

Web Services Admin API Specification

RemitONE Money Transfer Management Solution

Version 2.6
April 2020

Table of Contents

Introduction.....	4
Connection Parameters.....	4
Response Format.....	5
Web Service Method Details.....	6
Admin API.....	6
getSourceCountries.....	6
getDestinationCountries.....	6
getSellRates.....	7
getAgents.....	8
getAgentRates.....	9
getAgentCharges.....	9
getCountryCharges.....	11
getTransactionStatus.....	12
getTransactionStatusByAgentTransRef.....	13
getTransactionDetails.....	14
getDeliveryBanks.....	17
getDeliveryBankBranches.....	18
createDeliveryBankBranch.....	18
updateDeliveryBankBranch.....	19
getCollectionPoints.....	21
createCollectionPoint.....	22
updateCollectionPoint.....	23
getTransactions.....	24
assignTransactions.....	26
getErrorTransactions.....	26
updateErrorTransaction.....	27
processCashCollectionTransactionByBank.....	30
requestCancellation.....	31
getMobileNetworkOperators.....	32
createMobileNetworkOperators.....	32
updateMobileNetworkOperators.....	33
createMobileNetworkCreditType.....	34
updateMobileNetworkCreditType.....	35
getMobileNetworkCreditTypes.....	36
uploadCSVData.....	38
updateSellRates.....	39
updateConversionRates.....	40
addWalletCredit.....	41
errorTransaction.....	42
createRemitter.....	42
searchRemitter.....	45
updateRemitter.....	47
uploadRemitterKYCVideo.....	51
getRemitterKYCVideo.....	51
verifyRemitter.....	52

createBeneficiary.....	52
searchBeneficiary.....	55
updateBeneficiary.....	56
getPayoutTransactions.....	59
getPayoutTransactionDetails.....	62
acceptPayoutTransactions.....	65
getPendingPayoutTransactions.....	66
processPayoutTransaction.....	69
errorPayoutTransaction.....	70
getRemitterWhitelist.....	71
setRemitterWhitelist.....	71
getBeneficiaryWhitelist.....	72
setBeneficiaryWhitelist.....	72
transactionPaymentCleared.....	73
createPromotionRule.....	74
nameScreenCheck.....	76
updateBuyRates.....	77
createDeliveryBank.....	78
updateDeliveryBank.....	79
getDeliveryBanks.....	80
createUtilityCompany.....	82
updateUtilityCompany.....	84
getUtilityCompanies.....	86
updateTransactionDetails.....	87
checkCredentials.....	88
Web Services Functionality.....	90
Appendix A : Destination Country ISO Codes.....	91
Appendix B : Remitter ID Type Codes.....	98
Appendix C : Beneficiary ID Type Codes.....	98
Appendix D : Payment Method Codes.....	98
Appendix E : Service Level (Delivery Speed) Codes.....	99
Appendix F : Source of Income Codes.....	99
Appendix G : Remittance Purpose Codes.....	99
Appendix H : Transaction Statuses.....	100
Appendix I : Transfer Type*.....	101
Appendix J : Creation Channel.....	102

Introduction

The RemitONE WebServices Infrastructure uses the REST protocol. Our services use the POST method for invoking the service, and hence all data must be provided as POST variables. The response is provided as structured XML.

This document outlines the input and output parameters for each of the RemitONE WebServices.

Connection Parameters

The BASE_URL is the URL at which the WebServices are hosted. This BASE_URL is then followed by the required GROUP name, and then the required METHOD name.

For example,
BASE_URL = <https://www.remitone.com/ws/>

GROUP = transaction
METHOD = createTransaction

This would give a complete URL of :
<https://www.remitone.com/ws/transaction/createTransaction>

The client application would need to make a POST to this URL to invoke this particular WebService. *In the specification below, a * indicates a required input parameter.*

Every invocation must contain the following parameters (as POST variables) :

Name	Type	Notes
username *	Text	
password *	Text	
pin *	Text	

These will be provided to you when your WebService Account is activated.

All other parameters will depend on the particular WebService being invoked.

Response Format

A Successful Response will always contain the following XML structure :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>SUCCESS</status>
  <result>
    [Specific response for the particular Web Service]
  </result>
</response>
```

An Error Response is as follows :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>FAIL</status>
  <result>
    <message>[Error message]</message>
  </result>
</response>
```

An Error Response with Validation errors is as follows :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>FAIL</status>
  <result>
    <message>VALIDATION FAILED</message>
    <errors>
      <error>[Validation error message]</error>
      <error>[Validation error message]</error>
      <error>[Validation error message]</error>
    </errors>
  </result>
</response>
```

Web Service Method Details

Admin API

getSourceCountries

Group : **admin**

Method : **getSourceCountries**

This method provides a list of available source countries on the system.

Input fields :

none (apart from username, password and pin which are required in every invocation).

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>SUCCESS</status>
  <result>
    <countries>
      <country>
        <id>001</id>
        <name>United Kingdom</name>
        <iso_code>UK</iso_code>
        <currency>GBP</currency>
        <allowed_currencies>
          <currency>GBP</currency>
        </allowed_currencies>
      </country>
      <country>
        <id>002</id>
        <name>United States</name>
        <iso_code>US</iso_code>
        <currency>USD</currency>
        <allowed_currencies>
          <currency>USD</currency>
        </allowed_currencies>
      </country>
    </countries>
  </result>
</response>
```

getDestinationCountries

Group : **country**

Method : **getDestinationCountries**

This method provides a list of available destination countries on the system.

