minOccurs="1" name="dateFrom" type="xsd:dateTime"/>
                  <xsd:element minOccurs="1" name="dateTo" type="xsd:dateTime"/>
                  <xsd:element minOccurs="0" name="operationId" type="xsd:long"/>
                  <xsd:element minOccurs="0" name="amountFrom" type="tns:Money"/>
                  <xsd:element minOccurs="0" name="amountTo" type="tns:Money"/>
                  <xsd:element minOccurs="0" name="currencyCode" type="tns:Currency"/>
                  <xsd:element minOccurs="0" name="targetAccountId"
 type="tns:AccountId"/>
                  <xsd:element minOccurs="0" name="operationStatus"
 type="tns:OperationStatus"/>
                  <xsd:element minOccurs="0" name="clientTransaction" type="tns:CTID"/>
                  <xsd:element minOccurs="0" name="operationAmountTypeId"
 type="xsd:int"/>
                  <xsd:element minOccurs="0" name="propertyName" type="xsd:string"/>
                  <xsd:element minOccurs="0" name="propertyValue" type="xsd:string"/>
                  <xsd:element minOccurs="0" name="operationCategoryId"
 type="xsd:long"/>
                  <xsd:element minOccurs="0" name="refund" type="xsd:boolean"/>
                  <xsd:element minOccurs="0" name="accountingPeriodDate"
 type="xsd:dateTime"/>
                  <xsd:element default="false" minOccurs="0" name="searchInArchive"
 type="xsd:boolean"/>
                  <xsd:element minOccurs="0" name="attributeNamesInResponse"
 type="xsd:string"/>
                  <xsd:element minOccurs="0" name="showReceiptInfo" type="xsd:boolean"/
>
               </xsd:sequence>
            </xsd:complexType>
         </xsd:element>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

pager

Specifies the sequence number of the page that you want to retrieve if the list of transactions includes multiple pages.

Optional element.

Type: *Pager Complex Type* on page 88

filter

Filtering criteria.

unitId

Unique identifier of the user in MONETA.RU.

Optional element.

Type: long

accountType

Account type (example: Advanced account, Bank account).

Optional element.

Type: *AccountType Simple Type* on page 58

accountId

Account number.

Optional element.

Type: *AccountId Simple Type* on page 51

dateFrom

The start date of the period.

Required element.

Type: dateTime

dateTo

The end date of the period.

Required element.

Type: dateTime

operationId

MONETA.RU transaction ID.

Optional element.

Type: long

amountFrom

Minimum transaction amount.

If you include this element, you must also add the currencyCode filter parameter.

Optional element.

Type: *Money Simple Type* on page 77

amountTo

Maximum transaction amount.

If you include this element, you must also add the currencyCode filter parameter.

Optional element.

Type: *Money Simple Type* on page 77

currencyCode

Transaction currency.

Optional element.

Type: *Currency Simple Type* on page 66

targetAccountId

Correspondent account number in MONETA.RU.

Optional element.

Type: *AccountId Simple Type* on page 51

operationStatus

Transaction status.

Optional element.

Type: *OperationStatus Simple Type* on page 81

clientTransaction

Merchant transaction ID.

Optional element.

Type: *CTID Simple Type* on page 60

operationAmountTypeId

Specifies whether the transaction is debit or credit. Valid values are:

- **1**. Both credit and debit operations.
- **2**. Credit operations.
- **3**. Debit operations.

Optional element.

Type: int

propertyName

Additional operationInfo attributes of transactions. Valid attributes are:

- **ALFAIDCLIENT**. Specifies an AlfaBank client ID.
- **ALFAIDINVOICE**. Specifies an AlfaBank invoice ID.
- **CARDNUMBER**. Specifies the card number of the payer.
- **CONTACTTRANSFERORDERNUMBER**. Specifies the transfer order number in the Contact payment system.
- **CYBERPLATTRANSID**. Specifies a Cyberplat transaction ID.
- **DESCRIPTION**. Specifies a description for the transaction.
- **DMRBUYEREMAIL**. Specifies an email address of a client in dengi@mail.ru payment system.
- **MOSCOWBANKRRN**. Specifies an RRN transaction number in Moscow Bank.
- **PARENTID**. Specifies the parent transaction ID.
- **PAYEECARDNUMBER**. Specifies the card number of the payee.
- **PAYMENTTOKEN**. Specifies a payment token.
- **QIWIPHONE**. Specifies the phone number of a QIWI payment system user.
- **RAPIDATID**. Specifies a Rapida TID.
- **SBERBANKRECEIPT**. Specifies a Sberbank transaction ID.
- **SBSEVKAVRECEIPT**. Specifies a Sberbank SK transaction ID.
- **SUBPROVIDERID**. Specifies a provider ID of service payment.
- **SUBSCRIBERID**. Specifies a subscriber ID.
- **TELEPAYTXNID**. Specifies a transaction ID in the TelePay payment system.
- **WEBMONEYPURSE**. Specifies the WMR purse number of the specified user.
- **WEBMONEYWMID**. Specifies the identifier of a WebMoney user (WMID).
- **WIREDOCINDEX**. Specifies a unique identifier of charges.
- **WIRETRANSFERORDERNUMBER**. Specifies the number of the bank wired transfer order.
- **YANDEXPAYERCODE**. Specifies the Yandex Money wallet number of the payer.
- **YANDEXACCOUNT**. Specifies the Yandex Money wallet number of the payee.
- **PURPOSE:CASHBACK**. Specifies transaction as cashback. Available value is **1**.

Use the propertyValue element to specify the values of these additional attributes.

Optional element.

Type: string

propertyValue

Specifies a value for the additional transaction attribute that is specified by the propertyName element.

The response includes transactions which additional attribute values match the filter. You can use the following wildcards to specify masks: asterisk (*).

Optional element.

Type: string

operationCategoryId

Specifies the category of the transaction. Valid values are:

- **1** for deposit
- **2** for withdrawal
- **3** for transfer
- **4** for goods and services

Optional element.

Type: long

refund

Transactions with "Refund" type:

- no element - without restrictions;
- **true** - refunds only;
- **false** - except refunds.

Optional element.

Type: boolean

accountingPeriodDate

Settling date.

Optional element.

Type: dateTime

searchInArchive

Indicates whether to search for archived transactions.

Optional element.

Type: boolean

Default value: false

attributeNamesInResponse

List of comma separated transaction properties which will be returned in a response.

It is not obligatory to provide standard transaction properties since they are always returned. These properties are: modified, statusid, typeid, category, sourceaccountid, sourceamount, sourcecurrencycode, targetaccountid, targetamount, targetcurrencycode, description, clienttransaction.

If value is empty or contains only whitespaces, result includes only standard transaction properties.

If value is not set, result includes all available transaction properties.

For example, in case with bank transfer when it is required to receive not all properties in a response, you can use the following: wirebankbik, wirebankaccount, wireusername, customfield:email

Optional element.

Type: string

showReceiptInfo

Set true to get receipt status (if exists).

Optional element.

Type: boolean

Output message: FindOperationsListResponse

Response to FindOperationsListRequest.

The response might include multiple transactions. A long list of transactions might be split into multiple pages

Type: *OperationInfoList Complex Type* on page 80

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="FindOperationsListResponse">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:OperationInfoList"/>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

ForecastTransaction Endpoint

Input message: ForecastTransactionRequest

Request for estimating transaction amounts and fees by transaction parameters.

Type: *TransactionRequestType Complex Type* on page 103

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="ForecastTransactionRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:TransactionRequestType"/>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Output message: ForecastTransactionResponse

Response to a request for estimating transaction amounts and fees. Use information from this response to inform customers about the estimated amount and fee of transaction. Actual values may differ if newer financial rules and currency exchange rates are applied when processing a transaction.

Type: *ForecastTransactionResponseType Complex Type* on page 71

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="ForecastTransactionResponse">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:ForecastTransactionResponseType"/>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

GetAccountPaymentPasswordChallenge Endpoint

Input message: GetAccountPaymentPasswordChallengeRequest

Request for a challenge passcode. Use this request if a user gets payment passwords by SMS or uses a list of indexed transaction authentication numbers (TANs).

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             name="GetAccountPaymentPasswordChallengeRequest">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element minOccurs="1" name="accountId" type="tns:AccountId"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

accountId

Account number.

Required element.

Type: *AccountId Simple Type* on page 51

Output message: GetAccountPaymentPasswordChallengeResponse

Response to GetAccountPaymentPasswordChallengeRequest.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             name="GetAccountPaymentPasswordChallengeResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element minOccurs="0" name="paymentPasswordChallenge" type="xsd:string"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

paymentPasswordChallenge

Challenge passcode for users of indexed transaction authentication numbers (TANs).

Optional element.

Type: string

GetOperationDetailsById Endpoint

Input message: GetOperationDetailsByIdRequest

Request for transaction details by a MONETA.RU transaction ID.

If MONETA.RU cannot find a transaction with the specified transaction ID, an error occurs.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             name="GetOperationDetailsByIdRequest">
   <xsd:simpleType>
      <xsd:restriction base="xsd:long"/>
   </xsd:simpleType>
</xsd:element>
```

Output message: GetOperationDetailsByIdResponse

Response to the transaction details request.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             name="GetOperationDetailsByIdResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="operation" type="tns:OperationInfo"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

operation

Type: *OperationInfo Complex Type* on page 78

InvoiceBatch Endpoint

Input message: InvoiceBatchRequest

Request for creating invoices in batch mode.

Type: *InvoiceBatchRequestType Complex Type* on page 74

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="InvoiceBatchRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:InvoiceBatchRequestType"/>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Output message: InvoiceBatchResponse

Response to a request for creating invoices in batch mode.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="InvoiceBatchResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element maxOccurs="unbounded" minOccurs="1" name="transaction"
                      type="tns:TransactionBatchResponseType"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

transaction

Either information about completed transactions or error descriptions in the same order as in the InvoiceBatch request.

Required element.

Type: *TransactionBatchResponseType Complex Type* on page 102

Invoice Endpoint

Input message: InvoiceRequest

Request parameters for creating an invoice.

Type: *InvoiceRequestType Complex Type* on page 74

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="InvoiceRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:InvoiceRequestType"/>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Output message: InvoiceResponse

Response to a request for creating an invoice.

Type: *TransactionResponseType Complex Type* on page 104

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="InvoiceResponse">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:TransactionResponseType"/>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Payment Endpoint

Input message: PaymentRequest

Request for transferring money to another account. Unlike the Transfer request, the payee element in Payment request can specify not only the account number, but also a transaction ID, e-mail, or phone number.

Type: *PaymentRequestType Complex Type* on page 90

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="PaymentRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:PaymentRequestType"/>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Output message: PaymentResponse

Response to a request for transferring money to another account.

Type: *OperationInfo Complex Type* on page 78

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="PaymentResponse">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:OperationInfo"/>
      </xsd:complexContent>
```

```
   </xsd:complexType>
</xsd:element>
```

Description

Transaction information.

Transaction is completed if statusid is set to **SUCCEED**.

PaymentBatch Endpoint

Input message: PaymentBatchRequest

Request for transferring money to another account in batch mode.

Type: *PaymentBatchRequestType Complex Type* on page 89

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="PaymentBatchRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:PaymentBatchRequestType"/>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Output message: PaymentBatchResponse

Response to a request for transferring money to another account in batch mode.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="PaymentBatchResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element maxOccurs="unbounded" minOccurs="1" name="transaction"
                      type="tns:OperationInfoBatchResponseType"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

transaction

Either information about completed transactions or error descriptions in the same order as in the PaymentBatch request.

Required element.

Type: *OperationInfoBatchResponseType Complex Type* on page 79

Refund Endpoint

Input message: RefundRequest

Request for refunding a transaction.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="RefundRequest">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="transactionId" type="xsd:long"/>
         <xsd:element minOccurs="0" name="amount" type="tns:Money"/>
         <xsd:element minOccurs="0" name="paymentPassword" type="tns:Password"/>
```

```
          <xsd:element minOccurs="0" name="clientTransaction" type="tns:CTID"/>
          <xsd:element minOccurs="0" name="description" type="tns:Description"/>
          <xsd:element minOccurs="0" name="operationInfo" type="tns:OperationInfo"/>
          <xsd:element minOccurs="0" name="paymentPasswordChallenge" type="xsd:string"/>
       </xsd:sequence>
    </xsd:complexType>
 </xsd:element>
```

Description

transactionId

Unique identifier of the transaction in MONETA.RU that requires a refund.

Type: long

amount

Refund amount that is specified in the payee's currency of the original transaction. If the amount element is omitted, MONETA.RU refunds the amount that was specified by the payee in the original transaction.

Optional element.

Type: *Money Simple Type* on page 77

paymentPassword

Payment password for the payer's account.

Optional element.

Type: *Password Simple Type* on page 89

clientTransaction

Merchant transaction ID.

Optional element.

Type: *CTID Simple Type* on page 60

description

Transaction description or comments.

Optional element.

Type: *Description Simple Type* on page 66

operationInfo

Key-value pairs that will be saved as transaction attributes. For dates, use the following format: dd.MM.yyyy HH:mm:ss

Optional element.

Type: *OperationInfo Complex Type* on page 78

paymentPasswordChallenge

Challenge passcode that you received in the GetAccountPaymentPasswordChallenge response in the paymentPasswordChallenge element. Specify this element in the following cases:

- If a user gets payment passwords by SMS, set paymentPasswordChallenge to SMS. Set paymentPassword to the value that the user receives in the SMS from MONETA.RU.
- If a user gets a sequence number (index) for a password from a list of transaction authentication numbers (TANs), set paymentPasswordChallenge to the TAN index. Set paymentPassword to the TAN that has the specified index.

Optional element.

Type: string

Output message: RefundResponse

Response to a refund request.

Type: *OperationInfo Complex Type* on page 78

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="RefundResponse">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:OperationInfo"/>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Description

Transaction information.

Transaction is completed if statusid is set to **SUCCEED**.

SecureData Endpoint

Input message: SecureDataRequest

Attributes preservation request.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="SecureDataRequest">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="publicId" type="xsd:string"/>
         <xsd:element minOccurs="0" name="accountId" type="tns:AccountId"/>
         <xsd:element maxOccurs="unbounded" minOccurs="1" name="attribute"
                      type="tns:KeyValueAttribute"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

publicId

Public ID.

Type: string

accountId

Account ID.

Optional element.

Type: *AccountId Simple Type* on page 51

attribute

Attributes for preservation.

Required element.

Type: *KeyValueAttribute Complex Type* on page 76

Output message: SecureDataResponse

Response to SecureDataRequest.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="SecureDataResponse">
   <xsd:complexType>
```

```
      <xsd:sequence>
         <xsd:element name="secureToken" type="xsd:string"/>
         <xsd:element minOccurs="0" name="expirationDate" type="xsd:dateTime"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

secureToken

Secure Token.

Type: string

expirationDate

Secure token expiration date.

Optional element.

Type: dateTime

SecureDataStatus Endpoint

Input message: SecureDataStatusRequest

SECURETOKEN status request.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="SecureDataStatusRequest">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="secureToken" type="xsd:string"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

secureToken

Secure Token.

Type: string

Output message: SecureDataStatusResponse

Response to SecureDataStatusRequest.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="SecureDataStatusResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element minOccurs="0" name="expirationDate" type="xsd:dateTime"/>
         <xsd:element maxOccurs="unbounded" minOccurs="0" name="attribute"
                      type="tns:KeyValueAttribute"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

expirationDate

Secure token expiration date.

Optional element.

Type: dateTime

attribute

Saved attributes:

- **CARDNUMBER**. Masked value.
- **CARDEXPIRATION**
- **CARDHOLDER**
- **PAYEECARDNUMBER**. Masked value.

Optional element.

Type: *KeyValueAttribute Complex Type* on page 76

TransferBatch Endpoint

Input message: TransferBatchRequest

Request for multiple money transfer operations in batch mode.

Type: *TransactionBatchRequestType Complex Type* on page 102

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="TransferBatchRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:TransactionBatchRequestType"/>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Output message: TransferBatchResponse

Response to a request for multiple money transfer operations in batch mode.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="TransferBatchResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element maxOccurs="unbounded" minOccurs="1" name="transaction"
                      type="tns:TransactionBatchResponseType"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

transaction

Either information about completed transactions or error descriptions in the same order as in the TransferBatch request.

Required element.

Type: *TransactionBatchResponseType Complex Type* on page 102

Transfer Endpoint

Input message: TransferRequest

Request for transferring money to another account.

Type: *TransactionRequestType Complex Type* on page 103

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="TransferRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:TransactionRequestType"/>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Output message: TransferResponse

Response to a request for transferring money to another account.

Type: *TransactionResponseType Complex Type* on page 104

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="TransferResponse">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:TransactionResponseType"/>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

VerifyPayment Endpoint

Input message: VerifyPaymentRequest

Transaction validation request. Unlike VerifyTransferRequest, the payee element in this request can specify not only an account number, but also a transaction ID, e-mail address, or phone number.

Type: *PaymentRequestType Complex Type* on page 90

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="VerifyPaymentRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:PaymentRequestType"/>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Output message: VerifyPaymentResponse

Response to a transaction validation request.

Type: *VerifyTransactionResponseType Complex Type* on page 105

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="VerifyPaymentResponse">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:VerifyTransactionResponseType"/>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

VerifyPaymentBatch Endpoint

Input message: VerifyPaymentBatchRequest

Transaction validation request. Request in batch mode.

Type: *PaymentBatchRequestType Complex Type* on page 89

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="VerifyPaymentBatchRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:PaymentBatchRequestType"/>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Output message: VerifyPaymentBatchResponse

Response to a transaction validation request. Response in batch mode.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="VerifyPaymentBatchResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element maxOccurs="unbounded" minOccurs="1" name="transaction"
                      type="tns:VerifyTransactionResponseType"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

transaction

Either information about transactions or error descriptions in the same order as in the VerifyPaymentBatch request.

Required element.

Type: *VerifyTransactionResponseType Complex Type* on page 105

VerifyTransaction Endpoint

Input message: VerifyTransactionRequest

Transaction verification request.

Type: *Entity Complex Type* on page 69

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="VerifyTransactionRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:Entity">
            <xsd:sequence>
               <xsd:element name="transactionId" type="xsd:long"/>
               <xsd:element minOccurs="0" name="payer" type="xsd:string"/>
               <xsd:element minOccurs="0" name="amount" type="tns:Money"/>
               <xsd:element minOccurs="0" name="isPayerAmount" type="xsd:boolean"/>
               <xsd:element minOccurs="0" name="operationInfo"
 type="tns:OperationInfo"/>
               <xsd:element minOccurs="0" name="paymentPassword" type="tns:Password"/>
               <xsd:element minOccurs="0" name="paymentPasswordChallenge"
 type="xsd:string"/>
               <xsd:element minOccurs="0" name="clientTransaction" type="tns:CTID"/>
            </xsd:sequence>
         </xsd:extension>
```

```
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Description

transactionId

The unique identifier of transaction which you want to verify.

Type: long

payer

Account number of the payer.

Optional element.

Type: string

amount

Transaction amount. The currency of the transaction is specified by the isPayerAmount element. If a request does not include the isPayerAmount element, MONETA.RU uses the following rules to determine the currency of the transaction:

- If a user has access only to the payer's account, MONETA.RU uses the currency of the payer's account for the transaction.
- If a user has access only to the payee's account, MONETA.RU uses the currency of the payee's account for the transaction.

Optional element.

Type: *Money Simple Type* on page 77

isPayerAmount

This element is required if a user has access to the payer's and to the payee's accounts. Valid values:

- **true**. MONETA.RU uses the currency of the payer's account for the transaction amount.
- **false**. MONETA.RU uses the currency of the payee's account for the transaction amount.

Optional element.

Type: boolean

operationInfo

Key-value pairs that will be used as transaction attributes. For dates, use the following format: dd.MM.yyyy HH:mm:ss

Optional element.

Type: *OperationInfo Complex Type* on page 78

paymentPassword

Payment password for the payer's account.

Optional element.

Type: *Password Simple Type* on page 89

paymentPasswordChallenge

Challenge passcode that you received in the GetAccountPaymentPasswordChallenge response in the paymentPasswordChallenge element. Specify this element in the following cases:

- If a user gets payment passwords by SMS, set paymentPasswordChallenge to SMS. Set paymentPassword to the value that the user receives in the SMS from MONETA.RU.
- If a user gets a sequence number (index) for a password from a list of transaction authentication numbers (TANs), set paymentPasswordChallenge to the TAN index. Set paymentPassword to the TAN that has the specified index.

Optional element.

Type: string

clientTransaction

Merchant transaction ID.

Optional element.

Type: *CTID Simple Type* on page 60

Output message: VerifyTransactionResponse

Response to a request for verifying a transaction in MONETA.RU.

Type: *VerifyTransactionResponseType Complex Type* on page 105

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="VerifyTransactionResponse">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:VerifyTransactionResponseType"/>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

VerifyTransfer Endpoint

Input message: VerifyTransferRequest

Request for validating a transaction in MONETA.RU.

Type: *TransactionRequestType Complex Type* on page 103

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="VerifyTransferRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:TransactionRequestType"/>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Output message: VerifyTransferResponse

Response to a transaction validation request in MONETA.RU.

Type: *VerifyTransferResponseType Complex Type* on page 106

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="VerifyTransferResponse">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:VerifyTransferResponseType"/>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

PaymentSystemInfo Endpoint

Input message: PaymentSystemInfoRequest

Request information about payments system

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="PaymentSystemInfoRequest">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="accountId" type="tns:AccountId"/>
         <xsd:element minOccurs="0" name="amount" type="tns:Money"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

accountId

MONETA.RU account number.

Type: *AccountId Simple Type* on page 51

amount

Optional element.

Type: *Money Simple Type* on page 77

Output message: PaymentSystemInfoResponse

Response to a request PaymentSystemInfoRequest.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="PaymentSystemInfoResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element maxOccurs="unbounded" minOccurs="0" name="items"
                      type="tns:PaymentSystemInfoComplexType"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Operation templates

CreateOperationTemplate Endpoint

Input message: CreateOperationTemplateRequest

Type: *OperationTemplate Complex Type* on page 82

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             name="CreateOperationTemplateRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:OperationTemplate">
            <xsd:sequence>
               <xsd:element minOccurs="0" name="paymentPassword"
 type="tns:PaymentPassword"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Description

paymentPassword

Payment password for the payer's account.

Optional element.

Type: *PaymentPassword Complex Type* on page 90

Output message: CreateOperationTemplateResponse

Response to CreateOperationTemplateRequest.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             name="CreateOperationTemplateResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="id" type="xsd:long"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

id

Type: long

EditOperationTemplate Endpoint

Input message: EditOperationTemplateRequest

Type: *OperationTemplate Complex Type* on page 82

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="EditOperationTemplateRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:OperationTemplate">
            <xsd:sequence>
               <xsd:element minOccurs="0" name="paymentPassword"
 type="tns:PaymentPassword"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Description

paymentPassword

Payment password for the payer's account.

Optional element.

Type: *PaymentPassword Complex Type* on page 90

Output message: EditOperationTemplateResponse

Response to EditOperationTemplateRequest.

The response contains no data, if MONETA.RU applies your changes to the operation template. If MONETA.RU cannot apply your changes to the operation template, an error occurs.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="EditOperationTemplateResponse">
   <xsd:complexType/>
</xsd:element>
```

FindOperationTemplates Endpoint

Input message: FindOperationTemplatesRequest

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="FindOperationTemplatesRequest">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:choice>
            <xsd:element minOccurs="0" name="id" type="xsd:long"/>
            <xsd:sequence>
               <xsd:element minOccurs="0" name="unitId" type="xsd:long"/>
               <xsd:element minOccurs="0" name="operationTypeCategory"
 type="tns:OperationTypeCategory"/>
               <xsd:element minOccurs="0" name="type" type="tns:OperationTemplateType"/
>
               <xsd:element minOccurs="0" name="tag" type="xsd:string"/>
            </xsd:sequence>
         </xsd:choice>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

id

Optional element.

Type: long

unitId

User ID. If you omit this element, MONETA.RU uses the ID of the user who sends the request.

Optional element.

Type: long

operationTypeCategory

Transaction category.

Optional element.

Type: *OperationTypeCategory Simple Type* on page 88

type

Operation template type.

Optional element.

Type: *OperationTemplateType Simple Type* on page 87

tag

Optional element.

Type: string

Output message: FindOperationTemplatesResponse

Response to FindOperationTemplatesRequest.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             name="FindOperationTemplatesResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element maxOccurs="unbounded" minOccurs="0" name="operationTemplate"
                      type="tns:OperationTemplate"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

operationTemplate

Optional element.

Type: *OperationTemplate Complex Type* on page 82

DeleteOperationTemplate Endpoint

Input message: DeleteOperationTemplateRequest

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             name="DeleteOperationTemplateRequest">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="id" type="xsd:long"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

id

Type: long

Output message: DeleteOperationTemplateResponse

Response to DeleteOperationTemplateRequest.

The response contains no data. If MONETA.RU cannot delete operation template, an error occurs.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             name="DeleteOperationTemplateResponse">
   <xsd:complexType/>
</xsd:element>
```

Managing MONETA.RU User Profiles

FindProfileInfoByAccountId Endpoint

Input message: FindProfileInfoByAccountIdRequest

Request user profile information by an account number.

If MONETA.RU cannot find the specified account, an error occurs.

Type: *AccountId Simple Type* on page 51

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             name="FindProfileInfoByAccountIdRequest">
   <xsd:complexType>
      <xsd:simpleContent>
         <xsd:extension base="tns:AccountId">
            <xsd:attribute name="version" type="tns:Version" use="optional"/>
         </xsd:extension>
      </xsd:simpleContent>
   </xsd:complexType>
</xsd:element>
```

Description

Output message: FindProfileInfoByAccountIdResponse

Response to FindProfileInfoByAccountIdRequest.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             name="FindProfileInfoByAccountIdResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="accountId" type="tns:AccountId"/>
         <xsd:element name="currency" type="tns:Currency"/>
         <xsd:element name="profile" type="tns:Profile"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

accountId

MONETA.RU account number.

Type: *AccountId Simple Type* on page 51

currency

Account currency.

Type: *Currency Simple Type* on page 66

profile

Information about a MONETA.RU user profile.

Type: *Profile Complex Type* on page 93

EditProfile Endpoint

Input message: EditProfileRequest

Request for editing a MONETA.RU user profile.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="EditProfileRequest">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="unitId" type="xsd:long"/>
         <xsd:element minOccurs="0" name="profileId" type="xsd:long"/>
         <xsd:element minOccurs="0" name="profile" type="tns:Profile"/>
         <xsd:element minOccurs="0" name="parentId" type="xsd:long"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

unitId

The unique identifier of the MONETA.RU user that you want to edit.

Type: long

profileId

Optional element.

Type: long

profile

User profile information in the list of key-value pairs.

The list of supported keys depends on ProfileType.

Optional element.

Type: *Profile Complex Type* on page 93

parentId

The unique identifier of the parent MONETA.RU profile that must own the profile that you edit.

The user who sends this request must have access to the specified profile. And your agreement must allow creating profiles of this type.

Optional element.

Type: long

Output message: EditProfileResponse

Response to EditProfileRequest.

The response contains no data, if MONETA.RU applies your changes to the user profile. If MONETA.RU cannot apply your changes to the user profile, an error occurs.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="EditProfileResponse">
   <xsd:complexType/>
</xsd:element>
```

FindProfileInfo Endpoint

Input message: FindProfileInfoRequest

Request for information about a MONETA.RU user profile that matches the specified criteria.

MONETA.RU returns a list of profiles that might contain multiple pages.

Type: *Entity Complex Type* on page 69

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="FindProfileInfoRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:Entity">
            <xsd:sequence>
               <xsd:element minOccurs="0" name="pager" type="tns:Pager"/>
               <xsd:element minOccurs="1" name="filter">
                  <xsd:complexType>
                     <xsd:sequence>
                        <xsd:element minOccurs="0" name="unitId" type="xsd:long"/>
                        <xsd:element minOccurs="0" name="name" type="xsd:string"/>
                        <xsd:element minOccurs="0" name="email" type="xsd:string"/>
                        <xsd:element minOccurs="0" name="phone" type="xsd:string"/>
                        <xsd:element minOccurs="0" name="login" type="xsd:string"/>
                        <xsd:element minOccurs="0" name="accountBalanceFrom"
 type="tns:Money"/>
                        <xsd:element minOccurs="0" name="accountBalanceTo"
 type="tns:Money"/>
                        <xsd:element minOccurs="0" name="accountCurrencyCode"
 type="tns:Currency"/>
                        <xsd:element minOccurs="0" name="profileId" type="xsd:long"/>
                     </xsd:sequence>
                  </xsd:complexType>
               </xsd:element>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Description

pager

Specifies the sequence number of the page that you want to retrieve if the list of profiles includes multiple pages.

Optional element.

Type: *Pager Complex Type* on page 88

filter

Search criteria.

Required element.

unitId

Unique identifier of the parent MONETA.RU account that acts as a starting point of the search.

If you omit this element, MONETA.RU uses the ID of the user who sends the reuqest.

Optional element.

Type: long

name

The name of the user or organization.

Minimum length is two characters.

You can use the following wildcards to specify masks: asterisk (*) and question mark (?).

For organization profiles, MONETA.RU searches the organization_name element. For client profiles, MONETA.RU searches the last_name, first_name, and middle_initial_name elements.

Optional element.

Type: string

email

The email of the user profile that you want to find.

Minimum length is two characters.

You can use the following wildcards to specify masks: asterisk (*) and question mark (?).

Optional element.

Type: string

phone

The phone number of the user profile that you want to find.

Minimum length is two characters.

You can use the following wildcards to specify masks: asterisk (*) and question mark (?).

User profile can have different phones. To search specific phone (the search will be faster), you can specify prefix:

- **PHONE_DIRECTOR:** - Managing director office phone
- **PHONE_ACCOUNTANT:** - Financial support phone
- **PHONE_CONTACT:** - Contact phone number
- **PHONE_SUPPORT:** - Technical support phone number
- **PHONE:** - Telephone
- **CELL_PHONE:** - Mobile phone
- **LOGIN:** - Login
- **ADDRESS:** - Notifications (Cell phone number for notifications)

Example: **CELL_PHONE:79051234567**

Optional element.

Type: string

login

The username (login) that you want to find.

Minimum length is two characters.

You can use the following wildcards to specify masks: asterisk (*) and question mark (?).

Optional element.

Type: string

accountBalanceFrom

Specifies the minimum balance of the account that you want to find.

Optional element.

Type: *Money Simple Type* on page 77

accountBalanceTo

Specifies the maximum balance of the account that you want to find.

Optional element.

Type: *Money Simple Type* on page 77

accountCurrencyCode

If you include the maximum or minimum balance in the request, specifies the account currency.

Optional element.

Type: *Currency Simple Type* on page 66

profileId

Optional element.

Type: long

Output message: FindProfileInfoResponse

Response to FindProfileInfoRequest.

The response might contain multiple pages.

If the size element of the response is set to 0, MONETA.RU found no data.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="FindProfileInfoResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="pageSize" type="xsd:long"/>
         <xsd:element name="pageNumber" type="xsd:long"/>
         <xsd:element name="pagesCount" type="xsd:long"/>
         <xsd:element name="size" type="xsd:long"/>
         <xsd:element name="totalSize" type="xsd:long"/>
         <xsd:element maxOccurs="unbounded" minOccurs="0" name="profile"
 type="tns:Profile"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

pageSize

Specifies the number of profile to return per page.

Type: long

pageNumber

The current page number. Page numbering starts from 1.

Type: long

pagesCount

Total number of pages in the response.

Type: long

size

The number of profiles on the current page.

Type: long

totalSize

The total number of profiles in the response.

Type: long

profile

The list of profiles.

Optional element.

Type: *Profile Complex Type* on page 93

GetProfileInfo Endpoint

Input message: GetProfileInfoRequest

Request for profile information by a MONETA.RU user ID.

If MONETA.RU finds no information about the specified user, an error occurs.

Type: *Entity Complex Type* on page 69

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="GetProfileInfoRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:Entity">
            <xsd:sequence>
               <xsd:element minOccurs="0" name="unitId" type="xsd:long"/>
               <xsd:element minOccurs="0" name="profileId" type="xsd:long"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Description

unitId

MONETA.RU user ID.

If you omit this element, MONETA.RU uses the ID of the user who sends this request.

Optional element.

Type: long

profileId

Optional element.

Type: long

Output message: GetProfileInfoResponse

Response to GetProfileInfoRequest.

Type: *Profile Complex Type* on page 93

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="GetProfileInfoResponse">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:Profile"/>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Description

CreateProfile Endpoint

Input message: CreateProfileRequest

Request for creating a MONETA.RU account.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="CreateProfileRequest">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element minOccurs="0" name="unitId" type="xsd:long"/>
         <xsd:element minOccurs="0" name="profileId" type="xsd:long"/>
```

```
         <xsd:element name="profileType" type="tns:ProfileType"/>
         <xsd:element name="profile" type="tns:Profile"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

unitId

Unique identifier of the parent MONETA.RU account that will own the new account.

If you ommit this element, the new account will belong to the the user who sends this request.

Optional element.

Type: long

profileId

Optional element.

Type: long

profileType

Profile type. Valid values are:

- client
- organization

Type: *ProfileType Simple Type* on page 98

profile

User profile information in the list of key-value pairs.

The list of supported keys depends on ProfileType.

Type: *Profile Complex Type* on page 93

Output message: CreateProfileResponse

Response to CreateProfileRequest that contains the unique identifier of the new user.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="CreateProfileResponse">
   <xsd:simpleType>
      <xsd:restriction base="xsd:long"/>
   </xsd:simpleType>
</xsd:element>
```

CheckProfile Endpoint

Input message: CheckProfileRequest

Request to receive the status of the organization profile (attributes to be filled in) in MONETA.RU.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="CheckProfileRequest">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="unitId" type="xsd:long"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

unitId

The unique identifier of the MONETA.RU organization that you want to check.

Type: long

Output message: CheckProfileResponse

Response to CheckProfileRequest that contains the state of the profile.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="CheckProfileResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="status" type="xsd:string"/>
         <xsd:sequence>
            <xsd:element maxOccurs="unbounded" minOccurs="0" name="requestInfo">
               <xsd:complexType>
                  <xsd:sequence>
                     <xsd:element minOccurs="0" name="action" type="xsd:string"/>
                     <xsd:element minOccurs="0" name="scope" type="xsd:string"/>
                     <xsd:element minOccurs="0" name="method" type="xsd:string"/>
                     <xsd:choice>
                        <xsd:element minOccurs="0" name="profile">
                           <xsd:complexType>
                              <xsd:sequence>
                                 <xsd:element minOccurs="0" name="unitId"
 type="xsd:long"/>
                                 <xsd:element minOccurs="0" name="profileId"
 type="xsd:long"/>
                                 <xsd:element minOccurs="0" name="profileType"
 type="tns:ProfileType"/>
                                 <xsd:element minOccurs="0" name="profile"
 type="tns:Profile"/>
                              </xsd:sequence>
                           </xsd:complexType>
                        </xsd:element>
                        <xsd:element minOccurs="0" name="document">
                           <xsd:complexType>
                              <xsd:sequence>
                                 <xsd:element minOccurs="0" name="unitId"
 type="xsd:long"/>
                                 <xsd:element minOccurs="0" name="profileId"
 type="xsd:long"/>
                                 <xsd:element minOccurs="0" name="id" type="xsd:long"/>
                                 <xsd:element minOccurs="0" name="type"
 type="tns:DocumentType"/>
                                 <xsd:element maxOccurs="unbounded" minOccurs="0"
 name="attribute"
                                              type="tns:KeyValueAttribute"/>
                              </xsd:sequence>
                           </xsd:complexType>
                        </xsd:element>
                        <xsd:element minOccurs="0" name="bank">
                           <xsd:complexType>
                              <xsd:sequence>
                                 <xsd:element minOccurs="0" name="unitId"
 type="xsd:long"/>
                                 <xsd:element minOccurs="0" name="id" type="xsd:long"/>
                                 <xsd:element maxOccurs="unbounded" minOccurs="0"
 name="attribute"
                                              type="tns:KeyValueAttribute"/>
                              </xsd:sequence>
                           </xsd:complexType>
                        </xsd:element>
                        <xsd:element minOccurs="0" name="juridical">
                           <xsd:complexType>
                              <xsd:sequence>
```

```
                                    <xsd:element minOccurs="0" name="unitId"
 type="xsd:long"/>
                                    <xsd:element minOccurs="0" name="id" type="xsd:long"/>
                                    <xsd:element maxOccurs="unbounded" minOccurs="0"
 name="attribute"
                                                 type="tns:KeyValueAttribute"/>
                              </xsd:sequence>
                           </xsd:complexType>
                        </xsd:element>
                     </xsd:choice>
                  </xsd:sequence>
               </xsd:complexType>
            </xsd:element>
         </xsd:sequence>
         <xsd:element minOccurs="0" name="foundersTotalShare" type="xsd:float"/>
         <xsd:element minOccurs="0" name="daysBeforePartnerLock" type="xsd:long"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

status

Profile status:

NONE [no data required]

OK [all data entered]

DATA_REQUIRED [some data required]

Type: string

foundersTotalShare

The status of founders' shares (%)

Optional element.

Type: float

daysBeforePartnerLock

Remaining days to block partner - move in group Expired

Optional element.

Type: long

requestInfo

Request information

Optional element.

action

REQUEST [required to execute the request based on data below]

Optional element.

Type: string

scope

Request scope: Personal [personal data], Director [director profile], Director document, Founder, Beneficiary, Beneficiary document, Licence, Juridical, Bank

Optional element.

Type: string

method

API Method to execute

Optional element.

Type: string

profile

Request data [Personal, Director, Founder, Beneficiary] to fill profile

Optional element.

document

Request data [Director document, , Beneficiary document] to fill profile

Optional element.

bank

Request data [Bank] to fill profile

Optional element.

juridical

Request data [Juridical] to fill profile

Optional element.

unitId

Unique identifier of the parent MONETA.RU account that will own the new account.

If you ommit this element, the new account will belong to the the user who sends this request.

Optional element.

Type: long

profileId

ID main profile or child subprofile

Optional element.

Type: long

profileType

Profile type. Valid values are:

- client
- organization

Optional element.

Type: *ProfileType Simple Type* on page 98

profile

User profile information in the list of key-value pairs.

The list of supported keys depends on ProfileType.

Optional element.

Type: *Profile Complex Type* on page 93

unitId

The unique identifier of the user to whom the document belongs.

If you omit this element, MONETA.RU uses the ID of the user who sends the request.

Optional element.

Type: long

profileId

ID main profile or child subprofile

Optional element.

Type: long

id

Unique identifier of a document in MONETA.RU.

Optional element.

Type: long

type

Document type

Optional element.

Type: *DocumentType Simple Type* on page 68

attribute

Document attributes in MONETA.RU.

Document information is specified in a list of key-value pairs

Optional element.

Type: *KeyValueAttribute Complex Type* on page 76

unitId

Unique identifier of the parent MONETA.RU account that will own the bank data.

If you omit this element, the new account will belong to the the user who sends this request.

Optional element.

Type: long

id

Unique identifier of a object with bank data.

Optional element.

Type: long

attribute

Document attributes in MONETA.RU.

Document information is specified in a list of key-value pairs

Optional element.

Type: *KeyValueAttribute Complex Type* on page 76

unitId

Unique identifier of the parent MONETA.RU account that will own the juridical data.

If you omit this element, the new account will belong to the the user who sends this request.

Optional element.

Type: long

id

Unique identifier of a object with juridical data.

Optional element.

Type: long

attribute

Document attributes in MONETA.RU.

Document information is specified in a list of key-value pairs

Optional element.

Type: *KeyValueAttribute Complex Type* on page 76

Managing MONETA.RU Accounts

FindAccountByAlias Endpoint

Input message: FindAccountByAliasRequest

Request for searching an account by an alias.

The user that sends this request must own this account. The alias cannot be empty. If MONETA.RU cannot find an account, an error occurs.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="FindAccountByAliasRequest">
   <xsd:complexType>
      <xsd:simpleContent>
         <xsd:extension base="xsd:string">
            <xsd:attribute name="version" type="tns:Version" use="optional"/>
         </xsd:extension>
      </xsd:simpleContent>
   </xsd:complexType>
</xsd:element>
```

Output message: FindAccountByAliasResponse

Response to FindAccountByAliasRequest. The response contains information about the account that has the specified alias.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="FindAccountByAliasResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="account" type="tns:AccountInfo"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

FindAccountById Endpoint

Input message: FindAccountByIdRequest

Request for searching an account by an account number.

The user that sends this request must own this account. If MONETA.RU cannot find an account, an error occurs.

Type: *AccountId Simple Type* on page 51

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="FindAccountByIdRequest">
   <xsd:complexType>
      <xsd:simpleContent>
         <xsd:extension base="tns:AccountId">
            <xsd:attribute name="version" type="tns:Version" use="optional"/>
         </xsd:extension>
      </xsd:simpleContent>
```

```
   </xsd:complexType>
</xsd:element>
```

Output message: FindAccountByIdResponse

Response to FindAccountByIdRequest. The response contains information about the account that has the specified account number.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="FindAccountByIdResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="account" type="tns:AccountInfo"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

CreateAccount Endpoint

Input message: CreateAccountRequest

Request for creating a MONETA.RU account for the user that is specified in the unitId element. If unitId is omitted, an account is created for the current user.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="CreateAccountRequest">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element default="2" minOccurs="0" name="type" type="tns:AccountType"/>
         <xsd:element minOccurs="0" name="subTypeId" type="xsd:int"/>
         <xsd:element name="currency" type="tns:Currency"/>
         <xsd:element minOccurs="0" name="alias" type="xsd:string"/>
         <xsd:element minOccurs="0" name="paymentPasswordType"
 type="tns:AccountPaymentPasswordType"/>
         <xsd:element minOccurs="0" name="paymentPassword">
            <xsd:simpleType>
               <xsd:restriction base="xsd:string">
                  <xsd:minLength value="5"/>
                  <xsd:whiteSpace value="collapse"/>
               </xsd:restriction>
            </xsd:simpleType>
         </xsd:element>
         <xsd:element minOccurs="0" name="paymentPasswordExpirationDate"
 type="xsd:boolean"/>
         <xsd:element minOccurs="0" name="unitId" type="xsd:long"/>
         <xsd:element minOccurs="0" name="onSuccessfulDebitUrl" type="xsd:string"/>
         <xsd:element minOccurs="0" name="onSuccessfulCreditUrl" type="xsd:string"/>
         <xsd:element minOccurs="0" name="signature" type="xsd:string"/>
         <xsd:element minOccurs="0" name="lowBalanceThreshold" type="tns:Money"/>
         <xsd:element minOccurs="0" name="highBalanceThreshold" type="tns:Money"/>
         <xsd:element minOccurs="0" name="prototypeAccountId" type="xsd:long"/>
         <xsd:element minOccurs="0" name="onCancelledDebitUrl" type="xsd:string"/>
         <xsd:element minOccurs="0" name="onCancelledCreditUrl" type="xsd:string"/>
         <xsd:element minOccurs="0" name="onAuthoriseUrl" type="xsd:string"/>
         <xsd:element maxOccurs="unbounded" minOccurs="0" name="attribute"
                      type="tns:KeyValueApprovedAttribute"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

type

Account type.

Default value is 2 (Advanced account).

Optional element.