Input fields :

none (apart from username, password and pin which are required in every invocation).

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>SUCCESS</status>
  <result>
    <countries>
      <country>
        <id>998</id>
        <name>Bangladesh</name>
        <iso_code>BD</iso_code>
        <currency>BDT</currency>
        <allowed_currencies>
          <currency>BDT</currency>
        </allowed_currencies>
      </country>
      <country>
        <id>999</id>
        <name>Philippines</name>
        <iso_code>PH</iso_code>
        <currency>PHP</currency>
        <allowed_currencies>
          <currency>PHP</currency>
        </allowed_currencies>
      </country>
    </countries>
  </result>
</response>
```

getSellRates

Group : **admin**

Method : **getSellRates**

This method provides current rates from the system for the source and destination countries provided.

Input fields:

Name	Type	Notes
source_country *	Text	The name of the Source Country, as per the output of getSourceCountries .
destination_country *	Text	The name of the Destination Country, as per the output of getDestinationCountries .
source_currency	Text	The 3 letters ISO currency code, as listed in Appendix A
destination_currency	Text	The 3 letters ISO currency code,

		as listed in Appendix A
delivery_bank	Text	The name of the delivery bank, as per the output of getDeliveryBanks .

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>SUCCESS</status>
  <result>
    <rate>
      <account>39.5000</account>
      <cash_collection>39.0000</cash_collection>
      <card>39.5000</card>
      <home_delivery>38.0000</home_delivery>
      <utility_bill>39.6000</utility_bill>
      <mobile_transfer>36.0000</mobile_transfer>
    </rate>
  </result>
</response>
```

getAgents

Group : admin

Method : **getAgents**

This method provides a list of available agents on the system.

Input fields :

none (apart from username, password and pin which are required in every invocation).

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>4</responseId>
  <status>SUCCESS</status>
  <result>
    <agents>
      <agent>
        <name>Test Agent</name>
        <username>testagent</username>
        <code>2000</code>
        <country>Singapore</country>
      </agent>
      <agent>
        <name>PRIME EXCHANGE CO PTE LTD</name>
        <username>primex</username>
        <code>2007</code>
        <country>Singapore</country>
      </agent>
    </agents>
  </result>
</response>
```



```
</result>
</response>
```

getAgentRates

Group : admin

Method : **getAgentRates**

This method provides the current rates for an agent on the system.

Input fields:

Name	Type	Notes
agent_code *	Code	The Agent code, as per the output of getAgents .
dest_country *	Text	The name of the Destination Country, as per the output of getDestinationCountries .
source_currency	Text	The 3 letters ISO currency code, as listed in Appendix A
destination_currency	Text	The 3 letters ISO currency code, as listed in Appendix A
delivery_bank	Text	The name of the delivery bank, as per the output of getDeliveryBanks .

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>SUCCESS</status>
  <result>
    <rate>
      <account>39.5000</account>
      <cash_collection>39.0000</cash_collection>
      <card>39.5000</card>
      <home_delivery>38.0000</home_delivery>
      <utility_bill>39.6000</utility_bill>
      <mobile_transfer>36.0000</mobile_transfer>
    </rate>
  </result>
</response>
```

getAgentCharges

Group: admin

Method: **getAgentCharges**

Input fields:

Name	Type	Notes
agent_code *	Code	The Agent code, as per the output of getAgents .
destination_country *	Text	Name of country retrieved from calling getDestinationCountries
source_currency	Text	The 3 letters ISO currency code, as listed in Appendix A
destination_currency	Text	The 3 letters ISO currency code, as listed in Appendix A
trans_type *	Text	Account / Cash Collection / Card Transfer / Home Delivery / Mobile Transfer
payment_method *	Code	See Appendix D for codes.
service_level *	Code	See Appendix E for codes.
delivery_bank	Text	The name of the delivery bank, as per the output of getDeliveryBanks .
sms_confirmation *	Boolean	t / f
sms_notification *	Boolean	t / f
amount_type *		Whether the amount is specified in the source currency or the destination currency. Possible values : SOURCE / DESTINATION.
amount_to_send *	Numeric	