Type: *AccountType Simple Type* on page 58

Default value: 2

subTypeId

Account subtype. Optional field.

Please, contact Commercial department for details.

Valid values for type "Advanced account" (type = 2) are:

- **50** - Agent.

Optional element.

Type: int

currency

Account currency.

Type: *Currency Simple Type* on page 66

alias

Account alias. Specify a unique name among the other accounts of the specified user.

Optional element.

Type: string

paymentPasswordType

Payment password type. Default value is STATIC.

Optional element.

Type: *AccountPaymentPasswordType Simple Type* on page 54

paymentPassword

Payment password. Minimum length is 5 symbols. Use this element, when paymentPasswordType is STATIC.

Optional element.

Type: string

Minimum length: 5

paymentPasswordExpirationDate

Payment password expiration date. Use this element, when paymentPasswordType is STATIC.

Optional element.

Type: boolean

unitId

The owner of the account. If you omit this element, the account will belong to the current user.

Optional element.

Type: long

onSuccessfulDebitUrl

Specifies the URL of the script that MONETA.RU calls after debiting the payer's account.

Optional element.

Type: string

onSuccessfulCreditUrl

Specifies the URL of the script that MONETA.RU calls after crediting the payee's account.

Optional element.

Type: string

signature

Signature that MONETA.RU uses to verify data integrity submitted from a payment form.

Optional element.

Type: string

lowBalanceThreshold

Specifies the minimum balance threshold for an account. If the balance drops below the threshold, MONETA.RU sends daily notifications to the account owner.

Optional element.

Type: *Money Simple Type* on page 77

highBalanceThreshold

Specifies the maximum balance threshold for an account. If the balance exceeds the threshold, MONETA.RU sends daily notifications to the account owner.

Optional element.

Type: *Money Simple Type* on page 77

prototypeAccountId

Prototype account number. Settings from this account are used as default values.

Optional element.

Type: long

onCancelledDebitUrl

Specifies the URL of the script that MONETA.RU calls if a debit transaction is canceled.

Optional element.

Type: string

onCancelledCreditUrl

Specifies the URL of the script that MONETA.RU calls if a deposit transaction is canceled.

Optional element.

Type: string

onAuthoriseUrl

Specifies the URL of the script that MONETA.RU calls after authorising the payer's account.

Optional element.

Type: string

attribute

Account properties.

Information about an account contains a list of key-value pairs. Valid keys for account are:

- **alias**. Account alias. Specify a unique name among the other accounts of the specified user.
- **assistantVersion**. MONETA.Assistant version. Available values:

  v1

  v2

  v3
- **interfacetype**. Interface type.

1 - MONETA.Assistant.

- **testmode**. Test mode (true|false).
- **paymentsystem_limitids**. Payment method IDs.
- **paymentsystem_unitid**. Default payment method.
- **checkurl**. Check URL.
- **payurl**. Pay URL.
- **httpmethod**. HTTP method for PayUrl, CheckUrl (GET | POST).
- **issignaturemandatory**. Mandatory payment form signature (true|false).
- **redefinesettingsinurl**. Settings can be redefined in URL (true|false).
- **successurl**. Success URL.
- **failurl**. Fail URL.
- **inprogressurl**. InProgress URL.
- **returnurl**. Return URL.
- **assistantformtarget**. Target (redirect in iframe). Available values: _blank, _self, _top, _parent.

Optional element.

Type: *KeyValueApprovedAttribute Complex Type* on page 76

Output message: CreateAccountResponse

Response to CreateAccountRequest. The response contains the unique identifier (AccountId) of the created account.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="CreateAccountResponse">
   <xsd:simpleType>
      <xsd:restriction base="tns:AccountId"/>
   </xsd:simpleType>
</xsd:element>
```

EditAccount Endpoint

Input message: EditAccountRequest

Account modification request. It's allowed to change alias and payment password for account.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="EditAccountRequest">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="id" type="tns:AccountId"/>
         <xsd:element minOccurs="0" name="alias" type="xsd:string"/>
         <xsd:element minOccurs="0" name="paymentPassword">
            <xsd:simpleType>
               <xsd:restriction base="xsd:string">
                  <xsd:minLength value="5"/>
                  <xsd:whiteSpace value="collapse"/>
               </xsd:restriction>
            </xsd:simpleType>
         </xsd:element>
         <xsd:element minOccurs="0" name="paymentPasswordExpirationDate"
 type="xsd:boolean"/>
         <xsd:element minOccurs="0" name="oldPaymentPassword">
            <xsd:simpleType>
               <xsd:restriction base="xsd:string">
                  <xsd:minLength value="5"/>
                  <xsd:whiteSpace value="collapse"/>
               </xsd:restriction>
            </xsd:simpleType>
         </xsd:element>
```

```
          <xsd:element minOccurs="0" name="onSuccessfulDebitUrl" type="xsd:string"/>
          <xsd:element minOccurs="0" name="onSuccessfulCreditUrl" type="xsd:string"/>
          <xsd:element minOccurs="0" name="signature" type="xsd:string"/>
          <xsd:element minOccurs="0" name="lowBalanceThreshold" type="tns:Money"/>
          <xsd:element minOccurs="0" name="highBalanceThreshold" type="tns:Money"/>
          <xsd:element minOccurs="0" name="prototypeAccountId" type="xsd:long"/>
          <xsd:element minOccurs="0" name="oldPaymentPasswordChallenge"
 type="xsd:string"/>
          <xsd:element minOccurs="0" name="onCancelledDebitUrl" type="xsd:string"/>
          <xsd:element minOccurs="0" name="onCancelledCreditUrl" type="xsd:string"/>
          <xsd:element minOccurs="0" name="onAuthoriseUrl" type="xsd:string"/>
          <xsd:element maxOccurs="unbounded" minOccurs="0" name="attribute"
                       type="tns:KeyValueApprovedAttribute"/>
      </xsd:sequence>
    </xsd:complexType>
</xsd:element>
```

Description

id

The unique identifier of the account that you want to edit.

Type: *AccountId Simple Type* on page 51

alias

New alias of account.

Account alias. Specify a unique name among the other accounts of the specified user.

Optional element.

Type: string

paymentPassword

New payment password for the account.

You must specify the oldPaymentPassword element to specify a new payment password.

Optional element.

Type: string

Minimum length: 5

paymentPasswordExpirationDate

Switch on/off payment password expiration date.

Optional element.

Type: boolean

oldPaymentPassword

Current payment password.

This element is required only if the request includes the PaymentPassword element.

Optional element.

Type: string

Minimum length: 5

onSuccessfulDebitUrl

Specifies the URL of the script that MONETA.RU calls after debiting the funds from the account.

Optional element.

Type: string

onSuccessfulCreditUrl

Specifies the URL of the script that MONETA.RU calls after crediting funds to the account.

Optional element.

Type: string

signature

Signature that MONETA.RU uses to verify data integrity submitted from a payment form.

Optional element.

Type: string

lowBalanceThreshold

Specifies the minimum balance threshold for an account. If the balance drops below the threshold, MONETA.RU sends daily notifications to the account owner.

Optional element.

Type: *Money Simple Type* on page 77

highBalanceThreshold

Specifies the maximum balance threshold for an account. If the balance exceeds the threshold, MONETA.RU sends daily notifications to the account owner.

Optional element.

Type: *Money Simple Type* on page 77

prototypeAccountId

Prototype account number. Settings from this account are used as default values.

Optional element.

Type: long

oldPaymentPasswordChallenge

Challenge passcode that you received in the GetAccountPaymentPasswordChallenge response in the paymentPasswordChallenge element. Specify this element in the following cases:

- If a user gets payment passwords by SMS, set paymentPasswordChallenge to SMS. Set paymentPassword to the value that the user receives in the SMS from MONETA.RU.
- If a user gets a sequence number (index) for a password from a list of transaction authentication numbers (TANs), set paymentPasswordChallenge to the TAN index. Set paymentPassword to the TAN that has the specified index.

Optional element.

Type: string

onCancelledDebitUrl

Specifies the URL of the script that MONETA.RU calls if a debit transaction is canceled.

Optional element.

Type: string

onCancelledCreditUrl

Specifies the URL of the script that MONETA.RU calls if a deposit transaction is canceled.

Optional element.

Type: string

onAuthoriseUrl

Specifies the URL of the script that MONETA.RU calls after authorising the payer's account.

Optional element.

Type: string

attribute

Account properties.

Information about an account contains a list of key-value pairs. Valid keys for account are:

- **alias**. Account alias. Specify a unique name among the other accounts of the specified user.
- **assistantVersion**. MONETA.Assistant version. Available values:

  v1

  v2

  v3
- **interfacetype**. Interface type.

  1 - MONETA.Assistant.
- **testmode**. Test mode (true|false).
- **paymentsystem_limitids**. Payment method IDs.
- **paymentsystem_unitid**. Default payment method.
- **checkurl**. Check URL.
- **payurl**. Pay URL.
- **httpmethod**. HTTP method for PayUrl, CheckUrl (GET | POST).
- **issignaturemandatory**. Mandatory payment form signature (true|false).
- **redefinesettingsinurl**. Settings can be redefined in URL (true|false).
- **successurl**. Success URL.
- **failurl**. Fail URL.
- **inprogressurl**. InProgress URL.
- **returnurl**. Return URL.
- **assistantformtarget**. Target (redirect in iframe). Available values: _blank, _self, _top, _parent.

Optional element.

Type: *KeyValueApprovedAttribute Complex Type* on page 76

Output message: EditAccountResponse

Response to the request for editing an account. If you changed the account settings successfully, the response contains no data. If MONETA.RU cannot change the account settings, an error occurs.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="EditAccountResponse">
   <xsd:complexType/>
</xsd:element>
```

FindAccountsList Endpoint

Input message: FindAccountsListRequest

Request for a list of accounts that match the specified search criteria.

Type: *Entity Complex Type* on page 69

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="FindAccountsListRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:Entity">
            <xsd:sequence>
               <xsd:element minOccurs="0" name="unitId" type="xsd:long"/>
               <xsd:element minOccurs="0" name="alias" type="xsd:string"/>
               <xsd:element minOccurs="0" name="currency" type="tns:Currency"/>
```

```
               <xsd:element minOccurs="0" name="isDelegatedAccount" type="xsd:boolean"/
>
               <xsd:element minOccurs="0" name="showBankAccount" type="xsd:boolean"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Description

unitId

The owner of the accounts. If you omit this element, the request uses the name of the user who sends the request.

Optional element.

Type: long

alias

Account alias. An alias can include the following wildcards: asterisk (*) and question mark (?).

Optional element.

Type: string

currency

The currency of the account.

Optional element.

Type: *Currency Simple Type* on page 66

isDelegatedAccount

Indicates whether the account is delegated. Valid values are:

- **true**. Select only delegated accounts.
- **false**. Select only non-delegated (owned) accounts.
- Omit this element to select both delegated and non-delegated accounts.

Optional element.

Type: boolean

showBankAccount

If showBankAccount is **true**, response will have elements **bankAccountForCredits** and **bankAccountForDebits**.

Optional element.

Type: boolean

Output message: FindAccountsListResponse

Response to FindAccountsListRequest. The response includes the list of accounts.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="FindAccountsListResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element maxOccurs="unbounded" minOccurs="0" name="account"
 type="tns:AccountInfo"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

BlockAccount Endpoint

Input message: BlockAccountRequest

Block an account.

Type: *AbstractAttributeObject Complex Type* on page 50

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="BlockAccountRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:AbstractAttributeObject">
            <xsd:sequence>
               <xsd:element name="id" type="tns:AccountId"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Description

id

The identifier of the account.

Type: *AccountId Simple Type* on page 51

Output message: BlockAccountResponse

Response to the request for blocking an account. If you blocked the account successfully, the response contains no data. If MONETA.RU cannot block the account, an error occurs.

Type: *AbstractAttributeObject Complex Type* on page 50

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="BlockAccountResponse">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:AbstractAttributeObject"/>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

UnblockAccount Endpoint

Input message: UnblockAccountRequest

Unblock an account.

Type: *AbstractAttributeObject Complex Type* on page 50

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="UnblockAccountRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:AbstractAttributeObject">
            <xsd:sequence>
               <xsd:element name="id" type="tns:AccountId"/>
               <xsd:element name="secretAnswer" type="xsd:string"/>
               <xsd:element minOccurs="0" name="paymentPassword"
 type="tns:PaymentPassword"/>
            </xsd:sequence>
         </xsd:extension>
```

```
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Description

id

The identifier of the account.

Type: *AccountId Simple Type* on page 51

secretAnswer

Answer to secret question (in Security).

Type: string

paymentPassword

Payment password.

Optional element.

Type: *PaymentPassword Complex Type* on page 90

Output message: UnblockAccountResponse

Response to the request for unblocking an account. If you unblocked the account successfully, the response contains no data. If MONETA.RU cannot unblock the account, an error occurs.

Type: *AbstractAttributeObject Complex Type* on page 50

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="UnblockAccountResponse">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:AbstractAttributeObject"/>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Managing Profile Documents, Contracts, and Legal Information

CreateProfileDocument Endpoint

Input message: CreateProfileDocumentRequest

Request for creating a document.

Type: *Document Complex Type* on page 67

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="CreateProfileDocumentRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:Document">
            <xsd:sequence>
               <xsd:element minOccurs="0" name="unitId" type="xsd:long"/>
               <xsd:element minOccurs="0" name="profileId" type="xsd:long"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
```

```
</xsd:element>
```

Description

unitId

The unique identifier of the user to whom the document belongs.

If you omit this element, MONETA.RU uses the ID of the user who sends the request.

Optional element.

Type: long

profileId

Optional element.

Type: long

Output message: CreateProfileDocumentResponse

Response to CreateProfileDocumentRequest.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="CreateProfileDocumentResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="id" type="xsd:long"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

id

Unique identifier of the document in MONETA.RU.

Type: long

EditProfileDocument Endpoint

Input message: EditProfileDocumentRequest

Request for modifying a document.

Type: *Document Complex Type* on page 67

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="EditProfileDocumentRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:Document">
            <xsd:sequence>
               <xsd:element minOccurs="0" name="unitId" type="xsd:long"/>
               <xsd:element minOccurs="0" name="profileId" type="xsd:long"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Description

unitId

The unique identifier of the user to whom the document belongs.

If you omit this element, MONETA.RU uses the ID of the user who sends the request.

Optional element.

Type: long

profileId

Optional element.

Type: long

Output message: EditProfileDocumentResponse

Response to EditProfileDocumentRequest.

If MONETA.RU saves the document changes successfully, the response contains no data.

If MONETA.RU fails to save changes to the document, an error occurs.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="EditProfileDocumentResponse">
   <xsd:complexType/>
</xsd:element>
```

FindContracts Endpoint

Input message: FindContractsRequest

Request for contract information by a user ID. If MONETA.RU does not find information, the request contains an empty list.

Type: *Entity Complex Type* on page 69

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="FindContractsRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:Entity">
            <xsd:sequence>
               <xsd:element minOccurs="0" name="unitId" type="xsd:long"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Description

unitId

MONETA.RU user ID.

If you omit this element, MONETA.RU uses the ID of the user who sends the request.

Optional element.

Type: long

Output message: FindContractsResponse

Response to FindContractsRequest.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="FindContractsResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element maxOccurs="unbounded" minOccurs="0" name="contract"
 type="tns:Contract"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

FindLegalInformation Endpoint

Input message: FindLegalInformationRequest

Request for legal information by a user ID. The response contains an empty list, if MONETA.RU finds no data.

Type: *Entity Complex Type* on page 69

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="FindLegalInformationRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:Entity">
            <xsd:sequence>
               <xsd:element minOccurs="0" name="unitId" type="xsd:long"/>
               <xsd:element minOccurs="0" name="profileId" type="xsd:long"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Description

unitId

MONETA.RU user ID.

If you omit this element, MONETA.RU uses the ID of the user who sends the request.

Optional element.

Type: long

profileId

Optional element.

Type: long

Output message: FindLegalInformationResponse

Response to FindLegalInformationRequest.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="FindLegalInformationResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element maxOccurs="unbounded" minOccurs="0" name="legalInformation"
                      type="tns:LegalInformation"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

FindProfileDocumentFiles Endpoint

Input message: FindProfileDocumentFilesRequest

Request for binary data that is associated with the specified document.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             name="FindProfileDocumentFilesRequest">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="documentId" type="xsd:long"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Output message: FindProfileDocumentFilesResponse

Response to the request for binary data. The response contains binary data that is associated with the specified document.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             name="FindProfileDocumentFilesResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element maxOccurs="unbounded" minOccurs="0" name="file" type="tns:File"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

FindProfileDocuments Endpoint

Input message: FindProfileDocumentsRequest

Request for documents that are associated with the specified MONETA.RU user account.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="FindProfileDocumentsRequest">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="unitId" type="xsd:long"/>
         <xsd:element minOccurs="0" name="profileId" type="xsd:long"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

unitId

The unique identifier of the user whose documents you want to get.

Type: long

profileId

Optional element.

Type: long

Output message: FindProfileDocumentsResponse

Response to FindProfileDocumentsRequest.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="FindProfileDocumentsResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element maxOccurs="unbounded" minOccurs="0" name="document"
 type="tns:Document"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

document

The list of documents. IF MONETA.RU found no documents that belong to the specified user, the response contains an empty list.

Optional element.

Type: *Document Complex Type* on page 67

UploadProfileDocumentFile Endpoint

Input message: UploadProfileDocumentFileRequest

Request for adding or modifying a file.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             name="UploadProfileDocumentFileRequest">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="file" type="tns:File"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

file

Indicates whether to add or modify a file.

Specify the fileId element to modify an existing file. Omit this element to add a new file.

This request does not use the approved is element.

If you omit the mimeType element and specify a file name in the title element, MONETA.RU sets mimeType to match the extension of the file.

Type: *File Complex Type* on page 70

Output message: UploadProfileDocumentFileResponse

Response to the request for adding or modifying a file (UploadProfileDocumentFileRequest).

If MONETA.RU processes the request successfully, the response contains no data. If MONETA.RU cannot process the request, an error occurs.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

```
            name="UploadProfileDocumentFileResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="fileId" type="xsd:long"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

fileId

File ID.

Type: long

Managing Bank Accounts

FindBankAccounts Endpoint

Input message: FindBankAccountsRequest

Request for bank account details by a user ID. MONETA.RU returns an empty list if it cannot find bank account details.

Type: *Entity Complex Type* on page 69

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="FindBankAccountsRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:Entity">
            <xsd:sequence>
               <xsd:element minOccurs="0" name="unitId" type="xsd:long"/>
               <xsd:element minOccurs="0" name="profileId" type="xsd:long"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Description

unitId

User ID. If you omit this element, MONETA.RU uses the ID of the user who sends the request.

Optional element.

Type: long

profileId

Optional element.

Type: long

Output message: FindBankAccountsResponse

Response to FindBankAccountsRequest.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="FindBankAccountsResponse">
   <xsd:complexType>
```

```
      <xsd:sequence>
         <xsd:element maxOccurs="unbounded" minOccurs="0" name="bankAccount"
 type="tns:BankAccount"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

EditBankAccount Endpoint

Input message: EditBankAccountRequest

Request for editing a bank account.

Type: *BankAccount Complex Type* on page 59

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="EditBankAccountRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:BankAccount">
            <xsd:sequence>
               <xsd:element minOccurs="0" name="unitId" type="xsd:long"/>
               <xsd:element minOccurs="0" name="profileId" type="xsd:long"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Description

unitId

The unique identifier of the user who owns the bank account.

If you omit this element, MONETA.RU uses the ID of the user who sends the request.

Optional element.

Type: long

profileId

Optional element.

Type: long

Output message: EditBankAccountResponse

Response to EditBankAccountRequest.

The response contains no data, if MONETA.RU edited the bank account successfully.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="EditBankAccountResponse">
   <xsd:complexType/>
</xsd:element>
```

CreateBankAccount Endpoint

Input message: CreateBankAccountRequest

Request for adding information about your bank account to your MONETA.RU profile.

Type: *BankAccount Complex Type* on page 59

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="CreateBankAccountRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:BankAccount">
            <xsd:sequence>
               <xsd:element minOccurs="0" name="unitId" type="xsd:long"/>
               <xsd:element minOccurs="0" name="profileId" type="xsd:long"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Description

unitId

The unique identifier of the user who will own the bank account.

If you omit this element, MONETA.RU uses the ID of the user who sends the request.

Optional element.

Type: long

profileId

Optional element.

Type: long

Output message: CreateBankAccountResponse

Response to the request for creating a bank account.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="CreateBankAccountResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="id" type="xsd:long"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

id

Unique identifier of the bank account in MONETA.RU.

Type: long

Managing Delegated Access to MONETA.RU Accounts

FindAccountRelations Endpoint

Input message: FindAccountRelationsRequest

Request for a list of users who have delegated to the specified account ID.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="FindAccountRelationsRequest">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="accountId" type="tns:AccountId"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

accountId

MONETA.RU account number.

Type: *AccountId Simple Type* on page 51

Output message: FindAccountRelationsResponse

Response to FindAccountRelationRequest.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="FindAccountRelationsResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element maxOccurs="unbounded" minOccurs="0" name="accountRelation"
                      type="tns:AccountRelation"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

accountRelation

The list of users who have delegated access to the specified account. If MONETA.RU finds no users, the response contains an empty list.

Optional element.

Type: *AccountRelation Complex Type* on page 54

GetAccountRelation Endpoint

Input message: GetAccountRelationRequest

Request for delegated access details.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="GetAccountRelationRequest">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="accountId" type="tns:AccountId"/>
         <xsd:element name="principalEmail" type="tns:Email"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

accountId

MONETA.RU account number.

Type: *AccountId Simple Type* on page 51

principalEmail

Email address of the user.

Type: *Email Simple Type* on page 68

Output message: GetAccountRelationResponse

Response to GetAccountRelationRequest.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="GetAccountRelationResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element maxOccurs="1" minOccurs="0" name="accountRelation"
 type="tns:AccountRelation"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

SaveAccountRelation Endpoint

Input message: SaveAccountRelationRequest

Request for granting delegated access to the MONETA.RU account.

Type: *AccountRelation Complex Type* on page 54

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="SaveAccountRelationRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:AccountRelation">
            <xsd:sequence>
               <xsd:element name="paymentPassword" type="tns:Password"/>
               <xsd:element minOccurs="0" name="paymentPasswordChallenge"
 type="xsd:string"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Description

paymentPassword

Payment password for the account.

Type: *Password Simple Type* on page 89

paymentPasswordChallenge

Challenge passcode that you received in the GetAccountPaymentPasswordChallenge response in the paymentPasswordChallenge element. Specify this element in the following cases:

- If a user gets payment passwords by SMS, set paymentPasswordChallenge to SMS. Set paymentPassword to the value that the user receives in the SMS from MONETA.RU.
- If a user gets a sequence number (index) for a password from a list of transaction authentication numbers (TANs), set paymentPasswordChallenge to the TAN index. Set paymentPassword to the TAN that has the specified index.

Optional element.

Type: string

Output message: SaveAccountRelationResponse

Response to SaveAccountRelationRequest.

If MONETA.RU saves the information about delegated access successfully, the response contains no data. If MONETA.RU fails to save the information, an error occurs.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="SaveAccountRelationResponse">
   <xsd:complexType/>
</xsd:element>
```

DeleteAccountRelation Endpoint

Input message: DeleteAccountRelationRequest

Request for deletion of access delegation information.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="DeleteAccountRelationRequest">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="accountId" type="tns:AccountId"/>
         <xsd:element name="principalEmail" type="tns:Email"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

accountId

MONETA.RU account number.

Type: *AccountId Simple Type* on page 51

principalEmail

Email address of the user whose delegated access to your MONETA.RU account you want to revoke.

Type: *Email Simple Type* on page 68

Output message: DeleteAccountRelationResponse

Response to DeleteAccountRelationRequest.

If MONETA.RU deletes the information about delegated access successfully, the response contains no data. If MONETA.RU fails to delete the information, an error occurs.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="DeleteAccountRelationResponse">
   <xsd:complexType/>
</xsd:element>
```

MONETA.RU Reports

GetTurnoverList Endpoint

Input message: GetTurnoverListRequest

Request for the monthly turnover information.

If MONETA.RU finds no data, the size element in the response is set to 0.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="GetTurnoverListRequest">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element minOccurs="0" name="unitId" type="xsd:long"/>
         <xsd:element name="dateFrom" type="xsd:date"/>
         <xsd:element minOccurs="0" name="dateTo" type="xsd:date"/>
         <xsd:element maxOccurs="unbounded" minOccurs="0" name="accountType"
 type="tns:AccountType"/>
         <xsd:element maxOccurs="unbounded" minOccurs="0" name="accountIds"
 type="tns:AccountId"/>
         <xsd:element minOccurs="0" name="currencyCode" type="tns:Currency"/>
         <xsd:element default="true" minOccurs="0" name="groupByCurrency"
 type="xsd:boolean"/>
         <xsd:element minOccurs="0" name="operationTypeCategory"
 type="tns:OperationTypeCategory"/>
         <xsd:element minOccurs="0" name="categoryDetails" type="xsd:boolean"/>
         <xsd:element maxOccurs="unbounded" minOccurs="0" name="attribute"
                      type="tns:KeyValueAttribute"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

unitId

Unique identifier of a user in MONETA.RU.

If you omit this element, MONETA.RU sets the unitId value to the user who sends the request.

Optional element.

Type: long

dateFrom

The start date of the period. MONETA.RU always starts the period on the first day of the month.

Type: date

dateTo

The end date of the period. MONETA.RU always ends the period on the last day of the month. If you omit the dateTo element, MONETA.RU uses the last day of the month in the dateFrom element.

Optional element.

Type: date

accountType

Account type (example: Advanced account, Bank account).

Optional element.

Type: *AccountType Simple Type* on page 58

accountIds

The list of account numbers to include into the turnover.

Optional element.

Type: *AccountId Simple Type* on page 51

currencyCode

Currency to include into the turnover information.

Optional element.

Type: *Currency Simple Type* on page 66

groupByCurrency

Indicates whether to group turnovers by currency. Valid values are:

- **true**. Group by currency.
- **false**. Do not group by currency.

Default value: true

Optional element.

Type: boolean

Default value: true

operationTypeCategory

Category of transactions to include into the turnover information.

Optional element.

Type: *OperationTypeCategory Simple Type* on page 88

categoryDetails

Indicates whether to group turnovers by transaction categories. Valid values are:

- **true**. Group turnovers by category.
- **false**. Do not group turnovers.

Default value: false

Optional element.

Type: boolean

attribute

Additional request parameters.

Optional element.

Type: *KeyValueAttribute Complex Type* on page 76

Output message: GetTurnoverListResponse

Response to the monthly turnover request.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="GetTurnoverListResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element default="true" minOccurs="0" name="groupByCurrency"
 type="xsd:boolean"/>
         <xsd:element maxOccurs="unbounded" minOccurs="0" name="turnover">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element minOccurs="0" name="date" type="xsd:date"/>
                  <xsd:element minOccurs="0" name="account" type="xsd:string"/>
                  <xsd:element name="currency" type="tns:Currency"/>
                  <xsd:element minOccurs="0" name="operationTypeCategory"
 type="tns:OperationTypeCategory"/>
```

```
                    <xsd:element name="incomeAmount" type="tns:Money"/>
                    <xsd:element name="incomeCommission" type="tns:Money"/>
                    <xsd:element name="incomeTransactionsCount" type="xsd:long"/>
                    <xsd:element minOccurs="0" name="incomeLockedBalance"
 type="tns:Money"/>
                    <xsd:element name="expenseAmount" type="tns:Money"/>
                    <xsd:element name="expensesIncludingCommission" type="tns:Money"/>
                    <xsd:element name="expensesExtraCommission" type="tns:Money"/>
                    <xsd:element name="expenseTransactionsCount" type="xsd:long"/>
                    <xsd:element minOccurs="0" name="expenseLockedBalance"
 type="tns:Money"/>
                    <xsd:element name="total" type="tns:Money"/>
                    <xsd:element name="openingBalance" type="tns:Money"/>
                    <xsd:element name="closingBalance" type="tns:Money"/>
                </xsd:sequence>
            </xsd:complexType>
         </xsd:element>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

groupByCurrency

Indicates whether to group turnovers by currency. Valid values are:

- **true**. Group by currency.
- **false**. Do not group by currency.

> **Note:** If MONETA.RU returns more than 1000 results, the response is grouped by currency even if you set groupByCurrency to false in the request.

Optional element.

Type: boolean

Default value: true

turnover

Optional element.

date

Date.

Optional element.

Type: date

account

Account.

Optional element.

Type: string

currency

Currency.

Type: *Currency Simple Type* on page 66

operationTypeCategory

Transaction category (if categoryDetails=true in the request).

Optional element.

Type: *OperationTypeCategory Simple Type* on page 88

incomeAmount

Amount of income.

Type: *Money Simple Type* on page 77

incomeCommission

Income commission fee.

Type: *Money Simple Type* on page 77

incomeTransactionsCount

Income. Number of transactions.

Type: long

incomeLockedBalance

Income. Locked balance.

Optional element.

Type: *Money Simple Type* on page 77

expenseAmount

Amount of expenses.

Type: *Money Simple Type* on page 77

expensesIncludingCommission

Amount of expenses including the commission fee.

Type: *Money Simple Type* on page 77

expensesExtraCommission

Expenses. Extra commission fee.

Type: *Money Simple Type* on page 77

expenseTransactionsCount

Expenses. Number of transactions.

Type: long

expenseLockedBalance

Expenses. Locked balance.

Optional element.

Type: *Money Simple Type* on page 77

total

Total.

Type: *Money Simple Type* on page 77

openingBalance

Opening balance.

Type: *Money Simple Type* on page 77

closingBalance

Closing balance.

Type: *Money Simple Type* on page 77

GetFinancialFlowsList Endpoint

Input message: GetFinancialFlowsListRequest

Request for getting the financial flow information.

If the response contains no data, the size element of the response is set to 0.

Type: *Entity Complex Type* on page 69

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="GetFinancialFlowsListRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:Entity">
            <xsd:sequence>
               <xsd:element minOccurs="0" name="unitId" type="xsd:long"/>
               <xsd:element name="dateFrom" type="xsd:date"/>
               <xsd:element minOccurs="0" name="dateTo" type="xsd:date"/>
               <xsd:element maxOccurs="unbounded" minOccurs="0" name="accountType"
 type="tns:AccountType"/>
               <xsd:element maxOccurs="unbounded" minOccurs="0" name="accountIds"
 type="tns:AccountId"/>
               <xsd:element minOccurs="0" name="currencyCode" type="tns:Currency"/>
               <xsd:element minOccurs="0" name="operationTypeCategory"
 type="tns:OperationTypeCategory"/>
               <xsd:element minOccurs="0" name="operationAmountType"
 type="tns:OperationAmountType"/>
               <xsd:element minOccurs="0" name="categoryDetails" type="xsd:boolean"/>
               <xsd:element maxOccurs="unbounded" minOccurs="0" name="attribute"
                            type="tns:KeyValueAttribute"/>
               <xsd:element minOccurs="0" name="operationStatusState"
 type="tns:OperationStatusState"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Description

unitId

Unique identifier of a user in MONETA.RU.

If you omit this element, MONETA.RU sets the value of this element to the user who sends the request

Optional element.

Type: long

dateFrom

The start date of the financial flow.

Type: date

dateTo

The end date of the financial flow. If you omit the dateTo element, MONETA.RU uses the last day of the month in the dateFrom element. Maximum period of the financial flow is limited to three months.

Optional element.

Type: date

accountType

Account type (example: Advanced account, Bank account).

Optional element.

Type: *AccountType Simple Type* on page 58

accountIds

The list of account numbers to include into the financial flow.

Optional element.

Type: *AccountId Simple Type* on page 51

currencyCode

Currency that is used in the financial flow.

Optional element.

Type: *Currency Simple Type* on page 66

operationTypeCategory

Transaction category to include into the financial flow.

Optional element.

Type: *OperationTypeCategory Simple Type* on page 88

operationAmountType

Indicates whether to include only debit or credit transactions. If you omit this element, MONETA.RU includes both debit and credit transactions.

Optional element.

Type: *OperationAmountType Simple Type* on page 77

categoryDetails

Indicates whether the financial flow is grouped by transaction categories. Valid values are:

- **true**. Group by transaction categories.
- **false**. Do not group by transaction categories.

Default value: false

Optional element.

Type: boolean

attribute

Additional request parameters.

Optional element.

Type: *KeyValueAttribute Complex Type* on page 76

operationStatusState

The status of transactions to include into the financial flow. If you omit this element, all statuses are included.

Optional element.

Type: *OperationStatusState Simple Type* on page 81

Output message: GetFinancialFlowsListResponse

Response to the financial flow request.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="GetFinancialFlowsListResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element maxOccurs="unbounded" minOccurs="0" name="financialFlow">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element minOccurs="0" name="date" type="xsd:string"/>
                  <xsd:element name="payerSideAccess" type="xsd:boolean"/>
                  <xsd:element name="payerFlowId" type="xsd:long"/>
                  <xsd:element name="payerFlowName" type="xsd:string"/>
```

```
                    <xsd:element name="payerCurrencyCode" type="tns:Currency"/>
                    <xsd:element name="payeeSideAccess" type="xsd:boolean"/>
                    <xsd:element name="payeeFlowId" type="xsd:long"/>
                    <xsd:element name="payeeFlowName" type="xsd:string"/>
                    <xsd:element name="payeeCurrencyCode" type="tns:Currency"/>
                    <xsd:element minOccurs="0" name="payerDebited" type="tns:Money"/>
                    <xsd:element minOccurs="0" name="payerWithheld" type="tns:Money"/>
                    <xsd:element minOccurs="0" name="payerFee" type="tns:Money"/>
                    <xsd:element minOccurs="0" name="payerExchangeFee" type="tns:Money"/>
                    <xsd:element minOccurs="0" name="payeeCredited" type="tns:Money"/>
                    <xsd:element minOccurs="0" name="payeeFee" type="tns:Money"/>
                    <xsd:element name="transactionsCount" type="xsd:long"/>
                    <xsd:element minOccurs="0" name="operationTypeCategory"
 type="tns:OperationTypeCategory"/>
                    <xsd:element name="operationStatusState"
 type="tns:OperationStatusState"/>
                    <xsd:element minOccurs="0" name="reversal" type="xsd:boolean"/>
                    <xsd:element maxOccurs="unbounded" minOccurs="0" name="attribute"
                                 type="tns:KeyValueAttribute"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Description

financialFlow

Optional element.

date

Financial flow date in the following format: yyyy-mm

Optional element.

Type: string

payerSideAccess

Indicates whether the user has access to the payer's financial flow information.

Valid values are:

- **true**. The user has access to the payer's financial flow information.
- **false**. The user does not have access to the payer's financial flow information.

Type: boolean

payerFlowId

Financial flow ID of the payer.

Type: long

payerFlowName

The name of the payer's financial flow.

Type: string

payerCurrencyCode

The currency of the payer's financial flow.

Type: *Currency Simple Type* on page 66

payeeSideAccess

Indicates whether the user has access to the payee's financial flow.

Type: boolean

payeeFlowId

Payee's financial flow ID.

Type: long

payeeFlowName

Payee's financial flow name.

Type: string

payeeCurrencyCode

The currency of the payee's financial flow.

Type: *Currency Simple Type* on page 66

payerDebited

The transaction amount that was debited from the payer's account.

Optional element.

Type: *Money Simple Type* on page 77

payerWithheld

Withheld amount from the payer's account.

Optional element.

Type: *Money Simple Type* on page 77

payerFee

Payer's transaction fee.

Optional element.

Type: *Money Simple Type* on page 77

payerExchangeFee

Exchange fee for the payer.

Optional element.

Type: *Money Simple Type* on page 77

payeeCredited

Transaction amount that is transferred to the payee's account.

Optional element.

Type: *Money Simple Type* on page 77

payeeFee

Payee's transaction fee.

Optional element.

Type: *Money Simple Type* on page 77

transactionsCount

The number of transactions in a financial flow.

Type: long

operationTypeCategory

If the request set the categoryDetails element to true, specifies transaction categories that are included into this response.

Optional element.

Type: *OperationTypeCategory Simple Type* on page 88

operationStatusState

Statuses of transactions that are included into the financial flow.

Type: *OperationStatusState Simple Type* on page 81

reversal

- **false** - Normal postings.
- **true** - Reversal postings.

MONETA.RU returns this element only if you set the version attribute of your request to **VERSION_2** or higher.

Optional element.

Type: boolean

attribute

Additional response parameters.

MONETA.RU returns this element only if you set the version attribute of your request to **VERSION_2** or higher.

Optional element.

Type: *KeyValueAttribute Complex Type* on page 76

AccountStatement Endpoint

Input message: AccountStatementRequest

Request for getting an account statement.

Type: *Entity Complex Type* on page 69

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="AccountStatementRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:Entity">
            <xsd:sequence>
               <xsd:element name="accountId" type="tns:AccountId"/>
               <xsd:element minOccurs="0" name="accountRS">
                  <xsd:simpleType>
                     <xsd:restriction base="xsd:string">
                        <xsd:minLength value="20"/>
                        <xsd:maxLength value="20"/>
                     </xsd:restriction>
                  </xsd:simpleType>
               </xsd:element>
               <xsd:choice>
                  <xsd:sequence>
                     <xsd:element name="dateFrom" type="xsd:date"/>
                     <xsd:element name="dateTo" type="xsd:date"/>
                  </xsd:sequence>
                  <xsd:element name="selectPeriod">
                     <xsd:simpleType>
                        <xsd:restriction base="xsd:string">
                           <xsd:enumeration value="TODAY"/>
                           <xsd:enumeration value="YESTERDAY"/>
                           <xsd:enumeration value="CURRENT_WEEK"/>
                           <xsd:enumeration value="CURRENT_MONTH"/>
                           <xsd:enumeration value="PREVIOUS_MONTH"/>
                        </xsd:restriction>
                     </xsd:simpleType>
                  </xsd:element>
               </xsd:choice>
               <xsd:element default="false" minOccurs="0" name="searchInArchive"
 type="xsd:boolean"/>
```

```
                <xsd:element default="false" minOccurs="0" name="showActions"
 type="xsd:boolean"/>
                <xsd:element minOccurs="0" name="operationPropertyNames"
 type="xsd:string"/>
                <xsd:element default="25" name="pageSize" type="xsd:int"/>
                <xsd:element default="false" minOccurs="0" name="showPaging"
 type="xsd:boolean"/>
                <xsd:element minOccurs="0" name="paging"
 type="tns:AccountStatementPaging"/>
            </xsd:sequence>
          </xsd:extension>
       </xsd:complexContent>
    </xsd:complexType>
</xsd:element>
```

Description

accountId

Account number in MONETA.RU.

Type: *AccountId Simple Type* on page 51

accountRS

Account number (20 digits).

Optional element.

Type: string

Maximum length: 20

Minimum length: 20

selectPeriod

Interval for data sampling.

Type: string

searchInArchive

Indicates whether to search for archived transactions.

Optional element.

Type: boolean

Default value: false

showActions

Return *actions* property in response. By default is *false* - dont't return.

Optional element.

Type: boolean

Default value: false

operationPropertyNames

List of comma separated transaction properties which will be returned in a response.

For example, WIREKPP, WIRETRANSFERORDERNUMBER, WIRETRANSFERORDERDATE, WIREOKTMO

Optional element.

Type: string

pageSize

The number of elements to return per page.

Type: int

Default value: 25

showPaging

Return *paging* property in response. By default is *false* - dont't return.

Optional element.

Type: boolean

Default value: false

paging

For receiving the next page with operations you should send data from field *paging* in response *AccountStatementResponse*.

If you send field *paging* in the request, you will receive only information about operations (field *orders*) without common Account statement information.

Optional element.

Type: *AccountStatementPaging Complex Type* on page 55

dateFrom

The start date of the period.

Type: date

dateTo

The end date of the period.

Type: date

Output message: AccountStatementResponse

Response to accountStatementRequest. The response contains account statement.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="AccountStatementResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element minOccurs="0" name="ownerAccount" type="xsd:string"/>
         <xsd:element minOccurs="0" name="contractDate" type="xsd:dateTime"/>
         <xsd:element minOccurs="0" name="ownerAccountCurrency" type="xsd:string"/>
         <xsd:element minOccurs="0" name="ownerName" type="xsd:string"/>
         <xsd:element minOccurs="0" name="dateFrom" type="xsd:dateTime"/>
         <xsd:element minOccurs="0" name="dateTo" type="xsd:dateTime"/>
         <xsd:element minOccurs="0" name="lastOperationDate" type="xsd:dateTime"/>
         <xsd:element minOccurs="0" name="now" type="xsd:dateTime"/>
         <xsd:element minOccurs="0" name="inSaldo" type="tns:Money"/>
         <xsd:element minOccurs="0" name="outSaldo" type="tns:Money"/>
         <xsd:element minOccurs="0" name="inSaldoBlocked" type="tns:Money"/>
         <xsd:element minOccurs="0" name="outSaldoBlocked" type="tns:Money"/>
         <xsd:element minOccurs="0" name="debitAmountTotal" type="tns:Money"/>
         <xsd:element minOccurs="0" name="creditAmountTotal" type="tns:Money"/>
         <xsd:element maxOccurs="unbounded" minOccurs="0" name="attribute"
                      type="tns:KeyValueAttribute"/>
         <xsd:element minOccurs="0" name="totalSize" type="xsd:long"/>
         <xsd:element maxOccurs="unbounded" minOccurs="0" name="orders"
                      type="tns:AccountStatementRecordType"/>
         <xsd:element minOccurs="0" name="paging" type="tns:AccountStatementPaging"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

ownerAccount

Owner account.

Optional element.

Type: string

contractDate

Contract date.

Optional element.

Type: dateTime

ownerAccountCurrency

Account currency.

Optional element.

Type: string

ownerName

Account owner.

Optional element.

Type: string

dateFrom

Date from.

Optional element.

Type: dateTime

dateTo

Date to.

Optional element.

Type: dateTime

lastOperationDate

Date of last operation. This property is always empty. It is not used.

Optional element.

Type: dateTime

now

Relevance date.

Optional element.

Type: dateTime

inSaldo

Opening balance.

Optional element.

Type: *Money Simple Type* on page 77

outSaldo

Closing balance.

Optional element.

Type: *Money Simple Type* on page 77

inSaldoBlocked

Blocked opening balance.

Optional element.

Type: *Money Simple Type* on page 77

outSaldoBlocked

Blocked closing balance.

Optional element.

Type: *Money Simple Type* on page 77

debitAmountTotal

Total turnover on the debit.

Optional element.

Type: *Money Simple Type* on page 77

creditAmountTotal

Total turnover on the debit.

Optional element.

Type: *Money Simple Type* on page 77

attribute

Additional account statement attributes.

> **Important:** MONETA.RU returns this element only if you set the version attribute of your request to**VERSION_2** or higher.

Valid values for the key and value elements:

- **partner_agreement_payee_offer**

Optional element.

Type: *KeyValueAttribute Complex Type* on page 76

totalSize

The total number of records in the response.

Optional element.

Type: long

orders

Information about operations.

Optional element.

Type: *AccountStatementRecordType Complex Type* on page 55

paging

You can receive this field if set in request: **showPaging=true**.

You can send this data in the next request *AccountStatementRequest* for receiving next page with operations (field *orders*).

Optional element.

Type: *AccountStatementPaging Complex Type* on page 55

FindReports Endpoint

Input message: FindReportsRequest

Request for reports by a user ID. MONETA.RU returns an empty list if it cannot find reports.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="FindReportsRequest">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element minOccurs="0" name="unitId" type="xsd:long"/>
```

```
         <xsd:element minOccurs="0" name="reportId" type="xsd:long"/>
         <xsd:element minOccurs="0" name="year" type="xsd:int"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

unitId

User ID. If you omit this element, MONETA.RU uses the ID of the user who sends the request.

Optional element.

Type: long

reportId

Report ID.

Optional element.

Type: long

year

Report instances year.

Optional element.

Type: int

Output message: FindReportsResponse

Response to FindReportsRequest.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="FindReportsResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element maxOccurs="unbounded" minOccurs="0" name="report"
 type="tns:Report"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

report

Reports list.

Optional element.

Type: *Report Complex Type* on page 100

Verifying User Identity

According to the provision of Central Bank of Russia, all credit organizations, including MONETA.RU, must identify their users. This rule applies only to residents of Russia.

If you are a citizen of Russia, you must complete the simplified user identification procedure to raise the maximum sum of money that you can transfer or withdraw from MONETA.RU. Only authorized identification centers can perform the user identification procedure.

Authorized identification centers can use MONETA.RU Merchant API to complete a simplified user identification procedure.

1. Before users start the identification procedure, they must add the following information to their MONETA.RU profiles:

- First name
- Middle name
- Last name
- Date of birth
- Legal address
- INN (optional)
- Verified phone number
- Passport information

2. To get a unique identification code, a user can either go to *https://www.moneta.ru/personificationCode.htm* or an authorized identification center can use GetPersonificationCodeRequest to generate a unique identification code for a user.

   > **Note:** The code is valid for 7 days. If the user does not complete the identification procedure before the code expires, the user must get a new code.

3. An authorized identification center uses the identification code in the VerifyPersonificationCodeRequest to get information about the user from MONETA.RU. If the user did not provide the required information in the profile, MONETA.RU returns an error.
4. After the authorized identification center verifies that the provided user information is correct, the center uses ConfirmPersonificationRequest to complete the user identification procedure.
5. MONETA.RU notifies the authorized identification center whether the identification procedure was successful.

GetPersonificationCode Endpoint

Input message: GetPersonificationCodeRequest

Request for a user identification code.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="GetPersonificationCodeRequest">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:choice minOccurs="0">
            <xsd:element name="accountId" type="tns:AccountId"/>
            <xsd:element name="unitId" type="xsd:long"/>
         </xsd:choice>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

accountId

MONETA.RU account number.

Type: *AccountId Simple Type* on page 51

unitId

Unique identifier of the user.

If you omit this element, MONETA.RU uses the ID of the user who sends the request.

Type: long

Output message: GetPersonificationCodeResponse

Response to GetPersonificationCodeRequest.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

```
            name="GetPersonificationCodeResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="personificationCode" type="xsd:string"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

personificationCode

User identification code.

Type: string

VerifyPersonificationCode Endpoint

Input message: VerifyPersonificationCodeRequest

Request for verification of a user identification code.

Type: *Entity Complex Type* on page 69

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             name="VerifyPersonificationCodeRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:Entity">
            <xsd:sequence>
               <xsd:element name="personificationCode" type="xsd:string"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Description

personificationCode

User identification code.

Type: string

Output message: VerifyPersonificationCodeResponse

Response to VerifyPersonificationCodeRequest.

If the identification code is valid (that is MONETA.RU recognized the code and it is not expired), the response contains user information.

If the identification code is invalid, an error occurs.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             name="VerifyPersonificationCodeResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="personalInformation" type="tns:PersonalInformation"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

personalInformation

User information might include the following attributes:

- first_name
- middle_initial_name
- last_name
- date_of_birth

  Use the following format: yyyy-mm-dd
- legal_address
- inn
- snils

Type: *PersonalInformation Complex Type* on page 93

ConfirmPersonification Endpoint

Input message: ConfirmPersonificationRequest

Request for user identification.

Type: *Entity Complex Type* on page 69

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="ConfirmPersonificationRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:Entity">
            <xsd:sequence>
               <xsd:element name="personificationCode" type="xsd:string"/>
               <xsd:element name="personalInformation" type="tns:PersonalInformation"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Description

personificationCode

User identification code.

Type: string

personalInformation

User information must contain the following attributes:

- first_name
- middle_initial_name (optional)
- last_name
- date_of_birth (optional)

  Use the following format: yyyy-mm-dd
- legal_address (optional)
- place_of_birth (optional)
- inn (optional)
- snils (optional)

The document must contain the following attributes:

- type (PASSPORT)
- series
- number
- issuer (optional)

- issued (optional)
- department (optional)

Type: *PersonalInformation Complex Type* on page 93

Output message: ConfirmPersonificationResponse

Response to ConfirmPersonificationRequest.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             name="ConfirmPersonificationResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="personalInformation" type="tns:PersonalInformation"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

personalInformation

If user identification is successful, this element contains user information.

Type: *PersonalInformation Complex Type* on page 93

SimplifiedIdentification Endpoint

Input message: SimplifiedIdentificationRequest

Request for user simplified identification.

Type: *AbstractAttributeObject Complex Type* on page 50

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             name="SimplifiedIdentificationRequest">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:AbstractAttributeObject">
            <xsd:sequence>
               <xsd:element minOccurs="0" name="unitId" type="xsd:long"/>
               <xsd:element minOccurs="0" name="personalInformation"
 type="tns:PersonalInformation"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Description

unitId

User ID. If you omit this element, MONETA.RU uses the ID of the user who sends the request.

Optional element.

Type: long

personalInformation

User information must contain the following attributes:

- first_name
- middle_initial_name (optional)
- last_name

- cell_phone
- date_of_birth (optional)
- snils (optional)
- inn (optional)

The document must contain the following attributes:

- type (PASSPORT)
- series
- number
- issuer (optional)
- issued (optional)
- department (optional)

Optional element.

Type: *PersonalInformation Complex Type* on page 93

Output message: SimplifiedIdentificationResponse

Response to SimplifiedIdentificationRequest.

Type: *AbstractAttributeObject Complex Type* on page 50

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             name="SimplifiedIdentificationResponse">
   <xsd:complexType>
      <xsd:complexContent>
         <xsd:extension base="tns:AbstractAttributeObject">
            <xsd:sequence>
               <xsd:element name="success" type="xsd:boolean"/>
               <xsd:element minOccurs="0" name="error" type="xsd:string"/>
               <xsd:element minOccurs="0" name="errorCode" type="xsd:string"/>
               <xsd:element name="personalInformation" type="tns:PersonalInformation"/>
            </xsd:sequence>
         </xsd:extension>
      </xsd:complexContent>
   </xsd:complexType>
</xsd:element>
```

Description

success

- **true**. Simplified identification completed successfully.
- **false**. Simplified identification failed.

Type: boolean

error

If an error occurs when processing a simplified identification, the error element contains an error description.

If the simplified identification completes successfully, the error element is empty.

Optional element.

Type: string

errorCode

If an error occurs when processing a simplified identification, the errorCode element contains an error code.

See the faultDetail element for the error code descriptions.

Optional element.

Type: string

personalInformation

Current personal information.

User information contains the following attributes:

- first_name
- middle_initial_name (optional)
- last_name
- cell_phone
- date_of_birth (optional)
- snils (optional)
- inn (optional)

The document contains the following attributes:

- type (PASSPORT)
- series
- number
- issuer (optional)
- issued (optional)
- department (optional)

Type: *PersonalInformation Complex Type* on page 93

Verifying User Phone Numbers

ApprovePhoneApplyCode Endpoint

Input message: ApprovePhoneApplyCodeRequest

The second request for the confirmation of a user's phone number that is specified in the cell_phone element of tns:Profile.

After you send the ApprovePhoneSendConfirmation request, the user receives an SMS message with the verification code to the unconfirmed mobile phone number. Use the ApprovePhoneApplyCode request to send the verification code to MONETA.RU.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="ApprovePhoneApplyCodeRequest">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element default="true" minOccurs="0" name="approval" type="xsd:boolean"/>
         <xsd:element minOccurs="0" name="unitId" type="xsd:long"/>
         <xsd:element minOccurs="0" name="profileId" type="xsd:long"/>
         <xsd:element name="confirmationCode" type="xsd:string"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

approval

- **true**. Cell phone number confirmation.
- **false**. Cell phone number disapproval.

Default value is **true**.

Optional element.

Type: boolean

Default value: true

unitId

The unique identifier of the MONETA.RU user whose phone number must be confirmed. If you omit this element, MONETA.RU uses the ID of the user who sends the request.

Optional element.

Type: long

profileId

Optional element.

Type: long

confirmationCode

The confirmation code that the user received in the SMS message.

Type: string

Output message: ApprovePhoneApplyCodeResponse

Response to ApprovePhoneApplyCodeRequest.

If MONETA.RU confirms the phone number successfully, the response contains no data and the phone number gets the **approved** status. If MONETA.RU fails to confirm the phone number, the response contains an exception.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 name="ApprovePhoneApplyCodeResponse">
   <xsd:complexType/>
</xsd:element>
```

ApprovePhoneSendConfirmation Endpoint

Input message: ApprovePhoneSendConfirmationRequest

The first request for the confirmation of the user's phone number that is specified in the cell_phone element of tns:Profile.

Users must confirm their mobile phone numbers in the following cases:

- To receive payment passwords to their mobile phones.
- To complete the simplified identification procedure.

After you send the ApprovePhoneSendConfirmation request, MONETA.RU sends an SMS messages with a verification code to the specified mobile phone number. Use the verification code in the ApprovePhoneApplyCode request to confirm the phone number.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             name="ApprovePhoneSendConfirmationRequest">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element default="true" minOccurs="0" name="approval" type="xsd:boolean"/>
         <xsd:element minOccurs="0" name="unitId" type="xsd:long"/>
         <xsd:element minOccurs="0" name="profileId" type="xsd:long"/>
         <xsd:element minOccurs="0" name="text">
            <xsd:simpleType>
               <xsd:restriction base="xsd:string">
                  <xsd:maxLength value="160"/>
               </xsd:restriction>
```

```
            </xsd:simpleType>
         </xsd:element>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

approval

- **true**. Cell phone number confirmation.
- **false**. Cell phone number disapproval.

Default value is **true**.

Optional element.

Type: boolean

Default value: true

unitId

The unique identifier of the MONETA.RU user whose phone number must be confirmed. If you omit this element, MONETA.RU uses the ID of the user who sends the request.

Optional element.

Type: long

profileId

Optional element.

Type: long

text

Text of the SMS message to send to the user. The text must contain the {CODE} placeholder that MONETA.RU replaces with a real confirmation code.

> **Important:** The length of the message that contains non-ASCII characters must not exceed 70 symbols. The length of ASCII messages is limited to 160 symbols.

Default value for confirmation: `Confirmation code: {CODE}`

Default value for disapproval: `Disapproval code: {CODE}`

Optional element.

Type: string

Maximum length: 160

Output message: ApprovePhoneSendConfirmationResponse

Response to ApprovePhoneSendConfirmationRequest.

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             name="ApprovePhoneSendConfirmationResponse">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="phoneNumber" type="xsd:string"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

Description

phoneNumber

The phone number to which MONETA.RU sends the SMS message.

Type: string

Asynchronous requests

Async Endpoint

Input message: AsyncRequest

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="AsyncRequest">
   <xsd:complexType>
      <xsd:choice>
         <xsd:sequence>
            <xsd:choice>
               <xsd:element ref="tns:GetTurnoverListRequest"/>
               <xsd:element ref="tns:GetFinancialFlowsListRequest"/>
               <xsd:element ref="tns:SimplifiedIdentificationRequest"/>
               <xsd:element ref="tns:PingRequest"/>
               <xsd:element ref="tns:PaymentRequest"/>
               <xsd:element ref="tns:VerifyPaymentRequest"/>
               <xsd:element ref="tns:VerifyPaymentBatchRequest"/>
            </xsd:choice>
            <xsd:element minOccurs="0" name="callbackUrl" type="xsd:string"/>
         </xsd:sequence>
         <xsd:element name="asyncId" type="xsd:long"/>
      </xsd:choice>
   </xsd:complexType>
</xsd:element>
```

Description

callbackUrl

Optional element.

Type: string

Output message: AsyncResponse

XML Schema

```
<xsd:element xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="AsyncResponse">
   <xsd:complexType>
      <xsd:choice>
         <xsd:element ref="tns:GetTurnoverListResponse"/>
         <xsd:element ref="tns:GetFinancialFlowsListResponse"/>
         <xsd:element ref="tns:SimplifiedIdentificationResponse"/>
         <xsd:element ref="tns:PingResponse"/>
         <xsd:element ref="tns:PaymentResponse"/>
         <xsd:element ref="tns:VerifyPaymentResponse"/>
         <xsd:element ref="tns:VerifyPaymentBatchResponse"/>
         <xsd:sequence>
            <xsd:element name="asyncId" type="xsd:long"/>
            <xsd:element name="asyncStatus" type="tns:AsyncStatus"/>
            <xsd:element name="expirationDate" type="xsd:dateTime"/>
         </xsd:sequence>
      </xsd:choice>
   </xsd:complexType>
```

```
</xsd:element>
```

Chapter 6

Usage Examples

Topics:

Payments

Payments

Overview

You can process payments by using one of the following methods: Transfer or Payment. The Transfer method is deprecated. Unlike the deprecated Transfer method, the Payment method can take an operation ID as an argument if you want to continue processing a transaction. Also, the Payment method returns transaction details in the response. So you do not need to send the GetOperationDetailsById request after each transaction.

Before you initiate a transaction, you can optionally send the VerifyTransfer or VerifyPayment requests to ensure that the payment request is valid.

Payment Types

- Single-phase payments.
- Two-phase payments.
- Push payments.

Simple Payments

When processing a single-phase payment, the merchant accepts and captures a payment from a customer account by using a single Payment request.

Perform the following steps to process a single-phase payment:

1. Optionally, send a VerifyPayment request to ensure that the payment request is valid.

   For more information about the VerifyPayment endpoint, see *VerifyPayment Endpoint* on page 127.

   The following sample VerifyPayment request verifies whether a payment of 100 RUB from a customer's MONETA.RU account 12345678 to the merchant account 87654321 is valid:

```
<VerifyPaymentRequest version="VERSION_2">
  <payer>12345678</payer>
  <payee>87654321</payee>
  <amount>100</amount>
  <isPayerAmount>true</isPayerAmount>
  <paymentPassword>*****</paymentPassword>
</VerifyPaymentRequest>
```

   The `<isPayerAmount>true</isPayerAmount>` element in the request indicates that the payment amount is specified in the currency of the payer account.

   > **Important:** If the customer might receive payment passwords by SMS, set the version attribute of the request to VERSION_2. In this case, if the customer does use SMS passwords, the response includes the `<paymentPasswordChallengeRequired>true</paymentPasswordChallengeRequired>` element.

   The following sample XML response is returned if the payment is valid:

```
<VerifyPaymentResponse>
  <isTransactionValid>true</isTransactionValid>
  <forecast>
    <payer>12345678</payer>
    <payerCurrency>RUB</payerCurrency>
    <payerAmount>100</payerAmount>
    <payee>87654321</payee>
    <payeeCurrency>RUB</payeeCurrency>
    <payeeAmount>100</payeeAmount>
    <payeeFee>0</payeeFee>
    <payerAlias>RUB</payerAlias>
    <payeeAlias>Test Store</payeeAlias>
  </forecast>
</VerifyPaymentResponse>
```

   JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "VerifyPaymentRequest": {
            "version": "VERSION_2",
            "payer": "12345678",
            "payee": "87654321",
            "amount": 100,
            "isPayerAmount": true,
            "paymentPassword": "*****",
            "clientTransaction": "Merchant transaction Id"
        }
    }
}}
```

JSON response:

```
{"Envelope": {
    "Body": {
        "VerifyPaymentResponse": {
            "isTransactionValid": true,
            "forecast": {
                "payer": "12345678",
                "payerCurrency": "RUB",
                "payerAmount": 100,
                "payerFee": 0,
                "payee": "87654321",
                "payeeAmount": 100,
                "payeeCurrency": "RUB"
            },
            "operationInfo": {
                "attribute": [
                    {
                        "value": "true",
                        "key": "paymentPasswordChallengeRequired"
                    }
                ]
            }
        }
    }
}}
```

2. If the customer receives payment passwords by SMS, send the GetAccountPaymentPasswordChallenge request to obtain the payment password.

   For more information about the GetAccountPaymentPasswordChallenge endpoint, see *GetAccountPaymentPasswordChallenge Endpoint* on page 119

```
<GetAccountPaymentPasswordChallengeRequest>
  <accountId>12345678</accountId>
</GetAccountPaymentPasswordChallengeRequest>
```

   JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "GetAccountPaymentPasswordChallengeRequest": {
            "accountId": 12345678
        }
    }
}}
```

3. Send the Payment request.

   For more information about the Payment endpoint, see *Payment Endpoint* on page 121

   The following Payment request processes the payment from the customer account 12345678 to the merchant account 87654321:

```
<PaymentRequest>
  <payer>12345678</payer>
  <payee>87654321</payee>
  <amount>100</amount>
  <isPayerAmount>true</isPayerAmount>
  <paymentPassword>*****</paymentPassword>
  <clientTransaction>777777</clientTransaction>
```

```
</PaymentRequest>
```

The following sample XML response is returned if the payment is successful:

```
<PaymentResponse>
  <id>328268</id>
  <attribute>
    <key>category</key>
    <value>BUSINESS</value>
  </attribute>
  <attribute>
    <key>sourcecurrencycode</key>
    <value>RUB</value>
  </attribute>
  <attribute>
    <key>sourceamount</key>
    <value>100</value>
  </attribute>
  <attribute>
    <key>typeid</key>
    <value>10</value>
  </attribute>
  <attribute>
    <key>targetamount</key>
    <value>-100</value>
  </attribute>
  <attribute>
    <key>sourceamountfee</key>
    <value>0</value>
  </attribute>
  <attribute>
    <key>targetcurrencycode</key>
    <value>RUB</value>
  </attribute>
  <attribute>
    <key>statusid</key>
    <value>SUCCEED</value>
  </attribute>
  <attribute>
    <key>sourceamounttotal</key>
    <value>100</value>
  </attribute>
  <attribute>
    <key>modified</key>
    <value>2014-12-17T01:25:22.000+03:00</value>
  </attribute>
  <attribute>
    <key>clienttransaction</key>
    <value>777777</value>
  </attribute>
  <attribute>
    <key>targetaccountid</key>
    <value>12345678</value>
  </attribute>
  <attribute>
    <key>sourceaccountid</key>
    <value>87654321</value>
  </attribute>
  <attribute>
    <key>isreversed</key>
    <value>true</value>
  </attribute>
</PaymentResponse>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
```

```
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "PaymentRequest": {
            "payer": "12345678",
            "payee": "87654321",
            "amount": 100,
            "isPayerAmount": true,
            "paymentPassword": "*****",
            "clientTransaction": "Merchant transaction id"
        }
    }
}}
```

JSON response:

```
{"Envelope": {
    "Body": {
        "PaymentResponse": {
            "id": 328268,
            "attribute": [
...
                {
                    "value": "SUCCEED",
                    "key": "statusid"
                }
...
            ]
        }
    }
}}
```

Push Payments

Push payments are similar to simple payments. However, push payments can be processed by a merchant account that is different from the merchant account to which a customer transfers the funds. For example, a terminal can use this payment type to top up customer accounts in third-party services that use MONETA.RU.

To identify a customer account to which the funds are transferred, the Payment request must include the subscriberid element. This value must be a unique identifier of a customer in the credited system.

Perform the following steps to process a push payment:

1. Optionally, send a VerifyPayment request to ensure that the payment request is valid.

   For more information about the VerifyPayment endpoint, see *VerifyPayment Endpoint* on page 127.

   The following sample VerifyPayment request verifies whether a payment of 100 RUB from the merchant account 12345678 to the merchant account 87654321 is valid.

```
<VerifyPaymentRequest>
  <payer>12345678</payer>
  <payee>87654321</payee>
  <amount>100</amount>
  <isPayerAmount>true</isPayerAmount>
  <paymentPassword>*****</paymentPassword>
  <operationInfo>
    <attribute>
      <key>subscriberid</key>
      <value>007</value>
    </attribute>
  </operationInfo>
</VerifyPaymentRequest>
```

   The `<isPayerAmount>true</isPayerAmount>` element in the request indicates that the payment amount is specified in the currency of the payer account.

The following sample XML response is returned if the payment is valid:

```
<VerifyPaymentResponse>
  <isTransactionValid>true</isTransactionValid>
  <forecast>
    <payer>12345678</payer>
    <payerCurrency>RUB</payerCurrency>
    <payerAmount>100</payerAmount>
    <payee>87654321</payee>
    <payeeCurrency>RUB</payeeCurrency>
    <payeeAmount>100</payeeAmount>
    <payeeFee>0</payeeFee>
    <payerAlias>RUB</payerAlias>
    <payeeAlias>Test Store</payeeAlias>
  </forecast>
</VerifyPaymentResponse>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "VerifyPaymentRequest": {
            "version": "VERSION_2",
            "payer": "12345678",
            "payee": "87654321",
            "amount": 100,
            "isPayerAmount": true,
            "paymentPassword": "*****",
            "clientTransaction": "Merchant transaction Id",
            "operationInfo": {
                "attribute": [
                    {
                        "key": "subscriberid",
                        "value": "007"
                    }
                ]
            }
        }
    }
}}
```

JSON response:

```
{"Envelope": {
    "Body": {
        "VerifyPaymentResponse": {
            "isTransactionValid": true,
            "forecast": {
                "payer": "12345678",
                "payerCurrency": "RUB",
                "payerAmount": 100,
                "payerFee": 0,
                "payee": "87654321",
                "payeeAmount": 100,
                "payeeCurrency": "RUB"
            }
        }
    }
}}
```

2. Send the Payment request.

For more information about the Payment endpoint, see *Payment Endpoint* on page 121

The following Payment request processes the payment from the merchant account 12345678 to the merchant account 87654321:

```
<PaymentRequest>
  <payer>12345678</payer>
  <payee>87654321</payee>
  <amount>100</amount>
  <isPayerAmount>true</isPayerAmount>
  <paymentPassword>********</paymentPassword>
  <operationInfo>
    <attribute>
      <key>subscriberid</key>
      <value>007</value>
    </attribute>
  </operationInfo>
</PaymentRequest>
```

The following sample XML response is returned if the payment is successful:

```
<PaymentResponse>
  <id>333855</id>
  <attribute>
    <key>category</key>
    <value>BUSINESS</value>
  </attribute>
  <attribute>
    <key>sourcecurrencycode</key>
    <value>RUB</value>
  </attribute>
  <attribute>
    <key>sourceamount</key>
    <value>100</value>
  </attribute>
  <attribute>
    <key>typeid</key>
    <value>10</value>
  </attribute>
  <attribute>
    <key>targetamount</key>
    <value>-100</value>
  </attribute>
  <attribute>
    <key>subscriberid</key>
    <value>007</value>
  </attribute>
  <attribute>
    <key>sourceamountfee</key>
    <value>0</value>
  </attribute>
  <attribute>
    <key>targetcurrencycode</key>
    <value>RUB</value>
  </attribute>
  <attribute>
    <key>statusid</key>
    <value>SUCCEED</value>
  </attribute>
  <attribute>
    <key>sourceamounttotal</key>
    <value>100</value>
  </attribute>
  <attribute>
    <key>modified</key>
    <value>2015-01-31T17:15:46.000+03:00</value>
  </attribute>
  <attribute>
    <key>targetaccountid</key>
    <value>12345678</value>
```

```
  </attribute>
  <attribute>
    <key>sourceaccountid</key>
    <value>87654321</value>
  </attribute>
  <attribute>
    <key>isreversed</key>
    <value>true</value>
  </attribute>
</PaymentResponse>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "PaymentRequest": {
            "payer": "12345678",
            "payee": "87654321",
            "amount": 100,
            "isPayerAmount": true,
            "paymentPassword": "*****",
            "clientTransaction": "Merchant transaction id",
            "operationInfo": {
                "attribute": [
                    {
                        "key": "subscriberid",
                        "value": "007"
                    }
                ]
            }
        }
    }
}}
```

JSON response:

```
{"Envelope": {
    "Body": {
        "PaymentResponse": {
            "id": 328268,
            "attribute": [
...
                {
                    "value": "SUCCEED",
                    "key": "statusid"
                },
                {
                    "key": "subscriberid",
                    "value": "007"
                }
...
            ]
        }
    }
}}
```

Two-Phase Payments

The merchant can process a payment in two stages. First, the merchant puts an authorization hold on the payment amount in a customer's account. Later, the merchant captures the payment amount.

Phase One

1. Optionally, send a VerifyPayment request to ensure that the payment request is valid.

   For more information about the VerifyPayment endpoint, see *VerifyPayment Endpoint* on page 127.

   The following sample VerifyPayment request verifies whether a payment of 100 RUB from a customer's MONETA.RU account 12345678 to the merchant account 87654321 is valid.

```
<VerifyPaymentRequest version="VERSION_2">
  <payer>12345678</payer>
  <payee>87654321</payee>
  <amount>100</amount>
  <isPayerAmount>true</isPayerAmount>
  <paymentPassword>*****</paymentPassword>
</VerifyPaymentRequest>
```

   The `<isPayerAmount>true</isPayerAmount>` element in the request indicates that the payment amount is specified in the currency of the payer account.

   > **Important:** If the customer might receive payment passwords by SMS, set the version attribute of the request to VERSION_2. In this case, if the customer does use SMS passwords, the response includes the `<paymentPasswordChallengeRequired>true</paymentPasswordChallengeRequired>` element.

   The following sample XML response is returned if the payment is valid:

```
<VerifyPaymentResponse>
  <isTransactionValid>true</isTransactionValid>
  <forecast>
    <payer>12345678</payer>
    <payerCurrency>RUB</payerCurrency>
    <payerAmount>100</payerAmount>
    <payee>87654321</payee>
    <payeeCurrency>RUB</payeeCurrency>
    <payeeAmount>100</payeeAmount>
    <payeeFee>0</payeeFee>
    <payerAlias>RUB</payerAlias>
    <payeeAlias>Test Store</payeeAlias>
  </forecast>
</VerifyPaymentResponse>
```

   JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "VerifyPaymentRequest": {
            "version": "VERSION_2",
            "payer": "12345678",
            "payee": "87654321",
            "amount": 100,
            "isPayerAmount": true,
            "paymentPassword": "*****",
            "clientTransaction": "Merchant transaction Id"
        }
    }
}}
```

   JSON response:

```
{"Envelope": {
```

```
    "Body": {
        "VerifyPaymentResponse": {
            "isTransactionValid": true,
            "forecast": {
                "payer": "12345678",
                "payerCurrency": "RUB",
                "payerAmount": 100,
                "payerFee": 0,
                "payee": "87654321",
                "payeeAmount": 100,
                "payeeCurrency": "RUB"
            },
            "operationInfo": {
                "attribute": [
                    {
                        "value": "true",
                        "key": "paymentPasswordChallengeRequired"
                    }
                ]
            }
        }
    }
}}
```

2. If the customer receives payment passwords by SMS, send the GetAccountPaymentPasswordChallenge request to obtain the payment password.

   For more information about the GetAccountPaymentPasswordChallenge endpoint, see *GetAccountPaymentPasswordChallenge Endpoint* on page 119

```
<GetAccountPaymentPasswordChallengeRequest>
  <accountId>12345678</accountId>
</GetAccountPaymentPasswordChallengeRequest>
```

   JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "GetAccountPaymentPasswordChallengeRequest": {
            "accountId": 12345678
        }
    }
}}
```

3. Send the AuthoriseTransaction request to authorize the funds in the customer account.

   For more information about the AuthoriseTransaction endpoint, see *AuthoriseTransaction Endpoint* on page 108.

   The following sample AuthoriseTransaction request authorizes 100 RUB in the customer MONETA.RU account 12345678 for the merchant account 87654321.

```
<AuthoriseTransactionRequest>
  <payer>12345678</payer>
  <payee>87654321</payee>
  <amount>100</amount>
  <isPayerAmount>true</isPayerAmount>
  <paymentPassword>*****</paymentPassword>
  <clientTransaction>000002</clientTransaction>
  <description>Sample two-phase transaction</description>
</AuthoriseTransactionRequest>
```

The following sample XML response is returned if authorization is successful:

```
<AuthoriseTransactionResponse>
  <id>328805</id>
  <attribute>
    <key>category</key>
    <value>BUSINESS</value>
  </attribute>
  <attribute>
    <key>sourcecurrencycode</key>
    <value>RUB</value>
  </attribute>
  <attribute>
    <key>sourceamount</key>
    <value>100</value>
  </attribute>
  <attribute>
    <key>description</key>
    <value>Sample two-phase transaction</value>
  </attribute>
  <attribute>
    <key>typeid</key>
    <value>10</value>
  </attribute>
  <attribute>
    <key>targetamount</key>
    <value>-100</value>
  </attribute>
  <attribute>
    <key>protectioncode</key>
    <value>18721</value>
  </attribute>
  <attribute>
    <key>targetcurrencycode</key>
    <value>RUB</value>
  </attribute>
  <attribute>
    <key>statusid</key>
    <value>INPROGRESS</value>
  </attribute>
  <attribute>
    <key>sourceamounttotal</key>
    <value>100</value>
  </attribute>
  <attribute>
    <key>modified</key>
    <value>2014-12-22T23:23:20.000+03:00</value>
  </attribute>
  <attribute>
    <key>clienttransaction</key>
    <value>000002</value>
  </attribute>
  <attribute>
    <key>targetaccountid</key>
    <value>12345678</value>
  </attribute>
  <attribute>
    <key>protectioncodeexpirationdate</key>
    <value>2014-12-23T23:23:19.000+03:00</value>
  </attribute>
  <attribute>
    <key>sourceaccountid</key>
    <value>87654321</value>
  </attribute>
  <attribute>
    <key>isreversed</key>
    <value>true</value>
  </attribute>
</AuthoriseTransactionResponse>
```

Important: By default, the transaction is valid for 24 hours. If you do not complete the second phase of the transaction within this period, the transaction is canceled. You can include the protectioncodeexpirationdate element into the request to override the default expiration date. You can use the protectioncode attribute to specify a code for the ConfirmTransaction request.

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "AuthoriseTransactionRequest": {
            "payer": "12345678",
            "payee": "87654321",
            "amount": 100,
            "isPayerAmount": true,
            "paymentPassword": "******",
            "clientTransaction": "Merchant transaction Id",
            "description": "Sample two-phase transaction",
            "operationInfo": {
                "attribute": [
                    {
                        "key": "protectioncodeexpirationdate",
                        "value": "29.09.2015 15:25:07"
                    },
                    {
                        "key": "protectioncode",
                        "value": "12345"
                    }

                ]
            }
        }
    }
}}
```

JSON response:

```
{"Envelope": {
    "Body": {
        "AuthoriseTransactionResponse": {
            "id": 328805,
            "attribute": [
...
                {
                    "value": "INPROGRESS",
                    "key": "statusid"
                },
                {
                    "value": "12345",
                    "key": "protectioncode"
                },
                {
                    "value": "2015-09-29T15:25:07.000+03:00",
                    "key": "protectioncodeexpirationdate"
                }
...
            ]
        }
    }
}}
```

Phase Two

1. Send the ConfirmTransaction request to capture the funds.

   For more information about the ConfirmTransaction endpoint, see *ConfirmTransaction Endpoint* on page 110.

   The following sample ConfirmTransaction request captures the funds for the transaction ID of 328805 from the customer MONETA.RU account of 12345678.

```
<ConfirmTransactionRequest>
  <transactionId>328805</transactionId>
  <protectionCode>18721</protectionCode>
</ConfirmTransactionRequest>
```

   The following sample XML response is returned if the funds were captured:

```
<ConfirmTransactionResponse>
  <id>328805</id>
  <attribute>
    <key>category</key>
    <value>BUSINESS</value>
  </attribute>
  <attribute>
    <key>sourcecurrencycode</key>
    <value>RUB</value>
  </attribute>
  <attribute>
    <key>sourceamount</key>
    <value>100</value>
  </attribute>
  <attribute>
    <key>description</key>
    <value>Sample two-phase transaction</value>
  </attribute>
  <attribute>
    <key>typeid</key>
    <value>10</value>
  </attribute>
  <attribute>
    <key>targetamount</key>
    <value>-100</value>
  </attribute>
  <attribute>
    <key>protectioncode</key>
    <value>*****</value>
  </attribute>
  <attribute>
    <key>sourceamountfee</key>
    <value>0</value>
  </attribute>
  <attribute>
    <key>targetcurrencycode</key>
    <value>RUB</value>
  </attribute>
  <attribute>
    <key>statusid</key>
    <value>SUCCEED</value>
  </attribute>
  <attribute>
    <key>sourceamounttotal</key>
    <value>100</value>
  </attribute>
  <attribute>
    <key>modified</key>
    <value>2014-12-23T00:44:37.000+03:00</value>
  </attribute>
  <attribute>
    <key>clienttransaction</key>
    <value>000002</value>
  </attribute>
  <attribute>
```

```
    <key>targetaccountid</key>
    <value>12345678</value>
  </attribute>
  <attribute>
    <key>protectioncodeexpirationdate</key>
    <value>2014-12-23T23:23:19.000+03:00</value>
  </attribute>
  <attribute>
    <key>sourceaccountid</key>
    <value>87654321</value>
  </attribute>
  <attribute>
    <key>isreversed</key>
    <value>true</value>
  </attribute>
</ConfirmTransactionResponse>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "ConfirmTransactionRequest": {
            "transactionId": "328805",
            "protectionCode": "12345"
        }
    }
}}
```

JSON response:

```
{"Envelope": {
    "Body": {
        "ConfirmTransactionResponse": {
            "id": 328805,
            "attribute": [
...
                {
                    "value": "SUCCEED",
                    "key": "statusid"
                }
...
            ]
        }
    }
}}
```

Invoices

You can create an invoice and send it to your customer for payment. The customer can use a terminal to pay the invoice.

Creating an Invoice

1. Send the Invoice request to create the invoice.

   The following sample Invoice request creates an invoice that should be paid to the merchant account 87654321:

```
<InvoiceRequest>
  <payee>87654321</payee>
  <amount>100</amount>
```

```
  <clientTransaction>00001</clientTransaction>
  <description>Sample invoice</description>
</InvoiceRequest>
```

The following sample XML response is returned if the invoice is created successfully:

```
<InvoiceResponse>
  <status>CREATED</status>
  <dateTime>2014-12-18T23:36:27.000+03:00</dateTime>
  <transaction>328498</transaction>
  <clientTransaction>000001</clientTransaction>
</InvoiceResponse>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "InvoiceRequest": {
            "payee": "87654321",
            "amount": 100,
            "clientTransaction": "Merchant transaction Id"
        }
    }
}}
```

JSON response:

```
{"Envelope": {
    "Body": {
        "InvoiceResponse": {
            "transaction": 328498,
            "dateTime": "2015-06-17T09:39:41.000+0300",
            "status": "CREATED",
            "clientTransaction": "Merchant transaction Id"
        }
    }
}}
```

2. Get the transaction value from the Invoice response and send it to the customer. This transaction ID is required to pay the invoice.

> **Tip:** You can also specify a mobile number or an email of the customer in the Invoice request. In this case, the Merchant API notifies the customer automatically.

Processing an Invoice

A customer can use a terminal to pay the invoice. The following requests to the Merchant API must be sent by using a MONETA.RU account that is different from the account that was used to create the invoice..

1. Optionally, send a VerifyPayment request to ensure that the payment request is valid.

   For more information about the VerifyPayment endpoint, see *VerifyPayment Endpoint* on page 127

   The following sample VerifyPayment request verifies whether the invoice that has the transaction ID of 000001 from the customer MONETA.RU account of 12345678 is valid:

```
<VerifyPaymentRequest>
  <payer>12345678</payer>
  <payee>0328498</payee>
  <paymentPassword>*****</paymentPassword>
  <clientTransaction>000001</clientTransaction>
```

```
</VerifyPaymentRequest>
```

Important: To identify an invoice by a transaction ID, prepend a zero to the transaction ID and use this value in the Payee element of the VerifyPayment request.

The following sample XML response is returned if the payment is valid:

```
<VerifyPaymentResponse>
  <isTransactionValid>true</isTransactionValid>
  <forecast>
    <payer>12345678</payer>
    <payerCurrency>RUB</payerCurrency>
    <payerAmount>100</payerAmount>
    <payerFee>0</payerFee>
    <payee>87654321</payee>
    <payeeCurrency>RUB</payeeCurrency>
    <payeeAmount>100</payeeAmount>
    <payerAlias>RUB</payerAlias>
    <payeeAlias>Test Store</payeeAlias>
  </forecast>
  <transactionId>328498</transactionId>
  <operationStatus>CREATED</operationStatus>
</VerifyPaymentResponse>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "VerifyPaymentRequest": {
            "version": "VERSION_2",
            "payer": "12345678",
            "payee": "0328498",
            "paymentPassword": "*****",
            "clientTransaction": "Merchant transaction Id"
        }
    }
}}
```

JSON response:

```
{"Envelope": {
    "Body": {
        "VerifyPaymentResponse": {
            "isTransactionValid": true,
            "forecast": {
                "payer": "12345678",
                "payerCurrency": "RUB",
                "payerAmount": 100,
                "payerFee": 0,
                "payee": "87654321",
                "payeeAmount": 100,
                "payeeCurrency": "RUB"
            },
            "transactionId": "328498",
            "operationStatus": "CREATED"
        }
    }
}}
```

2. Send the Payment request.

For more information about the Payment endpoint, see *Payment Endpoint* on page 121

Important: An invoice cannot be paid by using the same merchant account that was used to create the invoice.

The following Payment request processes the invoice that has the transaction ID of 328498:

```
<PaymentRequest>
  <payer>12345678</payer>
  <payee>0328498</payee>
  <paymentPassword>*****</paymentPassword>
</PaymentRequest>
```

Important: To identify an invoice by a transaction ID, prepend a zero to the transaction ID and use this value in the Payee element of the Payment request.

The following sample XML response is returned if the payment is successful:

```
<PaymentResponse xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd">
  <id>328498</id>
  <attribute>
    <key>category</key>
    <value>BUSINESS</value>
  </attribute>
  <attribute>
    <key>isinvoice</key>
    <value>1</value>
  </attribute>
  <attribute>
    <key>sourcecurrencycode</key>
    <value>RUB</value>
  </attribute>
  <attribute>
    <key>sourceamount</key>
    <value>-100</value>
  </attribute>
  <attribute>
    <key>description</key>
    <value>Sample invoice</value>
  <attribute>
  <attribute>
    <key>typeid</key>
    <value>2</value>
  </attribute>
  <attribute>
    <key>targetamount</key>
    <value>100</value>
  </attribute>
  <attribute>
    <key>targetcurrencycode</key>
    <value>RUB</value>
  </attribute>
  <attribute>
    <key>statusid</key>
    <value>SUCCEED</value>
  </attribute>
  <attribute>
    <key>sourceamounttotal</key>
    <value>-100</value>
  </attribute>
  <attribute>
    <key>modified</key>
    <value>2014-12-21T01:39:42.000+03:00</value>
  </attribute>
  <attribute>
    <key>targetaccountid</key>
    <value>87654321</value>
  </attribute>
  <attribute>
    <key>targettransaction</key>
    <value>000001</value>
  </attribute>
```

```
  <attribute>
    <key>sourceaccountid</key>
    <value>12345678</value>
  </attribute>
</PaymentResponse>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "PaymentRequest": {
            "payer": "12345678",
            "payee": "0328498",
            "paymentPassword": "******"
        }
    }
}}
```

JSON response:

```
{"Envelope": {
    "Body": {
        "PaymentResponse": {
            "id": 328498,
            "attribute": [
...
                {
                    "value": "SUCCEED",
                    "key": "statusid"
                }
...
            ]
        }
    }
}}
```

Important: After processing the payment, MONETA.RU sends a processed payment report to the Pay URL that is specified in your merchant account settings. The script at this web address must return a plain text response in UTF-8 format. Valid responses are:

- SUCCESS. The merchant received the processed payment report. MONETA.RU registers that the merchant received the report and completes the funds transfer.
- FAIL. The merchant could not process the payment report. MONETA.RU resends the processed payment report.

For more information about processed payment reports, see the *MONETA.Assistant API Reference*.

Card payment

Card payment

The method used for credit card payment *Payment*.

The following fields should be passed in the request:

- CARDNUMBER
- CARDEXPIRATION
- CARDCVV2

SOAP request example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
        <wsse:Security
                xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd"
                xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                soap:mustUnderstand="1">
            <wsse:UsernameToken wsu:Id="UsernameToken-42">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <soap:Body>
        <PaymentRequest xmlns="http://www.moneta.ru/schemas/messages.xsd">
            <payer>Payment system account id *</payer>
            <payee>Merchant's account</payee>
            <amount>Amount</amount>
            <isPayerAmount>false</isPayerAmount>
            <clientTransaction>Merchant transaction ID</clientTransaction>
            <operationInfo>
                <attribute>
                    <key>CARDNUMBER</key>
                    <value>1111111111111111</value>
                </attribute>
                <attribute>
                    <key>CARDEXPIRATION</key>
                    <value>01/2001</value>
                </attribute>
                <attribute>
                    <key>CARDCVV2</key>
                    <value>012</value>
                </attribute>
            </operationInfo>
        </PaymentRequest>
    </soap:Body>
</soap:Envelope>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "PaymentRequest": {
            "payer": "Payment system account id *",
            "payee": "Merchant's account",
            "amount": "Amount",
            "isPayerAmount": false,
            "clientTransaction": "Merchant transaction id",
            "operationInfo": {
                "attribute": [
                    {
                        "key": "CARDNUMBER",
                        "value": "1111111111111111"
                    },
                    {
                        "key": "CARDEXPIRATION",
                        "value": "01/2001"
                    },
```

```
                    {
                        "key": "CARDCVV2",
                        "value": "012"
                    }
                ]
            }
        }
    }
}}
```

Note: * Payment system account id - You can check in the commercial department or in your personal account (Office -> Payment methods -> column Account numbers).

If the card does not support 3DS, the operation will be performed immediately. You just have to wait for the payment notification.

If 3DS is enabled for this card, the response will contain a link where the user should be redirected for 3DS authorization:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:PaymentResponse xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd">
            <ns2:id>Moneta_transaction_id</ns2:id>
            ...
            <ns2:attribute>
                <ns2:key>auth3dsurl</ns2:key>
                <ns2:value>https://www.moneta.ru/process3ds.htm?
operationId=Moneta_transaction_id&amp;clientTransaction=Merchant_transaction_ID</
ns2:value>
            </ns2:attribute>
            <ns2:attribute>
                <ns2:key>statusid</ns2:key>
                <ns2:value>INPROGRESS</ns2:value>
            </ns2:attribute>
            ...
        </ns2:PaymentResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON response:

```
{"Envelope": {
    "Body": {
        "PaymentResponse": {
            "id": "Moneta_transaction_id",
            "attribute": [
...
                {
                    "value": "https://www.moneta.ru/process3ds.htm?
operationId=Moneta_transaction_id&amp;clientTransaction=Merchant_transaction_ID",
                    "key": "auth3dsurl"
                },
                {
                    "value": "INPROGRESS",
                    "key": "statusid"
                }
...
            ]
        }
    }
}}
```

If payer's browser supports javascript, it is advisable to add the javascriptEnabled=true parameter to this link, to eliminate the need to draw an additional redirect screen.