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>SUCCESS</status>
  <result>
    <source_currency>GBP</source_currency>
    <source_amount>100</source_amount>
    <rate>37.53</rate>
    <destination_currency>PHP</destination_currency>
    <destination_amount>3753.00</destination_amount>
    <commission>10.50</commission>
    <agent_fee>0.00</agent_fee>
    <hq_fee>0.50</hq_fee>
    <total_charges>11.00</total_charges>
    <tax>3.40</tax>
  </result>
</response>
```

getCountryCharges

Group: admin

Method: getCountryCharges

Input fields:

Name	Type	Notes
source_country *	Text	Name of country retrieved from calling getSourceCountries
destination_country *	Text	Name of country retrieved from calling getDestinationCountries
source_currency	Text	The 3 letters ISO currency code, as listed in Appendix A
destination_currency	Text	The 3 letters ISO currency code, as listed in Appendix A
trans_type *	Text	Account / Cash Collection / Card Transfer / Home Delivery / Mobile Transfer
payment_method *	Code	See Appendix D for codes.
service_level *	Code	See Appendix E for codes.
delivery_bank	Text	The name of the delivery bank, as per the output of getDeliveryBanks .
sms_confirmation *	Boolean	t / f
sms_notification *	Boolean	t / f
amount_type *		Whether the amount is specified in the source currency or the destination currency. Possible values : SOURCE / DESTINATION.
amount_to_send *	Numeric	

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>SUCCESS</status>
  <result>
    <source_currency>GBP</source_currency>
    <source_amount>100</source_amount>
    <rate>37.53</rate>
    <destination_currency>PHP</destination_currency>
    <destination_amount>3753.00</destination_amount>
```

```

    <commission>10.50</commission>
    <agent_fee>0.00</agent_fee>
    <hq_fee>0.50</hq_fee>
    <total_charges>11.00</total_charges>
    <tax>3.40</tax>
  </result>
</response>

```

getTransactionStatus

Group: transaction

Method: getTransactionStatus

Input fields:

Name	Type	Notes
trans_ref *	Text Array	If Text, it is searching for the single transaction status, returning all the information about the requested transaction. If Array, it will execute the search for all the transactions specified.

Example Output XML :

```

<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>SUCCESS</status>
  <result>
    <trans_ref>RA0003101</trans_ref>
    <agent_trans_ref>ABC12345678</agent_trans_ref>
    <benef_trans_ref>ABC12345678</benef_trans_ref>
    <status>SENT_FOR_PAY</status>
    <compliance_check_required>f</compliance_check_required>
    <compliance_checked>f</compliance_checked>
  </result>
</response>

```

```

<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>468</responseId>
  <status>SUCCESS</status>
  <result>
    <trans_refs>

```

```

<trans_ref>
  <trans_ref>RC000247000101</trans_ref>
  <agent_trans_ref></agent_trans_ref>
  <benef_trans_ref>RC000247000101</benef_trans_ref>
  <status>SENT_FOR_PAY</status>
  <compliance_check_required>t</compliance_check_required>
  <compliance_checked>t</compliance_checked>
  <ext_compliance_check_required>f</ext_compliance_check_required>
  <ext_compliance_checked>f</ext_compliance_checked>
</trans_ref>
<trans_ref>
  <trans_ref>RC000224000100</trans_ref>
  <agent_trans_ref></agent_trans_ref>
  <benef_trans_ref>RC000224000112</benef_trans_ref>
  <status>SENT_FOR_DELIVERY</status>
  <compliance_check_required>f</compliance_check_required>
  <compliance_checked>f</compliance_checked>
  <ext_compliance_check_required>f</ext_compliance_check_required>
  <ext_compliance_checked>f</ext_compliance_checked>
</trans_ref>
<trans_ref>
  <trans_ref>RB000270000102</trans_ref>
  <agent_trans_ref></agent_trans_ref>
  <benef_trans_ref>RB000270000102</benef_trans_ref>
  <status>PROCESSED</status>
  <compliance_check_required>f</compliance_check_required>
  <compliance_checked>f</compliance_checked>
  <ext_compliance_check_required>f</ext_compliance_check_required>
  <ext_compliance_checked>f</ext_compliance_checked>
</trans_ref>
</trans_refs>
</result>
</response>

```

getTransactionStatusByAgentTransRef

Group: transaction

Method: getTransactionStatusByAgentTransRef

Input fields:

Name	Type	Notes
trans_ref *	Text Array	Agent's unique transaction reference (if provided at the time of creation). See getTransactionStatus above to request the transaction status using the RemitONE's transaction reference. If Array, the same search will be processed for all the given transactions references

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>SUCCESS</status>
  <result>
    <trans_ref>RA0003101</trans_ref>
    <agent_trans_ref>ABC12345678</agent_trans_ref>
    <benef_trans_ref>ABC12345678</benef_trans_ref>
    <status>SENT_FOR_PAY</status>
    <compliance_check_required>f</compliance_check_required>
    <compliance_checked>f</compliance_checked>
  </result>
</response>
```

getTransactionDetails

Group: transaction

Method: getTransactionDetails

Get details of a single transaction.

Input fields:

Name	Type	Notes
trans_ref *	Text	

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>SUCCESS</status>
  <result>
    <transaction>
      <trans_ref>RA20079127</trans_ref>
      <status>HQ_OK</status>
      <creation_date>2010-05-11 10:28:02.244553</creation_date>
      <bank_branch_name></bank_branch_name>
      <bank_branch_code></bank_branch_code>
      <bank_accept_date>2010-05-11
10:28:35.435216</bank_accept_date>
      <bank_branch_accept_date></bank_branch_accept_date>
      <delivery_date>2010-05-18 00:00:00</delivery_date>
      <processed_date></processed_date>
      <processed_by></processed_by>
      <delivery_date>2010-05-18 00:00:00</delivery_date>
      <trans_type>Account</trans_type>
      <benef_name>
        <![CDATA[ZIAD HASSAN SYFULLAH]]>
      </benef_name>
      <benef_tel>
        <![CDATA[]]>
      </benef_tel>
    </transaction>
  </result>
</response>
```

```
</benef_tel>
<benef_mobile>
  <![CDATA[07512487412]]>
</benef_mobile>
<benef_email>
  <![CDATA[]]>
</benef_email>
<collection_point>
  <![CDATA[BOGRA]]>
</collection_point>
<collection_point_code>
  <![CDATA[]]>
</collection_point_code>
<collection_pin>
  <![CDATA[]]>
</collection_pin>
<benef_id_type>
  <![CDATA[NONE]]>
</benef_id_type>
<benef_id_detail>
  <![CDATA[]]>
</benef_id_detail>
<benef_ac>
  <![CDATA[1234567890]]>
</benef_ac>
<benef_bank>
  <![CDATA[PRIME BANK LIMITED]]>
</benef_bank>
<benef_bank_swift_code>
  <![CDATA[ABC123]]>
</benef_bank_swift_code>
<benef_bank_ifsc_code>12340678912</benef_bank_ifsc_code>
<benef_branch>
  <![CDATA[AGRABAD]]>
</benef_branch>
<benef_bank_city>
  <![CDATA[CHITTAGONG]]>
</benef_bank_city>
<benef_bank_state>
  <![CDATA[CHITTAGONG]]>
</benef_bank_state>
<benef_card_number>
  <![CDATA[]]>
</benef_card_number>
<benef_address1>
  <![CDATA[BOGRA]]>
</benef_address1>
<benef_address2>
  <![CDATA[]]>
</benef_address2>
<benef_address3>
  <![CDATA[]]>
</benef_address3>
<benef_city>
  <![CDATA[BOGRA]]>
</benef_city>
<benef_state>
  <![CDATA[]]>
</benef_state>
<benef_postcode>
  <![CDATA[]]>
```

```

</benef_postcode>
<benef_mobiletransfer_number>
  <![CDATA[01234567]]>
</benef_mobiletransfer_number>
<benef_mobiletransfer_network>
  <![CDATA[mobileNetwork]]>
</benef_mobiletransfer_network>
<benef_mobiletransfer_network_credit_type>
  <![CDATA[]]>
</benef_mobiletransfer_network_credit_type>
<benef_mobiletransfer_network_credit_type_code>
  <![CDATA[]]>
</benef_mobiletransfer_network_credit_type_code>
<delivery_notes>
  <![CDATA[]]>
</delivery_notes>
<utility_company><![CDATA[]]></utility_company>
<utility_account_no><![CDATA[]]></utility_account_no>
<utility_invoice><![CDATA[]]></utility_invoice>
<utility_bank><![CDATA[]]></utility_bank>
<utility_bank_code><![CDATA[]]></utility_bank_code>
<utility_description><![CDATA[]]></utility_description>
<remitter_name>
  <![CDATA[ALI IMAN]]>
</remitter_name>
<remitter_id>1210</remitter_id>
<remitter_address>
  <![CDATA[#26-02, 9 TEMASEK BOULEVARD, SINGAPORE,
SINGAPORE]]>
</remitter_address>
<remitter_tel>
  <![CDATA[]]>
</remitter_tel>
<remitter_mobile>
  <![CDATA[]]>
</remitter_mobile>
<send_country>SG</send_country>
<send_currency>SGD</send_currency>
<send_amount>157.57</send_amount>
<rate>49.2800</rate>
<commission>5.00</commission>
<fees>0</fees>
<paid>162.57</paid>
<receive_country>BD</receive_country>
<receive_currency>BDT</receive_currency>
<receive_amount>7765.05</receive_amount>
<bank_sequence>2929</bank_sequence>
<pay_method></pay_method>
<issue_date></issue_date>
<bank_ref></bank_ref>
<benef_trans_ref>GHT33762634</benef_trans_ref>
<bank_comments></bank_comments>
<bank_credit_date></bank_credit_date>
<bank_clear_date></bank_clear_date>
<deleted_reason></deleted_reason>
<admin_comments><![CDATA[]]></admin_comments>
<agent_comments><![CDATA[]]></agent_comments>
<payout_comments><![CDATA[]]></payout_comments>
</transaction>
</result>
</response>

```

getDeliveryBanks

Group: **deliveryBank**

Method: **getDeliveryBanks**

Input fields:

Name	Type	Notes
dest_country	Text	The name of the Destination Country, as per the output of getDestinationCountries . You can provide destination country name or code.
country_code	Text	The code of the Destination Country, as per the output of getDestinationCountries . You can provide destination country name or code.
bank_code	Text	The code of the processing bank connected to the delivery bank. If no dest_country, country_code and bank_code is given, all the delivery banks are returned.