Then you should wait for the payment notification.

Tip: You can also check the status of an operation using the method *GetOperationDetailsById* or wait for a request upon crediting, which is configured in the acccount/prototype in "Actions upon crediting/debiting".

Request for recurring payments without entering card data

> **Important:** If you want to make recurring payments, then you should ask the commercial department, as additional configuration is required in the acquiring bank.

Execute method *Invoice* with parameter PAYMENTTOKEN:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
        <wsse:Security
                xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd"
                xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                soap:mustUnderstand="1">
            <wsse:UsernameToken wsu:Id="UsernameToken-42">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <soap:Body>
        <InvoiceRequest xmlns="http://www.moneta.ru/schemas/messages.xsd">
            <payer>Payment system account id *</payer>
            <payee>Merchant's account</payee>
            <amount>Amount</amount>
            <clientTransaction>Merchant transaction ID</clientTransaction>
            <operationInfo>
                <attribute>
                    <key>PAYMENTTOKEN</key>
                    <value>request</value>
                </attribute>
            </operationInfo>
        </InvoiceRequest>
    </soap:Body>
</soap:Envelope>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "InvoiceRequest": {
            "payer": "Payment system account id *",
            "payee": "Merchant's account",
            "amount": "Amount",
            "clientTransaction": "Merchant transaction ID",
            "operationInfo": {
                "attribute": [
                    {
                        "key": "PAYMENTTOKEN",
                        "value": "request"
                    }
                ]
            }
        }
    }
}}
```

> **Tip:** The PAYMENTTOKEN=request attribute means that if the payment is successful, then we must generate a token for recurring payments and add it to the same PAYMENTTOKEN attribute. After the operation, the generated PAYMENTTOKEN will come in a payment notification. Or you can get it by calling the method

GetOperationDetailsById. Then this PAYMENTTOKEN may be passed in the Payment method instead of card data.

Note: * Payment system account id - You can check in the commercial department or in your personal account (Office -> Payment methods -> column Account numbers).

As a result of calling the *Invoice* method, we return the transactionId - the transaction number that must be used when redirecting to the Moneta.Assistant payment form as: https://www.payanyway.ru/assistant.htm?operationId=[transactionId]& paymentSystem.unitId=card&paymentSystem.limitIds=card&followup=true

Invoice request with subsequent authorization hold without confirmation

If you want as a result of payment that the money is not debited from the card, but only blocked, then the method should be called first *Invoice* with parameter AUTHORIZEONLY:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
        <wsse:Security
                xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd"
                xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                soap:mustUnderstand="1">
            <wsse:UsernameToken wsu:Id="UsernameToken-42">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <soap:Body>
        <InvoiceRequest xmlns="http://www.moneta.ru/schemas/messages.xsd">
            <payer>Payment system account id *</payer>
            <payee>Merchant's account</payee>
            <amount>Amount</amount>
            <clientTransaction>Merchant transaction ID</clientTransaction>
            <operationInfo>
                <attribute>
                    <key>AUTHORIZEONLY</key>
                    <value>1</value>
                </attribute>
            </operationInfo>
        </InvoiceRequest>
    </soap:Body>
</soap:Envelope>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "InvoiceRequest": {
            "payer": "Payment system account id *",
            "payee": "Merchant's account",
            "amount": "Amount",
            "clientTransaction": "Merchant transaction ID",
            "operationInfo": {
                "attribute": [
                    {
                        "key": "AUTHORIZEONLY",
                        "value": "1"
                    }
```

```
                ]
            }
        }
    }
}}
```

Note: * Payment system account id - You can check in the commercial department or in your personal account (Office -> Payment methods -> column Account numbers).

As a result of calling the *Invoice* method, we return transactionId - the transaction number that should be used when redirecting to the Moneta.Assistant payment form as: https://www.payanyway.ru/assistant.htm?operationId=[transactionId]& paymentSystem.unitId=card&paymentSystem.limitIds=card&followup=true

Refund

Refund examples

SOAP request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="XWSSGID-1453884616001-1949076214"
 xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:RefundRequest xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd">
            <ns2:transactionId>102314549</ns2:transactionId>
            <ns2:amount>2.16</ns2:amount>
            <ns2:paymentPassword>*****</ns2:paymentPassword>
            <ns2:clientTransaction>CLIENT_TRANSACTION</ns2:clientTransaction>
            <ns2:description>DESCRIPTION</ns2:description>
            <ns2:operationInfo>
                <ns2:attribute>
                    <ns2:key>CUSTOMFIELD:comment</ns2:key>
                    <ns2:value>***</ns2:value>
                </ns2:attribute>
            </ns2:operationInfo>
        </ns2:RefundRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Short SOAP request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="XWSSGID-1453884616001-1949076214"
 xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
```

```
        <ns2:RefundRequest xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd">
            <ns2:transactionId>102314549</ns2:transactionId>
            <ns2:paymentPassword>*****</ns2:paymentPassword>
        </ns2:RefundRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:RefundResponse xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd">
            <ns2:id>102314550</ns2:id>
            ...
            <ns2:attribute>
                <ns2:key>parentid</ns2:key>
                <ns2:value>102314549</ns2:value>
            </ns2:attribute>
            <ns2:attribute>
                <ns2:key>statusid</ns2:key>
                <ns2:value>SUCCEED</ns2:value>
            </ns2:attribute>
            <ns2:attribute>
                <ns2:key>isrefund</ns2:key>
                <ns2:value>1</ns2:value>
            </ns2:attribute>
            ...
        </ns2:RefundResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{
  "Envelope": {
    "Body": {
      "RefundRequest": {
        "amount": 2.16,
        "paymentPassword": "*****",
        "transactionId": 102314549,
        "description": "DESCRIPTION",
        "clientTransaction": "CLIENT_TRANSACTION",
        "operationInfo": {
          "attribute": [
            {
              "value": "***",
              "key": "CUSTOMFIELD:comment"
            }
          ]
        }
      }
    },
    "Header": {
      "Security": {
        "UsernameToken": {
          "Password": "PASSWORD",
          "Username": "USERNAME"
        }
      }
    }
  }
}
```

Short JSON request:

```
{
  "Envelope": {
    "Body": {
```

```
      "RefundRequest": {
        "paymentPassword": "******",
        "transactionId": 102314549
      }
    },
    "Header": {
      "Security": {
        "UsernameToken": {
          "Password": "PASSWORD",
          "Username": "USERNAME"
        }
      }
    }
  }
}
```

JSON response:

```
{
  "Envelope": {
    "Body": {
      "RefundResponse": {
        "id": 102314550,
        "attribute": [
...
          {
            "value": "102314549",
            "key": "parentid"
          },
          {
            "value": "SUCCEED",
            "key": "statusid"
          },
          {
            "value": "1",
            "key": "isrefund"
          }
...
        ]
      }
    }
  }
}
```

Withdrawal

Overview

You can use the Payment request to withdraw money from your MONETA.RU account to a third-party payment system.

The Payment request to withdraw money from your MONETA.RU must include additional parameters that are required by the payment system to which you want to transfer the funds. These parameters are specified within the operationInfo element.

Another difference from the processing payments is the payee element. The payee element in the Payment request for withdrawal must specify the unique identifier of the target payment system.

Before you can withdraw money from your MONETA.RU account, you must determine the unique identifier of the target payment system and all of the additional parameters that this system requires in the Payment requests. Review the following information:

- *Additional Payment Request Parameters* on page 219
- *Payment Request Examples* on page 221
- *Getting an XML Request Template for Withdrawal from MONETA.RU* on page 219

Withdrawing Money from MONETA.RU

Perform the following steps to withdraw money from your MONETA.RU account to a third-party payment system:

1. Optionally, send a VerifyPayment request to ensure that the payment request is valid.

   For more information about the VerifyPayment endpoint, see *VerifyPayment Endpoint* on page 127.

   The following sample XML response is returned if the payment is valid:

```
<VerifyPaymentResponse>
  <isTransactionValid>true</isTransactionValid>
  <forecast>
    <payer>12******78</payer>
    <payerCurrency>RUB</payerCurrency>
    <payerAmount>1.23</payerAmount>
    <payerFee>0</payerFee>
    <payee>9000</payee>
    <payeeCurrency>RUB</payeeCurrency>
    <payeeAmount>1.23</payeeAmount>
    <payeeFee>0</payeeFee>
    <payeeAlias>Municipal Enterprise of Water Supply and Waste Water Treatment
 (Yoshkar-Ola)</payeeAlias>
  </forecast>
</VerifyPaymentResponse>
```

   JSON response:

```
{"Envelope": {
    "Body": {
        "VerifyPaymentResponse": {
            "isTransactionValid": true,
            "forecast": {
                "payer": "12******78",
                "payerCurrency": "RUB",
                "payerAmount": 1.23,
                "payerFee": 0,
                "payee": "9000",
                "payeeAmount": 1.23,
                "payeeCurrency": "RUB",
                "payeeAlias": "Municipal Enterprise of Water Supply and Waste Water
 Treatment (Yoshkar-Ola)"
            }
        }
    }
}}
```

2. If the payer account receives payment passwords by SMS, send the GetAccountPaymentPasswordChallenge request to obtain the payment password.

   For more information about the GetAccountPaymentPasswordChallenge endpoint, see *GetAccountPaymentPasswordChallenge Endpoint* on page 119

3. Send the Payment request.

   For more information about the Payment endpoint, see *Payment Endpoint* on page 121

   The following sample XML response is returned if the withdrawal is successful:

```
<PaymentResponse>
    <id>6********</id>
    <attribute>
        <key>category</key>
        <value>BUSINESS</value>
    </attribute>
    <attribute>
        <key>sourcecurrencycode</key>
        <value>RUB</value>
    </attribute>
    <attribute>
        <key>sourceamount</key>
```

```
        <value>-103</value>
    </attribute>
    <attribute>
        <key>description</key>
        <value>Beeline</value>
    </attribute>
    <attribute>
        <key>typeid</key>
        <value>2</value>
    </attribute>
    <attribute>
        <key>targetamount</key>
        <value>100</value>
    </attribute>
    <attribute>
        <key>sourceamountcompensation</key>
        <value>3</value>
    </attribute>
    <attribute>
        <key>sourceamountfee</key>
        <value>0</value>
   </attribute>
    <attribute>
        <key>targetcurrencycode</key>
        <value>RUB</value>
    </attribute>
    <attribute>
        <key>statusid</key>
        <value>SUCCEED</value>
    </attribute>
    <attribute>
        <key>sourceamounttotal</key>
        <value>-103</value>
    </attribute>
    <attribute>
        <key>customfield:100</key>
        <value>9********4</value>
    </attribute>
    <attribute>
        <key>modified</key>
        <value>2015-01-26T15:00:00.000+03:00</value>
    </attribute>
    <attribute>
        <key>targetalias</key>
        <value>Mobile Communication</value>
    </attribute>
    <attribute>
        <key>targetaccountid</key>
        <value>1144</value>
    </attribute>
    <attribute>
        <key>targettransaction</key>
        <value>4**************0</value>
    </attribute>
    <attribute>
        <key>subproviderid</key>
        <value>2</value>
    </attribute>
    <attribute>
        <key>sourceaccountid</key>
        <value>8*******7</value>
    </attribute>
</PaymentResponse>
```

JSON response:

```
{"Envelope": {
    "Body": {
        "PaymentResponse": {
            "id": "6********",
```

```
            "attribute": [
...
                {
                    "value": "SUCCEED",
                    "key": "statusid"
                }
...
            ]
        }
    }
}}
```

Getting an XML Request Template for Withdrawal from MONETA.RU

Transfer requests for withdrawal from MONETA.RU vary depending on the destination payment system.

1. Withdraw a small amount of money from MONETA.RU by using the web interface to obtain a unique identifier of the operation.
2. Use the GetOperationDetailsById method to get information about the transaction.

   Tip: You can use curl to send the GetOperationDetailsById request. For more information, see *Using CURL to Send SOAP Requests and Get Responses* on page 45 or *Using CURL to Send JSON Requests and Get Responses* on page 47.
3. Extract the OperationInfo object from the response. This object contains all parameters that are required for withdrawal by using the Merchant API.

Now you can edit the OperationInfo object and then use it in the Transfer request for withdrawal.

Additional Payment Request Parameters

The following table describes the Payee values and operationInfo attributes that a Payment request must include for withdrawing money from your MONETA.RU account to other payment systems:

Payment System	Payee	OperationInfo Attributes
Yandex Money	13	• **YANDEXACCOUNT** Specifies the Yandex Money wallet number.
VKontakte	139	• **VKONTAKTEID** Specifies the email address or user name of the VKontakte account.
WebMoney	2 or 3 or 4	• **WEBMONEYWMID** Specifies the identifier of a WebMoney user (WMID). • **WEBMONEYPURSE** Specifies the WMR purse number (if payee = 2). Specifies the WMZ purse number (if payee = 3). Specifies the WME purse number (if payee = 4).
CyberPlat	1144	• **CYBERPLATPROVIDERID** Specifies the CYBERPLATPROVIDERID value. • **CUSTOMFIELD:100**
Bank Account	5	• **WIREUSERNAME** The name of the beneficiary. • **WIREUSERINN** Tax Identification Number (TIN) of the bank. • **WIREPAYMENTPURPOSE**

Payment System	Payee	OperationInfo Attributes
		The description of the payment purpose. • **WIREBANKACCOUNT** The number of the beneficiary bank account. • **WIREBANKBIK** Russian Central Bank identifier code. • **WIREBANKNAME** The name of the bank to which you want to transfer the funds. • **WIREBANKKS** Bank correspondent account. • **WIREKPP** Tax registration reason code. • **WIREKBK** Budgetary classification code. • **WIREOKTMO** Russian classifier of municipal unit territories. • **WIREDOCINDEX** Unique identifier of charges.
Credit Card via Russian Standard Bank	275	• **PAYERCOUNTRY** The country code of the bank to which you want to transfer the funds. Valid values are: • **RUS**. Transfer the funds to a credit card that is issued by a bank in Russia. • **PAYEECARDNUMBER** Full credit card number.
Credit Card via Alpha Bank	279	• **PAYEECARDNUMBER** Full credit card number. • **CARDEXPIRATION** Expiration date of the credit card in the following format: `MM/YYYY`
Money@Mail.ru	268	• **CUSTOMFIELD:RECIPIENT** Specifies an email address or the 16-digit Money@MAil.ru account number. • **CUSTOMFIELD:PAYMENT_PURPOSE** The description of the payment purpose.
QIWI	255	• **EXTERNALACCOUNTID** Specifies the QIWI account.

Payment Request Examples

Review the sample Payment requests to withdraw money from your MONETA.RU account to a bank account or another payment system.

Yandex.Money

SOAP request:

```
<PaymentRequest xmlns="http://www.moneta.ru/schemas/messages.xsd">
  <payer>MONETA.RU Account Number</payer>
  <payee>13</payee>
  <amount>withdrawal_amount</amount>
  <isPayerAmount>true</isPayerAmount>
  <paymentPassword>payment_password</paymentPassword>
  <operationInfo>
    <attribute>
      <key>YANDEXACCOUNT</key>
      <value>Yandex.Money_Wallet_Number</value>
    </attribute>
  </operationInfo>
</PaymentRequest>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "PaymentRequest": {
            "payer": "MONETA.RU Account Number",
            "payee": "13",
            "amount": "withdrawal_amount",
            "isPayerAmount": true,
            "paymentPassword": "payment_password",
            "clientTransaction": "Merchant transaction Id",
            "description": "Transaction description",
            "operationInfo": {
                "attribute": [
                    {
                        "key": "YANDEXACCOUNT",
                        "value": "Yandex.Money_Wallet_Number"
                    }
                ]
            }
        }
    }
}}
```

VKontakte

SOAP request:

```
<PaymentRequest xmlns="http://www.moneta.ru/schemas/messages.xsd">
  <payer>MONETA.RU Account Number</payer>
  <payee>139</payee>
  <amount>payment_amount</amount>
  <isPayerAmount>true</isPayerAmount>
  <paymentPassword>payment_password</paymentPassword>
  <operationInfo>
    <attribute>
      <key>VKONTAKTEID</key>
      <value>E-mail or ID of the VKontakte user</value>
```

```
    </attribute>
  </operationInfo>
</PaymentRequest>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "PaymentRequest": {
            "payer": "MONETA.RU Account Number",
            "payee": "139",
            "amount": "withdrawal_amount",
            "isPayerAmount": true,
            "paymentPassword": "payment_password",
            "clientTransaction": "Merchant transaction Id",
            "description": "Transaction description",
            "operationInfo": {
                "attribute": [
                    {
                        "key": "VKONTAKTEID",
                        "value": "E-mail or ID of the VKontakte user"
                    }
                ]
            }
        }
    }
}}
```

WebMoney

SOAP request:

```
<PaymentRequest xmlns="http://www.moneta.ru/schemas/messages.xsd">
  <payer>MONETA.RU Account Number</payer>
  <payee>2</payee>
  <amount>withdrawal_amount</amount>
  <isPayerAmount>true</isPayerAmount>
  <paymentPassword>payment_password</paymentPassword>
  <operationInfo>
    <attribute>
      <key>WEBMONEYWMID</key>
      <value>Identifier of WebMoney User (WMID)</value>
    </attribute>
    <attribute>
      <key>WEBMONEYPURSE</key>
      <value>WMR Purse Number</value>
    </attribute>
  </operationInfo>
</PaymentRequest>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
```

```
    "Body": {
        "PaymentRequest": {
            "payer": "MONETA.RU Account Number",
            "payee": "2",
            "amount": "withdrawal_amount",
            "isPayerAmount": true,
            "paymentPassword": "payment_password",
            "clientTransaction": "Merchant transaction Id",
            "description": "Transaction description",
            "operationInfo": {
                "attribute": [
                    {
                        "key": "WEBMONEYWMID",
                        "value": "Identifier of WebMoney User (WMID)"
                    },
                    {
                        "key": "WEBMONEYPURSE",
                        "value": "WMR Purse Number"
                    }
                ]
            }
        }
    }
}}
```

CyberPlat

SOAP request:

```
<PaymentRequest xmlns="http://www.moneta.ru/schemas/messages.xsd">
  <payer>MONETA.RU Account Number</payer>
  <payee>1144</payee>
  <amount>withdrawal_amount</amount>
  <isPayerAmount>true</isPayerAmount>
  <paymentPassword>payment_password</paymentPassword>
  <operationInfo>
    <attribute>
      <key>CYBERPLATPROVIDERID</key>
      <value>CYBERPLATPROVIDERID_value</value>
    </attribute>
    <attribute>
      <key>CUSTOMFIELD:100</key>
      <value>8888888888</value>
    </attribute>
  </operationInfo>
</PaymentRequest>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "PaymentRequest": {
            "payer": "MONETA.RU Account Number",
            "payee": "1144",
            "amount": "withdrawal_amount",
            "isPayerAmount": true,
            "paymentPassword": "payment_password",
            "clientTransaction": "Merchant transaction Id",
            "description": "Transaction description",
            "operationInfo": {
                "attribute": [
```

```
                    {
                        "key": "CYBERPLATPROVIDERID",
                        "value": "CYBERPLATPROVIDERID_value"
                    },
                    {
                        "key": "CUSTOMFIELD:100",
                        "value": "8888888888"
                    }
                ]
            }
        }
    }
}}
```

Bank

SOAP request:

```
<PaymentRequest xmlns="http://www.moneta.ru/schemas/messages.xsd">
  <payer>MONETA.RU Account Number</payer>
  <payee>5</payee>
  <amount>withdrawal_amount</amount>
  <isPayerAmount>true</isPayerAmount>
  <paymentPassword>payment_password</paymentPassword>
  <operationInfo>
    <attribute>
      <key>WIREUSERNAME</key>
      <value>beneficiary_name</value>
    </attribute>
    <attribute>
      <key>WIREUSERINN</key>
      <value>Tax Identification Number (TIN) of the bank</value>
    </attribute>
    <attribute>
      <key>WIREPAYMENTPURPOSE</key>
      <value>payment_purpose_description</value>
    </attribute>
    <attribute>
      <key>WIREBANKACCOUNT</key>
      <value>beneficiary_account</value>
    </attribute>
    <attribute>
      <key>WIREBANKBIK</key>
      <value>russian_central_bank_identifier_code</value>
    </attribute>
    <attribute>
      <key>WIREBANKNAME</key>
      <value>bank_name</value>
    </attribute>
    <attribute>
      <key>WIREBANKKS</key>
      <value>bank_correspondent_account</value>
    </attribute>
    <attribute>
      <key>WIREKPP</key>
      <value>tax_registration_reason_code</value>
    </attribute>
    <attribute>
      <key>WIREKBK</key>
      <value>budgetary_classification_code</value>
    </attribute>
    <attribute>
      <key>WIREOKTMO</key>
      <value>russian_classifier_of_municipal_unit_territories</value>
    </attribute>
    <attribute>
      <key>WIREDOCINDEX</key>
      <value>unique_identifier_of_charges</value>
    </attribute>
  </operationInfo>
```

```
</PaymentRequest>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "PaymentRequest": {
            "payer": "MONETA.RU Account Number",
            "payee": "5",
            "amount": "withdrawal_amount",
            "isPayerAmount": true,
            "paymentPassword": "payment_password",
            "clientTransaction": "Merchant transaction Id",
            "operationInfo": {
                "attribute": [
                    {
                        "key": "WIREUSERNAME",
                        "value": "beneficiary_name"
                    },
                    {
                        "key": "WIREUSERINN",
                        "value": "Tax Identification Number (TIN) of the bank"
                    },
                    {
                        "key": "WIREPAYMENTPURPOSE",
                        "value": "payment_purpose_description"
                    },
                    {
                        "key": "WIREBANKACCOUNT",
                        "value": "beneficiary_account"
                    },
                    {
                        "key": "WIREBANKBIK",
                        "value": "russian_central_bank_identifier_code"
                    },
                    {
                        "key": "WIREBANKNAME",
                        "value": "bank_name"
                    },
                    {
                        "key": "WIREBANKKS",
                        "value": "bank_correspondent_account"
                    },
                    {
                        "key": "WIREKPP",
                        "value": "tax_registration_reason_code"
                    },
                    {
                        "key": "WIREKBK",
                        "value": "budgetary_classification_code"
                    },
                    {
                        "key": "WIREOKTMO",
                        "value": "russian_classifier_of_municipal_unit_territories"
                    },
                    {
                        "key": "WIREDOCINDEX",
                        "value": "unique_identifier_of_charges"
                    }
                ]
            }
        }
```

```
    }
}}
```

Credit Card via Russian Standard Bank

SOAP request:

```
<PaymentRequest xmlns="http://www.moneta.ru/schemas/messages.xsd">
  <payer>MONETA.RU Account Number</payer>
  <payee>275</payee>
  <amount>withdrawal_amount</amount>
  <isPayerAmount>true</isPayerAmount>
  <paymentPassword>payment_password</paymentPassword>
  <clientTransaction>unique_transaction_identifier</clientTransaction>
  <operationInfo>
    <attribute>
      <key>PAYERCOUNTRY</key>
      <value>RUS</value>
    </attribute>
    <!-- Withdrawal to a Credit Card that was Issued in Russia -->
    <attribute>
      <key>PAYEECARDNUMBER</key>
      <value>full_credit_card_number</value>
    </attribute>
  </operationInfo>
</PaymentRequest>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "PaymentRequest": {
            "payer": "MONETA.RU Account Number",
            "payee": "275",
            "amount": "withdrawal_amount",
            "isPayerAmount": true,
            "paymentPassword": "payment_password",
            "clientTransaction": "Merchant transaction Id",
            "description": "Transaction description",
            "operationInfo": {
                "attribute": [
                    {
                        "key": "PAYERCOUNTRY",
                        "value": "RUS"
                    },
                    {
                        "key": "PAYEECARDNUMBER",
                        "value": "full_credit_card_number"
                    }
                ]
            }
        }
    }
}}
```

Credit Card via Alpha Bank

SOAP request:

```
<PaymentRequest xmlns="http://www.moneta.ru/schemas/messages.xsd">
```

```
  <payer>MONETA.RU Account Number</payer>
  <payee>279</payee>
  <amount>withdrawal_amount</amount>
  <isPayerAmount>true</isPayerAmount>
  <paymentPassword>payment_password</paymentPassword>
  <clientTransaction>unique_transaction_identifier</clientTransaction>
  <operationInfo>
    <attribute>
      <key>PAYEECARDNUMBER</key>
      <value>full_credit_card_number</value>
    </attribute>
    <attribute>
      <key>CARDEXPIRATION</key>
      <value>credit_card_expiration_date</value>
    </attribute>
    <!-- MM/YYYY -->
  </operationInfo>
</PaymentRequest>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "PaymentRequest": {
            "payer": "MONETA.RU Account Number",
            "payee": "279",
            "amount": "withdrawal_amount",
            "isPayerAmount": true,
            "paymentPassword": "payment_password",
            "clientTransaction": "Merchant transaction Id",
            "description": "Transaction description",
            "operationInfo": {
                "attribute": [
                    {
                        "key": "PAYEECARDNUMBER",
                        "value": "full_credit_card_number"
                    },
                    {
                        "key": "CARDEXPIRATION",
                        "value": "MM/YYYY"
                    }
                ]
            }
        }
    }
}}
```

Money@Mail.ru

SOAP request:

```
<PaymentRequest xmlns="http://www.moneta.ru/schemas/messages.xsd">
  <payer>MONETA.RU Account Number</payer>
  <payee>268</payee>
  <amount>withdrawal_amount</amount>
  <!-- Withdrawal amount format: 12.34 -->
  <isPayerAmount>true</isPayerAmount>
  <paymentPassword>payment_password</paymentPassword>
  <clientTransaction>unique_transaction_identifier</clientTransaction>
  <operationInfo>
    <attribute>
```

```
      <key>CUSTOMFIELD:RECIPIENT</key>
      <value>E-mail address or 16-digit account number at Money@Mail.ru</value>
    </attribute>
    <attribute>
      <key>CUSTOMFIELD:PAYMENT_PURPOSE</key>
      <value>payment_purpose_description</value>
    </attribute>
  </operationInfo>
</PaymentRequest>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "PaymentRequest": {
            "payer": "MONETA.RU Account Number",
            "payee": "268",
            "amount": "withdrawal_amount",
            "isPayerAmount": true,
            "paymentPassword": "payment_password",
            "clientTransaction": "Merchant transaction Id",
            "description": "Transaction description",
            "operationInfo": {
                "attribute": [
                    {
                        "key": "CUSTOMFIELD:RECIPIENT",
                        "value": "E-mail address or 16-digit account number at
 Money@Mail.ru"
                    },
                    {
                        "key": "CUSTOMFIELD:PAYMENT_PURPOSE",
                        "value": "payment_purpose_description"
                    }
                ]
            }
        }
    }
}}
```

QIWI

SOAP request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken
                    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:PaymentRequest xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd"
 ns2:version="VERSION_2">
```

```
            <ns2:payer>MONETA.RU Account Number</ns2:payer>
            <ns2:payee>255</ns2:payee>
            <ns2:amount>AMOUNT</ns2:amount>
            <ns2:isPayerAmount>true</ns2:isPayerAmount>
            <ns2:paymentPassword>PAYMENT_PASSWORD</ns2:paymentPassword>
            <ns2:operationInfo>
                <ns2:attribute>
                    <ns2:key>EXTERNALACCOUNTID</ns2:key>
                    <ns2:value>QIWI_Account_Id</ns2:value>
                </ns2:attribute>
            </ns2:operationInfo>
        </ns2:PaymentRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "PaymentRequest": {
            "payer": "MONETA.RU Account Number",
            "payee": "255",
            "amount": "withdrawal_amount",
            "isPayerAmount": true,
            "paymentPassword": "payment_password",
            "clientTransaction": "Merchant transaction Id",
            "description": "Transaction description",
            "operationInfo": {
                "attribute": [
                    {
                        "key": "EXTERNALACCOUNTID",
                        "value": "QIWI_Account_Id"
                    }
                ]
            }
        }
    }
}}
```

Batch requests

VerifyPaymentBatch

Transaction validation request. Request in batch mode.

Examples

SOAP request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-1456389563171198030372 3"
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
```

```
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:VerifyPaymentBatchRequest xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd" ns2:version="VERSION_2" xmlns="">
            <ns2:transactional>false</ns2:transactional>
            <ns2:exitOnFailure>false</ns2:exitOnFailure>
            <ns2:transaction ns2:version="VERSION_2">
                <ns2:payer>12345678</ns2:payer>
                <ns2:payee>87654321</ns2:payee>
                <ns2:amount>100.87</ns2:amount>
                <ns2:isPayerAmount>false</ns2:isPayerAmount>
                <ns2:clientTransaction>Merchant transaction Id</ns2:clientTransaction>
            </ns2:transaction>
            <ns2:transaction ns2:version="VERSION_2">
                <ns2:payer>12345678</ns2:payer>
                <ns2:payee>87654321</ns2:payee>
                <ns2:amount>200.87</ns2:amount>
                <ns2:isPayerAmount>false</ns2:isPayerAmount>
                <ns2:clientTransaction>Merchant transaction Id</ns2:clientTransaction>
                <ns2:operationInfo>
                    <ns2:attribute>
                        <ns2:key>ISINVOICE</ns2:key>
                        <ns2:value>1</ns2:value>
                    </ns2:attribute>
                </ns2:operationInfo>
            </ns2:transaction>
        </ns2:VerifyPaymentBatchRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:VerifyPaymentBatchResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
            <ns2:transaction>
                <ns2:isTransactionValid>true</ns2:isTransactionValid>
                <ns2:forecast>
                    <ns2:payer>12345678</ns2:payer>
                    <ns2:payerCurrency>USD</ns2:payerCurrency>
                    <ns2:payerAmount>3.56</ns2:payerAmount>
                    <ns2:payee>87654321</ns2:payee>
                    <ns2:payeeCurrency>RUB</ns2:payeeCurrency>
                    <ns2:payeeAmount>100.87</ns2:payeeAmount>
                </ns2:forecast>
            </ns2:transaction>
            <ns2:transaction>
                <ns2:isTransactionValid>true</ns2:isTransactionValid>
                <ns2:description>Transaction description</ns2:description>
                <ns2:forecast>
                    <ns2:payer>12345678</ns2:payer>
                    <ns2:payerCurrency>USD</ns2:payerCurrency>
                    <ns2:payerAmount>6.17</ns2:payerAmount>
                    <ns2:payee>87654321</ns2:payee>
                    <ns2:payeeCurrency>RUB</ns2:payeeCurrency>
                    <ns2:payeeAmount>200.87</ns2:payeeAmount>
                </ns2:forecast>
                <ns2:operationInfo>
                    <ns2:attribute>
                        <ns2:key>isinvoice</ns2:key>
                        <ns2:value>1</ns2:value>
                    </ns2:attribute>
                </ns2:operationInfo>
            </ns2:transaction>
```

```
        </ns2:VerifyPaymentBatchResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response with error:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:VerifyPaymentBatchResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
            <ns2:transaction>
                <ns2:isTransactionValid>true</ns2:isTransactionValid>
                <ns2:forecast>
                    <ns2:payer>12345678</ns2:payer>
                    <ns2:payerCurrency>USD</ns2:payerCurrency>
                    <ns2:payerAmount>3.56</ns2:payerAmount>
                    <ns2:payee>87654321</ns2:payee>
                    <ns2:payeeCurrency>RUB</ns2:payeeCurrency>
                    <ns2:payeeAmount>100.87</ns2:payeeAmount>
                </ns2:forecast>
            </ns2:transaction>
            <ns2:transaction>
                <ns2:isTransactionValid>false</ns2:isTransactionValid>
                <ns2:description>Error description</ns2:description>
                <ns2:errorCode>500</ns2:errorCode>
            </ns2:transaction>
        </ns2:VerifyPaymentBatchResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{
  "Envelope": {
    "Body": {
      "VerifyPaymentBatchRequest": {
        "version": "VERSION_2",
        "transactional": false,
        "exitOnFailure": false,
        "transaction": [
          {
            "version": "VERSION_2",
            "payer": "12345678",
            "payee": "87654321",
            "amount": 100.87,
            "isPayerAmount": false,
            "clientTransaction": "Merchant transaction Id"
          },
          {
            "version": "VERSION_2",
            "payer": "12345678",
            "payee": "87654321",
            "amount": 200.87,
            "isPayerAmount": false,
            "clientTransaction": "Merchant transaction Id",
            "operationInfo": {
              "attribute": [
                {
                  "value": "1",
                  "key": "ISINVOICE"
                }
              ]
            }
          }
        ]
      }
    },
    "Header": {
      "Security": {
```

```
        "UsernameToken": {
          "Password": "PASSWORD",
          "Username": "USERNAME"
        }
      }
    }
  }
}
```

JSON response:

```
{
  "Envelope": {
    "Body": {
      "VerifyPaymentBatchResponse": {
        "transaction": [
          {
            "isTransactionValid": true,
            "forecast": {
              "payer": 12345678,
              "payerCurrency": "USD",
              "payerAmount": 3.56,
              "payee": 87654321,
              "payeeCurrency": "RUB",
              "payeeAmount": 100.87
            }
          },
          {
            "isTransactionValid": true,
            "description": "Transaction description",
            "forecast": {
              "payer": 12345678,
              "payerCurrency": "USD",
              "payerAmount": 6.17,
              "payee": 87654321,
              "payeeCurrency": "RUB",
              "payeeAmount": 200.87
            },
            "operationInfo": {
              "attribute": [
                {
                  "value": "1",
                  "key": "isinvoice"
                }
              ]
            }
          }
        ]
      }
    }
  }
}
```

JSON response with error:

```
{
  "Envelope": {
    "Body": {
      "VerifyPaymentBatchResponse": {
        "transaction": [
          {
            "isTransactionValid": true,
            "forecast": {
              "payer": 12345678,
              "payerCurrency": "USD",
              "payerAmount": 3.56,
              "payee": 87654321,
              "payeeCurrency": "RUB",
              "payeeAmount": 100.87
            }
```

```
          },
          {
            "isTransactionValid": false,
            "description": "Error description",
            "errorCode": "500"
          }
        ]
      }
    }
  }
}
```

SecureData requests

SecureData

Attributes preservation request

See: *SecureData Endpoint* on page 124.

JSON request:

```
{
  "SecureDataRequest": {
    "accountId": 22222222,
    "publicId": "d21ba7f4-1272-43c9-bbc5-e8c7cb75789e",
    "attribute": [
      {
        "value": "5417151111116825",
        "key": "CARDNUMBER"
      },
      {
        "value": "01\/2020",
        "key": "CARDEXPIRATION"
      },
      {
        "value": "Ivanov Ivan",
        "key": "CARDHOLDER"
      },
      {
        "value": "123",
        "key": "CARDCVV2"
      }
    ]
  }
}
```

JSON response:

```
{
  "SecureDataResponse": {
    "expirationDate": "2018-08-16T09:06:05.000+03:00",
    "secureToken": "jz8QdLCXQTeWXHNS-1075-788270e2fcf170c60136a2a2ee06dd8110"
  }
}
```

SecureDataStatus

Secure Token status request

See: *SecureDataStatus Endpoint* on page 125.

SOAP request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
```

```
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-1534398665524-497643242" xmlns:wsse="http://
docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:SecureDataStatusRequest xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd" xmlns="">
            <ns2:secureToken>jz8QdLCXQTeWXHNS-1075-788270e2fcf170c60136a2a2ee06dd8110</
ns2:secureToken>
        </ns2:SecureDataStatusRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:SecureDataStatusResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
            <ns2:expirationDate>2018-08-16T09:06:05.000+03:00</ns2:expirationDate>
            <ns2:attribute>
                <ns2:key>CARDNUMBER</ns2:key>
                <ns2:value>541715******6825</ns2:value>
            </ns2:attribute>
            <ns2:attribute>
                <ns2:key>CARDEXPIRATION</ns2:key>
                <ns2:value>01/2020</ns2:value>
            </ns2:attribute>
            <ns2:attribute>
                <ns2:key>CARDHOLDER</ns2:key>
                <ns2:value>Ivanov Ivan</ns2:value>
            </ns2:attribute>
        </ns2:SecureDataStatusResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
          "Password": "PASSWORD"
        }
      }
    },
    "Body": {
      "SecureDataStatusRequest": {
        "secureToken": "jz8QdLCXQTeWXHNS-1075-788270e2fcf170c60136a2a2ee06dd8110"
      }
    }
  }
}
```

JSON response:

```
{
  "Envelope": {
```

```
    "Body": {
      "SecureDataStatusResponse": {
        "expirationDate": "2018-08-16T09:13:28.000+03:00",
        "attribute": [
          {
            "value": "541715******6825",
            "key": "CARDNUMBER"
          },
          {
            "value": "01\/2020",
            "key": "CARDEXPIRATION"
          },
          {
            "value": "Ivanov Ivan",
            "key": "CARDHOLDER"
          }
        ]
      }
    }
  }
}
```

Payment with SECURETOKEN

See: *Payment Endpoint* on page 121.

SOAP request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd"
                       SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken
                    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-1534416862068-2072311100"
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:PaymentRequest xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd"
 ns2:version="VERSION_2" xmlns="">
            <ns2:payer>303</ns2:payer>
            <ns2:payee>22222222</ns2:payee>
            <ns2:amount>100.00</ns2:amount>
            <ns2:isPayerAmount>true</ns2:isPayerAmount>
            <ns2:clientTransaction>ctid</ns2:clientTransaction>
            <ns2:operationInfo>
                <ns2:attribute>
                    <ns2:key>SECURETOKEN</ns2:key>

 <ns2:value>jz8QdLCXQTeWXHNS-1075-788270e2fcf170c60136a2a2ee06dd8110</ns2:value>
                </ns2:attribute>
            </ns2:operationInfo>
        </ns2:PaymentRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{
  "Envelope": {
    "Header": {
```

```
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
          "Password": "PASSWORD"
        }
      }
    },
    "Body": {
      "PaymentRequest": {
        "payer": "303",
        "payee": "22222222",
        "amount": 100.00,
        "isPayerAmount": true,
        "clientTransaction": "ctid",
        "operationInfo": {
          "attribute": [
            {
              "value": "jz8QdLCXQTeWXHNS-1075-788270e2fcf170c60136a2a2ee06dd8110",
              "key": "SECURETOKEN"
            }
          ]
        },
        "version": "VERSION_2"
      }
    }
  }
}
```

Payment history

GetOperationDetailsById - transaction details

Request for transaction details by a MONETA.RU transaction ID.

Examples

SOAP request:

```
<soapenv:Envelope xmlns:mes="http://www.moneta.ru/schemas/messages.xsd"
                  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Header>
        <wsse:Security soapenv:mustUnderstand="1"
                       xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
            <wsse:UsernameToken wsu:Id="UsernameToken-1"
                                xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-utility-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password
                        Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </soapenv:Header>
    <soapenv:Body>
        <mes:GetOperationDetailsByIdRequest>101370592</
mes:GetOperationDetailsByIdRequest>
    </soapenv:Body>
</soapenv:Envelope>
```

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <SOAP-ENV:Body>
      <ns2:GetOperationDetailsByIdResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
```

```
         <ns2:operation>
            <ns2:id>101370592</ns2:id>
            <ns2:attribute>
               <ns2:key>category</ns2:key>
               <ns2:value>BUSINESS</ns2:value>
            </ns2:attribute>
            <ns2:attribute>
               <ns2:key>sourcecurrencycode</ns2:key>
               <ns2:value>RUB</ns2:value>
            </ns2:attribute>
            <ns2:attribute>
               <ns2:key>sourceamount</ns2:key>
               <ns2:value>2.02</ns2:value>
            </ns2:attribute>
            <ns2:attribute>
               <ns2:key>typeid</ns2:key>
               <ns2:value>3</ns2:value>
            </ns2:attribute>
            <ns2:attribute>
               <ns2:key>targetamount</ns2:key>
               <ns2:value>-2.02</ns2:value>
            </ns2:attribute>
            <ns2:attribute>
               <ns2:key>sourceamountfee</ns2:key>
               <ns2:value>-0.03</ns2:value>
            </ns2:attribute>
            <ns2:attribute>
               <ns2:key>targetcurrencycode</ns2:key>
               <ns2:value>RUB</ns2:value>
            </ns2:attribute>
            <ns2:attribute>
               <ns2:key>statusid</ns2:key>
               <ns2:value>SUCCEED</ns2:value>
            </ns2:attribute>
            <ns2:attribute>
               <ns2:key>sourceamounttotal</ns2:key>
               <ns2:value>1.99</ns2:value>
            </ns2:attribute>
            <ns2:attribute>
               <ns2:key>modified</ns2:key>
               <ns2:value>2015-07-13T12:00:46.000+03:00</ns2:value>
            </ns2:attribute>
            <ns2:attribute>
               <ns2:key>targetaccountid</ns2:key>
               <ns2:value>14692426</ns2:value>
            </ns2:attribute>
            <ns2:attribute>
               <ns2:key>sourceaccountid</ns2:key>
               <ns2:value>96801571</ns2:value>
            </ns2:attribute>
            <ns2:attribute>
               <ns2:key>isreversed</ns2:key>
               <ns2:value>true</ns2:value>
            </ns2:attribute>
         </ns2:operation>
      </ns2:GetOperationDetailsByIdResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
```

```
    "Body": {
        "GetOperationDetailsByIdRequest": 101370592
    }
}}
```

JSON response:

```
{"Envelope": {
    "Body": {
        "GetOperationDetailsByIdResponse": {
            "operation": {
                "id": 101370592,
                "attribute": [
                    {
                        "value": "BUSINESS",
                        "key": "category"
                    },
                    {
                        "value": "RUB",
                        "key": "sourcecurrencycode"
                    },
                    {
                        "value": "2.02",
                        "key": "sourceamount"
                    },
                    {
                        "value": "3",
                        "key": "typeid"
                    },
                    {
                        "value": "-2.02",
                        "key": "targetamount"
                    },
                    {
                        "value": "-0.03",
                        "key": "sourceamountfee"
                    },
                    {
                        "value": "RUB",
                        "key": "targetcurrencycode"
                    },
                    {
                        "value": "SUCCEED",
                        "key": "statusid"
                    },
                    {
                        "value": "1.99",
                        "key": "sourceamounttotal"
                    },
                    {
                        "value": "2015-07-13T12:00:46.000+03:00",
                        "key": "modified"
                    },
                    {
                        "value": "87654321",
                        "key": "targetaccountid"
                    },
                    {
                        "value": "12345678",
                        "key": "sourceaccountid"
                    },
                    {
                        "value": "true",
                        "key": "isreversed"
                    }
                ]
            }
        }
    }
}}
```

FindOperationsList - list of transactions

Request for getting a list of transactions by the specified filter.