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>42</responseId>
  <status>SUCCESS</status>
  <result>
    <delivery_banks>
      <delivery_bank>
        <bank_id>14</bank_id>
        <name>delivery bank</name>
        <country_id>999</country_id>
        <bank_code>del123</bank_code>
        <swift_code>sw123</swift_code>
        <address>address 123</address>
        <city>city123</city>
        <state>state123</state>
        <telephone>123</telephone>
        <account_number_mask>123</account_number_mask>
        <enabled>t</enabled>
      </delivery_bank>
    </delivery_banks>
  </result>
</response>
```

getDeliveryBankBranches

Group: **deliveryBankBranch**

Method: **getDeliveryBankBranches**

Input fields:

Name	Type	Notes
delivery_bank	Text	The name of the delivery bank, as per the output of getDeliveryBanks .
destination_country	Text	Name of country retrieved from calling getDestinationCountries . This will be used if the destination_country_code is not provided.
destination_country_code	Text	Iso code of country retrieved from calling getDestinationCountries . This will be used in priority over the destination_country value.

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>44</responseId>
  <status>SUCCESS</status>
  <result>
    <delivery_bank_branches>
      <delivery_bank_branch>
        <id>1</id>
        <name>Branch123</name>
        <branch_code>BRANCH-123</branch_code>
        <enabled>t</enabled>
        <delivery_bank>1</delivery_bank>
        <city>city123</city>
        <state>state123</state>
        <telephone>012345678</telephone>
        <manager>manager123</manager>
      </delivery_bank_branch>
    </delivery_bank_branches>
  </result>
</response>
```

createDeliveryBankBranch

Group: **deliveryBankBranch**

Method: **createDeliveryBankBranch**

Input fields:

Name	Type	Notes
name	Text	The name of the delivery bank branch.
branch_code	Text	The code of the delivery bank branch.
enabled	Boolean	Enable/Disable this delivery bank branch.
bank	Text	The name of the delivery bank, as per the output of getDeliveryBanks .
address	Text	The address of the delivery bank branch.
city	Text	The city of the delivery bank branch.
state	Text	The state of the delivery bank branch.
telephone	Text	The telephone number of the delivery bank branch prefixed by + or 00 followed by the country code.
manager	Text	The manager's name of the delivery bank branch.
email	Text	The email of the delivery bank branch.
third_party_id	Text	An optional third party identifier.

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>44</responseId>
  <status>SUCCESS</status>
  <result>
    <id>1</id>
    <name>My Collection Point</name>
    <branch_code>CODE</branch_code>
    <enabled>t</enabled>
    <delivery_bank>1</delivery_bank>
    <address>123 Street</address>
    <city>London</city>
    <state>London</state>
    <telephone>004412356789</telephone>
    <manager>John Doe</manager>
    <email>delivery@bank.branch</email>
    <third_party_id>123</third_party_id>
  </result>
</response>
```

updateDeliveryBankBranch

Group: **deliveryBankBranch**

Method: `updateDeliveryBankBranch`

Input fields:

Name	Type	Notes
id	Integer	The id of the delivery bank branch.
name	Text	The name of the delivery bank branch.
branch_code	Text	The code of the delivery bank branch.
enabled	Boolean	Enable/Disable this delivery bank branch.
bank	Text	The name of the delivery bank, as per the output of <code>getDeliveryBanks</code> .
address	Text	The address of the delivery bank branch.
city	Text	The city of the delivery bank branch.
state	Text	The state of the delivery bank branch.
telephone	Text	The telephone number of the delivery bank branch prefixed by + or 00 followed by the country code.
manager	Text	The manager's name of the delivery bank branch.
email	Text	The email of the delivery bank branch.
third_party_id	Text	An optional third party identifier.

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>44</responseId>
  <status>SUCCESS</status>
  <result>
    <id>1</id>
    <name>My Collection Point</name>
    <branch_code>CODE</branch_code>
    <enabled>t</enabled>
    <delivery_bank>1</delivery_bank>
    <address>123 Street</address>
    <city>London</city>
    <state>London</state>
    <telephone>004412356789</telephone>
    <manager>John Doe</manager>
    <email>delivery@bank.branch</email>
    <third_party_id>123</third_party_id>
  </result>
</response>
```

getCollectionPoints

Group: collectionPoint

Method: getCollectionPoints

Input fields:

Name	Type	Notes
delivery_bank	Text	The name of the delivery bank, as per the output of getDeliveryBanks .
destination_country	Text	Name of country retrieved from calling getDestinationCountries . This will be used if the destination_country_code is not provided.
destination_country_code	Text	Iso code of country retrieved from calling getDestinationCountries . This will be used in priority over the destination_country value.