Examples

SOAP request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:mes="http://www.moneta.ru/schemas/messages.xsd">
   <soapenv:Header>
      <wsse:Security soapenv:mustUnderstand="1" xmlns:wsse="http://docs.oasis-open.org/
wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
         <wsse:UsernameToken wsu:Id="UsernameToken-1" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
            <wsse:Username>USERNAME</wsse:Username>
            <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
         </wsse:UsernameToken>
      </wsse:Security>
   </soapenv:Header>
   <soapenv:Body>
      <mes:FindOperationsListRequest>
         <mes:pager>
            <mes:pageNumber>1</mes:pageNumber>
            <mes:pageSize>10</mes:pageSize>
         </mes:pager>
         <mes:filter>
            <mes:unitId>123456</mes:unitId>
            <mes:dateFrom>2015-05-01T00:00:00.000+03:00</mes:dateFrom>
            <mes:dateTo>2015-05-03T23:59:59.999+03:00</mes:dateTo>
         </mes:filter>
      </mes:FindOperationsListRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:FindOperationsListResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
            <ns2:pageSize>10</ns2:pageSize>
            <ns2:pageNumber>1</ns2:pageNumber>
            <ns2:pagesCount>57</ns2:pagesCount>
            <ns2:size>10</ns2:size>
            <ns2:totalSize>570</ns2:totalSize>
            <ns2:operation>
                <ns2:id>101093434</ns2:id>
                <ns2:attribute>
                    <ns2:key>statusid</ns2:key>
                    <ns2:value>SUCCEED</ns2:value>
                </ns2:attribute>
                ...
            </ns2:operation>
            ...
            <ns2:operation>
                <ns2:id>101093432</ns2:id>
                <ns2:attribute>
                    <ns2:key>statusid</ns2:key>
                    <ns2:value>SUCCEED</ns2:value>
                </ns2:attribute>
                ...
            </ns2:operation>
            ...
        </ns2:FindOperationsListResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "FindOperationsListRequest": {
            "pager": {
                "pageNumber": 1,
                "pageSize": 10
            },
            "filter": {
                "unitId": 123456,
                "dateFrom": "2015-05-01T00:00:00.000+03:00",
                "dateTo": "2015-05-03T23:59:59.999+03:00"
            }
        }
    }
}}
```

JSON response:

```
{"Envelope": {
    "Body": {
        "FindOperationsListResponse": {
            "pageSize": 10,
            "pagesCount": 57,
            "pageNumber": 1,
            "totalSize": 570,
            "size": 10,
            "operation": [
                {
                    "id": 101093434,
                    "attribute": [
                        {
                            "value": "SUCCEED",
                            "key": "statusid"
                        }
                        ...
                    ]
                },
                ...
                {
                    "id": 101093433,
                    "attribute": [
                        {
                            "value": "SUCCEED",
                            "key": "statusid"
                        }
                        ...
                    ]
                }
                ...
            ]
        }
    }
}}
```

FindLastOperationsList - list of the last transactions

Request for getting the list of the last transactions.

Examples

SOAP request:

```
<soapenv:Envelope xmlns:mes="http://www.moneta.ru/schemas/messages.xsd"
                  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Header>
        <wsse:Security soapenv:mustUnderstand="1"
                       xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
            <wsse:UsernameToken wsu:Id="UsernameToken-1"
                                xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-utility-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password
                         Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </soapenv:Header>
    <soapenv:Body>
        <mes:FindLastOperationsListRequest>
            <mes:unitId>123456</mes:unitId>
            <mes:transactionsQuantity>5</mes:transactionsQuantity>
        </mes:FindLastOperationsListRequest>
    </soapenv:Body>
</soapenv:Envelope>
```

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:FindLastOperationsListResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
            <ns2:pageSize>5</ns2:pageSize>
            <ns2:pageNumber>1</ns2:pageNumber>
            <ns2:pagesCount>1</ns2:pagesCount>
            <ns2:size>5</ns2:size>
            <ns2:totalSize>5</ns2:totalSize>
            <ns2:operation>
                <ns2:id>101851409</ns2:id>
                <ns2:attribute>
                    <ns2:key>statusid</ns2:key>
                    <ns2:value>TAKENIN_NOTSENT</ns2:value>
                </ns2:attribute>
                ...
            </ns2:operation>
            ...
            <ns2:operation>
                <ns2:id>101851410</ns2:id>
                <ns2:attribute>
                    <ns2:key>statusid</ns2:key>
                    <ns2:value>CREATED</ns2:value>
                </ns2:attribute>
                ...
            </ns2:operation>
            ...
        </ns2:FindLastOperationsListResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{"Envelope": {
```

```
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "FindLastOperationsListRequest": {
            "unitId": 123456,
            "transactionsQuantity": 5
        }
    }
}}
```

JSON response:

```
{"Envelope": {
    "Body": {
        "FindLastOperationsListResponse": {
            "pageSize": 5,
            "pagesCount": 1,
            "pageNumber": 1,
            "totalSize": 5,
            "size": 5,
            "operation": [
                {
                    "id": 101851409,
                    "attribute": [
                        {
                            "value": "TAKENIN_NOTSENT",
                            "key": "statusid"
                        }
                        ...
                    ]
                },
                ...
                {
                    "id": 101851410,
                    "attribute": [
                        {
                            "value": "CREATED",
                            "key": "statusid"
                        }
                        ...
                    ]
                }
                ...
            ]
        }
    }
}}
```

FindOperationsListByCTID - transaction details by a merchant transaction Id

Request for retrieving transaction details by a merchant transaction Id.

Examples

SOAP request:

```
<soapenv:Envelope xmlns:mes="http://www.moneta.ru/schemas/messages.xsd"
                  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Header>
        <wsse:Security soapenv:mustUnderstand="1"
                       xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
            <wsse:UsernameToken wsu:Id="UsernameToken-1"
```

```
                                xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-utility-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password
                        Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </soapenv:Header>
    <soapenv:Body>
        <mes:FindOperationsListByCTIDRequest>
            <mes:pager>
                <mes:pageNumber>1</mes:pageNumber>
                <mes:pageSize>10</mes:pageSize>
            </mes:pager>
            <mes:accountId>12345678</mes:accountId>
            <mes:clientTransaction>Merchant transaction Id</mes:clientTransaction>
        </mes:FindOperationsListByCTIDRequest>
    </soapenv:Body>
</soapenv:Envelope>
```

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:FindOperationsListByCTIDResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
            <ns2:pageSize>10</ns2:pageSize>
            <ns2:pageNumber>1</ns2:pageNumber>
            <ns2:pagesCount>1</ns2:pagesCount>
            <ns2:size>1</ns2:size>
            <ns2:totalSize>1</ns2:totalSize>
            <ns2:operation>
                <ns2:id>100944043</ns2:id>
                <ns2:attribute>
                    <ns2:key>statusid</ns2:key>
                    <ns2:value>SUCCEED</ns2:value>
                </ns2:attribute>
                <ns2:attribute>
                    <ns2:key>clienttransaction</ns2:key>
                    <ns2:value>Merchant transaction Id</ns2:value>
                </ns2:attribute>
                ...
            </ns2:operation>
        </ns2:FindOperationsListByCTIDResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "FindOperationsListByCTIDRequest": {
            "pager": {
                "pageNumber": 1,
                "pageSize": 10
            },
            "accountId": 12345678,
            "clientTransaction": "Merchant transaction Id"

        }
```

```
    }
}}
```

JSON response:

```
{"Envelope": {
    "Body": {
        "FindOperationsListByCTIDResponse": {
            "pageSize": 10,
            "pagesCount": 1,
            "pageNumber": 1,
            "totalSize": 1,
            "size": 1,
            "operation": [
                {
                    "id": 100944043,
                    "attribute": [
                        {
                            "value": "SUCCEED",
                            "key": "statusid"
                        },
                        {
                            "value": "Merchant transaction Id",
                            "key": "clienttransaction"
                        }
                        ...
                    ]
                }
            ]
        }
    }
}}
```

GetTurnoverList - monthly total

Request for the monthly turnover information.

Examples

SOAP request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken
                    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-1491387795271-184820942"
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">
                    PASSWORD
                </wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:GetTurnoverListRequest xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd" xmlns="">
            <ns2:unitId>123456</ns2:unitId>
            <ns2:dateFrom>2017-03-01+03:00</ns2:dateFrom>
            <ns2:dateTo>2017-03-31+03:00</ns2:dateTo>
            <ns2:accountIds>12345678</ns2:accountIds>
            <ns2:currencyCode>RUB</ns2:currencyCode>
            <ns2:groupByCurrency>false</ns2:groupByCurrency>
            <ns2:operationTypeCategory>BUSINESS</ns2:operationTypeCategory>
```

```
            <ns2:categoryDetails>true</ns2:categoryDetails>
        </ns2:GetTurnoverListRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:GetTurnoverListResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
            <ns2:groupByCurrency>false</ns2:groupByCurrency>
            <ns2:turnover>
                <ns2:date>2017-03-01+03:00</ns2:date>
                <ns2:account>12345678</ns2:account>
                <ns2:currency>RUB</ns2:currency>
                <ns2:operationTypeCategory>BUSINESS</ns2:operationTypeCategory>
                <ns2:incomeAmount>253675.62</ns2:incomeAmount>
                <ns2:incomeCommission>-2924.27</ns2:incomeCommission>
                <ns2:incomeTransactionsCount>2547</ns2:incomeTransactionsCount>
                <ns2:expenseAmount>3999.58</ns2:expenseAmount>
                <ns2:expensesIncludingCommission>5.19</ns2:expensesIncludingCommission>
                <ns2:expensesExtraCommission>115.66</ns2:expensesExtraCommission>
                <ns2:expenseTransactionsCount>323</ns2:expenseTransactionsCount>
                <ns2:total>246636.11</ns2:total>
                <ns2:openingBalance>0</ns2:openingBalance>
                <ns2:closingBalance>0</ns2:closingBalance>
            </ns2:turnover>
        </ns2:GetTurnoverListResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
          "Password": "PASSWORD"
        }
      }
    },
    "Body": {
      "GetTurnoverListRequest": {
        "unitId": 123456,
        "dateFrom": "2017-03-01T00:00:00.000+03:00",
        "dateTo": "2017-03-31T00:00:00.000+03:00",
        "accountIds": [
          12345678
        ],
        "currencyCode": "RUB",
        "groupByCurrency": false,
        "categoryDetails": true,
        "operationTypeCategory": "BUSINESS"
      }
    }
  }
}
```

JSON response:

```
{
  "Envelope": {
    "Body": {
      "GetTurnoverListResponse": {
        "turnover": [
```

```
          {
            "total": 246636.11,
            "incomeCommission": -2924.27,
            "expenseAmount": 3999.58,
            "expenseTransactionsCount": 323,
            "date": "2017-03-01T00:00:00.000+03:00",
            "operationTypeCategory": "BUSINESS",
            "currency": "RUB",
            "expensesIncludingCommission": 5.19,
            "openingBalance": 0,
            "incomeAmount": 253675.62,
            "account": "12345678",
            "expensesExtraCommission": 115.66,
            "incomeTransactionsCount": 2547,
            "closingBalance": 0
          }
        ],
        "groupByCurrency": false
      }
    }
  }
}
```

GetTurnoverList (AsyncRequest) - monthly total

Asynchronous request for the monthly turnover information.

Examples

SOAP request (send request):

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken
                    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-1500891606651-1147232686"
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:AsyncRequest xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd"
 xmlns="">
            <ns2:GetTurnoverListRequest xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd" xmlns="">
                <ns2:unitId>123456</ns2:unitId>
                <ns2:dateFrom>2017-03-01+03:00</ns2:dateFrom>
                <ns2:dateTo>2017-03-31+03:00</ns2:dateTo>
                <ns2:accountIds>12345678</ns2:accountIds>
                <ns2:currencyCode>RUB</ns2:currencyCode>
                <ns2:groupByCurrency>false</ns2:groupByCurrency>
                <ns2:operationTypeCategory>BUSINESS</ns2:operationTypeCategory>
                <ns2:categoryDetails>true</ns2:categoryDetails>
            </ns2:GetTurnoverListRequest>
        </ns2:AsyncRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response (status=CREATED):

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
```

```
    <SOAP-ENV:Body>
        <ns2:AsyncResponse xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd">
            <ns2:asyncId>333333</ns2:asyncId>
            <ns2:asyncStatus>CREATED</ns2:asyncStatus>
            <ns2:expirationDate>2017-07-24T14:20:06.887+03:00</ns2:expirationDate>
        </ns2:AsyncResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request (send request):

```
{
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
          "Password": "PASSWORD"
        }
      }
    },
    "Body": {
      "AsyncRequest": {
        "GetTurnoverListRequest": {
          "unitId": 123456,
          "dateFrom": "2017-03-01T00:00:00.000+03:00",
          "dateTo": "2017-03-31T00:00:00.000+03:00",
          "accountIds": [
            12345678
          ],
          "currencyCode": "RUB",
          "groupByCurrency": false,
          "categoryDetails": true,
          "operationTypeCategory": "BUSINESS"
        }
      }
    }
  }
}
```

JSON response (status=CREATED):

```
{
  "Envelope": {
    "Body": {
      "AsyncResponse": {
        "expirationDate": "2017-07-24T14:44:15.122+03:00",
        "asyncId": 333333,
        "asyncStatus": "CREATED"
      }
    }
  }
}
```

SOAP request (get result by asyncId):

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken
                    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-1500891607067-1295353757"
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
```

```
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:AsyncRequest xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd"
 xmlns="">
            <ns2:asyncId>333333</ns2:asyncId>
        </ns2:AsyncRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response (status=INPROGRESS):

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:AsyncResponse xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd">
            <ns2:asyncId>333333</ns2:asyncId>
            <ns2:asyncStatus>INPROGRESS</ns2:asyncStatus>
            <ns2:expirationDate>2017-07-24T14:39:55.000+03:00</ns2:expirationDate>
        </ns2:AsyncResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response (result):

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:AsyncResponse xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd">
            <ns2:GetTurnoverListResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
                <ns2:groupByCurrency>false</ns2:groupByCurrency>
                <ns2:turnover>
                    <ns2:date>2017-03-01+03:00</ns2:date>
                    <ns2:account>12345678</ns2:account>
                    <ns2:currency>RUB</ns2:currency>
                    <ns2:operationTypeCategory>BUSINESS</ns2:operationTypeCategory>
                    <ns2:incomeAmount>253675.62</ns2:incomeAmount>
                    <ns2:incomeCommission>-2924.27</ns2:incomeCommission>
                    <ns2:incomeTransactionsCount>2547</ns2:incomeTransactionsCount>
                    <ns2:expenseAmount>3999.58</ns2:expenseAmount>
                    <ns2:expensesIncludingCommission>5.19</
ns2:expensesIncludingCommission>
                    <ns2:expensesExtraCommission>115.66</ns2:expensesExtraCommission>
                    <ns2:expenseTransactionsCount>323</ns2:expenseTransactionsCount>
                    <ns2:total>246636.11</ns2:total>
                    <ns2:openingBalance>0</ns2:openingBalance>
                    <ns2:closingBalance>0</ns2:closingBalance>
                </ns2:turnover>
            </ns2:GetTurnoverListResponse>
        </ns2:AsyncResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request (get result by asyncId):

```
{
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
          "Password": "PASSWORD"
        }
      }
    },
    "Body": {
      "AsyncRequest": {
```

```
        "asyncId": 333333
      }
    }
  }
}
```

JSON response (status=INPROGRESS):

```
{
  "Envelope": {
    "Body": {
      "AsyncResponse": {
        "expirationDate": "2017-07-24T14:44:15.000+03:00",
        "asyncId": 333333,
        "asyncStatus": "INPROGRESS"
      }
    }
  }
}
```

JSON response (result):

```
{
  "Envelope": {
    "Body": {
      "AsyncResponse": {
        "GetTurnoverListResponse": {
          "turnover": [
            {
              "total": 246636.11,
              "incomeCommission": -2924.27,
              "expenseAmount": 3999.58,
              "expenseTransactionsCount": 323,
              "date": "2017-03-01T00:00:00.000+03:00",
              "operationTypeCategory": "BUSINESS",
              "currency": "RUB",
              "expensesIncludingCommission": 5.19,
              "openingBalance": 0,
              "incomeAmount": 253675.62,
              "account": "12345678",
              "expensesExtraCommission": 115.66,
              "incomeTransactionsCount": 2547,
              "closingBalance": 0
            }
          ],
          "groupByCurrency": false
        }
      }
    }
  }
}
```

GetFinancialFlowsList - financial flows

Request for getting the financial flow information.

Examples

SOAP request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken
                    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-1491386045251210836932 3"
```

```
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">
                    PASSWORD
                </wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:GetFinancialFlowsListRequest xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd" xmlns="">
            <ns2:unitId>123456</ns2:unitId>
            <ns2:dateFrom>2017-03-01+03:00</ns2:dateFrom>
            <ns2:dateTo>2017-03-31+03:00</ns2:dateTo>
            <ns2:accountIds>12345678</ns2:accountIds>
            <ns2:currencyCode>RUB</ns2:currencyCode>
            <ns2:operationTypeCategory>BUSINESS</ns2:operationTypeCategory>
            <ns2:operationAmountType>INCOME</ns2:operationAmountType>
            <ns2:categoryDetails>true</ns2:categoryDetails>
            <ns2:operationStatusState>COMPLETED</ns2:operationStatusState>
        </ns2:GetFinancialFlowsListRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:GetFinancialFlowsListResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
            <ns2:financialFlow>
                <ns2:date>2017-03</ns2:date>
                <ns2:payerSideAccess>false</ns2:payerSideAccess>
                <ns2:payerFlowId>654321</ns2:payerFlowId>
                <ns2:payerFlowName>Payer name</ns2:payerFlowName>
                <ns2:payerCurrencyCode>RUB</ns2:payerCurrencyCode>
                <ns2:payeeSideAccess>true</ns2:payeeSideAccess>
                <ns2:payeeFlowId>123456</ns2:payeeFlowId>
                <ns2:payeeFlowName>Payee name</ns2:payeeFlowName>
                <ns2:payeeCurrencyCode>RUB</ns2:payeeCurrencyCode>
                <ns2:payeeCredited>1481.4</ns2:payeeCredited>
                <ns2:payeeFee>180</ns2:payeeFee>
                <ns2:transactionsCount>12</ns2:transactionsCount>
                <ns2:operationTypeCategory>BUSINESS</ns2:operationTypeCategory>
                <ns2:operationStatusState>COMPLETED</ns2:operationStatusState>
            </ns2:financialFlow>
        </ns2:GetFinancialFlowsListResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
          "Password": "PASSWORD"
        }
      }
    },
    "Body": {
      "GetFinancialFlowsListRequest": {
        "unitId": 123456,
        "dateFrom": "2017-03-01T00:00:00.000+03:00",
        "dateTo": "2017-03-31T00:00:00.000+03:00",
```

```
        "accountIds": [
          12345678
        ],
        "currencyCode": "RUB",
        "operationTypeCategory": "BUSINESS",
        "operationAmountType": "INCOME",
        "categoryDetails": true,
        "operationStatusState": "COMPLETED"
      }
    }
  }
}
```

JSON response:

```
{
  "Envelope": {
    "Body": {
      "GetFinancialFlowsListResponse": {
        "financialFlow": [
          {
            "payeeFlowId": 123456,
            "payerCurrencyCode": "RUB",
            "date": "2017-03",
            "operationTypeCategory": "BUSINESS",
            "transactionsCount": 12,
            "payerSideAccess": false,
            "payeeCurrencyCode": "RUB",
            "payeeFee": 180,
            "payeeSideAccess": true,
            "operationStatusState": "COMPLETED",
            "payeeCredited": 1481.4,
            "payerFlowName": "Payer name",
            "payeeFlowName": "Payee name",
            "payerFlowId": 654321
          }
        ]
      }
    }
  }
}
```

GetFinancialFlowsList (AsyncRequest) - financial flows

Asynchronous request for getting the financial flow information.

Examples

SOAP request (send request):

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken
                    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-1500901469981-677067966"
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:AsyncRequest xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd"
 xmlns="">
```

```
            <ns2:GetFinancialFlowsListRequest xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd" xmlns="">
                <ns2:unitId>123456</ns2:unitId>
                <ns2:dateFrom>2017-03-01+03:00</ns2:dateFrom>
                <ns2:dateTo>2017-03-31+03:00</ns2:dateTo>
                <ns2:accountIds>12345678</ns2:accountIds>
                <ns2:currencyCode>RUB</ns2:currencyCode>
                <ns2:operationTypeCategory>BUSINESS</ns2:operationTypeCategory>
                <ns2:operationAmountType>INCOME</ns2:operationAmountType>
                <ns2:categoryDetails>true</ns2:categoryDetails>
                <ns2:operationStatusState>COMPLETED</ns2:operationStatusState>
            </ns2:GetFinancialFlowsListRequest>
        </ns2:AsyncRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response (status=CREATED):

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:AsyncResponse xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd">
            <ns2:asyncId>333333</ns2:asyncId>
            <ns2:asyncStatus>CREATED</ns2:asyncStatus>
            <ns2:expirationDate>2017-07-24T18:04:32.357+03:00</ns2:expirationDate>
        </ns2:AsyncResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request (send request):

```
{
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
          "Password": "PASSWORD"
        }
      }
    },
    "Body": {
      "AsyncRequest": {
        "GetFinancialFlowsListRequest": {
          "unitId": 123456,
          "dateFrom": "2017-03-01T00:00:00.000+03:00",
          "dateTo": "2017-03-31T00:00:00.000+03:00",
          "accountIds": [
            12345678
          ],
          "currencyCode": "RUB",
          "operationTypeCategory": "BUSINESS",
          "operationAmountType": "INCOME",
          "categoryDetails": true,
          "operationStatusState": "COMPLETED"
        }
      }
    }
  }
}
```

JSON response (status=CREATED):

```
{
  "Envelope": {
    "Body": {
      "AsyncResponse": {
        "expirationDate": "2017-07-24T18:07:12.536+03:00",
        "asyncId": 333333,
```

```
        "asyncStatus": "CREATED"
      }
    }
  }
}
```

SOAP request (get result by asyncId):

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken
                    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-1500901472966-1144616882"
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:AsyncRequest xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd"
 xmlns="">
            <ns2:asyncId>333333</ns2:asyncId>
        </ns2:AsyncRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response (status=INPROGRESS):

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:AsyncResponse xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd">
            <ns2:asyncId>333333</ns2:asyncId>
            <ns2:asyncStatus>INPROGRESS</ns2:asyncStatus>
            <ns2:expirationDate>2017-07-24T18:04:32.000+03:00</ns2:expirationDate>
        </ns2:AsyncResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response (result):

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:AsyncResponse xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd">
            <ns2:GetFinancialFlowsListResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
                <ns2:financialFlow>
                    <ns2:date>2017-03</ns2:date>
                    <ns2:payerSideAccess>false</ns2:payerSideAccess>
                    <ns2:payerFlowId>654321</ns2:payerFlowId>
                    <ns2:payerFlowName>Payer name</ns2:payerFlowName>
                    <ns2:payerCurrencyCode>RUB</ns2:payerCurrencyCode>
                    <ns2:payeeSideAccess>true</ns2:payeeSideAccess>
                    <ns2:payeeFlowId>123456</ns2:payeeFlowId>
                    <ns2:payeeFlowName>Payee name</ns2:payeeFlowName>
                    <ns2:payeeCurrencyCode>RUB</ns2:payeeCurrencyCode>
                    <ns2:payeeCredited>1481.4</ns2:payeeCredited>
                    <ns2:payeeFee>180</ns2:payeeFee>
                    <ns2:transactionsCount>12</ns2:transactionsCount>
                    <ns2:operationTypeCategory>BUSINESS</ns2:operationTypeCategory>
                    <ns2:operationStatusState>COMPLETED</ns2:operationStatusState>
                </ns2:financialFlow>
```

```
            </ns2:GetFinancialFlowsListResponse>
        </ns2:AsyncResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request (get result by asyncId):

```
{
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
          "Password": "PASSWORD"
        }
      }
    },
    "Body": {
      "AsyncRequest": {
        "asyncId": 333333
      }
    }
  }
}
```

JSON response (status=INPROGRESS):

```
{
  "Envelope": {
    "Body": {
      "AsyncResponse": {
        "expirationDate": "2017-07-24T18:07:12.000+03:00",
        "asyncId": 333333,
        "asyncStatus": "INPROGRESS"
      }
    }
  }
}
```

JSON response (result):

```
{
  "Envelope": {
    "Body": {
      "AsyncResponse": {
        "GetFinancialFlowsListResponse": {
          "financialFlow": [
            {
              "payeeFlowId": 123456,
              "payerCurrencyCode": "RUB",
              "date": "2017-03",
              "operationTypeCategory": "BUSINESS",
              "transactionsCount": 12,
              "payerSideAccess": false,
              "payeeCurrencyCode": "RUB",
              "payeeFee": 180,
              "payeeSideAccess": true,
              "operationStatusState": "COMPLETED",
              "payeeCredited": 1481.4,
              "payerFlowName": "Payer name",
              "payeeFlowName": "Payee name",
              "payerFlowId": 654321
            }
          ]
        }
      }
    }
  }
}
```

AccountStatement

Request for getting an account statement.

See: *AccountStatement Endpoint* on page 176.

Examples

SOAP request (common information and first page with operations):

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd"
                       SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken
                    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-159437199330116126427 3"
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password
                        Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordText">
                    PASSWORD
                </wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:AccountStatementRequest xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd" xmlns="">
            <ns2:accountId>12345678</ns2:accountId>
            <ns2:dateFrom>2019-02-06+03:00</ns2:dateFrom>
            <ns2:dateTo>2019-02-06+03:00</ns2:dateTo>
            <ns2:showActions>true</ns2:showActions>
            <ns2:pageSize>2</ns2:pageSize>
            <ns2:showPaging>true</ns2:showPaging>
        </ns2:AccountStatementRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response (common information and first page with operations):

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:AccountStatementResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
            <ns2:ownerAccount>30232810608012345678</ns2:ownerAccount>
            <ns2:contractDate>2017-10-28T00:00:00.000+03:00</ns2:contractDate>
            <ns2:ownerAccountCurrency>RUB</ns2:ownerAccountCurrency>
            <ns2:ownerName>NCO «MONETA» (LLC)</ns2:ownerName>
            <ns2:dateFrom>2019-02-06T00:00:00.000+03:00</ns2:dateFrom>
            <ns2:dateTo>2019-02-06T23:59:59.999+03:00</ns2:dateTo>
            <ns2:lastOperationDate>2019-02-02T00:00:00.000+03:00</
ns2:lastOperationDate>
            <ns2:now>2020-07-10T12:06:34.221+03:00</ns2:now>
            <ns2:inSaldo>108232.77</ns2:inSaldo>
            <ns2:outSaldo>108408.53</ns2:outSaldo>
            <ns2:debitAmountTotal>11.93</ns2:debitAmountTotal>
            <ns2:creditAmountTotal>187.69</ns2:creditAmountTotal>
            <ns2:totalSize>10</ns2:totalSize>
            <ns2:orders>
                <ns2:date>2019-02-06T08:48:29.000+03:00</ns2:date>
                <ns2:operationId>1000256633</ns2:operationId>
                <ns2:operationView>17</ns2:operationView>
                <ns2:correspondentName>HKO "MOHETA"(OOO)</ns2:correspondentName>
                <ns2:correspondentAccountId>87654321</ns2:correspondentAccountId>
```

```
                <ns2:correspondentAccount/>
                <ns2:debitAmount>0.14</ns2:debitAmount>
                <ns2:payerINN>1234567890</ns2:payerINN>
                <ns2:payeeINN>1215192632</ns2:payeeINN>
                <ns2:documentNumber>1000256633</ns2:documentNumber>
                <ns2:BIK>048860734</ns2:BIK>
                <ns2:description>Комиссия за осуществление переводов денежных
 средств ...</ns2:description>
                <ns2:actions>form0401067</ns2:actions>
            </ns2:orders>
            <ns2:orders>
                <ns2:date>2019-02-06T00:00:00.000+03:00</ns2:date>
                <ns2:operationId>1000256633</ns2:operationId>
                <ns2:operationView>1</ns2:operationView>
                <ns2:correspondentName>Общество с ограниченной ответственностью "..."</
ns2:correspondentName>
                <ns2:correspondentAccountId>87654321</ns2:correspondentAccountId>
                <ns2:correspondentAccount>40702770383986912345</
ns2:correspondentAccount>
                <ns2:debitAmount>4.45</ns2:debitAmount>
                <ns2:payerINN>1234567890</ns2:payerINN>
                <ns2:payeeINN>1234567890</ns2:payeeINN>
                <ns2:documentNumber>269891</ns2:documentNumber>
                <ns2:BIK>044525214</ns2:BIK>
                <ns2:description>Назначение платежа</ns2:description>
                <ns2:actions>receipt</ns2:actions>
            </ns2:orders>
            <ns2:paging>
                <ns2:date>2019-02-06T08:48:29.000+03:00</ns2:date>
                <ns2:id>1000256633</ns2:id>
            </ns2:paging>
        </ns2:AccountStatementResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP request (no common information and operations paging):

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd"
                       SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken
                    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-159437199439627907940 5"
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password
                        Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordText">
                    PASSWORD
                </wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:AccountStatementRequest xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd" xmlns="">
            <ns2:accountId>12345678</ns2:accountId>
            <ns2:dateFrom>2019-02-06+03:00</ns2:dateFrom>
            <ns2:dateTo>2019-02-06+03:00</ns2:dateTo>
            <ns2:showActions>true</ns2:showActions>
            <ns2:pageSize>2</ns2:pageSize>
            <ns2:showPaging>true</ns2:showPaging>
            <ns2:paging>
                <ns2:date>2019-02-06T08:48:29.000+03:00</ns2:date>
                <ns2:id>1000256633</ns2:id>
            </ns2:paging>
```

```
        </ns2:AccountStatementRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response (no common information and operations paging):

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:AccountStatementResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
            <ns2:orders>
                <ns2:date>2019-02-06T00:00:00.000+03:00</ns2:date>
                <ns2:operationId>1000256634</ns2:operationId>
                <ns2:operationView>1</ns2:operationView>
                <ns2:correspondentName>ПАО "ПРОМСВЯЗЬБАНК" г. Москва</
ns2:correspondentName>
                <ns2:correspondentAccountId>87654321</ns2:correspondentAccountId>
                <ns2:correspondentAccount>40911810490000012345</
ns2:correspondentAccount>
                <ns2:creditAmount>183.8</ns2:creditAmount>
                <ns2:payerINN>1234567890</ns2:payerINN>
                <ns2:payeeINN>1234567890</ns2:payeeINN>
                <ns2:documentNumber>537152</ns2:documentNumber>
                <ns2:BIK>044525555</ns2:BIK>
                <ns2:description>Назначение платежа</ns2:description>
                <ns2:actions>receipt</ns2:actions>
            </ns2:orders>
            <ns2:orders>
                <ns2:date>2019-02-06T08:50:18.000+03:00</ns2:date>
                <ns2:operationId>1000256634</ns2:operationId>
                <ns2:operationView>17</ns2:operationView>
                <ns2:correspondentName>НКО "МОНЕТА"(ООО)</ns2:correspondentName>
                <ns2:correspondentAccountId>87654321</ns2:correspondentAccountId>
                <ns2:correspondentAccount/>
                <ns2:debitAmount>1.84</ns2:debitAmount>
                <ns2:payerINN>1234567890</ns2:payerINN>
                <ns2:payeeINN>1215192632</ns2:payeeINN>
                <ns2:documentNumber>1000256634</ns2:documentNumber>
                <ns2:BIK>048860734</ns2:BIK>
                <ns2:description>Комиссия за осуществление переводов денежных
 средств ...</ns2:description>
                <ns2:actions>form0401067</ns2:actions>
            </ns2:orders>
            <ns2:paging>
                <ns2:date>2019-02-06T08:50:18.000+03:00</ns2:date>
                <ns2:id>1000256634</ns2:id>
            </ns2:paging>
        </ns2:AccountStatementResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request (common information and first page with operations):

```
{
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
          "Password": "PASSWORD"
        }
      }
    },
    "Body": {
      "AccountStatementRequest": {
        "accountId": 12345678,
        "dateFrom": "2019-02-06T00:00:00.000+03:00",
        "dateTo": "2019-02-06T00:00:00.000+03:00",
        "showActions": true,
```

```
        "showPaging": true,
        "pageSize": 2
      }
    }
  }
}
```

JSON response (common information and first page with operations):

```
{
  "Envelope": {
    "Body": {
      "AccountStatementResponse": {
        "contractDate": "2017-10-28T00:00:00.000+03:00",
        "inSaldo": 108232.77,
        "creditAmountTotal": 187.69,
        "ownerAccount": "30232810608012345678",
        "dateFrom": "2019-02-06T00:00:00.000+03:00",
        "lastOperationDate": "2019-02-02T00:00:00.000+03:00",
        "debitAmountTotal": 11.93,
        "totalSize": 10,
        "ownerName": "NCO «MONETA» (LLC)",
        "outSaldo": 108408.53,
        "now": "2020-07-10T12:15:53.812+03:00",
        "dateTo": "2019-02-06T23:59:59.999+03:00",
        "ownerAccountCurrency": "RUB",
        "orders": [
          {
            "date": "2019-02-06T08:48:29.000+03:00",
            "BIK": "048860734",
            "documentNumber": "1000256633",
            "description": "Комиссия за осуществление переводов денежных средств ...",
            "payerINN": "1234567890",
            "debitAmount": 0.14,
            "operationView": 17,
            "correspondentAccount": "",
            "operationId": 1000256633,
            "correspondentAccountId": 87654321,
            "payeeINN": "1215192632",
            "correspondentName": "НКО \"МОНЕТА\"(ООО)",
            "actions": [
              "form0401067"
            ]
          },
          {
            "date": "2019-02-06T00:00:00.000+03:00",
            "BIK": "044525214",
            "documentNumber": "269891",
            "description": "Назначение платежа",
            "payerINN": "1234567890",
            "debitAmount": 4.45,
            "operationView": 1,
            "correspondentAccount": "40702770383986912345",
            "operationId": 1000256633,
            "correspondentAccountId": 87654321,
            "payeeINN": "1234567890",
            "correspondentName": "Общество с ограниченной ответственностью \"...\"",
            "actions": [
              "receipt"
            ]
          }
        ],
        "paging": {
          "date": "2019-02-06T08:48:29.000+03:00",
          "id": 1000256633
        }
      }
    }
  }
}
```

JSON request (no common information and operations paging):

```
{
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
          "Password": "PASSWORD"
        }
      }
    },
    "Body": {
      "AccountStatementRequest": {
        "accountId": 12345678,
        "dateFrom": "2019-02-06T00:00:00.000+03:00",
        "dateTo": "2019-02-06T00:00:00.000+03:00",
        "showActions": true,
        "showPaging": true,
        "pageSize": 2,
        "paging": {
          "date": "2019-02-06T08:48:29.000+03:00",
          "id": 1000256633
        }
      }
    }
  }
}
```

JSON response (no common information and operations paging):

```
{
  "Envelope": {
    "Body": {
      "AccountStatementResponse": {
        "orders": [
          {
            "date": "2019-02-06T00:00:00.000+03:00",
            "BIK": "044525555",
            "documentNumber": "537152",
            "description": "Назначение платежа",
            "payerINN": "1234567890",
            "operationView": 1,
            "correspondentAccount": "40911810490000012345",
            "operationId": 1000256634,
            "correspondentAccountId": 87654321,
            "payeeINN": "1234567890",
            "correspondentName": "ПАО \"ПРОМСВЯЗЬБАНК\" г. Москва",
            "creditAmount": 183.8,
            "actions": [
              "receipt"
            ]
          },
          {
            "date": "2019-02-06T08:50:18.000+03:00",
            "BIK": "048860734",
            "documentNumber": "1000256634",
            "description": "Комиссия за осуществление переводов денежных средств ...",
            "payerINN": "1234567890",
            "debitAmount": 1.84,
            "operationView": 17,
            "correspondentAccount": "",
            "operationId": 1000256634,
            "correspondentAccountId": 87654321,
            "payeeINN": "1215192632",
            "correspondentName": "НКО \"МОНЕТА\"(ООО)",
            "actions": [
              "form0401067"
            ]
          }
```

```
        ],
        "paging": {
          "date": "2019-02-06T08:50:18.000+03:00",
          "id": 1000256634
        }
      }
    }
  }
}
```

Operation templates

CreateOperationTemplate

CreateOperationTemplate Endpoint on page 131

Common operation templates

Examples

SOAP min request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken
                    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-1495190341499-967371235"
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:CreateOperationTemplateRequest xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd" xmlns="">
            <ns2:type>COMMON</ns2:type>
            <ns2:name>TemplateName</ns2:name>
            <ns2:payer>12345678</ns2:payer>
            <ns2:payee>87654321</ns2:payee>
            <ns2:commonParameters>
                <ns2:amount>10.00</ns2:amount>
                <ns2:isPayerAmount>true</ns2:isPayerAmount>
            </ns2:commonParameters>
        </ns2:CreateOperationTemplateRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON min request:

```
{
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
          "Password": "PASSWORD"
        }
      }
    },
    "Body": {
```

```
      "CreateOperationTemplateRequest": {
        "payer": 12345678,
        "name": "TemplateName",
        "payee": 87654321,
        "commonParameters": {
          "amount": 10.0,
          "isPayerAmount": true
        },
        "operationInfo": [],
        "type": "COMMON"
      }
    }
  }
}
```

SOAP full request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken
                    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-1495196689869-1728719953"
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:CreateOperationTemplateRequest xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd" xmlns="">
            <ns2:unitId>11111111</ns2:unitId>
            <ns2:type>COMMON</ns2:type>
            <ns2:name>TemplateName</ns2:name>
            <ns2:payer>12345678</ns2:payer>
            <ns2:payee>87654321</ns2:payee>
            <ns2:description>Template description</ns2:description>
            <ns2:tags>transfer,John,work</ns2:tags>
            <ns2:favorite>true</ns2:favorite>
            <ns2:commonParameters>
                <ns2:amount>10.00</ns2:amount>
                <ns2:isPayerAmount>true</ns2:isPayerAmount>
            </ns2:commonParameters>
        </ns2:CreateOperationTemplateRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON full request:

```
{
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
          "Password": "PASSWORD"
        }
      }
    },
    "Body": {
      "CreateOperationTemplateRequest": {
        "unitId": 11111111,
        "type": "COMMON",
        "name": "TemplateName",
        "payer": 12345678,
```

```
        "payee": 87654321,
        "description": "Template description",
        "tags": "transfer,John,work",
        "favorite": true,
        "commonParameters": {
          "amount": 10.0,
          "isPayerAmount": true
        },
        "operationInfo": []
      }
    }
  }
}
```

Regular operation templates

Examples

SOAP request (amountType = AMOUNT):

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken
                    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-1495197812473-214663471"
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:CreateOperationTemplateRequest xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd" xmlns="">
            <ns2:unitId>11111111</ns2:unitId>
            <ns2:type>REGULAR</ns2:type>
            <ns2:name>Template name</ns2:name>
            <ns2:payer>12345678</ns2:payer>
            <ns2:payee>87654321</ns2:payee>
            <ns2:regularParameters>
                <ns2:amountInfo>
                    <ns2:type>AMOUNT</ns2:type>
                    <ns2:amount>
                        <ns2:amount>10.00</ns2:amount>
                        <ns2:isPayerAmount>true</ns2:isPayerAmount>
                    </ns2:amount>
                </ns2:amountInfo>
                <ns2:timeInfo>
                    <ns2:type>EVERY_DAY</ns2:type>
                    <ns2:startDateTime>2099-09-04T13:00:00.000+03:00</
ns2:startDateTime>
                    <ns2:endDateTime>2099-11-26T02:00:00.000+03:00</ns2:endDateTime>
                </ns2:timeInfo>
                <ns2:reminderInfo>
                    <ns2:remind>true</ns2:remind>
                    <ns2:hoursBeforeExecution>25</ns2:hoursBeforeExecution>
                </ns2:reminderInfo>
            </ns2:regularParameters>
            <ns2:paymentPassword>
                <ns2:paymentPassword>11111</ns2:paymentPassword>
            </ns2:paymentPassword>
        </ns2:CreateOperationTemplateRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request (amountType = AMOUNT):

```
{
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
          "Password": "PASSWORD"
        }
      }
    },
    "Body": {
      "CreateOperationTemplateRequest": {
        "payer": 12345678,
        "name": "Template name",
        "payee": 87654321,
        "operationInfo": [],
        "type": "REGULAR",
        "unitId": 11111111,
        "paymentPassword": {
          "paymentPassword": "11111"
        },
        "regularParameters": {
          "timeInfo": {
            "startDateTime": "2017-09-04T13:00:00.000+03:00",
            "type": "EVERY_DAY",
            "endDateTime": "2017-11-26T00:00:00.000+03:00"
          },
          "amountInfo": {
            "amount": {
              "amount": 10.0,
              "isPayerAmount": true
            },
            "type": "AMOUNT"
          },
          "reminderInfo": {
            "remind": true,
            "hoursBeforeExecution": 25
          }
        }
      }
    }
  }
}
```

SOAP request (amountType = BALANCE):

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken
                    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-1495197814475-906210923"
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:CreateOperationTemplateRequest xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd" xmlns="">
            <ns2:unitId>11111111</ns2:unitId>
            <ns2:type>REGULAR</ns2:type>
            <ns2:name>Template name</ns2:name>
```

```
            <ns2:payer>12345678</ns2:payer>
            <ns2:payee>87654321</ns2:payee>
            <ns2:regularParameters>
                <ns2:amountInfo>
                    <ns2:type>BALANCE</ns2:type>
                </ns2:amountInfo>
                <ns2:timeInfo>
                    <ns2:type>EVERY_WORKDAY</ns2:type>
                    <ns2:startDateTime>2099-09-04T13:00:00.000+03:00</
ns2:startDateTime>
                    <ns2:endDateTime>2099-11-26T02:00:00.000+03:00</ns2:endDateTime>
                </ns2:timeInfo>
            </ns2:regularParameters>
            <ns2:paymentPassword>
                <ns2:paymentPassword>11111</ns2:paymentPassword>
            </ns2:paymentPassword>
        </ns2:CreateOperationTemplateRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request (amountType = BALANCE):

```
{
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
          "Password": "PASSWORD"
        }
      }
    },
    "Body": {
      "CreateOperationTemplateRequest": {
        "paymentPassword": {
          "paymentPassword": "11111"
        },
        "regularParameters": {
          "amountInfo": {
            "type": "BALANCE"
          },
          "timeInfo": {
            "type": "EVERY_WORKDAY",
            "startDateTime": "2099-09-04T13:00:00.000+03:00",
            "endDateTime": "2099-11-26T02:00:00.000+03:00"
          }
        },
        "payer": 12345678,
        "name": "Template name",
        "payee": 87654321,
        "operationInfo": [],
        "type": "REGULAR",
        "unitId": 11111111
      }
    }
  }
}
```

SOAP request (amountType = PAYMENTS):

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken
                    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-1495197814475-906210923"
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
```

```
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:CreateOperationTemplateRequest xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd" xmlns="">
            <ns2:unitId>11111111</ns2:unitId>
            <ns2:type>REGULAR</ns2:type>
            <ns2:name>Template name</ns2:name>
            <ns2:payer>12345678</ns2:payer>
            <ns2:payee>87654321</ns2:payee>
            <ns2:regularParameters>
                <ns2:amountInfo>
                    <ns2:type>PAYMENTS</ns2:type>
                </ns2:amountInfo>
                <ns2:timeInfo>
                    <ns2:type>EVERY_MONTH</ns2:type>
                    <ns2:startDateTime>2099-09-04T13:00:00.000+03:00</
ns2:startDateTime>
                    <ns2:endDateTime>2099-11-26T02:00:00.000+03:00</ns2:endDateTime>
                </ns2:timeInfo>
                <ns2:operationsReportNotifications>
                    <ns2:id>222222</ns2:id>
                    <ns2:selected>true</ns2:selected>
                </ns2:operationsReportNotifications>
            </ns2:regularParameters>
            <ns2:paymentPassword>
                <ns2:paymentPassword>11111</ns2:paymentPassword>
            </ns2:paymentPassword>
        </ns2:CreateOperationTemplateRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request (amountType = PAYMENTS):