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>45</responseId>
  <status>SUCCESS</status>
  <result>
    <collection_points>
      <collection_point>
        <collection_id>1</collection_id>
        <name>colltion123</name>
        <bank>Bank 123</bank>
        <delivery_bank>1</delivery_bank>
        <address>address123</address>
        <city>city123</city>
        <state>state123</state>
        <country_id>123</country_id>
        <code>COLLECTION-123</code>
        <telephone>012345678</telephone>
        <fax>fax123</fax>
        <email>email123</email>
        <working_hours>1</working_hours>
        <contact_person>person123</contact_person>
        <default_in_country>t</default_in_country>
        <enabled>t</enabled>
        <collection_pin_prefix>123</collection_pin_prefix>
        <rate_markup>1</rate_markup>
      </collection_point>
    </collection_points>
  </result>
</response>
```

createCollectionPoint

Group: collectionPoint

Method: createCollectionPoint

Input fields:

Name	Type	Notes
name	Text	The name of the collection point.
bank	Text	The name of the delivery bank, as per the output of getDeliveryBanks .
country_id	Integer	The id of the Destination Country, as per the output of getDestinationCountries .
code	Text	The code of the collection point.
address	Text	The address of the collection point.
city	Text	The city of the collection point.
state	Text	The state of the collection point.
telephone	Text	The telephone number of the collection point prefixed by + or 00 followed by the country code.
fax	Text	The fax number of the collection point prefixed by + or 00 followed by the country code.
email	Text	The email of the collection point.
working_hours	Text	The working hours of the collection point.
contact_person	Text	The name of the person to contact in the collection point.
default_in_country	Boolean	Set this collection point as the default one for the specified country.
enabled	Boolean	Enable/Disable this collection point.

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>42</responseId>
  <status>SUCCESS</status>
  <result>
    <collection_id>1</collection_id>
    <name>My Collection Point</name>
    <bank>My delivery Bank</bank>
    <delivery_bank>1</delivery_bank>
    <address>123 Street</address>
    <city>London</city>
    <state>London</state>
```

```

    <country_id>999</country_id>
    <code>CODE</code>
    <telephone>0044123456789</telephone>
    <fax>0044123456789</fax>
    <email>collection@point.com</email>
    <working_hours>8am to 7pm</working_hours>
    <contact_person>John Doe</contact_person>
    <default_in_country>f</default_in_country>
    <enabled>t</enabled>
    <collection_pin_prefix>CP</collection_pin_prefix>
    <rate_markup></rate_markup>
  </result>
</response>

```

updateCollectionPoint

Group: **collectionPoint**

Method: **updateCollectionPoint**

Input fields:

Name	Type	Notes
collection_id	Integer	The id of the collection point.
name	Text	The name of the collection point.
bank	Text	The name of the delivery bank, as per the output of getDeliveryBanks .
country_id	Integer	The id of the Destination Country, as per the output of getDestinationCountries .
code	Text	The code of the collection point.
address	Text	The address of the collection point.
city	Text	The city of the collection point.
state	Text	The state of the collection point.
telephone	Text	The telephone number of the collection point prefixed by + or 00 followed by the country code.
fax	Text	The fax number of the collection point prefixed by + or 00 followed by the country code.
email	Text	The email of the collection point.
working_hours	Text	The working hours of the collection point.
contact_person	Text	The name of the person to contact in the

		collection point.
default_in_country	Boolean	Set this collection point as the default one for the specified country.
enabled	Boolean	Enable/Disable this collection point.

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>42</responseId>
  <status>SUCCESS</status>
  <result>
    <collection_id>1</collection_id>
    <name>My Collection Point</name>
    <bank>My delivery Bank</bank>
    <delivery_bank>1</delivery_bank>
    <address>123 Street</address>
    <city>London</city>
    <state>London</state>
    <country_id>999</country_id>
    <code>CODE</code>
    <telephone>0044123456789</telephone>
    <fax>0044123456789</fax>
    <email>collection@point.com</email>
    <working_hours>8am to 7pm</working_hours>
    <contact_person>John Doe</contact_person>
    <default_in_country>f</default_in_country>
    <enabled>t</enabled>
    <collection_pin_prefix>CP</collection_pin_prefix>
    <rate_markup></rate_markup>
  </result>
</response>
```

getTransactions

Group : transaction

Method : getTransactions

Input fields :

Name	Type	Notes
from_date	Date	YYYY-MM-DD (Either this date is required or the processed_from_date)
to_date	Date	YYYY-MM-DD (Either this date is required or the processed_to_date)
processed_from_date	Date Time	YYYY-MM-DD HH:MM
processed_to_date	Date Time	YYYY-MM-DD HH:MM
country_code	Text	2 letter country ISO code. (Country code

		as set-up in the system)
country_type	Text	SOURCE / DESTINATION (Required if country_code is provided)
trans_corridor	Text	ALL / INTERNATIONAL / DOMESTIC (If not provided, will default to ALL)
compliance_needed	Boolean	t/f

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>SUCCESS</status>
  <result>
    <count>1</count>
    <transactions>
      <transaction>
        <trans_ref>RA20079112</trans_ref>
        <trans_type>Account</trans_type>
        <status>HQ_OK</status>
        <creation_date>2010-05-07 16:25:12.26266</creation_date>
        <processed_date>2010-05-09 16:25:12</processed_date>
        <originating_country>SG</originating_country>
        <destination_country>BD</destination_country>
        <source_currency>SGD</source_currency>
        <source_amount>85.00</source_amount>
        <commission>5.00</commission>
        <tax>5.00</tax>
        <dest_currency>BDT</dest_currency>
        <dest_amount>4188.80</dest_amount>
        <rate>49.2800</rate>
        <remitter_name>ALI IMAN</remitter_name>
        <remitter_id>1210</remitter_id>
        <payment_method>1</payment_method>
        <benef_name>HASSAN SYFULLAH</benef_name>
        <benef_mobile>07512457841</benef_mobile>
        <source_agent_name>JOE BLOGGS</source_agent_name>
        <source_agent_teller_name>JOHN
TELLER</source_agent_teller_name>
        <bank_buy_rate>53.3200</rate>
        <source_conv_rate>48.1400</source_conv_rate>
        <dest_conv_rate>49.2800</dest_conv_rate>
        <compliance_needed>f</compliance_needed>
        <compliance_checked>f</compliance_checked>
        <deleted_reason></deleted_reason>
        <admin_comments><![CDATA[]]></admin_comments>
        <agent_comments><![CDATA[]]></agent_comments>
        <payout_comments><![CDATA[]]></payout_comments>
      </transaction>
    </transactions>
  </result>
</response>
```

assignTransactions

Group : transaction

Method : assignTransactions

Input fields :

Name	Type	Notes
trans_refs	Text	Comma separated list of transaction references. I.e. RA123451234 or RA123451234, RA123451251, RA123453614
proc_bank_name	Text	Name of the Processing Bank (Abbreviated Bank Name)
proc_bank_branch_code	Text	The Processing Bank Branch Code

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>326</responseId>
  <status>SUCCESS</status>
  <result>
    <transactions>
      <transaction>
        <trans_ref>RC000647000434</trans_ref>
        <proc_bank_name>HSBC_UK</proc_bank_name>
      </transaction>
      <transaction>
        <trans_ref>RA000606000286</trans_ref>
        <proc_bank_name>HSBC_UK</proc_bank_name>
        <proc_bank_branch>HSBC_LUTON</proc_bank_branch>
      </transaction>
    </transactions>
  </result>
</response>
```

getErrorTransactions

Group : transaction

Method : getErrorTransactions

Input fields :

Name	Type	Notes
from_date	Date	YYYY-MM-DD
to_date	Date	YYYY-MM-DD
bank_name	Text	Abbreviated Bank Name

Example Output XML :

```

<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>326</responseId>
  <status>SUCCESS</status>
  <result>
    <count>{$trans_count}</count>
    <transactions>
      <transaction>
        <trans_ref>CC123456789</trans_ref>
        <agent_trans_ref></agent_trans_ref>
        <trans_type>CASH_COLLECTION</trans_type>
        <status>ERROR</status>
        <error_date>2016-02-26</error_date>
        <error_reason>Invalid Beneficiary ID</error_reason>
<previous_status_before_error>SENT_FOR_PAY</previous_status_before_error>
        <error_user_id>1</error_user_id>
        <error_username>admin</error_username>
        <error_agent_code>123</error_agent_code>
        <error_agent_name>Admin</error_agent_name>
        <remitter_id>10000</remitter_id>
        <remitter_name>John Doe</remitter_name>
        <beneficiary_id>1</beneficiary_id>
        <beneficiary_name>Jane Doe</beneficiary_name>
        <destination_country>China</destination_country>
        <originating_country>France</originating_country>
        <source_currency>EUR</source_currency>
        <source_transfer_amount>42</source_transfer_amount>
        <rate>13.37</rate>
        <destination_currency>CNY</destination_currency>
        <destination_amount>561.54</destination_amount>
        <delivery_date></delivery_date>
        <admin_comments><![CDATA[]]></admin_comments>
        <agent_comments><![CDATA[]]></agent_comments>
        <payout_comments><![CDATA[]]></payout_comments>
      </transaction>
    </transactions>
  </result>
</response>

```

updateErrorTransaction

Group: transaction

Method: updateErrorTransaction

Input fields:

Name	Type	Notes
trans_ref*	Text	Transaction reference from the sending system.
purpose	Text	See 'Description' on Appendix G
source_of_income	Text	See 'Description' on Appendix F
benef_name	Text	
benef_fname	Text	
benef_lname	Text	
benef_address	Text	
benef_address2	Text	
benef_address3	Text	
benef_city	Text	
benef_state	Text	
benef_postcode	Text	
benef_tel	Text	
benef_mobile	Text	
benef_id_type	Text	
benef_id_details	Text	
benef_ac	Text	
benef_ac_type	Text	
benef_bank	Text	
benef_bank_swift_code	Text	
benef_branch	Text	
benef_branch_code	Text	
benef_branch_address	Text	
benef_branch_tel	Text	
benef_branch_city	Text	
benef_branch_state	Text	
benef_inter_bank_account	Text	
benef_inter_bank_swift_code	Text	

benef_inter_bank_name	Text	
benef_inter_bank_address	Text	
benef_bank_routing_transit_number	Text	
benef_card_number	Text	
collection_point	Text	
collection_point_code	Text	
collection_point_bank	Text	
collection_point_addresses	Text	
collection_point_city	Text	
collection_pin	Text	
collection_telephone	Numeric	
benef_utilitybillcompany	Text	Name of the Utility Company where money will be paid. Required if transaction is of type Utility Bill.
benef_utilitybillaccountno	Text	Customer Account No. Required if transaction is of type Utility Bill.
benef_utilitybillinvoice	Text	Invoice No.
benef_utilitybillbank	Text	Bank Name
benef_utilitybillbankcode	Text	Bank IBAN/SWIFT Code.
benef_utilitybillbankbic	Text	
benef_utilitybilldescription	Text	Bill Description.
benef_utilitybill_address1	Text	
benef_utilitybill_address2	Text	
benef_utilitybill_address3	Text	
benef_utilitybill_city	Text	
benef_utilitybill_state	Text	
benef_utilitybill_postcode	Text	
benef_mobiletransfer_number	Text	

benef_mobiletransfer_network	Text	
benef_mobiletransfer_network_credit_type	Text	
sms_notification	Boolean	t / f
sms_confirmation	Boolean	t / f
sms_benef_confirmation	Boolean	t / f
agent_comments	Text	
comments_to_beneficiary		

NOTE: The only Required field is trans_ref the rest are optional. You can use any of the fields according to the changes you want to apply to the transaction send to another REMITONE System.

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>SUCCESS</status>
  <result>
    <trans_ref>RA0003101</trans_ref>
    <agent_trans_ref>ABC1234567sss8</agent_trans_ref>
    <remitter_id>1111</remitter_id>
    <remitter_name>John Smith</remitter_name>
    <beneficiary_id>2222</beneficiary_id>
    <beneficiary_name>Jane Doe</beneficiary_name>
    <trans_type>Cash Collection</trans_type>
    <destination_country>GH</destination_country>
    <source_currency>USD</source_currency>
    <source_transfer_amount>94</source_transfer_amount>
    <rate>1.9900</rate>
    <destination_currency>GHS</destination_currency>
    <destination_amount>187.06</destination_amount>
    <commission>2.50</commission>
    <tax>1.00</tax>
    <agent_fee></agent_fee>
    <hq_fee>0.00</hq_fee>
    <remitter_pay_amount>97.50</remitter_pay_amount>
    <agent_deduction>0.00</agent_deduction>
    <agent_to_pay_hq>97.50</agent_to_pay_hq>
    <delivery_date>2012-01-01 00:00:00+00</delivery_date>
  </result>
</response>
```