```
{
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
          "Password": "PASSWORD"
        }
      }
    },
    "Body": {
      "CreateOperationTemplateRequest": {
        "paymentPassword": {
          "paymentPassword": "11111"
        },
        "regularParameters": {
          "amountInfo": {
            "type": "PAYMENTS"
          },
          "timeInfo": {
            "type": "EVERY_MONTH",
            "startDateTime": "2099-09-04T13:00:00.000+03:00",
            "endDateTime": "2099-11-26T02:00:00.000+03:00"
          },
          "operationsReportNotifications": [
            {
              "id": 222222,
              "selected": true
            }
          ]
        },
        "payer": 12345678,
```

```
        "name": "Template name",
        "payee": 87654321,
        "operationInfo": [],
        "type": "REGULAR",
        "unitId": 11111111
      }
    }
  }
}
```

SOAP request (amountType = RANGE):

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken
                    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-149519781545048507591 5"
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:CreateOperationTemplateRequest xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd" xmlns="">
            <ns2:unitId>11111111</ns2:unitId>
            <ns2:type>REGULAR</ns2:type>
            <ns2:name>Template name</ns2:name>
            <ns2:payer>12345678</ns2:payer>
            <ns2:payee>87654321</ns2:payee>
            <ns2:regularParameters>
                <ns2:amountInfo>
                    <ns2:type>RANGE</ns2:type>
                    <ns2:range>
                        <ns2:amountMinValue>10.00</ns2:amountMinValue>
                        <ns2:amountMaxValue>20.00</ns2:amountMaxValue>
                    </ns2:range>
                </ns2:amountInfo>
                <ns2:timeInfo>
                    <ns2:type>EVERY_LAST_DAY_OF_MONTH</ns2:type>
                    <ns2:startDateTime>2099-09-30T13:00:00.000+03:00</
ns2:startDateTime>
                    <ns2:endDateTime>2099-11-26T02:00:00.000+03:00</ns2:endDateTime>
                </ns2:timeInfo>
            </ns2:regularParameters>
            <ns2:paymentPassword>
                <ns2:paymentPassword>11111</ns2:paymentPassword>
            </ns2:paymentPassword>
        </ns2:CreateOperationTemplateRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request (amountType = RANGE):

```
{
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
          "Password": "PASSWORD"
        }
      }
    },
```

```
    "Body": {
      "CreateOperationTemplateRequest": {
        "paymentPassword": {
          "paymentPassword": "11111"
        },
        "regularParameters": {
          "amountInfo": {
            "type": "RANGE",
            "range": {
              "amountMinValue": 10.0,
              "amountMaxValue": 20.0
            }
          },
          "timeInfo": {
            "type": "EVERY_LAST_DAY_OF_MONTH",
            "startDateTime": "2099-09-30T13:00:00.000+03:00",
            "endDateTime": "2099-11-26T02:00:00.000+03:00"
          }
        },
        "payer": 12345678,
        "name": "Template name",
        "payee": 87654321,
        "operationInfo": [],
        "type": "REGULAR",
        "unitId": 11111111
      }
    }
  }
}
```

SOAP request (amountType = REST):

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken
                    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-1495197816260838370495"
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:CreateOperationTemplateRequest xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd" xmlns="">
            <ns2:unitId>11111111</ns2:unitId>
            <ns2:type>REGULAR</ns2:type>
            <ns2:name>Template name</ns2:name>
            <ns2:payer>12345678</ns2:payer>
            <ns2:payee>87654321</ns2:payee>
            <ns2:regularParameters>
                <ns2:amountInfo>
                    <ns2:type>REST</ns2:type>
                    <ns2:rest>
                        <ns2:amount>10.00</ns2:amount>
                    </ns2:rest>
                </ns2:amountInfo>
                <ns2:timeInfo>
                    <ns2:type>ONCE</ns2:type>
                    <ns2:startDateTime>2099-09-04T13:00:00.000+03:00</
ns2:startDateTime>
                </ns2:timeInfo>
            </ns2:regularParameters>
            <ns2:paymentPassword>
```

```
                <ns2:paymentPassword>11111</ns2:paymentPassword>
            </ns2:paymentPassword>
        </ns2:CreateOperationTemplateRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request (amountType = REST):

```
{
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
          "Password": "PASSWORD"
        }
      }
    },
    "Body": {
      "CreateOperationTemplateRequest": {
        "paymentPassword": {
          "paymentPassword": "11111"
        },
        "regularParameters": {
          "amountInfo": {
            "type": "REST",
            "rest": {
              "amount": 10.0
            }
          },
          "timeInfo": {
            "type": "ONCE",
            "startDateTime": "2099-09-04T13:00:00.000+03:00"
          }
        },
        "payer": 12345678,
        "name": "Template name",
        "payee": 87654321,
        "operationInfo": [],
        "type": "REGULAR",
        "unitId": 11111111
      }
    }
  }
}
```

SOAP request (amountType = CREDIT):

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken
                    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-1495197814475-906210923"
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:CreateOperationTemplateRequest xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd" xmlns="">
            <ns2:unitId>11111111</ns2:unitId>
            <ns2:type>REGULAR</ns2:type>
            <ns2:name>Template name</ns2:name>
```

```
            <ns2:payer>12345678</ns2:payer>
            <ns2:payee>87654321</ns2:payee>
            <ns2:regularParameters>
                <ns2:amountInfo>
                    <ns2:type>CREDIT</ns2:type>
                </ns2:amountInfo>
                <ns2:timeInfo>
                    <ns2:type>EVERY_WEEK</ns2:type>
                    <ns2:startDateTime>2099-09-04T13:00:00.000+03:00</
ns2:startDateTime>
                    <ns2:endDateTime>2099-11-26T02:00:00.000+03:00</ns2:endDateTime>
                </ns2:timeInfo>
                <ns2:operationsReportNotifications>
                    <ns2:id>222222</ns2:id>
                    <ns2:selected>true</ns2:selected>
                </ns2:operationsReportNotifications>
            </ns2:regularParameters>
            <ns2:paymentPassword>
                <ns2:paymentPassword>11111</ns2:paymentPassword>
            </ns2:paymentPassword>
        </ns2:CreateOperationTemplateRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request (amountType = CREDIT):

```
{
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
          "Password": "PASSWORD"
        }
      }
    },
    "Body": {
      "CreateOperationTemplateRequest": {
        "paymentPassword": {
          "paymentPassword": "11111"
        },
        "regularParameters": {
          "amountInfo": {
            "type": "CREDIT"
          },
          "timeInfo": {
            "type": "EVERY_WEEK",
            "startDateTime": "2099-09-04T13:00:00.000+03:00",
            "endDateTime": "2099-11-26T02:00:00.000+03:00"
          },
          "operationsReportNotifications": [
            {
              "id": 222222,
              "selected": true
            }
          ]
        },
        "payer": 12345678,
        "name": "Template name",
        "payee": 87654321,
        "operationInfo": [],
        "type": "REGULAR",
        "unitId": 11111111
      }
    }
  }
}
```

Create template by transaction id

Examples

SOAP min request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken
                    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-1496927745339-258987445"
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:CreateOperationTemplateRequest xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd" xmlns="">
            <ns2:type>COMMON</ns2:type>
            <ns2:name>Template Name</ns2:name>
            <ns2:prototypeOperationId>88888888</ns2:prototypeOperationId>
        </ns2:CreateOperationTemplateRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON min request:

```
{
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
          "Password": "PASSWORD"
        }
      }
    },
    "Body": {
      "CreateOperationTemplateRequest": {
        "type": "COMMON",
        "name": "Template Name",
        "prototypeOperationId": 88888888
      }
    }
  }
}
```

SOAP request with parameters:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken
                    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-1496928543165-1438707752"
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
```

```
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:CreateOperationTemplateRequest xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd" xmlns="">
            <ns2:type>REGULAR</ns2:type>
            <ns2:name>Template Name</ns2:name>
            <ns2:payer>12345678</ns2:payer>
            <ns2:prototypeOperationId>88888888</ns2:prototypeOperationId>
            <ns2:regularParameters>
                <ns2:amountInfo>
                    <ns2:type>AMOUNT</ns2:type>
                    <ns2:amount>
                        <ns2:amount>10.00</ns2:amount>
                        <ns2:isPayerAmount>false</ns2:isPayerAmount>
                    </ns2:amount>
                </ns2:amountInfo>
                <ns2:timeInfo>
                    <ns2:type>EVERY_DAY</ns2:type>
                    <ns2:startDateTime>2099-09-04T13:29:52.000+03:00</
ns2:startDateTime>
                    <ns2:endDateTime>2099-11-26T02:01:39.000+03:00</ns2:endDateTime>
                </ns2:timeInfo>
            </ns2:regularParameters>
            <ns2:operationInfo>
                <ns2:key>WIREPAYMENTPURPOSE</ns2:key>
                <ns2:value>payment purpose value</ns2:value>
            </ns2:operationInfo>
            <ns2:paymentPassword>
                <ns2:paymentPassword>11111</ns2:paymentPassword>
            </ns2:paymentPassword>
        </ns2:CreateOperationTemplateRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request with parameters:

```
{
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
          "Password": "PASSWORD"
        }
      }
    },
    "Body": {
      "CreateOperationTemplateRequest": {
        "paymentPassword": {
          "paymentPassword": "11111"
        },
        "type": "REGULAR",
        "name": "Template Name",
        "prototypeOperationId": 88888888,
        "payer": 12345678,
        "regularParameters": {
          "timeInfo": {
            "type": "EVERY_DAY",
            "startDateTime": "2099-09-04T13:29:52.000+03:00",
            "endDateTime": "2099-11-26T02:01:39.000+03:00"
          },
          "amountInfo": {
            "type": "AMOUNT",
            "amount": {
              "amount": 10.0,
              "isPayerAmount": false
            }
          }
```

```
        },
        "operationInfo": [
          {
            "key": "WIREPAYMENTPURPOSE",
            "value": "payment purpose value"
          }
        ]
      }
    }
  }
}
```

CreateOperationTemplateResponse

Examples

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:CreateOperationTemplateResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
            <ns2:id>12345</ns2:id>
        </ns2:CreateOperationTemplateResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON response:

```
{
  "Envelope": {
    "Body": {
      "CreateOperationTemplateResponse": {
        "id": 12345
      }
    }
  }
}
```

EditOperationTemplate

EditOperationTemplate Endpoint on page 132

EditOperationTemplateRequest

Examples

SOAP min request (change template name):

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken
                    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-149794456465427186054 7"
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
```

```
        <ns2:EditOperationTemplateRequest xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd" xmlns="">
            <ns2:id>12345</ns2:id>
            <ns2:name>New name</ns2:name>
        </ns2:EditOperationTemplateRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON min request (change template name):

```
{
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
          "Password": "PASSWORD"
        }
      }
    },
    "Body": {
      "EditOperationTemplateRequest": {
        "id": 12345,
        "name": "New name"
      }
    }
  }
}
```

SOAP full request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken
                    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-1497945234621116739080"
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:EditOperationTemplateRequest xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd" xmlns="">
            <ns2:id>12345</ns2:id>
            <ns2:type>REGULAR</ns2:type>
            <ns2:name>New name</ns2:name>
            <ns2:payer>12345678</ns2:payer>
            <ns2:description>New description</ns2:description>
            <ns2:tags>New tag</ns2:tags>
            <ns2:favorite>true</ns2:favorite>
            <ns2:regularParameters>
                <ns2:amountInfo>
                    <ns2:type>PAYMENTS</ns2:type>
                </ns2:amountInfo>
                <ns2:timeInfo>
                    <ns2:type>EVERY_WEEK</ns2:type>
                    <ns2:startDateTime>2099-09-04T13:29:52.000+03:00</
ns2:startDateTime>
                    <ns2:endDateTime>2099-11-26T02:01:39.000+03:00</ns2:endDateTime>
                </ns2:timeInfo>
                <ns2:operationsReportNotifications>
                    <ns2:id>222222</ns2:id>
                    <ns2:selected>false</ns2:selected>
```

```
                </ns2:operationsReportNotifications>
                <ns2:operationsReportNotifications>
                    <ns2:id>333333</ns2:id>
                    <ns2:selected>true</ns2:selected>
                </ns2:operationsReportNotifications>
            </ns2:regularParameters>
            <ns2:paymentPassword>
                <ns2:paymentPassword>11111</ns2:paymentPassword>
            </ns2:paymentPassword>
        </ns2:EditOperationTemplateRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON full request:

```
{
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
          "Password": "PASSWORD"
        }
      }
    },
    "Body": {
      "EditOperationTemplateRequest": {
        "id": 12345,
        "type": "REGULAR",
        "name": "New name",
        "payer": 12345678,
        "description": "New description",
        "tags": "New tag",
        "favorite": true,
        "regularParameters": {
          "amountInfo": {
            "type": "PAYMENTS"
          },
          "timeInfo": {
            "type": "EVERY_WEEK",
            "startDateTime": "2099-09-04T13:29:52.000+03:00",
            "endDateTime": "2099-11-26T02:01:39.000+03:00"
          },
          "operationsReportNotifications": [
            {
              "id": 222222,
              "selected": false
            },
            {
              "id": 333333,
              "selected": true
            }
          ]
        },
        "operationInfo": [],
        "paymentPassword": {
          "paymentPassword": "11111"
        }
      }
    }
  }
}
```

EditOperationTemplateResponse

Examples

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:EditOperationTemplateResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd"/>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON response:

```
{
  "Envelope": {
    "Body": {
      "EditOperationTemplateResponse": {}
    }
  }
}
```

FindOperationTemplates

FindOperationTemplates Endpoint on page 133

Search by ID

Examples

SOAP request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken
                    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-1495190343812165566418"
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:FindOperationTemplatesRequest xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd" xmlns="">
            <ns2:id>12345</ns2:id>
        </ns2:FindOperationTemplatesRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
          "Password": "PASSWORD"
```

```
        }
      }
    },
    "Body": {
      "FindOperationTemplatesRequest": {
        "id": 12345
      }
    }
  }
}
```

Common operation templates

Examples

SOAP min response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:FindOperationTemplatesResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
            <ns2:operationTemplate>
                <ns2:id>12345</ns2:id>
                <ns2:unitId>11111111</ns2:unitId>
                <ns2:type>COMMON</ns2:type>
                <ns2:name>TemplateName</ns2:name>
                <ns2:payer>12345678</ns2:payer>
                <ns2:payee>87654321</ns2:payee>
                <ns2:operationTypeCategory>TRANSFER</ns2:operationTypeCategory>
                <ns2:description/>
                <ns2:favorite>false</ns2:favorite>
                <ns2:commonParameters>
                    <ns2:amount>10</ns2:amount>
                    <ns2:isPayerAmount>true</ns2:isPayerAmount>
                </ns2:commonParameters>
            </ns2:operationTemplate>
        </ns2:FindOperationTemplatesResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON min response:

```
{
  "Envelope": {
    "Body": {
      "FindOperationTemplatesResponse": {
        "operationTemplate": [
          {
            "payee": 87654321,
            "commonParameters": {
              "amount": 10,
              "isPayerAmount": true
            },
            "type": "COMMON",
            "favorite": false,
            "additionalInfo": [],
            "operationTypeCategory": "TRANSFER",
            "id": 12345,
            "payer": 12345678,
            "description": "",
            "name": "TemplateName",
            "operationInfo": [],
            "unitId": 11111111
          }
        ]
      }
    }
  }
```

```
}
```

SOAP full response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:FindOperationTemplatesResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
            <ns2:operationTemplate>
                <ns2:id>12345</ns2:id>
                <ns2:unitId>11111111</ns2:unitId>
                <ns2:type>COMMON</ns2:type>
                <ns2:name>TemplateName</ns2:name>
                <ns2:payer>12345678</ns2:payer>
                <ns2:payee>87654321</ns2:payee>
                <ns2:operationTypeCategory>TRANSFER</ns2:operationTypeCategory>
                <ns2:description>Template description</ns2:description>
                <ns2:tags>work,John,transfer</ns2:tags>
                <ns2:favorite>true</ns2:favorite>
                <ns2:commonParameters>
                    <ns2:amount>10</ns2:amount>
                    <ns2:isPayerAmount>true</ns2:isPayerAmount>
                </ns2:commonParameters>
            </ns2:operationTemplate>
        </ns2:FindOperationTemplatesResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON full response:

```
{
  "Envelope": {
    "Body": {
      "FindOperationTemplatesResponse": {
        "operationTemplate": [
          {
            "id": 12345,
            "unitId": 11111111,
            "type": "COMMON",
            "name": "templateName",
            "payer": 12345678,
            "payee": 87654321,
            "operationTypeCategory": "TRANSFER",
            "description": "Template description",
            "tags": "work,John,transfer",
            "favorite": true,
            "commonParameters": {
              "amount": 10,
              "isPayerAmount": true
            },
            "operationInfo": [],
            "additionalInfo": []
          }
        ]
      }
    }
  }
}
```

Regular operation templates

Examples

SOAP response (amountType = AMOUNT):

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
```

```
        <ns2:FindOperationTemplatesResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
            <ns2:operationTemplate>
                <ns2:id>12345</ns2:id>
                <ns2:unitId>11111111</ns2:unitId>
                <ns2:type>REGULAR</ns2:type>
                <ns2:name>Template name</ns2:name>
                <ns2:payer>12345678</ns2:payer>
                <ns2:payee>87654321</ns2:payee>
                <ns2:operationTypeCategory>TRANSFER</ns2:operationTypeCategory>
                <ns2:description/>
                <ns2:favorite>false</ns2:favorite>
                <ns2:regularParameters>
                    <ns2:amountInfo>
                        <ns2:type>AMOUNT</ns2:type>
                        <ns2:amount>
                            <ns2:amount>10</ns2:amount>
                            <ns2:isPayerAmount>true</ns2:isPayerAmount>
                        </ns2:amount>
                    </ns2:amountInfo>
                    <ns2:timeInfo>
                        <ns2:type>EVERY_DAY</ns2:type>
                        <ns2:startDateTime>2099-09-04T13:00:00.000+03:00</
ns2:startDateTime>
                        <ns2:endDateTime>2099-11-26T02:01:39.000+03:00</
ns2:endDateTime>
                    </ns2:timeInfo>
                    <ns2:reminderInfo>
                        <ns2:remind>true</ns2:remind>
                        <ns2:hoursBeforeExecution>25</ns2:hoursBeforeExecution>
                    </ns2:reminderInfo>
                </ns2:regularParameters>
                <ns2:additionalInfo>
                    <ns2:key>executionlastdatetime</ns2:key>
                    <ns2:value>2099-09-05T13:00:00.000+03:00</ns2:value>
                </ns2:additionalInfo>
                <ns2:additionalInfo>
                    <ns2:key>executionlastoperationid</ns2:key>
                    <ns2:value>22222222</ns2:value>
                </ns2:additionalInfo>
            </ns2:operationTemplate>
        </ns2:FindOperationTemplatesResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON response (amountType = AMOUNT):

```
{
  "Envelope": {
    "Body": {
      "FindOperationTemplatesResponse": {
        "operationTemplate": [
          {
            "payee": 87654321,
            "type": "REGULAR",
            "favorite": false,
            "additionalInfo": [],
            "operationTypeCategory": "TRANSFER",
            "id": 12345,
            "payer": 12345678,
            "description": "",
            "name": "Template name",
            "operationInfo": [],
            "unitId": 11111111,
            "regularParameters": {
              "timeInfo": {
                "type": "EVERY_DAY",
                "startDateTime": "2099-09-04T13:00:00.000+03:00",
                "endDateTime": "2099-11-26T02:01:39.000+03:00"
              },
```

```
              "amountInfo": {
                "type": "AMOUNT",
                "amount": {
                  "amount": 10,
                  "isPayerAmount": true
                }
              },
              "reminderInfo": {
                "remind": true,
                "hoursBeforeExecution": 25,
                "notification": []
              }
            },
            "additionalInfo": [
              {
                "value": "2099-09-05T13:00:00.000+03:00",
                "key": "executionlastdatetime"
              },
              {
                "value": "22222222",
                "key": "executionlastoperationid"
              }
            ]
          }
        ]
      }
    }
  }
}
```

SOAP response (amountType = BALANCE):

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:FindOperationTemplatesResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
            <ns2:operationTemplate>
                <ns2:id>12345</ns2:id>
                <ns2:unitId>11111111</ns2:unitId>
                <ns2:type>REGULAR</ns2:type>
                <ns2:name>Template name</ns2:name>
                <ns2:payer>12345678</ns2:payer>
                <ns2:payee>87654321</ns2:payee>
                <ns2:operationTypeCategory>TRANSFER</ns2:operationTypeCategory>
                <ns2:description/>
                <ns2:favorite>false</ns2:favorite>
                <ns2:regularParameters>
                    <ns2:amountInfo>
                        <ns2:type>BALANCE</ns2:type>
                    </ns2:amountInfo>
                    <ns2:timeInfo>
                        <ns2:type>EVERY_WORKDAY</ns2:type>
                        <ns2:startDateTime>2099-09-04T13:00:00.000+03:00</
ns2:startDateTime>
                        <ns2:endDateTime>2099-11-26T02:01:39.000+03:00</
ns2:endDateTime>
                    </ns2:timeInfo>
                </ns2:regularParameters>
            </ns2:operationTemplate>
        </ns2:FindOperationTemplatesResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON response (amountType = BALANCE):

```
{
  "Envelope": {
    "Body": {
      "FindOperationTemplatesResponse": {
```

```
        "operationTemplate": [
          {
            "id": 12345,
            "unitId": 11111111,
            "payer": 12345678,
            "payee": 87654321,
            "type": "REGULAR",
            "favorite": false,
            "additionalInfo": [],
            "operationTypeCategory": "TRANSFER",
            "description": "",
            "name": "Template name",
            "operationInfo": [],
            "regularParameters": {
              "amountInfo": {
                "type": "BALANCE"
              },
              "timeInfo": {
                "type": "EVERY_WORKDAY",
                "startDateTime": "2099-09-04T13:00:00.000+03:00",
                "endDateTime": "2099-11-26T02:01:39.000+03:00"
              }
            }
          }
        ]
      }
    }
  }
}
```


SOAP response (amountType = PAYMENTS):


```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:FindOperationTemplatesResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
            <ns2:operationTemplate>
                <ns2:id>12345</ns2:id>
                <ns2:unitId>11111111</ns2:unitId>
                <ns2:type>REGULAR</ns2:type>
                <ns2:name>Template name</ns2:name>
                <ns2:payer>12345678</ns2:payer>
                <ns2:payee>87654321</ns2:payee>
                <ns2:operationTypeCategory>TRANSFER</ns2:operationTypeCategory>
                <ns2:description/>
                <ns2:favorite>false</ns2:favorite>
                <ns2:regularParameters>
                    <ns2:amountInfo>
                        <ns2:type>PAYMENTS</ns2:type>
                    </ns2:amountInfo>
                    <ns2:timeInfo>
                        <ns2:type>EVERY_MONTH</ns2:type>
                        <ns2:startDateTime>2099-09-04T13:00:00.000+03:00</
ns2:startDateTime>
                        <ns2:endDateTime>2099-11-26T02:01:39.000+03:00</
ns2:endDateTime>
                    </ns2:timeInfo>
                    <ns2:operationsReportNotifications>
                        <ns2:id>222222</ns2:id>
                        <ns2:type>EMAIL</ns2:type>
                        <ns2:recipient>report@site.com</ns2:recipient>
                        <ns2:selected>true</ns2:selected>
                    </ns2:operationsReportNotifications>
                </ns2:regularParameters>
            </ns2:operationTemplate>
        </ns2:FindOperationTemplatesResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON response (amountType = PAYMENTS):


```
{
  "Envelope": {
    "Body": {
      "FindOperationTemplatesResponse": {
        "operationTemplate": [
          {
            "payee": 87654321,
            "type": "REGULAR",
            "favorite": false,
            "additionalInfo": [],
            "operationTypeCategory": "TRANSFER",
            "id": 12345,
            "regularParameters": {
              "amountInfo": {
                "type": "PAYMENTS"
              },
              "timeInfo": {
                "type": "EVERY_MONTH",
                "startDateTime": "2099-09-04T13:00:00.000+03:00",
                "endDateTime": "2099-11-26T02:01:39.000+03:00"
              },
              "operationsReportNotifications": [
                {
                  "id": 222222,
                  "type": "EMAIL",
                  "recipient": [
                    "report@site.com"
                  ],
                  "selected": true
                }
              ]
            },
            "payer": 12345678,
            "description": "",
            "name": "Template name",
            "operationInfo": [],
            "unitId": 11111111
          }
        ]
      }
    }
  }
}
```


SOAP response (amountType = RANGE):


```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:FindOperationTemplatesResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
            <ns2:operationTemplate>
                <ns2:id>12345</ns2:id>
                <ns2:unitId>11111111</ns2:unitId>
                <ns2:type>REGULAR</ns2:type>
                <ns2:name>Template name</ns2:name>
                <ns2:payer>12345678</ns2:payer>
                <ns2:payee>87654321</ns2:payee>
                <ns2:operationTypeCategory>TRANSFER</ns2:operationTypeCategory>
                <ns2:description/>
                <ns2:favorite>false</ns2:favorite>
                <ns2:regularParameters>
                    <ns2:amountInfo>
                        <ns2:type>RANGE</ns2:type>
                        <ns2:range>
                            <ns2:amountMinValue>10</ns2:amountMinValue>
                            <ns2:amountMaxValue>20</ns2:amountMaxValue>
                        </ns2:range>
```

```
                    </ns2:amountInfo>
                    <ns2:timeInfo>
                        <ns2:type>EVERY_LAST_DAY_OF_MONTH</ns2:type>
                        <ns2:startDateTime>2099-09-30T13:00:00.000+03:00</
ns2:startDateTime>
                        <ns2:endDateTime>2099-11-26T02:01:39.000+03:00</
ns2:endDateTime>
                    </ns2:timeInfo>
                </ns2:regularParameters>
            </ns2:operationTemplate>
        </ns2:FindOperationTemplatesResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON response (amountType = RANGE):

```
{
  "Envelope": {
    "Body": {
      "FindOperationTemplatesResponse": {
        "operationTemplate": [
          {
            "payee": 87654321,
            "type": "REGULAR",
            "favorite": false,
            "additionalInfo": [],
            "operationTypeCategory": "TRANSFER",
            "id": 12345,
            "regularParameters": {
              "amountInfo": {
                "type": "RANGE",
                "range": {
                  "amountMinValue": 10,
                  "amountMaxValue": 20
                }
              },
              "timeInfo": {
                "type": "EVERY_LAST_DAY_OF_MONTH",
                "startDateTime": "2099-09-30T13:00:00.000+03:00",
                "endDateTime": "2099-11-26T02:01:39.000+03:00"
              }
            },
            "payer": 12345678,
            "description": "",
            "name": "Template name",
            "operationInfo": [],
            "unitId": 11111111
          }
        ]
      }
    }
  }
}
```

SOAP response (amountType = REST):

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:FindOperationTemplatesResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
            <ns2:operationTemplate>
                <ns2:id>12345</ns2:id>
                <ns2:unitId>11111111</ns2:unitId>
                <ns2:type>REGULAR</ns2:type>
                <ns2:name>Template name</ns2:name>
                <ns2:payer>12345678</ns2:payer>
                <ns2:payee>87654321</ns2:payee>
                <ns2:operationTypeCategory>TRANSFER</ns2:operationTypeCategory>
                <ns2:description/>
```

```
                <ns2:favorite>false</ns2:favorite>
                <ns2:regularParameters>
                    <ns2:amountInfo>
                        <ns2:type>REST</ns2:type>
                        <ns2:rest>
                            <ns2:amount>10</ns2:amount>
                        </ns2:rest>
                    </ns2:amountInfo>
                    <ns2:timeInfo>
                        <ns2:type>ONCE</ns2:type>
                        <ns2:startDateTime>2099-09-04T13:00:00.000+03:00</
ns2:startDateTime>
                    </ns2:timeInfo>
                </ns2:regularParameters>
            </ns2:operationTemplate>
        </ns2:FindOperationTemplatesResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON response (amountType = REST):

```
{
  "Envelope": {
    "Body": {
      "FindOperationTemplatesResponse": {
        "operationTemplate": [
          {
            "payee": 87654321,
            "type": "REGULAR",
            "favorite": false,
            "additionalInfo": [],
            "operationTypeCategory": "TRANSFER",
            "id": 12345,
            "regularParameters": {
              "amountInfo": {
                "type": "REST",
                "rest": {
                  "amount": 10
                }
              },
              "timeInfo": {
                "type": "ONCE",
                "startDateTime": "2099-09-04T13:00:00.000+03:00"
              }
            },
            "payer": 12345678,
            "description": "",
            "name": "Template name",
            "operationInfo": [],
            "unitId": 11111111
          }
        ]
      }
    }
  }
}
```

SOAP response (amountType = CREDIT):

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:FindOperationTemplatesResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
            <ns2:operationTemplate>
                <ns2:id>12345</ns2:id>
                <ns2:unitId>11111111</ns2:unitId>
                <ns2:type>REGULAR</ns2:type>
                <ns2:name>Template name</ns2:name>
                <ns2:payer>12345678</ns2:payer>
```

```
                <ns2:payee>87654321</ns2:payee>
                <ns2:operationTypeCategory>TRANSFER</ns2:operationTypeCategory>
                <ns2:description/>
                <ns2:favorite>false</ns2:favorite>
                <ns2:regularParameters>
                    <ns2:amountInfo>
                        <ns2:type>CREDIT</ns2:type>
                    </ns2:amountInfo>
                    <ns2:timeInfo>
                        <ns2:type>EVERY_WEEK</ns2:type>
                        <ns2:startDateTime>2099-09-04T13:00:00.000+03:00</
ns2:startDateTime>
                        <ns2:endDateTime>2099-11-26T02:01:39.000+03:00</
ns2:endDateTime>
                    </ns2:timeInfo>
                    <ns2:operationsReportNotifications>
                        <ns2:id>222222</ns2:id>
                        <ns2:type>EMAIL</ns2:type>
                        <ns2:recipient>report@site.com</ns2:recipient>
                        <ns2:selected>true</ns2:selected>
                    </ns2:operationsReportNotifications>
                </ns2:regularParameters>
            </ns2:operationTemplate>
        </ns2:FindOperationTemplatesResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON response (amountType = CREDIT):

```
{
  "Envelope": {
    "Body": {
      "FindOperationTemplatesResponse": {
        "operationTemplate": [
          {
            "payee": 87654321,
            "type": "REGULAR",
            "favorite": false,
            "additionalInfo": [],
            "operationTypeCategory": "TRANSFER",
            "id": 12345,
            "regularParameters": {
              "amountInfo": {
                "type": "CREDIT"
              },
              "timeInfo": {
                "type": "EVERY_WEEK",
                "startDateTime": "2099-09-04T13:00:00.000+03:00",
                "endDateTime": "2099-11-26T02:01:39.000+03:00"
              },
              "operationsReportNotifications": [
                {
                  "id": 222222,
                  "type": "EMAIL",
                  "recipient": [
                    "report@site.com"
                  ],
                  "selected": true
                }
              ]
            },
            "payer": 12345678,
            "description": "",
            "name": "Template name",
            "operationInfo": [],
            "unitId": 11111111
          }
        ]
      }
    }
```

```
  }
}
```

Search by filter

Examples

SOAP request (search by unitId):

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken
                    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-1495523855803162105608"
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:FindOperationTemplatesRequest xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd" xmlns="">
            <ns2:unitId>11111111</ns2:unitId>
        </ns2:FindOperationTemplatesRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request (search by unitId):

```
{
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
          "Password": "PASSWORD"
        }
      }
    },
    "Body": {
      "FindOperationTemplatesRequest": {
        "unitId": 11111111
      }
    }
  }
}
```

SOAP request (search by filter):

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken
                    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-1495523855803162105608"
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
```

```
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:FindOperationTemplatesRequest xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd" xmlns="">
            <ns2:unitId>11111111</ns2:unitId>
            <ns2:operationTypeCategory>WITHDRAWAL</ns2:operationTypeCategory>
            <ns2:type>COMMON</ns2:type>
            <ns2:tag>work</ns2:tag>
        </ns2:FindOperationTemplatesRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request (search by filter):

```
{
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
          "Password": "PASSWORD"
        }
      }
    },
    "Body": {
      "FindOperationTemplatesRequest": {
        "unitId": 11111111,
        "operationTypeCategory": "WITHDRAWAL",
        "type": "COMMON",
        "tag": "work"
      }
    }
  }
}
```

Response for "search by filter"

Examples

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:FindOperationTemplatesResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
            <ns2:operationTemplate>
                <ns2:id>12345</ns2:id>
                <ns2:unitId>11111111</ns2:unitId>
                <ns2:type>COMMON</ns2:type>
                <ns2:name>Template Name</ns2:name>
                <ns2:payer>12345678</ns2:payer>
                <ns2:payee>87654321</ns2:payee>
                <ns2:operationTypeCategory>WITHDRAWAL</ns2:operationTypeCategory>
                <ns2:favorite>false</ns2:favorite>
                <ns2:commonParameters>
                    <ns2:amount>10</ns2:amount>
                    <ns2:isPayerAmount>false</ns2:isPayerAmount>
                </ns2:commonParameters>
                <ns2:additionalInfo>
                    <ns2:key>payer_alias</ns2:key>
                    <ns2:value>Payer alias</ns2:value>
                </ns2:additionalInfo>
                <ns2:additionalInfo>
                    <ns2:key>payee_alias</ns2:key>
                    <ns2:value>Payee alias</ns2:value>
                </ns2:additionalInfo>
                <ns2:additionalInfo>
```

```
                    <ns2:key>payer_currency</ns2:key>
                    <ns2:value>RUB</ns2:value>
                </ns2:additionalInfo>
                <ns2:additionalInfo>
                    <ns2:key>payee_currency</ns2:key>
                    <ns2:value>RUB</ns2:value>
                </ns2:additionalInfo>
            </ns2:operationTemplate>
        </ns2:FindOperationTemplatesResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON response:

```
{
  "Envelope": {
    "Body": {
      "FindOperationTemplatesResponse": {
        "operationTemplate": [
          {
            "id": 12345,
            "unitId": 11111111,
            "name": "Template Name",
            "type": "COMMON",
            "payer": 12345678,
            "payee": 87654321,
            "operationTypeCategory": "WITHDRAWAL",
            "favorite": false,
            "commonParameters": {
              "amount": 10,
              "isPayerAmount": false
            },
            "operationInfo": [],
            "additionalInfo": [
              {
                "value": "Payer alias",
                "key": "payer_alias"
              },
              {
                "value": "Payee alias",
                "key": "payee_alias"
              },
              {
                "value": "RUB",
                "key": "payer_currency"
              },
              {
                "value": "RUB",
                "key": "payee_currency"
              }
            ]
          }
        ]
      }
    }
  }
}
```

Withdrawal operation templates

Wiretransfer

Available attributes:

- **WIREBANKBIK** - BIN
- **WIREBANKACCOUNT** - Bank account
- **WIREPAYMENTPURPOSE** - Payment purpose
- **WIREUSERNAME** - Payee name

- **WIREUSERINN** - Payee Tax ID
- **WIREKBK** - KBK
- **WIREOKTMO** - OKTMO
- **WIREKPP** - KPP

Examples

SOAP create request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken
                    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-1495439085421168452053 2"
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:CreateOperationTemplateRequest xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd" xmlns="">
            <ns2:unitId>11111111</ns2:unitId>
            <ns2:type>REGULAR</ns2:type>
            <ns2:name>Template Name</ns2:name>
            <ns2:payer>12345678</ns2:payer>
            <ns2:payee>5</ns2:payee>
            <ns2:regularParameters>
                <ns2:amountInfo>
                    <ns2:type>PAYMENTS</ns2:type>
                </ns2:amountInfo>
                <ns2:timeInfo>
                    <ns2:type>EVERY_DAY</ns2:type>
                    <ns2:startDateTime>2099-09-04T13:00:00.000+03:00</
ns2:startDateTime>
                </ns2:timeInfo>
            </ns2:regularParameters>
            <ns2:operationInfo>
                <ns2:key>WIREBANKBIK</ns2:key>
                <ns2:value>044525214</ns2:value>
            </ns2:operationInfo>
            <ns2:operationInfo>
                <ns2:key>WIREBANKACCOUNT</ns2:key>
                <ns2:value>40101678901234567896</ns2:value>
            </ns2:operationInfo>
            <ns2:operationInfo>
                <ns2:key>WIREPAYMENTPURPOSE</ns2:key>
                <ns2:value></ns2:value>
            </ns2:operationInfo>
            <ns2:paymentPassword>
                <ns2:paymentPassword>11111</ns2:paymentPassword>
            </ns2:paymentPassword>
        </ns2:CreateOperationTemplateRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON create request:

```
{
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
```

```
          "Username": "USERNAME",
          "Password": "PASSWORD"
        }
      }
    },
    "Body": {
      "CreateOperationTemplateRequest": {
        "unitId": 11111111,
        "type": "REGULAR",
        "name": "Template Name",
        "payer": 12345678,
        "payee": 5,
        "regularParameters": {
          "amountInfo": {
            "type": "PAYMENTS"
          },
          "timeInfo": {
            "type": "EVERY_DAY",
            "startDateTime": "2099-09-04T13:00:00.000+03:00"
          }
        },
        "operationInfo": [
          {
            "value": "044525214",
            "key": "WIREBANKBIK"
          },
          {
            "value": "40101678901234567896",
            "key": "WIREBANKACCOUNT"
          },
          {
            "value": "",
            "key": "WIREPAYMENTPURPOSE"
          }
        ],
        "paymentPassword": {
          "paymentPassword": "11111"
        }
      }
    }
  }
}
```

SOAP find response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:FindOperationTemplatesResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
            <ns2:operationTemplate>
                <ns2:id>12345</ns2:id>
                <ns2:unitId>11111111</ns2:unitId>
                <ns2:type>REGULAR</ns2:type>
                <ns2:name>Template Name</ns2:name>
                <ns2:payer>12345678</ns2:payer>
                <ns2:payee>5</ns2:payee>
                <ns2:operationTypeCategory>WITHDRAWAL</ns2:operationTypeCategory>
                <ns2:description/>
                <ns2:favorite>false</ns2:favorite>
                <ns2:regularParameters>
                    <ns2:amountInfo>
                        <ns2:type>PAYMENTS</ns2:type>
                    </ns2:amountInfo>
                    <ns2:timeInfo>
                        <ns2:type>EVERY_DAY</ns2:type>
                        <ns2:startDateTime>2099-09-04T13:00:00.000+03:00</
ns2:startDateTime>
                    </ns2:timeInfo>
                </ns2:regularParameters>
```

```
                <ns2:operationInfo>
                    <ns2:key>wirebankbik</ns2:key>
                    <ns2:value>044525214</ns2:value>
                </ns2:operationInfo>
                <ns2:operationInfo>
                    <ns2:key>wirepaymentpurpose</ns2:key>
                    <ns2:value></ns2:value>
                </ns2:operationInfo>
                <ns2:operationInfo>
                    <ns2:key>wirebankaccount</ns2:key>
                    <ns2:value>40101678901234567896</ns2:value>
                </ns2:operationInfo>
                <ns2:additionalInfo>
                    <ns2:key>executionlastdatetime</ns2:key>
                    <ns2:value>2099-09-05T13:00:00.000+03:00</ns2:value>
                </ns2:additionalInfo>
                <ns2:additionalInfo>
                    <ns2:key>executionlastoperationid</ns2:key>
                    <ns2:value>22222222</ns2:value>
                </ns2:additionalInfo>
            </ns2:operationTemplate>
        </ns2:FindOperationTemplatesResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON find response:

```
{
  "Envelope": {
    "Body": {
      "FindOperationTemplatesResponse": {
        "operationTemplate": [
          {
            "id": 12345,
            "unitId": 11111111,
            "type": "REGULAR",
            "name": "Template Name",
            "payer": 12345678,
            "payee": 5,
            "operationTypeCategory": "WITHDRAWAL",
            "description": "",
            "favorite": false,
            "regularParameters": {
              "timeInfo": {
                "type": "EVERY_DAY",
                "startDateTime": "2099-09-04T13:00:00.000+03:00"
              },
              "amountInfo": {
                "type": "PAYMENTS"
              }
            },
            "operationInfo": [
              {
                "value": "044525214",
                "key": "wirebankbik"
              },
              {
                "value": "",
                "key": "wirepaymentpurpose"
              },
              {
                "value": "40101678901234567896",
                "key": "wirebankaccount"
              }
            ],
            "additionalInfo": [
              {
                "value": "2099-09-05T13:00:00.000+03:00",
                "key": "executionlastdatetime"
              },
```

```
            {
              "value": "22222222",
              "key": "executionlastoperationid"
            }
          ]
        }
      ]
    }
  }
}
```

Credit cards

Examples

SOAP create request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken
                    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-1495544753255-363074483"
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:CreateOperationTemplateRequest xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd" xmlns="">
            <ns2:type>COMMON</ns2:type>
            <ns2:name>Template Name</ns2:name>
            <ns2:payer>12345678</ns2:payer>
            <ns2:payee>279</ns2:payee>
            <ns2:commonParameters>
                <ns2:amount>10.00</ns2:amount>
                <ns2:isPayerAmount>false</ns2:isPayerAmount>
            </ns2:commonParameters>
            <ns2:operationInfo>
                <ns2:key>PAYEECARDNUMBER</ns2:key>
                <ns2:value>4444441111114444</ns2:value>
            </ns2:operationInfo>
            <ns2:operationInfo>
                <ns2:key>CARDEXPIRATION</ns2:key>
                <ns2:value>01/2020</ns2:value>
            </ns2:operationInfo>
        </ns2:CreateOperationTemplateRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON create request:

```
{
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
          "Password": "PASSWORD"
        }
      }
    },
```

```
    "Body": {
      "CreateOperationTemplateRequest": {
        "payer": 12345678,
        "name": "Template Name",
        "payee": 279,
        "commonParameters": {
          "amount": 10.0,
          "isPayerAmount": false
        },
        "operationInfo": [
          {
            "value": "4444441111114444",
            "key": "PAYEECARDNUMBER"
          },
          {
            "value": "01\/2020",
            "key": "CARDEXPIRATION"
          }
        ],
        "type": "COMMON"
      }
    }
  }
}
```

SOAP find response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:FindOperationTemplatesResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
            <ns2:operationTemplate>
                <ns2:id>12345</ns2:id>
                <ns2:unitId>11111111</ns2:unitId>
                <ns2:type>COMMON</ns2:type>
                <ns2:name>Template Name</ns2:name>
                <ns2:payer>12345678</ns2:payer>
                <ns2:payee>279</ns2:payee>
                <ns2:operationTypeCategory>WITHDRAWAL</ns2:operationTypeCategory>
                <ns2:description/>
                <ns2:favorite>false</ns2:favorite>
                <ns2:commonParameters>
                    <ns2:amount>10</ns2:amount>
                    <ns2:isPayerAmount>false</ns2:isPayerAmount>
                </ns2:commonParameters>
                <ns2:operationInfo>
                    <ns2:key>cardexpiration</ns2:key>
                    <ns2:value>01/2020</ns2:value>
                </ns2:operationInfo>
                <ns2:operationInfo>
                    <ns2:key>payeecardnumber</ns2:key>
                    <ns2:value>444444******4444</ns2:value>
                </ns2:operationInfo>
            </ns2:operationTemplate>
        </ns2:FindOperationTemplatesResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON find response:

```
{
  "Envelope": {
    "Body": {
      "FindOperationTemplatesResponse": {
        "operationTemplate": [
          {
            "payee": 279,
            "commonParameters": {
              "amount": 10,
```

```
              "isPayerAmount": false
            },
            "type": "COMMON",
            "favorite": false,
            "additionalInfo": [],
            "operationTypeCategory": "WITHDRAWAL",
            "id": 12345,
            "payer": 12345678,
            "description": "",
            "name": "Template Name",
            "operationInfo": [
              {
                "value": "01\/2020",
                "key": "cardexpiration"
              },
              {
                "value": "444444******4444",
                "key": "payeecardnumber"
              }
            ],
            "unitId": 11111111
          }
        ]
      }
    }
  }
}
```

QIWI

Examples

SOAP create request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken
                    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-1495544443076-462207690"
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:CreateOperationTemplateRequest xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd" xmlns="">
            <ns2:type>COMMON</ns2:type>
            <ns2:name>Template Name</ns2:name>
            <ns2:payer>12345678</ns2:payer>
            <ns2:payee>255</ns2:payee>
            <ns2:commonParameters>
                <ns2:amount>10.00</ns2:amount>
                <ns2:isPayerAmount>false</ns2:isPayerAmount>
            </ns2:commonParameters>
            <ns2:operationInfo>
                <ns2:key>EXTERNALACCOUNTID</ns2:key>
                <ns2:value>111111111111</ns2:value>
            </ns2:operationInfo>
        </ns2:CreateOperationTemplateRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON create request:

```
{
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
          "Password": "PASSWORD"
        }
      }
    },
    "Body": {
      "CreateOperationTemplateRequest": {
        "payer": 12345678,
        "name": "Template Name",
        "payee": 255,
        "commonParameters": {
          "amount": 10.0,
          "isPayerAmount": false
        },
        "operationInfo": [
          {
            "value": "11111111111",
            "key": "EXTERNALACCOUNTID"
          }
        ],
        "type": "COMMON"
      }
    }
  }
}
```

SOAP find response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:FindOperationTemplatesResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
            <ns2:operationTemplate>
                <ns2:id>12345</ns2:id>
                <ns2:unitId>11111111</ns2:unitId>
                <ns2:type>COMMON</ns2:type>
                <ns2:name>Template Name</ns2:name>
                <ns2:payer>12345678</ns2:payer>
                <ns2:payee>255</ns2:payee>
                <ns2:operationTypeCategory>WITHDRAWAL</ns2:operationTypeCategory>
                <ns2:description/>
                <ns2:favorite>false</ns2:favorite>
                <ns2:commonParameters>
                    <ns2:amount>10</ns2:amount>
                    <ns2:isPayerAmount>false</ns2:isPayerAmount>
                </ns2:commonParameters>
                <ns2:operationInfo>
                    <ns2:key>externalaccountid</ns2:key>
                    <ns2:value>111111111111</ns2:value>
                </ns2:operationInfo>
            </ns2:operationTemplate>
        </ns2:FindOperationTemplatesResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON find response:

```
{
  "Envelope": {
    "Body": {
      "FindOperationTemplatesResponse": {
        "operationTemplate": [
```

```
          {
            "payee": 255,
            "commonParameters": {
              "amount": 10,
              "isPayerAmount": false
            },
            "type": "COMMON",
            "favorite": false,
            "additionalInfo": [],
            "operationTypeCategory": "WITHDRAWAL",
            "id": 12345,
            "payer": 12345678,
            "description": "",
            "name": "Template Name",
            "operationInfo": [
              {
                "value": "111111111111",
                "key": "externalaccountid"
              }
            ],
            "unitId": 11111111
          }
        ]
      }
    }
  }
}
```

DeleteOperationTemplate

DeleteOperationTemplate Endpoint on page 134

Examples

SOAP request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken
                    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-14951903443231747031771"
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:DeleteOperationTemplateRequest xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd" xmlns="">
            <ns2:id>12345</ns2:id>
        </ns2:DeleteOperationTemplateRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:DeleteOperationTemplateResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd"/>
    </SOAP-ENV:Body>
```

```
</SOAP-ENV:Envelope>
```

JSON request:

```
{
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
          "Password": "PASSWORD"
        }
      }
    },
    "Body": {
      "DeleteOperationTemplateRequest": {
        "id": 12345
      }
    }
  }
}
```

JSON response:

```
{
  "Envelope": {
    "Body": {
      "DeleteOperationTemplateResponse": {}
    }
  }
}
```

Profile examples

Read profile

SOAP request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
        <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="XWSSGID-1417525998002-1416764885"
 xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd">
            <wsse:Username>USERNAME</wsse:Username>
            <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
        </wsse:UsernameToken>
    </wsse:Security>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
    <ns2:GetProfileInfoRequest xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd"
 ns2:version="VERSION_2" xmlns="">
        <ns2:unitId>10054789</ns2:unitId>
    </ns2:GetProfileInfoRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
    <ns2:GetProfileInfoResponse xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd">
```

```
        <ns2:attribute>
            <ns2:key>last_name</ns2:key>
            <ns2:value>Last name</ns2:value>
            <ns2:approved>false</ns2:approved>
            <ns2:published>false</ns2:published>
        </ns2:attribute>
        <ns2:attribute>
            <ns2:key>first_name</ns2:key>
            <ns2:value>First name</ns2:value>
            <ns2:approved>false</ns2:approved>
            <ns2:published>false</ns2:published>
        </ns2:attribute>
        <ns2:attribute>
            <ns2:key>date_of_birth</ns2:key>
            <ns2:value>1981-07-23</ns2:value>
            <ns2:approved>false</ns2:approved>
            <ns2:published>false</ns2:published>
        </ns2:attribute>
        <ns2:attribute>
            <ns2:key>profileType</ns2:key>
            <ns2:value>client</ns2:value>
            <ns2:approved>false</ns2:approved>
            <ns2:published>false</ns2:published>
        </ns2:attribute>
        <ns2:attribute>
            <ns2:key>cell_phone</ns2:key>
            <ns2:value>71234567890</ns2:value>
            <ns2:approved>false</ns2:approved>
            <ns2:published>false</ns2:published>
        </ns2:attribute>
        <ns2:attribute>
            <ns2:key>unitid</ns2:key>
            <ns2:value>10054789</ns2:value>
            <ns2:approved>false</ns2:approved>
            <ns2:published>false</ns2:published>
        </ns2:attribute>
    </ns2:GetProfileInfoResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "GetProfileInfoRequest": {
            "version": "VERSION_2",
            "unitId": 10060344
        }
    }
}}
```

JSON response:

```
{"Envelope": {
    "Body": {
        "GetProfileInfoResponse": {
            "attribute": [
                {
                    "approved": false,
                    "value": "Last name",
                    "published": false,
                    "key": "last_name"
```

```
                },
                {
                    "approved": false,
                    "value": "First name",
                    "published": false,
                    "key": "first_name"
                },
                {
                    "approved": false,
                    "value": "1981-07-23",
                    "published": false,
                    "key": "date_of_birth"
                },
                {
                    "approved": false,
                    "value": "client",
                    "published": false,
                    "key": "profileType"
                },
                {
                    "approved": false,
                    "value": "+71234567890",
                    "published": false,
                    "key": "cell_phone"
                },
                {
                    "approved": false,
                    "value": "10060344",
                    "published": false,
                    "key": "unitid"
                },
                {
                    "approved": false,
                    "value": "10065755",
                    "published": false,
                    "key": "profileid"
                }
            ]
        }
    }
}}
```


Create profile

SOAP request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="XWSSGID-14175259948916951504 69"
 xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:CreateProfileRequest xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd"
 xmlns="">
            <ns2:unitId>10053304</ns2:unitId>
            <ns2:profileType>client</ns2:profileType>
            <ns2:profile>
                <ns2:attribute>
                    <ns2:key>LAST_NAME</ns2:key>
                    <ns2:value>Last name</ns2:value>
                </ns2:attribute>
```

```
                <ns2:attribute>
                    <ns2:key>FIRST_NAME</ns2:key>
                    <ns2:value>First name</ns2:value>
                </ns2:attribute>
                <ns2:attribute>
                    <ns2:key>CELL_PHONE</ns2:key>
                    <ns2:value>71234567899</ns2:value>
                </ns2:attribute>
                <ns2:attribute>
                    <ns2:key>DATE_OF_BIRTH</ns2:key>
                    <ns2:value>1981-05-24</ns2:value>
                </ns2:attribute>
            </ns2:profile>
        </ns2:CreateProfileRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:CreateProfileResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">10054789</ns2:CreateProfileResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "CreateProfileRequest": {
            "unitId": 10053304,
            "profileType": "CLIENT",
            "profile": {
                "attribute": [
                    {
                        "key": "LAST_NAME",
                        "value": "Last name"
                    },
                    {
                        "key": "FIRST_NAME",
                        "value": "First name"
                    },
                    {
                        "key": "CELL_PHONE",
                        "value": "+71234567899"
                    },
                    {
                        "key": "DATE_OF_BIRTH",
                        "value": "1981-05-24"
                    }
                ]
            }
        }
    }
}}
```

JSON response:

```
{"Envelope": {
```

```
    "Body": {
        "CreateProfileResponse": 10060344
    }
}}
```

Edit profile

SOAP request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
        <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="XWSSGID-14175259972751278772983"
 xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd">
            <wsse:Username>USERNAME</wsse:Username>
            <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
        </wsse:UsernameToken>
    </wsse:Security>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
    <ns2:EditProfileRequest xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd"
 xmlns="">
        <ns2:unitId>10054789</ns2:unitId>
        <ns2:profile>
            <ns2:attribute>
                <ns2:key>LAST_NAME</ns2:key>
                <ns2:value>Last name</ns2:value>
            </ns2:attribute>
            <ns2:attribute>
                <ns2:key>FIRST_NAME</ns2:key>
                <ns2:value>First name</ns2:value>
            </ns2:attribute>
            <ns2:attribute>
                <ns2:key>CELL_PHONE</ns2:key>
                <ns2:value>71234567890</ns2:value>
            </ns2:attribute>
            <ns2:attribute>
                <ns2:key>DATE_OF_BIRTH</ns2:key>
                <ns2:value>1981-07-23</ns2:value>
            </ns2:attribute>
        </ns2:profile>
    </ns2:EditProfileRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:EditProfileResponse xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd"/
>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
```

```
    },
    "Body": {
        "EditProfileRequest": {
            "unitId": 10060344,
            "profile": {
                "attribute": [
                    {
                        "key": "LAST_NAME",
                        "value": "Last name"
                    },
                    {
                        "key": "FIRST_NAME",
                        "value": "Fist name"
                    },
                    {
                        "key": "CELL_PHONE",
                        "value": "+71234567890"
                    },
                    {
                        "key": "DATE_OF_BIRTH",
                        "value": "1981-07-23"
                    }
                ]
            }
        }
    }
}}
```

JSON response:

```
{"Envelope": {
    "Body": {
        "EditProfileResponse": {}
    }
}}
```

Check profile

SOAP request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-156585797757810318348 2" xmlns:wsse="http://
docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:CheckProfileRequest xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd"
 xmlns="">
            <ns2:unitId>10037569</ns2:unitId>
        </ns2:CheckProfileRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:CheckProfileResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
```

```
            <ns2:status>DATA_REQUIRED</ns2:status>
            <ns2:requestInfo>
                <ns2:action>REQUEST</ns2:action>
                <ns2:scope>Founder</ns2:scope>
                <ns2:method>EditProfile</ns2:method>
                <ns2:profile>
                    <ns2:unitId>10037569</ns2:unitId>
                    <ns2:profileId>10073968</ns2:profileId>
                    <ns2:profile>
                        <ns2:attribute>
                            <ns2:key>FOUNDER_SHARE_PERCENTAGE</ns2:key>
                            <ns2:value>number{0...100}</ns2:value>
                        </ns2:attribute>
                        <ns2:attribute>
                            <ns2:key>FOUNDER_SHARE_NUMERATOR</ns2:key>
                            <ns2:value>number{0...}</ns2:value>
                        </ns2:attribute>
                        <ns2:attribute>
                            <ns2:key>FOUNDER_SHARE_DENOMINATOR</ns2:key>
                            <ns2:value>number{0...}</ns2:value>
                        </ns2:attribute>
                    </ns2:profile>
                </ns2:profile>
            </ns2:requestInfo>
            <ns2:requestInfo>
                <ns2:action>REQUEST</ns2:action>
                <ns2:scope>Personal</ns2:scope>
                <ns2:method>EditProfile</ns2:method>
                <ns2:profile>
                    <ns2:unitId>10037569</ns2:unitId>
                    <ns2:profileId>10042500</ns2:profileId>
                    <ns2:profile>
                        <ns2:attribute>
                            <ns2:key>CONDITIONS_CORRECT_DATA</ns2:key>
                            <ns2:value>Y|N</ns2:value>
                        </ns2:attribute>
                    </ns2:profile>
                </ns2:profile>
            </ns2:requestInfo>
            <ns2:requestInfo>
                <ns2:action>CALL_SERVICE_SUPPORT</ns2:action>
                <ns2:scope>Juridical</ns2:scope>
                <ns2:juridical>
                    <ns2:attribute>
                        <ns2:key>OKATO</ns2:key>
                        <ns2:value>number{11 [КЧ | 222 [КЧ | 333 [КЧ | 444]]]}</
ns2:value>
                    </ns2:attribute>
                </ns2:juridical>
            </ns2:requestInfo>
            <ns2:foundersTotalShare>5.0</ns2:foundersTotalShare>
        </ns2:CheckProfileResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
          "Password": "PASSWORD"
        }
      }
    },
    "Body": {
      "CheckProfileRequest": {
        "unitId": 10037569
```

```
      }
    }
  }
}
```

JSON response:

```
{
  "Envelope": {
    "Body": {
      "CheckProfileResponse": {
        "foundersTotalShare": 5.0,
        "requestInfo": [
          {
            "method": "EditProfile",
            "scope": "Founder",
            "profile": {
              "profileId": 10073968,
              "profile": {
                "attribute": [
                  {
                    "value": "number{0...100}",
                    "key": "FOUNDER_SHARE_PERCENTAGE"
                  },
                  {
                    "value": "number{0...}",
                    "key": "FOUNDER_SHARE_NUMERATOR"
                  },
                  {
                    "value": "number{0...}",
                    "key": "FOUNDER_SHARE_DENOMINATOR"
                  }
                ]
              },
              "unitId": 10037569
            },
            "action": "REQUEST"
          },
          {
            "method": "EditProfile",
            "scope": "Personal",
            "profile": {
              "profileId": 10042500,
              "profile": {
                "attribute": [
                  {
                    "value": "Y|N",
                    "key": "CONDITIONS_CORRECT_DATA"
                  }
                ]
              },
              "unitId": 10037569
            },
            "action": "REQUEST"
          },
          {
            "juridical": {
              "attribute": [
                {
                  "value": "number{11 [КЧ | 222 [КЧ | 333 [КЧ | 444]]]}",
                  "key": "OKATO"
                }
              ]
            },
            "scope": "Juridical",
            "action": "CALL_SERVICE_SUPPORT"
          }
        ],
        "status": "DATA_REQUIRED"
      }
```

```
    }
  }
}
```

Accounts examples

Create account

SOAP request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="XWSSGID-1419574020701190066033 4"
 xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:CreateAccountRequest xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd"
 xmlns="">
            <ns2:currency>RUB</ns2:currency>
            <ns2:alias>Alias</ns2:alias>
            <ns2:paymentPassword>12345</ns2:paymentPassword>
            <ns2:unitId>10055203</ns2:unitId>
            <ns2:onSuccessfulDebitUrl>https://server.com/moneta_callback</
ns2:onSuccessfulDebitUrl>
            <ns2:onSuccessfulCreditUrl>https://server.com/moneta_callback</
ns2:onSuccessfulCreditUrl>
            <ns2:signature>QWERTY001</ns2:signature>
            <ns2:lowBalanceThreshold>0</ns2:lowBalanceThreshold>
            <ns2:highBalanceThreshold>200000</ns2:highBalanceThreshold>
            <ns2:prototypeAccountId>86360536</ns2:prototypeAccountId>
            <ns2:onCancelledDebitUrl>https://server.com/moneta_callback</
ns2:onCancelledDebitUrl>
            <ns2:onCancelledCreditUrl>https://server.com/moneta_callback</
ns2:onCancelledCreditUrl>
            <ns2:attribute>
                <ns2:key>ALIAS</ns2:key>
                <ns2:value>Alias</ns2:value>
                <ns2:published>true</ns2:published>
            </ns2:attribute>
        </ns2:CreateAccountRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
    <ns2:CreateAccountResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">86323445</ns2:CreateAccountResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{"Envelope": {
    "Header": {
```

```
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "CreateAccountRequest": {
            "currency": "RUB",
            "alias": "Alias",
            "paymentPassword": "12345",
            "unitId": 10060344,
            "onSuccessfulDebitUrl": "https://server.com/moneta_callback",
            "onSuccessfulCreditUrl": "https://server.com/moneta_callback",
            "signature": "QWERTY001",
            "lowBalanceThreshold": 50000,
            "highBalanceThreshold": 100000,
            "prototypeAccountId": 86360536,
            "onCancelledDebitUrl": "https://server.com/moneta_callback",
            "onCancelledCreditUrl": "https://server.com/moneta_callback",
            "attribute": [
                {
                    "key": "ALIAS",
                    "value": "Alias",
                    "published": true
                }
            ]
        }
    }
}}
```

JSON response:

```
{"Envelope": {
    "Body": {
        "CreateAccountResponse": 87198713
    }
}}
```

Edit account

SOAP request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
        <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="XWSSGID-1419574021946-353601982"
 xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd">
            <wsse:Username>USERNAME</wsse:Username>
            <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
        </wsse:UsernameToken>
    </wsse:Security>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
    <ns2:EditAccountRequest xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd"
 xmlns="">
        <ns2:id>86323445</ns2:id>
        <ns2:alias>Alias</ns2:alias>
        <ns2:paymentPassword>11111</ns2:paymentPassword>
        <ns2:oldPaymentPassword>12345</ns2:oldPaymentPassword>
        <ns2:onSuccessfulDebitUrl>https://server.com/moneta_callback</
ns2:onSuccessfulDebitUrl>
        <ns2:onSuccessfulCreditUrl>https://server.com/moneta_callback</
ns2:onSuccessfulCreditUrl>
```

```
        <ns2:signature>QWERTY123</ns2:signature>
        <ns2:lowBalanceThreshold>10000</ns2:lowBalanceThreshold>
        <ns2:highBalanceThreshold>500000</ns2:highBalanceThreshold>
        <ns2:onCancelledDebitUrl>https://server.com/moneta_callback</
ns2:onCancelledDebitUrl>
        <ns2:onCancelledCreditUrl>https://server.com/moneta_callback</
ns2:onCancelledCreditUrl>
        <ns2:attribute>
            <ns2:key>ALIAS</ns2:key>
            <ns2:value>Alias</ns2:value>
            <ns2:published>true</ns2:published>
        </ns2:attribute>
    </ns2:EditAccountRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
    <ns2:EditAccountResponse xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd"/>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "EditAccountRequest": {
            "id": 87198713,
            "alias": "Alias",
            "paymentPassword": "11111",
            "oldPaymentPassword": "12345",
            "onSuccessfulDebitUrl": "https://server.com/moneta_callback",
            "onSuccessfulCreditUrl": "https://server.com/moneta_callback",
            "signature": "QWERTY123",
            "lowBalanceThreshold": 10000,
            "highBalanceThreshold": 500000,
            "onCancelledDebitUrl": "https://server.com/moneta_callback",
            "onCancelledCreditUrl": "https://server.com/moneta_callback",
            "attribute": [
                {
                    "key": "ALIAS",
                    "value": "Alias",
                    "published": true
                }
            ]
        }
    }
}}
```

JSON response:

```
{"Envelope": {
    "Body": {
        "EditAccountResponse": {}
    }
}}
```

Read account

Read list of accounts

SOAP request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
        <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="XWSSGID-1419574025786-400683759"
 xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd">
            <wsse:Username>USERNAME</wsse:Username>
            <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
        </wsse:UsernameToken>
    </wsse:Security>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
    <ns2:FindAccountsListRequest xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd"
 ns2:version="VERSION_2" xmlns="">
        <ns2:unitId>10055203</ns2:unitId>
    </ns2:FindAccountsListRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
    <ns2:FindAccountsListResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
        <ns2:account>
            <ns2:id>26939899</ns2:id>
            <ns2:currency>RUB</ns2:currency>
            <ns2:balance>0</ns2:balance>
            <ns2:availableBalance>0</ns2:availableBalance>
            <ns2:type>2</ns2:type>
            <ns2:status>1</ns2:status>
            <ns2:alias>Alias</ns2:alias>
            <ns2:attribute>
                <ns2:key>paymentPasswordType</ns2:key>
                <ns2:value>SMS_SIMPLE</ns2:value>
            </ns2:attribute>
            <ns2:attribute>
                <ns2:key>paymentPasswordChallengeRequired</ns2:key>
                <ns2:value>true</ns2:value>
            </ns2:attribute>
            <ns2:attribute>
                <ns2:key>ALIAS</ns2:key>
                <ns2:value>Alias</ns2:value>
                <ns2:approved>false</ns2:approved>
                <ns2:published>false</ns2:published>
            </ns2:attribute>
        </ns2:account>
        <ns2:account>
            <ns2:id>86323445</ns2:id>
            <ns2:currency>RUB</ns2:currency>
            <ns2:balance>0</ns2:balance>
            <ns2:availableBalance>0</ns2:availableBalance>
            <ns2:type>2</ns2:type>
            <ns2:status>1</ns2:status>
            <ns2:alias>Alias</ns2:alias>
            <ns2:onSuccessfulDebitUrl>https://server.com/moneta_callback</
ns2:onSuccessfulDebitUrl>
```

```
            <ns2:onSuccessfulCreditUrl>https://server.com/moneta_callback</
ns2:onSuccessfulCreditUrl>
            <ns2:signature>QWERTY123</ns2:signature>
            <ns2:lowBalanceThreshold>-10000</ns2:lowBalanceThreshold>
            <ns2:highBalanceThreshold>500000</ns2:highBalanceThreshold>
            <ns2:prototypeAccountId>86360536</ns2:prototypeAccountId>
            <ns2:onCancelledDebitUrl>https://server.com/moneta_callback</
ns2:onCancelledDebitUrl>
            <ns2:onCancelledCreditUrl>https://server.com/moneta_callback</
ns2:onCancelledCreditUrl>
            <ns2:attribute>
                <ns2:key>paymentPasswordType</ns2:key>
                <ns2:value>STATIC</ns2:value>
            </ns2:attribute>
            <ns2:attribute>
                <ns2:key>ALIAS</ns2:key>
                <ns2:value>Alias</ns2:value>
                <ns2:approved>false</ns2:approved>
                <ns2:published>true</ns2:published>
            </ns2:attribute>
        </ns2:account>
    </ns2:FindAccountsListResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "FindAccountsListRequest": {
            "version": "VERSION_2",
            "unitId": 10060344
        }
    }
}}
```

JSON response:

```
{"Envelope": {
    "Body": {
        "FindAccountsListResponse": {
            "account": [
                {
                    "highBalanceThreshold": 500000,
                    "lowBalanceThreshold": -10000,
                    "status": 1,
                    "alias": "Alias",
                    "attribute": [
                        {
                            "key": "paymentPasswordType",
                            "value": "SMS_SIMPLE"
                        },
                        {
                            "key": "paymentPasswordChallengeRequired",
                            "value": "true"
                        },
                        {
                            "key": "ALIAS",
                            "value": "Alias",
                            "approved": false,
                            "published": false
                        }
```

```
                    ],
                    "availableBalance": 0,
                    "type": 2,
                    "currency": "RUB",
                    "id": 30523002,
                    "balance": 0
                },
                {
                    "highBalanceThreshold": 500000,
                    "lowBalanceThreshold": -10000,
                    "status": 1,
                    "alias": "Alias",
                    "attribute": [
                        {
                            "key": "paymentPasswordType",
                            "value": "STATIC"
                        },
                        {
                            "key": "ALIAS",
                            "value": "Alias",
                            "approved": false,
                            "published": true
                        }
                    ],
                    "onSuccessfulCreditUrl": "",
                    "availableBalance": 0,
                    "type": 2,
                    "onCancelledCreditUrl": "",
                    "currency": "RUB",
                    "id": 87198713,
                    "balance": 0,
                    "onCancelledDebitUrl": "",
                    "prototypeAccountId": 86360536,
                    "onSuccessfulDebitUrl": "",
                    "signature": "QWERTY123"
                }
            ]
        }
    }
}}
```


Read account

SOAP request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
        <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="XWSSGID-1419574022832875529758"
 xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd">
            <wsse:Username>USERNAME</wsse:Username>
            <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
        </wsse:UsernameToken>
    </wsse:Security>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
    <ns2:FindAccountByIdRequest xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd"
 ns2:version="VERSION_2" xmlns="">86323445</ns2:FindAccountByIdRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
```

```
<SOAP-ENV:Body>
    <ns2:FindAccountByIdResponse xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd">
        <ns2:account>
            <ns2:id>86323445</ns2:id>
            <ns2:currency>RUB</ns2:currency>
            <ns2:balance>0</ns2:balance>
            <ns2:availableBalance>0</ns2:availableBalance>
            <ns2:type>2</ns2:type>
            <ns2:status>1</ns2:status>
            <ns2:alias>Alias</ns2:alias>
            <ns2:onSuccessfulDebitUrl>https://server.com/moneta_callback</
ns2:onSuccessfulDebitUrl>
            <ns2:onSuccessfulCreditUrl>https://server.com/moneta_callback</
ns2:onSuccessfulCreditUrl>
            <ns2:signature>QWERTY123</ns2:signature>
            <ns2:lowBalanceThreshold>-10000</ns2:lowBalanceThreshold>
            <ns2:highBalanceThreshold>500000</ns2:highBalanceThreshold>
            <ns2:prototypeAccountId>86360536</ns2:prototypeAccountId>
            <ns2:onCancelledDebitUrl>https://server.com/moneta_callback</
ns2:onCancelledDebitUrl>
            <ns2:onCancelledCreditUrl>https://server.com/moneta_callback</
ns2:onCancelledCreditUrl>
            <ns2:attribute>
                <ns2:key>paymentPasswordType</ns2:key>
                <ns2:value>STATIC</ns2:value>
            </ns2:attribute>
            <ns2:attribute>
                <ns2:key>ALIAS</ns2:key>
                <ns2:value>Alias</ns2:value>
                <ns2:approved>false</ns2:approved>
                <ns2:published>true</ns2:published>
            </ns2:attribute>
        </ns2:account>
    </ns2:FindAccountByIdResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "FindAccountByIdRequest": {
            "value": 87198713,
            "version": "VERSION_2"
        }
    }
}}
```

JSON response:

```
{"Envelope": {
    "Body": {
        "FindAccountByIdResponse": {
            "account": {
                "highBalanceThreshold": 500000,
                "lowBalanceThreshold": -10000,
                "status": 1,
                "alias": "Alias",
                "attribute": [
                    {
                        "key": "paymentPasswordType",
                        "value": "STATIC"
```

```
                    },
                    {
                        "key": "ALIAS",
                        "value": "Alias",
                        "approved": false,
                        "published": true
                    }
                ],
                "onSuccessfulCreditUrl": "",
                "availableBalance": 0,
                "type": 2,
                "onCancelledCreditUrl": "",
                "currency": "RUB",
                "id": 87198713,
                "balance": 0,
                "onCancelledDebitUrl": "",
                "prototypeAccountId": 86360536,
                "onSuccessfulDebitUrl": "",
                "signature": "QWERTY123"
            }
        }
    }
}}
```

Account with SMS payment password

Create account with SMS payment password

SOAP request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
        <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="XWSSGID-1419574023021-692096092"
 xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd">
            <wsse:Username>USERNAME</wsse:Username>
            <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
        </wsse:UsernameToken>
    </wsse:Security>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
    <ns2:CreateAccountRequest xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd"
 xmlns="">
        <ns2:currency>RUB</ns2:currency>
        <ns2:alias>Alias</ns2:alias>
        <ns2:paymentPasswordType>SMS_SIMPLE</ns2:paymentPasswordType>
        <ns2:unitId>10055203</ns2:unitId>
    </ns2:CreateAccountRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
    <ns2:CreateAccountResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">26939899</ns2:CreateAccountResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{"Envelope": {
```

```
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "CreateAccountRequest": {
            "currency": "RUB",
            "alias": "Alias",
            "paymentPasswordType": "SMS_SIMPLE",
            "unitId": 10060344
        }
    }
}}
```

JSON response:

```
{"Envelope": {
    "Body": {
        "CreateAccountResponse": 30523002
    }
}}
```

Send SMS payment password

SOAP request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
        <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="XWSSGID-14195740239361078121633"
 xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd">
            <wsse:Username>USERNAME</wsse:Username>
            <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
        </wsse:UsernameToken>
    </wsse:Security>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
    <ns2:GetAccountPaymentPasswordChallengeRequest xmlns:ns2="http://www.moneta.ru/
schemas/messages.xsd" xmlns="">
        <ns2:accountId>26939899</ns2:accountId>
    </ns2:GetAccountPaymentPasswordChallengeRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
    <ns2:GetAccountPaymentPasswordChallengeResponse xmlns:ns2="http://www.moneta.ru/
schemas/messages.xsd">
        <ns2:paymentPasswordChallenge>SMS</ns2:paymentPasswordChallenge>
    </ns2:GetAccountPaymentPasswordChallengeResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{"Envelope": {
    "Header": {
```

```
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "GetAccountPaymentPasswordChallengeRequest": {
            "accountId": 30523002
        }
    }
}}
```

JSON response:

```
{"Envelope": {
    "Body": {
        "GetAccountPaymentPasswordChallengeResponse": {
            "paymentPasswordChallenge": "SMS"
        }
    }
}}
```

Block/unblock account

Block an account

SOAP request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken
                    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-1505895829924102345938 4"
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:BlockAccountRequest xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd"
 xmlns="">
            <ns2:id>12345678</ns2:id>
        </ns2:BlockAccountRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:BlockAccountResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd"/>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{
```

```
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
          "Password": "PASSWORD"
        }
      }
    },
    "Body": {
      "BlockAccountRequest": {
        "id": 12345678
      }
    }
  }
}
```

JSON response:

```
{
  "Envelope": {
    "Body": {
      "BlockAccountResponse": {}
    }
  }
}
```

Unblock an account

SOAP request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken
                    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-1505895831583159455049 5"
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:UnblockAccountRequest xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd" xmlns="">
            <ns2:id>12345678</ns2:id>
            <ns2:secretAnswer>ANSWER</ns2:secretAnswer>
            <ns2:paymentPassword>
                <ns2:paymentPassword>12345</ns2:paymentPassword>
            </ns2:paymentPassword>
        </ns2:UnblockAccountRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:UnblockAccountResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd"/>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
          "Password": "PASSWORD"
        }
      }
    },
    "Body": {
      "UnblockAccountRequest": {
        "id": 12345678,
        "secretAnswer": "ANSWER",
        "paymentPassword": {
          "paymentPassword": "12345"
        }
      }
    }
  }
}
```

JSON response:

```
{
  "Envelope": {
    "Body": {
      "UnblockAccountResponse": {}
    }
  }
}
```

Information about available payment systems

SOAP request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken
                    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-1504855068128-1992070225"
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:PaymentSystemInfoRequest xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd" xmlns="">
            <ns2:accountId>12345678</ns2:accountId>
            <ns2:amount>100</ns2:amount>
        </ns2:PaymentSystemInfoRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
```

```
        <ns2:PaymentSystemInfoResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
            <ns2:items>
                <ns2:unitId>11111111</ns2:unitId>
                <ns2:name>VISA, MasterCard</ns2:name>
                <ns2:icon>visa_mc_mir.jpg</ns2:icon>
                <ns2:logicalGroup>Cards</ns2:logicalGroup>
                <ns2:infoUrl>
                    <ns2:href></ns2:href>
                    <ns2:text></ns2:text>
                </ns2:infoUrl>
                <ns2:referenceData>
                    <ns2:period>1 min.</ns2:period>
                    <ns2:percentage>0,0%</ns2:percentage>
                    <ns2:sourceFee>0</ns2:sourceFee>
                    <ns2:sourceFeeExt>0</ns2:sourceFeeExt>
                    <ns2:targetFee>0.01</ns2:targetFee>
                    <ns2:targetFeeExt>0</ns2:targetFeeExt>
                </ns2:referenceData>
                <ns2:infoTariff>
                    <ns2:targetRangeCurrency>RUB</ns2:targetRangeCurrency>
                    <ns2:targetAmountMax>10000000</ns2:targetAmountMax>
                </ns2:infoTariff>
                <ns2:currencies>RUB</ns2:currencies>
                <ns2:psAccountIds>317</ns2:psAccountIds>
            </ns2:items>
        </ns2:PaymentSystemInfoResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{
  "Header": {
    "Security": {
      "UsernameToken": {
        "Username": "USERNAME",
        "Password": "PASSWORD"
      }
    }
  },
  "Envelope": {
    "Body": {
      "PaymentSystemInfoRequest": {
        "amount": 100,
        "accountId": 12345678
      }
    }
  }
}
```

JSON response:

```
{
  "Envelope": {
    "Body": {
      "PaymentSystemInfoResponse": {
        "items": [
          {
            "icon": "visa_mc_mir.jpg",
            "currencies": "RUB",
            "logicalGroup": "Cards",
            "name": "VISA, MasterCard",
            "infoTariff": {
              "targetAmountMax": 10000000,
              "targetRangeCurrency": "RUB"
            },
            "referenceData": {
              "targetFeeExt": 0,
              "percentage": "0,0%",
```

```
              "sourceFeeExt": 0,
              "period": "1 min.",
              "targetFee": 0.01,
              "sourceFee": 0
            },
            "infoUrl": {
              "text": "",
              "href": ""
            },
            "psAccountIds": "317",
            "unitId": 11111111
          }
        ]
      }
    }
  }
}
```

Documents examples

Read documents

SOAP request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
        <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="XWSSGID-14175259995878204332O"
 xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd">
            <wsse:Username>USERNAME</wsse:Username>
            <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
        </wsse:UsernameToken>
    </wsse:Security>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
    <ns2:FindProfileDocumentsRequest xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd" xmlns="">
        <ns2:unitId>10054789</ns2:unitId>
    </ns2:FindProfileDocumentsRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
    <ns2:FindProfileDocumentsResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
        <ns2:document>
            <ns2:id>9996</ns2:id>
            <ns2:type>PASSPORT</ns2:type>
            <ns2:attribute>
                <ns2:key>series</ns2:key>
                <ns2:value>8804</ns2:value>
                <ns2:approved>false</ns2:approved>
            </ns2:attribute>
            <ns2:attribute>
                <ns2:key>issuer</ns2:key>
                <ns2:value>issuer</ns2:value>
                <ns2:approved>false</ns2:approved>
            </ns2:attribute>
            <ns2:attribute>
```

```
                <ns2:key>department</ns2:key>
                <ns2:value>123-001</ns2:value>
                <ns2:approved>false</ns2:approved>
            </ns2:attribute>
            <ns2:attribute>
                <ns2:key>issued</ns2:key>
                <ns2:value>2004-07-19</ns2:value>
                <ns2:approved>false</ns2:approved>
            </ns2:attribute>
            <ns2:attribute>
                <ns2:key>number</ns2:key>
                <ns2:value>424000</ns2:value>
                <ns2:approved>false</ns2:approved>
            </ns2:attribute>
            <ns2:attribute>
                <ns2:key>modificationdate</ns2:key>
                <ns2:value>2014-12-02T16:12:48.000+03:00</ns2:value>
                <ns2:approved>true</ns2:approved>
            </ns2:attribute>
            <ns2:hasAttachedFiles>false</ns2:hasAttachedFiles>
        </ns2:document>
    </ns2:FindProfileDocumentsResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "FindProfileDocumentsRequest": {
            "unitId": 10060344
        }
    }
}}
```

JSON response:

```
{"Envelope": {
    "Body": {
        "FindProfileDocumentsResponse": {
            "document": [
                {
                    "id": 11334,
                    "hasAttachedFiles": false,
                    "attribute": [
                        {
                            "approved": false,
                            "value": "8804",
                            "key": "series"
                        },
                        {
                            "approved": false,
                            "value": "issuer",
                            "key": "issuer"
                        },
                        {
                            "approved": false,
                            "value": "123-001",
                            "key": "department"
                        },
                        {
                            "approved": false,
```

```
                            "value": "2004-07-19",
                            "key": "issued"
                        },
                        {
                            "approved": false,
                            "value": "424000",
                            "key": "number"
                        },
                        {
                            "approved": true,
                            "value": "2015-06-02T15:48:48.000+03:00",
                            "key": "modificationdate"
                        }
                    ],
                    "type": "PASSPORT"
                }
            ]
        }
    }
}}
```

Create document

SOAP request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
        <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="XWSSGID-1417525998490-750226304"
 xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd">
            <wsse:Username>USERNAME</wsse:Username>
            <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
        </wsse:UsernameToken>
    </wsse:Security>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
    <ns2:CreateProfileDocumentRequest xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd" xmlns="">
        <ns2:type>PASSPORT</ns2:type>
        <ns2:attribute>
            <ns2:key>SERIES</ns2:key>
            <ns2:value>8803</ns2:value>
        </ns2:attribute>
        <ns2:attribute>
            <ns2:key>NUMBER</ns2:key>
            <ns2:value>424001</ns2:value>
        </ns2:attribute>
        <ns2:attribute>
            <ns2:key>ISSUER</ns2:key>
            <ns2:value>issuer</ns2:value>
        </ns2:attribute>
        <ns2:attribute>
            <ns2:key>ISSUED</ns2:key>
            <ns2:value>2004-05-18</ns2:value>
        </ns2:attribute>
        <ns2:attribute>
            <ns2:key>DEPARTMENT</ns2:key>
            <ns2:value>123-002</ns2:value>
        </ns2:attribute>
        <ns2:unitId>10054789</ns2:unitId>
    </ns2:CreateProfileDocumentRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
    <ns2:CreateProfileDocumentResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
        <ns2:id>9996</ns2:id>
    </ns2:CreateProfileDocumentResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "CreateProfileDocumentRequest": {
            "type": "PASSPORT",
            "attribute": [
                {
                    "key": "SERIES",
                    "value": "8803"
                },
                {
                    "key": "NUMBER",
                    "value": "424001"
                },
                {
                    "key": "ISSUER",
                    "value": "issuer"
                },
                {
                    "key": "ISSUED",
                    "value": "2004-05-18"
                },
                {
                    "key": "DEPARTMENT",
                    "value": "123-002"
                }
            ],
            "unitId": 10060344
        }
    }
}}
```

JSON response:

```
{"Envelope": {
    "Body": {
        "CreateProfileDocumentResponse": {
            "id": 11334
        }
    }
}}
```

Edit document

SOAP request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
```

```
<SOAP-ENV:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
        <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="XWSSGID-1417525999030128341455 6"
 xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd">
            <wsse:Username>USERNAME</wsse:Username>
            <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
        </wsse:UsernameToken>
    </wsse:Security>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
    <ns2:EditProfileDocumentRequest xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd" xmlns="">
        <ns2:id>9996</ns2:id>
        <ns2:attribute>
            <ns2:key>SERIES</ns2:key>
            <ns2:value>8804</ns2:value>
        </ns2:attribute>
        <ns2:attribute>
            <ns2:key>NUMBER</ns2:key>
            <ns2:value>424000</ns2:value>
        </ns2:attribute>
        <ns2:attribute>
            <ns2:key>ISSUER</ns2:key>
            <ns2:value>issuer</ns2:value>
        </ns2:attribute>
        <ns2:attribute>
            <ns2:key>ISSUED</ns2:key>
            <ns2:value>2004-07-19</ns2:value>
        </ns2:attribute>
        <ns2:attribute>
            <ns2:key>DEPARTMENT</ns2:key>
            <ns2:value>123-001</ns2:value>
        </ns2:attribute>
        <ns2:unitId>10054789</ns2:unitId>
    </ns2:EditProfileDocumentRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
    <ns2:EditProfileDocumentResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd"/>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "EditProfileDocumentRequest": {
            "id": 11334,
            "attribute": [
                {
                    "key": "SERIES",
                    "value": "8804"
```

```
                },
                {
                    "key": "NUMBER",
                    "value": "424000"
                },
                {
                    "key": "ISSUER",
                    "value": "issuer"
                },
                {
                    "key": "ISSUED",
                    "value": "2004-07-19"
                },
                {
                    "key": "DEPARTMENT",
                    "value": "123-001"
                }
            ],
            "unitId": 10060344
        }
    }
}}
```

JSON response:

```
{"Envelope": {
    "Body": {
        "EditProfileDocumentResponse": {}
    }
}}
```

Bank accounts examples

Read bank account details

SOAP request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
        <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="XWSSGID-14175260021691023607342"
 xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd">
            <wsse:Username>USERNAME</wsse:Username>
            <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
        </wsse:UsernameToken>
    </wsse:Security>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
    <ns2:FindBankAccountsRequest xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd"
 ns2:version="VERSION_2" xmlns="">
        <ns2:unitId>10054789</ns2:unitId>
    </ns2:FindBankAccountsRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
    <ns2:FindBankAccountsResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
```

```
        <ns2:bankAccount>
            <ns2:id>10000914</ns2:id>
            <ns2:attribute>
                <ns2:key>bank</ns2:key>
                <ns2:value>Bank name</ns2:value>
                <ns2:approved>false</ns2:approved>
                <ns2:published>false</ns2:published>
            </ns2:attribute>
            <ns2:attribute>
                <ns2:key>account</ns2:key>
                <ns2:value>12345678901234567890</ns2:value>
                <ns2:approved>false</ns2:approved>
                <ns2:published>false</ns2:published>
            </ns2:attribute>
            <ns2:attribute>
                <ns2:key>bik</ns2:key>
                <ns2:value>040037469</ns2:value>
                <ns2:approved>false</ns2:approved>
                <ns2:published>false</ns2:published>
            </ns2:attribute>
            <ns2:attribute>
                <ns2:key>is_international</ns2:key>
                <ns2:value>false</ns2:value>
                <ns2:approved>false</ns2:approved>
                <ns2:published>false</ns2:published>
            </ns2:attribute>
            <ns2:attribute>
                <ns2:key>corr_account</ns2:key>
                <ns2:value>30101810900000000469</ns2:value>
                <ns2:approved>false</ns2:approved>
                <ns2:published>false</ns2:published>
            </ns2:attribute>
        </ns2:bankAccount>
    </ns2:FindBankAccountsResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "FindBankAccountsRequest": {
            "version": "VERSION_2",
            "unitId": 10060344
        }
    }
}}
```

JSON response:

```
{"Envelope": {
    "Body": {
        "FindBankAccountsResponse": {
            "bankAccount": [
                {
                    "id": 10001185,
                    "attribute": [
                        {
                            "approved": false,
                            "value": "Bank name",
                            "published": false,
                            "key": "bank"
```

```
                        },
                        {
                            "approved": false,
                            "value": "12345678901234567897",
                            "published": false,
                            "key": "account"
                        },
                        {
                            "approved": false,
                            "value": "044585214",
                            "published": false,
                            "key": "bik"
                        },
                        {
                            "approved": false,
                            "value": "30101810800000000214",
                            "published": false,
                            "key": "corr_account"
                        },
                        {
                            "approved": false,
                            "value": "false",
                            "published": false,
                            "key": "is_international"
                        }
                    ]
                }
            ]
        }
    }
}}
```

Create bank account details

SOAP request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd"SOAP-ENV:mustUnderstand="1">
        <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="XWSSGID-1417526001372-735973564"
 xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd">
            <wsse:Username>USERNAME</wsse:Username>
            <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
        </wsse:UsernameToken>
    </wsse:Security>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
    <ns2:CreateBankAccountRequest xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd"
 xmlns="">
        <ns2:attribute>
            <ns2:key>BIK</ns2:key>
            <ns2:value>040037469</ns2:value>
        </ns2:attribute>
        <ns2:attribute>
            <ns2:key>ACCOUNT</ns2:key>
            <ns2:value>12345678901234567890</ns2:value>
        </ns2:attribute>
        <ns2:unitId>10054789</ns2:unitId>
    </ns2:CreateBankAccountRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
```

```
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
    <ns2:CreateBankAccountResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
        <ns2:id>10000914</ns2:id>
    </ns2:CreateBankAccountResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "CreateBankAccountRequest": {
            "attribute": [
                {
                    "key": "BIK",
                    "value": "044585214"
                },
                {
                    "key": "ACCOUNT",
                    "value": "12345678901234567897"
                }
            ],
            "unitId": 10060344
        }
    }
}}
```

JSON response:

```
{"Envelope": {
    "Body": {
        "CreateBankAccountResponse": {
            "id": 10001185
        }
    }
}}
```

Edit bank account details

SOAP request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd"SOAP-ENV:mustUnderstand="1">
        <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="XWSSGID-1417526001803-917345112"
            xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd">
            <wsse:Username>USERNAME</wsse:Username>
            <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
        </wsse:UsernameToken>
    </wsse:Security>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
    <ns2:EditBankAccountRequest xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd"
 xmlns="">
        <ns2:id>10000914</ns2:id>
```

```
        <ns2:attribute>
            <ns2:key>BIK</ns2:key>
            <ns2:value>040037469</ns2:value>
        </ns2:attribute>
        <ns2:attribute>
            <ns2:key>ACCOUNT</ns2:key>
            <ns2:value>12345678901234567890</ns2:value>
        </ns2:attribute>
        <ns2:unitId>10054789</ns2:unitId>
    </ns2:EditBankAccountRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
    <ns2:EditBankAccountResponse xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd"/
>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "EditBankAccountRequest": {
            "id": 10001185,
            "attribute": [
                {
                    "key": "BIK",
                    "value": "044585214"
                },
                {
                    "key": "ACCOUNT",
                    "value": "12345678901234567897"
                }
            ],
            "unitId": 10060344
        }
    }
}}
```

JSON response:

```
{"Envelope": {
    "Body": {
        "EditBankAccountResponse": {}
    }
}}
```

Legal information examples

Read legal information

See: *FindLegalInformation Endpoint* on page 159.

SOAP request:

```
<soapenv:Envelope xmlns:mes="http://www.moneta.ru/schemas/messages.xsd"
 xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Header>
        <wsse:Security soapenv:mustUnderstand="1" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
                       xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-utility-1.0.xsd">
            <wsse:UsernameToken wsu:Id="UsernameToken-
DC84E436F5096EBAA115281025930841">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </soapenv:Header>
    <soapenv:Body>
        <mes:FindLegalInformationRequest mes:version="VERSION_2">
            <mes:unitId>11111</mes:unitId>
        </mes:FindLegalInformationRequest>
    </soapenv:Body>
</soapenv:Envelope>
```

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:FindLegalInformationResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
            <ns2:legalInformation>
                <ns2:id>22222222</ns2:id>
                <ns2:attribute>
                    <ns2:key>ogrn</ns2:key>
                    <ns2:value>OGRN</ns2:value>
                    <ns2:approved>true</ns2:approved>
                    <ns2:published>true</ns2:published>
                </ns2:attribute>
                <ns2:attribute>
                    <ns2:key>okved</ns2:key>
                    <ns2:value>OKVED</ns2:value>
                    <ns2:approved>true</ns2:approved>
                    <ns2:published>false</ns2:published>
                </ns2:attribute>
                <ns2:attribute>
                    <ns2:key>okpo</ns2:key>
                    <ns2:value>OKPO</ns2:value>
                    <ns2:approved>true</ns2:approved>
                    <ns2:published>false</ns2:published>
                </ns2:attribute>
                <ns2:attribute>
                    <ns2:key>kpp</ns2:key>
                    <ns2:value>KPP</ns2:value>
                    <ns2:approved>true</ns2:approved>
                    <ns2:published>true</ns2:published>
                </ns2:attribute>
            </ns2:legalInformation>
        </ns2:FindLegalInformationResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
```

```
          "Password": "PASSWORD"
        }
      }
    },
    "Body": {
      "FindLegalInformationRequest": {
        "version": "VERSION_2",
        "unitId": 11111
      }
    }
  }
}
```

JSON response:

```
{
  "Envelope": {
    "Body": {
      "FindLegalInformationResponse": {
        "legalInformation": [
          {
            "id": 22222222,
            "attribute": [
              {
                "approved": true,
                "value": "OGRN",
                "published": true,
                "key": "ogrn"
              },
              {
                "approved": true,
                "value": "OKVED",
                "published": false,
                "key": "okved"
              },
              {
                "approved": true,
                "value": "OKPO",
                "published": false,
                "key": "okpo"
              },
              {
                "approved": true,
                "value": "KPP",
                "published": true,
                "key": "kpp"
              }
            ]
          }
        ]
      }
    }
  }
}
```

Reports examples

Read reports list

SOAP request:

```
<soapenv:Envelope xmlns:mes="http://www.moneta.ru/schemas/messages.xsd"
 xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Header>
        <wsse:Security soapenv:mustUnderstand="1" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
```

```
            <wsse:UsernameToken wsu:Id="UsernameToken-1" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </soapenv:Header>
    <soapenv:Body>
        <mes:FindReportsRequest>
            <mes:unitId>12345678</mes:unitId>
            <mes:year>2016</mes:year>
        </mes:FindReportsRequest>
    </soapenv:Body>
</soapenv:Envelope>
```

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:FindReportsResponse xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd">
            <ns2:report>
                <ns2:id>1084</ns2:id>
                <ns2:name>Act of acceptance</ns2:name>
                <ns2:unitId>12345678</ns2:unitId>
                <ns2:typeId>3</ns2:typeId>
                <ns2:reportInstance>
                    <ns2:id>2010</ns2:id>
                    <ns2:reportId>1084</ns2:reportId>
                    <ns2:year>2016</ns2:year>
                    <ns2:month>4</ns2:month>
                    <ns2:url>https://moneta.ru/downloadReport.htm?
reportInstanceId=2010&amp;date=2016-06-29_12-21&amp;
publicId=82739741-3968-4068-b620-328cdf4ce351&amp;
signature=4f93d845fe70d23e8afd062b941d1f547be29a6ee6c778ea2fcdbc53ac7db5ff
                    </ns2:url>
                    <ns2:urlExpirationDate>2016-06-29T12:21:00.000+03:00</
ns2:urlExpirationDate>
                </ns2:reportInstance>
            </ns2:report>
        </ns2:FindReportsResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
          "Password": "PASSWORD"
        }
      }
    },
    "Body": {
      "FindReportsRequest": {
        "unitId": 12345678,
        "year": 2016
      }
    }
  }
}
```

JSON response:

```
{
```

```
  "Envelope": {
    "Body": {
      "FindReportsResponse": {
        "report": [
          {
            "id": 1084,
            "reportInstance": [
              {
                "id": 2010,
                "month": 4,
                "year": 2016,
                "reportId": 1084,
                "url": "https://moneta.ru/downloadReport.htm?
reportInstanceId=2010&date=2016-06-29_12-04&
publicId=82739741-3968-4068-b620-328cdf4ce351&
signature=2757ef81cedb2b37ae14136a2f05b698541b6084f2b7be09ccce4de2ad83d726",
                "urlExpirationDate": "2016-06-29T12:04:00.000+03:00",
                "attribute": []
              }
            ],
            "name": "Act of acceptance",
            "attribute": [],
            "typeId": 3,
            "unitId": 12345678
          }
        ]
      }
    }
  }
}
```


Approve cell phone

Send confirmation code

SOAP request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
        <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="XWSSGID-1417526002418-988211293"
 xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd">
            <wsse:Username>USERNAME</wsse:Username>
            <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
        </wsse:UsernameToken>
    </wsse:Security>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
    <ns2:ApprovePhoneSendConfirmationRequest xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd" xmlns="">
        <ns2:unitId>10054789</ns2:unitId>
        <ns2:text>{CODE}</ns2:text>
    </ns2:ApprovePhoneSendConfirmationRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
    <ns2:ApprovePhoneSendConfirmationResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd">
```

```
        <ns2:phoneNumber>71234567890</ns2:phoneNumber>
    </ns2:ApprovePhoneSendConfirmationResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "ApprovePhoneSendConfirmationRequest": {
            "unitId": 10060344,
            "text": "{CODE}"
        }
    }
}}
```

JSON response:

```
{"Envelope": {
    "Body": {
        "ApprovePhoneSendConfirmationResponse": {
            "phoneNumber": "+71234567890"
        }
    }
}}
```

Approve cell phone

SOAP request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
        <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="XWSSGID-1417526003898-834549441"
 xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd">
            <wsse:Username>USERNAME</wsse:Username>
            <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
        </wsse:UsernameToken>
    </wsse:Security>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
    <ns2:ApprovePhoneApplyCodeRequest xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd" xmlns="">
        <ns2:unitId>10054789</ns2:unitId>
        <ns2:confirmationCode>999739</ns2:confirmationCode>
    </ns2:ApprovePhoneApplyCodeRequest>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
    <ns2:ApprovePhoneApplyCodeResponse xmlns:ns2="http://www.moneta.ru/schemas/
messages.xsd"/>
```

```
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request:

```
{"Envelope": {
    "Header": {
        "Security": {
            "UsernameToken": {
                "Username": "USERNAME",
                "Password": "PASSWORD"
            }
        }
    },
    "Body": {
        "ApprovePhoneApplyCodeRequest": {
            "unitId": 10060344,
            "confirmationCode": "27924"
        }
    }
}}
```

JSON response:

```
{"Envelope": {
    "Body": {
        "ApprovePhoneApplyCodeResponse": {}
    }
}}
```

SimplifiedIdentificationRequest (AsyncRequest) - simplified identification

Request for user simplified identification.

SimplifiedIdentification Endpoint on page 185

Examples

SOAP request (send request):

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken
                    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-150364571998148884173 7"
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:AsyncRequest xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd"
 xmlns="">
            <ns2:SimplifiedIdentificationRequest>
                <ns2:unitId>11111</ns2:unitId>
                <ns2:personalInformation>
                    <ns2:profile>
                        <ns2:attribute>
                            <ns2:key>LAST_NAME</ns2:key>
                            <ns2:value>Last name</ns2:value>
```

```
                        </ns2:attribute>
                        <ns2:attribute>
                            <ns2:key>FIRST_NAME</ns2:key>
                            <ns2:value>First name</ns2:value>
                        </ns2:attribute>
                        <ns2:attribute>
                            <ns2:key>MIDDLE_INITIAL_NAME</ns2:key>
                            <ns2:value>Middle initial</ns2:value>
                        </ns2:attribute>
                        <ns2:attribute>
                            <ns2:key>CELL_PHONE</ns2:key>
                            <ns2:value>79001234567</ns2:value>
                        </ns2:attribute>
                        <ns2:attribute>
                            <ns2:key>SNILS</ns2:key>
                            <ns2:value>000-000-000 00</ns2:value>
                        </ns2:attribute>
                        <ns2:attribute>
                            <ns2:key>INN</ns2:key>
                            <ns2:value>000000000000</ns2:value>
                        </ns2:attribute>
                    </ns2:profile>
                    <ns2:document>
                        <ns2:id>123</ns2:id>
                        <ns2:type>PASSPORT</ns2:type>
                        <ns2:attribute>
                            <ns2:key>SERIES</ns2:key>
                            <ns2:value>0000</ns2:value>
                        </ns2:attribute>
                        <ns2:attribute>
                            <ns2:key>NUMBER</ns2:key>
                            <ns2:value>000000</ns2:value>
                        </ns2:attribute>
                        <ns2:attribute>
                            <ns2:key>ISSUER</ns2:key>
                            <ns2:value>Issuer name</ns2:value>
                        </ns2:attribute>
                        <ns2:attribute>
                            <ns2:key>ISSUED</ns2:key>
                            <ns2:value>2017-08-15</ns2:value>
                        </ns2:attribute>
                        <ns2:attribute>
                            <ns2:key>DEPARTMENT</ns2:key>
                            <ns2:value>000-000</ns2:value>
                        </ns2:attribute>
                    </ns2:document>
                </ns2:personalInformation>
            </ns2:SimplifiedIdentificationRequest>
        </ns2:AsyncRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request (send request):

```
{
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
          "Password": "PASSWORD"
        }
      }
    },
    "Body": {
      "AsyncRequest": {
        "SimplifiedIdentificationRequest": {
          "unitId": 11111,
          "personalInformation": {
            "profile": {
```

```
              "attribute": [
                {
                  "value": "Last name",
                  "key": "LAST_NAME"
                },
                {
                  "value": "First name",
                  "key": "FIRST_NAME"
                },
                {
                  "value": "Middle initial",
                  "key": "MIDDLE_INITIAL_NAME"
                },
                {
                  "value": "79001234567",
                  "key": "CELL_PHONE"
                },
                {
                  "value": "000-000-000 00",
                  "key": "SNILS"
                },
                {
                  "value": "000000000000",
                  "key": "INN"
                }
              ]
            },
            "document": {
              "id": 123,
              "type": "PASSPORT",
              "attribute": [
                {
                  "value": "0000",
                  "key": "SERIES"
                },
                {
                  "value": "000000",
                  "key": "NUMBER"
                },
                {
                  "value": "Issuer name",
                  "key": "ISSUER"
                },
                {
                  "value": "2017-08-15",
                  "key": "ISSUED"
                },
                {
                  "value": "000-000",
                  "key": "DEPARTMENT"
                }
              ]
            }
          }
        }
      }
    }
  }
}
```


SOAP request (get result by asyncId):

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd" SOAP-ENV:mustUnderstand="1">
            <wsse:UsernameToken
                    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd"
                    wsu:Id="XWSSGID-1503577859855116690135 5"
```

```
                    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-secext-1.0.xsd">
                <wsse:Username>USERNAME</wsse:Username>
                <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-username-token-profile-1.0#PasswordText">PASSWORD</wsse:Password>
            </wsse:UsernameToken>
        </wsse:Security>
    </SOAP-ENV:Header>
    <SOAP-ENV:Body>
        <ns2:AsyncRequest xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd"
 xmlns="">
            <ns2:asyncId>333333</ns2:asyncId>
        </ns2:AsyncRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response (success=true):

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:AsyncResponse xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd">
            <ns2:SimplifiedIdentificationResponse>
                <ns2:success>true</ns2:success>
                <ns2:personalInformation>
                    <ns2:profile>
                        <ns2:attribute>
                            <ns2:key>last_name</ns2:key>
                            <ns2:value>Last name</ns2:value>
                            <ns2:approved>true</ns2:approved>
                            <ns2:published>false</ns2:published>
                        </ns2:attribute>
                        <ns2:attribute>
                            <ns2:key>snils</ns2:key>
                            <ns2:value>000-000-000 00</ns2:value>
                            <ns2:approved>true</ns2:approved>
                            <ns2:published>false</ns2:published>
                        </ns2:attribute>
                        <ns2:attribute>
                            <ns2:key>first_name</ns2:key>
                            <ns2:value>First name</ns2:value>
                            <ns2:approved>true</ns2:approved>
                            <ns2:published>false</ns2:published>
                        </ns2:attribute>
                        <ns2:attribute>
                            <ns2:key>inn</ns2:key>
                            <ns2:value>000000000000</ns2:value>
                            <ns2:approved>true</ns2:approved>
                            <ns2:published>false</ns2:published>
                        </ns2:attribute>
                        <ns2:attribute>
                            <ns2:key>cell_phone</ns2:key>
                            <ns2:value>79001234567</ns2:value>
                            <ns2:approved>true</ns2:approved>
                            <ns2:published>false</ns2:published>
                        </ns2:attribute>
                        <ns2:attribute>
                            <ns2:key>middle_initial_name</ns2:key>
                            <ns2:value>Middle initial</ns2:value>
                            <ns2:approved>true</ns2:approved>
                            <ns2:published>false</ns2:published>
                        </ns2:attribute>
                        <ns2:attribute>
                            <ns2:key>unitid</ns2:key>
                            <ns2:value>11111</ns2:value>
                            <ns2:approved>false</ns2:approved>
                            <ns2:published>false</ns2:published>
                        </ns2:attribute>
                        <ns2:attribute>
                            <ns2:key>profileid</ns2:key>
```

```
                            <ns2:value>22222</ns2:value>
                            <ns2:approved>false</ns2:approved>
                            <ns2:published>false</ns2:published>
                        </ns2:attribute>
                    </ns2:profile>
                    <ns2:document>
                        <ns2:id>123</ns2:id>
                        <ns2:type>PASSPORT</ns2:type>
                        <ns2:attribute>
                            <ns2:key>series</ns2:key>
                            <ns2:value>0000</ns2:value>
                            <ns2:approved>true</ns2:approved>
                            <ns2:published>false</ns2:published>
                        </ns2:attribute>
                        <ns2:attribute>
                            <ns2:key>issuer</ns2:key>
                            <ns2:value>Issuer name</ns2:value>
                            <ns2:approved>false</ns2:approved>
                            <ns2:published>false</ns2:published>
                        </ns2:attribute>
                        <ns2:attribute>
                            <ns2:key>department</ns2:key>
                            <ns2:value>000-000</ns2:value>
                            <ns2:approved>false</ns2:approved>
                            <ns2:published>false</ns2:published>
                        </ns2:attribute>
                        <ns2:attribute>
                            <ns2:key>issued</ns2:key>
                            <ns2:value>2017-08-15</ns2:value>
                            <ns2:approved>false</ns2:approved>
                            <ns2:published>false</ns2:published>
                        </ns2:attribute>
                        <ns2:attribute>
                            <ns2:key>number</ns2:key>
                            <ns2:value>000000</ns2:value>
                            <ns2:approved>true</ns2:approved>
                            <ns2:published>false</ns2:published>
                        </ns2:attribute>
                        <ns2:attribute>
                            <ns2:key>modificationdate</ns2:key>
                            <ns2:value>2017-08-08T13:53:22.000+03:00</ns2:value>
                            <ns2:approved>true</ns2:approved>
                        </ns2:attribute>
                        <ns2:hasAttachedFiles>false</ns2:hasAttachedFiles>
                    </ns2:document>
                </ns2:personalInformation>
            </ns2:SimplifiedIdentificationResponse>
        </ns2:AsyncResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP response (success=false):

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header/>
    <SOAP-ENV:Body>
        <ns2:AsyncResponse xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd">
            <ns2:SimplifiedIdentificationResponse>
                <ns2:success>false</ns2:success>
                <ns2:error>Customer data did not pass ESIA check.</ns2:error>
                <ns2:errorCode>500.7.3</ns2:errorCode>
                <ns2:personalInformation>
                    <ns2:profile>
                        <ns2:attribute>
                            <ns2:key>last_name</ns2:key>
                            <ns2:value>Last name</ns2:value>
                            <ns2:approved>false</ns2:approved>
                            <ns2:published>false</ns2:published>
                        </ns2:attribute>
                        <ns2:attribute>
```

```
                        <ns2:key>snils</ns2:key>
                        <ns2:value>000-000-000 00</ns2:value>
                        <ns2:approved>false</ns2:approved>
                        <ns2:published>false</ns2:published>
                    </ns2:attribute>
                    <ns2:attribute>
                        <ns2:key>first_name</ns2:key>
                        <ns2:value>First name</ns2:value>
                        <ns2:approved>false</ns2:approved>
                        <ns2:published>false</ns2:published>
                    </ns2:attribute>
                    <ns2:attribute>
                        <ns2:key>inn</ns2:key>
                        <ns2:value>000000000000</ns2:value>
                        <ns2:approved>false</ns2:approved>
                        <ns2:published>false</ns2:published>
                    </ns2:attribute>
                    <ns2:attribute>
                        <ns2:key>cell_phone</ns2:key>
                        <ns2:value>79001234567</ns2:value>
                        <ns2:approved>true</ns2:approved>
                        <ns2:published>false</ns2:published>
                    </ns2:attribute>
                    <ns2:attribute>
                        <ns2:key>middle_initial_name</ns2:key>
                        <ns2:value>Middle initial</ns2:value>
                        <ns2:approved>false</ns2:approved>
                        <ns2:published>false</ns2:published>
                    </ns2:attribute>
                    <ns2:attribute>
                        <ns2:key>unitid</ns2:key>
                        <ns2:value>11111</ns2:value>
                        <ns2:approved>false</ns2:approved>
                        <ns2:published>false</ns2:published>
                    </ns2:attribute>
                    <ns2:attribute>
                        <ns2:key>profileid</ns2:key>
                        <ns2:value>22222</ns2:value>
                        <ns2:approved>false</ns2:approved>
                        <ns2:published>false</ns2:published>
                    </ns2:attribute>
                </ns2:profile>
                <ns2:document>
                    <ns2:id>123</ns2:id>
                    <ns2:type>PASSPORT</ns2:type>
                    <ns2:attribute>
                        <ns2:key>series</ns2:key>
                        <ns2:value>0000</ns2:value>
                        <ns2:approved>false</ns2:approved>
                        <ns2:published>false</ns2:published>
                    </ns2:attribute>
                    <ns2:attribute>
                        <ns2:key>issuer</ns2:key>
                        <ns2:value>Issuer name</ns2:value>
                        <ns2:approved>false</ns2:approved>
                        <ns2:published>false</ns2:published>
                    </ns2:attribute>
                    <ns2:attribute>
                        <ns2:key>department</ns2:key>
                        <ns2:value>000-000</ns2:value>
                        <ns2:approved>false</ns2:approved>
                        <ns2:published>false</ns2:published>
                    </ns2:attribute>
                    <ns2:attribute>
                        <ns2:key>issued</ns2:key>
                        <ns2:value>22017-08-15</ns2:value>
                        <ns2:approved>false</ns2:approved>
                        <ns2:published>false</ns2:published>
                    </ns2:attribute>
                    <ns2:attribute>
                        <ns2:key>number</ns2:key>
```

```
                            <ns2:value>000000</ns2:value>
                            <ns2:approved>false</ns2:approved>
                            <ns2:published>false</ns2:published>
                        </ns2:attribute>
                        <ns2:attribute>
                            <ns2:key>modificationdate</ns2:key>
                            <ns2:value>2017-08-17T10:23:15.000+03:00</ns2:value>
                            <ns2:approved>true</ns2:approved>
                        </ns2:attribute>
                        <ns2:hasAttachedFiles>false</ns2:hasAttachedFiles>
                    </ns2:document>
                </ns2:personalInformation>
            </ns2:SimplifiedIdentificationResponse>
        </ns2:AsyncResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

JSON request (get result by asyncId):

```
{
  "Envelope": {
    "Header": {
      "Security": {
        "UsernameToken": {
          "Username": "USERNAME",
          "Password": "PASSWORD"
        }
      }
    },
    "Body": {
      "AsyncRequest": {
        "asyncId": 333333
      }
    }
  }
}
```

JSON response (success=true):

```
{
  "Envelope": {
    "Body": {
      "AsyncResponse": {
        "SimplifiedIdentificationResponse": {
          "success": true,
          "personalInformation": {
            "profile": {
              "attribute": [
                {
                  "approved": true,
                  "value": "Last name",
                  "published": false,
                  "key": "last_name"
                },
                {
                  "approved": true,
                  "value": "000-000-000 00",
                  "published": false,
                  "key": "snils"
                },
                {
                  "approved": true,
                  "value": "First name",
                  "published": false,
                  "key": "first_name"
                },
                {
                  "approved": true,
                  "value": "000000000000",
                  "published": false,
```

```
            "key": "inn"
          },
          {
            "approved": true,
            "value": "79001234567",
            "published": false,
            "key": "cell_phone"
          },
          {
            "approved": true,
            "value": "Middle initial",
            "published": false,
            "key": "middle_initial_name"
          },
          {
            "approved": false,
            "value": "11111",
            "published": false,
            "key": "unitid"
          },
          {
            "approved": false,
            "value": "22222",
            "published": false,
            "key": "profileid"
          }
        ]
      },
      "document": {
        "id": 123,
        "type": "PASSPORT",
        "hasAttachedFiles": false,
        "attribute": [
          {
            "approved": true,
            "value": "0000",
            "published": false,
            "key": "series"
          },
          {
            "approved": false,
            "value": "Issuer name",
            "published": false,
            "key": "issuer"
          },
          {
            "approved": false,
            "value": "000-000",
            "published": false,
            "key": "department"
          },
          {
            "approved": false,
            "value": "2017-08-15",
            "published": false,
            "key": "issued"
          },
          {
            "approved": true,
            "value": "000000",
            "published": false,
            "key": "number"
          },
          {
            "approved": true,
            "value": "2017-08-08T13:53:22.000+03:00",
            "key": "modificationdate"
          }
        ]
      }
    }
```

```
        }
      }
    }
  }
}
```

JSON response (success=false):

```
{
  "Envelope": {
    "Body": {
      "AsyncResponse": {
        "SimplifiedIdentificationResponse": {
          "success": false,
          "error": "Customer data did not pass ESIA check.",
          "errorCode": "500.7.3",
          "personalInformation": {
            "document": {
              "id": 123,
              "hasAttachedFiles": false,
              "attribute": [
                {
                  "approved": false,
                  "value": "0000",
                  "published": false,
                  "key": "series"
                },
                {
                  "approved": false,
                  "value": "Issuer name",
                  "published": false,
                  "key": "issuer"
                },
                {
                  "approved": false,
                  "value": "000-000",
                  "published": false,
                  "key": "department"
                },
                {
                  "approved": false,
                  "value": "2017-08-15",
                  "published": false,
                  "key": "issued"
                },
                {
                  "approved": false,
                  "value": "000000",
                  "published": false,
                  "key": "number"
                },
                {
                  "approved": true,
                  "value": "2017-08-17T10:23:15.000+03:00",
                  "key": "modificationdate"
                }
              ],
              "type": "PASSPORT"
            },
            "profile": {
              "attribute": [
                {
                  "approved": false,
                  "value": "Last name",
                  "published": false,
                  "key": "last_name"
                },
                {
                  "approved": false,
                  "value": "000-000-000 00",
```

```
                  "published": false,
                  "key": "snils"
                },
                {
                  "approved": false,
                  "value": "First name",
                  "published": false,
                  "key": "first_name"
                },
                {
                  "approved": false,
                  "value": "000000000000",
                  "published": false,
                  "key": "inn"
                },
                {
                  "approved": true,
                  "value": "79001234567",
                  "published": false,
                  "key": "cell_phone"
                },
                {
                  "approved": false,
                  "value": "Middle initial",
                  "published": false,
                  "key": "middle_initial_name"
                },
                {
                  "approved": false,
                  "value": "11111",
                  "published": false,
                  "key": "unitid"
                },
                {
                  "approved": false,
                  "value": "22222",
                  "published": false,
                  "key": "profileid"
                }
              ]
            }
          }
        }
      }
    }
  }
}
```

Chapter 7

Sample Code

Topics:

Java API for XML Web Services (JAX-WS) Example

The following Java example uses JAX-WS portable artifacts to get the account ID by using the account number of 12345678. Then, the account ID is used to get the account balance.

JAX-WS Portable Artifacts

Run the following command to generate JAX-WS portable articfacts based on the Merchant API WSDL file:

```
wsimport -clientjar moneta.jar https://service.moneta.ru:8443/services/x509.wsdl
```

This command generates the moneta.jar file with JAX-WS artifacts that you can use in your Java applications.

Java Code

After you generate the artifacts, you can create a Java class that uses these artifacts to create SOAP messages, call the Merchant API, and process API responses.

1. Ensure that the generated moneta.jar file is in the class path of your application.
2. Import the artifacts from the moneta.jar file.

```
import ru.moneta.schemas.messages.*;
import ru.moneta.services.*;
```

3. Create an instance of the FindAccountByIdRequest class. This class represents the FindAccountById SOAP request. Use this instance to create a request with the account number of 12345678.

```
FindAccountByIdRequest msg = new FindAccountByIdRequest();
msg.setValue(12345678L);
```

4. Create an instance of the generated MessagesService class to send requests to the Merchant API.

```
MessagesService service = new MessagesService();
```

5. Invoke the getMessagesSoap11 method on the service to get the proxy, or a port, to the service. This port contains all of the methods from the WSDL file.

```
Messages port = service.getMessagesSoap11();
```

6. Invoke the findAccountById method of the port with the SOAP message as an argument. Then pass the result of the call to the getBalance method.

```
FindAccountByIdResponse result = port.findAccountById(msg);
Number balance = result.getAccount().getBalance();
```

The following code shows the full source of this Java example:

```
package ru.moneta.soap.wsimport;

import ru.moneta.schemas.messages.*;
import ru.moneta.services.*;


public class Main {

    public static void main(String[] args) {

        FindAccountByIdRequest msg = new FindAccountByIdRequest();
        msg.setValue(12345678L);

        MessagesService service = new MessagesService();
        Messages port = service.getMessagesSoap11();
        FindAccountByIdResponse result = port.findAccountById(msg);
        Number balance = result.getAccount().getBalance();
        System.out.print(balance);
    }
}
```

SSL Certificate Authentication

To authenticate your SOAP requests to the Merchant API, you can use an SSL certificate that you can obtain from MONETA.RU.

1. Obtain an SSL certificate from MONETA.RU and add the certificate to a PKCS12 keystore. For more information, see *Obtaining a Certificate* on page 11.
2. Specify the keystore in the Java Virtual Machine (JVM) arguments. Use the following arguments:

```
-Djavax.net.ssl.keyStore=keystore.p12 -Djavax.net.ssl.keyStoreType=pkcs12 -
Djavax.net.ssl.keyStorePassword=keystore_password
```

SOAP with Attachments for Java (SAAJ) Example

1. Import necessary classes.

```
import javax.xml.namespace.QName;
import javax.xml.soap.*;

import javax.xml.transform.TransformerFactory;
import javax.xml.transform.Transformer;
import javax.xml.transform.Source;

import javax.xml.transform.stream.StreamResult;
```

2. Create an instance of the SOAPConnection class that you can use to send and receive SOAP messages.

```
SOAPConnectionFactory soapConnFactory =
    SOAPConnectionFactory.newInstance();
SOAPConnection connection =
    soapConnFactory.createConnection();
```

3. Create a SOAP request.

a. Create an instance of the SOAPMessage class that represents a SOAP request.

```
MessageFactory messageFactory =
    MessageFactory.newInstance();
SOAPMessage message =
    messageFactory.createMessage();
```

b. Create the envelope, body, and header objects for the message.

```
SOAPPart soapPart =
    message.getSOAPPart();
SOAPEnvelope envelope =
    soapPart.getEnvelope();
SOAPBody body =
    envelope.getBody();
SOAPHeader header =
    envelope.getHeader();
```

c. To authenticate your SOAP request to the Merchant API, this example uses a security header that includes a *username* and *password*,

```
SOAPElement security =
    header.addChildElement("Security", "wsse",
    "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd");

SOAPElement usernameToken =
    security.addChildElement("UsernameToken", "wsse");
usernameToken.addAttribute(new QName("xmlns:wsu"),
    "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd");

SOAPElement username =
    usernameToken.addChildElement("Username", "wsse");
username.addTextNode("username");

SOAPElement password =
    usernameToken.addChildElement("Password", "wsse");
password.setAttribute("Type",
    "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-
profile-1.0#PasswordText");
            password.addTextNode("password");
```

d. Create the body of the message. The following example, creates the FindProfileInfoByAccountRequest xml request for the account number of 12345678:

```
SOAPElement bodyElement =

 body.addChildElement(envelope.createName("FindProfileInfoByAccountIdRequest" ,
        "mes",
        "http://www.moneta.ru/schemas/messages.xsd"));

bodyElement.addTextNode("12345678");

message.saveChanges();
```

The following listing shows the generated XML request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-secext-1.0.xsd">
      <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-utility-1.0.xsd">
 <wsse:Username>user_name</wsse:Username>
 <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
username-token-profile-1.0#PasswordText">password</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
```

```
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <mes:FindProfileInfoByAccountIdRequest xmlns:mes="http://www.moneta.ru/
schemas/messages.xsd">12345678</mes:FindProfileInfoByAccountIdRequest>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

4. Use the connection object to send the SOAP request and receive the response.

```
String destination =
    "https://demo.moneta.ru/services";
SOAPMessage reply = connection.call(message, destination);
```

> **Note:** The sample code above connects to the sandbox Merchant API. For production code, use the following address:
>
> ```
> https://service.moneta.ru/services
> ```

5. Process the response.

   - Create a new instance of the Tranformer class and use it to print the XML response:

     ```
     TransformerFactory transformerFactory =
         TransformerFactory.newInstance();
     Transformer transformer =
         transformerFactory.newTransformer();

     Source sourceContent =
         reply.getSOAPPart().getContent();

     StreamResult result =
         new StreamResult(System.out);
     transformer.transform(sourceContent, result);
     System.out.println();
     ```

     The following listing shows the generated XML response:

     ```
     <?xml version="1.0" encoding="UTF-8"?>
     <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
       <SOAP-ENV:Header/>
       <SOAP-ENV:Body>
         <ns2:FindProfileInfoByAccountIdResponse
                 xmlns:ns2="http://www.moneta.ru/schemas/messages.xsd">
           <ns2:accountId>12345678</ns2:accountId>
           <ns2:currency>RUB</ns2:currency>
           <ns2:profile>
          <ns2:attribute>
            <ns2:key>unitid</ns2:key>
            <ns2:value>55555</ns2:value>
          </ns2:attribute>
           </ns2:profile>
         </ns2:FindProfileInfoByAccountIdResponse>
       </SOAP-ENV:Body>
     </SOAP-ENV:Envelope>
     ```

   - Extract data from the specified tag in the XML response.

     ```
     String accountId =

      reply.getSOAPBody().getElementsByTagName("ns2:accountId").item(0).getFirstChild().getNodeValu
     System.out.println("Account ID: "+accountId);
     ```

6. Close the connection.

```
connection.close();
```

The following code shows the full source of this Java example:

```
package ru.moneta;
```

```
import javax.xml.namespace.QName;
import javax.xml.soap.*;

import javax.xml.transform.TransformerFactory;
import javax.xml.transform.Transformer;
import javax.xml.transform.Source;

import javax.xml.transform.stream.StreamResult;

public class Main {

    public static void main(String args[]) {
        try {
            SOAPConnectionFactory soapConnFactory =
 SOAPConnectionFactory.newInstance();
            SOAPConnection connection = soapConnFactory.createConnection();

            MessageFactory messageFactory = MessageFactory.newInstance();
            SOAPMessage message = messageFactory.createMessage();

            SOAPPart soapPart = message.getSOAPPart();
            SOAPEnvelope envelope = soapPart.getEnvelope();
            SOAPBody body = envelope.getBody();
            SOAPHeader header = envelope.getHeader();


            SOAPElement security = header.addChildElement("Security", "wsse", "http://
docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd");

            SOAPElement usernameToken = security.addChildElement("UsernameToken",
 "wsse");
            usernameToken.addAttribute(new QName("xmlns:wsu"),
                    "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd");

            SOAPElement username = usernameToken.addChildElement("Username", "wsse");
            username.addTextNode("user_name");

            SOAPElement password = usernameToken.addChildElement("Password", "wsse");
            password.setAttribute("Type",
                    "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-
token-profile-1.0#PasswordText");
            password.addTextNode("password");


            SOAPElement bodyElement =

 body.addChildElement(envelope.createName("FindProfileInfoByAccountIdRequest" ,
                            "mes",
                            "http://www.moneta.ru/schemas/messages.xsd"));

            bodyElement.addTextNode("12345678");

            message.saveChanges();


            String destination = "https://demo.moneta.ru/services";
            SOAPMessage reply = connection.call(message, destination);

            String accountId = reply.getSOAPBody().
                    getElementsByTagName("ns2:accountId").item(0)
                    .getFirstChild().getNodeValue();
            System.out.println("Account ID: "+accountId);

            connection.close();
        } catch(Exception e) {
            System.out.println(e.getMessage());
        }
    }
```

```
}
```

SOAP C# example

WSE3

WSE3 has to be installed and referenced in project.

Microsoft.Web.Services3 has to be visible and accessible.

Client stubs generation via wsdl fix

```
1) public partial class FindProfileInfoResponse : object,
 System.ComponentModel.INotifyPropertyChanged {

        private long pageSizeField;

        private long pageNumberField;

        private long pagesCountField;

        private long sizeField;

        private long totalSizeField;

        private KeyValueApprovedAttribute[] profileField;



2)      [System.Xml.Serialization.XmlArrayAttribute(Order=5)]
        [System.Xml.Serialization.XmlArrayItemAttribute("attribute",
 typeof(KeyValueApprovedAttribute), IsNullable=false)]
        public KeyValueApprovedAttribute[] profile {
            get {
                return this.profileField;
            }
            set {
                this.profileField = value;
                this.RaisePropertyChanged("profile");
            }
        }
```

SSL connection and HTTP version

```
protected override WebRequest GetWebRequest(Uri uri)
{
  HttpWebRequest request = (HttpWebRequest) base.GetWebRequest(uri);
  request.KeepAlive = false;
  request.ProtocolVersion = new Version(1, 0); //this was the big deal...
  return request;
}


ServicePointManager.ServerCertificateValidationCallback = new
 RemoteCertificateValidationCallback(
    delegate(object sender2, X509Certificate certificate, X509Chain chain,
 SslPolicyErrors sslPolicyErrors)
        {
            return true;
    }
);
```

Call example

```
class Program {
        public static bool ValidateServerCertificate(
              object sender,
              X509Certificate certificate,
              X509Chain chain,
              SslPolicyErrors sslPolicyErrors)
        {
            // Do not allow this client to communicate with unauthenticated servers.
            return true;
        }
        static void doWork() {
// installing custom server certificate validator
            ServicePointManager.ServerCertificateValidationCallback = new
 RemoteCertificateValidationCallback(ValidateServerCertificate);
            ru.moneta.www.MessagesServiceWse ms = new
 ru.moneta.www.MessagesServiceWse();
// using client certificate for authentication
//            X509Certificate cert = new X509Certificate("c:\\your-client-
certificate.p12", "");
//            ms.ClientCertificates.Add(cert);
// using login and password for authentication
            ms.RequestSoapContext.Security.Tokens.Add(new UsernameToken(V_LOGIN,
 V_PASSWORD, PasswordOption.SendPlainText));
            ru.moneta.www.FindProfileInfoByAccountIdResponse o =
 ms.FindProfileInfoByAccountId(12345678);
            System.Console.Out.WriteLine(o.currency);
        }
        static void Main(string[] args) {
            doWork();
        }
    }
```

SOAP PHP Example

The following PHP example uses the MonetaWebService library to get the account balance.

1. Import the MonetaWebService library.

```
require_once 'MonetaWebService.php';
```

2. Create an instance of the MonetaWebService class. This class requires three arguments: the URL of the WSDL file, a user name, and a password for your merchant account.

```
$service = new MonetaWebService("https://demo.moneta.ru/services.wsdl", "user_name",
 "password");
```

> **Note:** The sample code above connects to the sandbox Merchant API. For production code, use the following address:
>
> ```
> https://service.moneta.ru/services.wsdl
> ```

3. Call the FindAccountById method of the service instance. This method sends the FindAccountByIdRequest xml request to the Merchant API and returns a response.

```
$response = $service->FindAccountById(account_ID);
```

4. Process the response object.

```
echo "balance: " . $response->account->availableBalance;
echo "currency: " . $response->account->currency;
```

The following code listing shows the full source of this PHP example:

```
<?php
```

```
require_once 'MonetaWebService.php';

$service = new MonetaWebService("https://demo.moneta.ru/services.wsdl",
 "merchant_email", "merchant_password");

try
{
 $response = $service->FindAccountById(12345678);
 echo "balance: " . $response->account->availableBalance;
 echo "currency: " . $response->account->currency;
}
catch (Exception $e)
{
 echo $e->getMessage();
}

?>
```

SOAP Python Example

The following Python example uses the Suds library to get information about the profile of your merchant account.

1. Import necessary Suds classes.

```
from suds.client import Client
from suds.wsse import *
```

2. Optionally, configure the suds library to log SOAP requests and responses.

```
import logging
logging.basicConfig(level=logging.INFO)
logging.getLogger('suds.client').setLevel(logging.DEBUG)
```

3. Create an instance of the Suds.Client class. Use this instance to send SOAP requests to the Merchant API and receive responses.

```
wsdl='https://demo.moneta.ru/services.wsdl'
client = Client(wsdl,cache=None)
```

> **Note:** The sample code above connects to the sandbox Merchant API. For production code, use the following address:
>
> ```
> https://service.moneta.ru/services.wsdl
> ```

4. To authenticate your SOAP request to the Merchant API, this example uses a security header that includes a *username* and a *password*.

```
security = Security()
token = UsernameToken('username', 'password')
security.tokens.append(token)
client.set_options(wsse=security)
```

> **Tip:** You can now use the Suds Client instance to print the list of available methods:
>
> ```
> print client
> ```
>
> The following fragment of the output shows the information about the GetProfileInfo method that requires the unitId value as an argument:
>
> ```
> GetProfileInfo(xs:long unitId)
> ```

5. Use the GetProfileInfo method of the client instance to send the request to the Merchant API and get a response.

```
response = client.service['MessagesSoap11']['GetProfileInfo'](unitId=40488)
```

6. Parse the response. The following code prints the list of key-value pairs from the response:

```
for attr in response:
    print attr['key']+": "+attr['value']
```

The following code listing shows the full source of this Python example:

```
#!/usr/bin/python2.7

from suds.client import Client
from suds.wsse import *

import logging
logging.basicConfig(level=logging.INFO)
logging.getLogger('suds.client').setLevel(logging.DEBUG)

wsdl='https://demo.moneta.ru/services.wsdl'
client = Client(wsdl,cache=None)

security = Security()
token = UsernameToken('username', 'password')
security.tokens.append(token)
client.set_options(wsse=security)

response = client.service['MessagesSoap11']['GetProfileInfo'](unitId=40488)

for attr in response:
    print attr['key']+": "+attr['value']
```