processCashCollectionTransactionByBank

Group : transaction

Method : processingCashCollectionTransactionByBank

Input fields :

Name	Type	Notes
trans_refs	Text	text
bank_name	Text	Name of the Processing Bank (Abbreviated Bank Name)

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>SUCCESS</status>
  <result>
    <trans_ref>RA20079121</trans_ref>
    <process_result>SUCCESS</process_result>
    <message></message>
  </result>
</response>
```

requestCancellation

Group : transaction

Method : requestCancellation

Input fields :

Name	Type	Notes
trans_ref	Text	Transaction reference I.e. RA123451234
reason	Text	Reason for the cancellation/deletion

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>326</responseId>
  <status>SUCCESS</status>
  <result>
    <trans_ref>RA123451234</trans_ref>
    <message>Transaction successfully aborted</message>
  </result>
</response>
```

Depending on the configuration of your system which is on a per user-type / transaction-status basis, the transaction will either be ABORTED or DELETED after this call.

getMobileNetworkOperators

Group : mobileNetworkOperator

Method : getMobileNetworkOperators

Input fields :

Name	Type	Notes
country_id	Integer	The id of the Destination Country, as per the output of getDestinationCountries .
country_code	Text	The ISO code of the Destination Country.

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>42</responseId>
  <status>SUCCESS</status>
  <result>
    <count>1</count>
    <mnos>
      <mno>
        <mobile_network_id>1</mobile_network_id>
        <name>AT&T</name>
        <code>ATT</code>
        <country_id>999</country_id>
        <number_pattern></number_pattern>
        <regex_pattern></regex_pattern>
        <enabled>t</enabled>
      </mno>
    </mnos>
  </result>
</response>
```

createMobileNetworkOperators

Group : mobileNetworkOperator

Method : createMobileNetworkOperators

Input fields :

Name	Type	Notes
name	Text	The name of the Mobile Network Operator.
code	Text	The code of the Mobile Network Operator.
country_id	Text	The country id of the destination country
third_party_id	Text	The third party id.
number_pattern	Text	The number pattern.
regex_pattern	Text	The regex pattern.
enabled	Boolean	The enabled field. Supported options are t or f.
default_in_country	Boolean	The default in country field. Supported options are t or f.

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>42</responseId>
  <status>SUCCESS</status>
  <result>
    <id>1</credit>
    <name>My Mobile Network Operator</name>
    <code>CODE</code>
    <country_id>999</country_id>
    <third_party_id></third_party_id>
    <number_pattern></number_pattern>
    <regex_pattern></regex_pattern>
    <enabled>t</enabled>
    <default_in_country>t</default_in_country>
  </result>
</response>
```

updateMobileNetworkOperators

Group : mobileNetworkOperator

Method : updateMobileNetworkOperators

Input fields :

Name	Type	Notes
name	Text	The name of the Mobile Network Operator.
code	Text	The code of the Mobile Network Operator.

country_id	Text	The country id of the destination country
third_party_id	Text	The third party id.
number_pattern	Text	The number pattern.
regex_pattern	Text	The regex pattern.
enabled	Boolean	The enabled field. Supported options are t or f.
default_in_country	Boolean	The default in country field. Supported options are t or f.

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>42</responseId>
  <status>SUCCESS</status>
  <result>
    <id>1</credit>
    <name>My Mobile Network Operator</name>
    <code>CODE</code>
    <country_id>999</country_id>
    <third_party_id></third_party_id>
    <number_pattern></number_pattern>
    <regex_pattern></regex_pattern>
    <enabled>t</enabled>
    <default_in_country>t</default_in_country>
  </result>
</response>
```

createMobileNetworkCreditType

Group : mobileNetworkCreditType

Method : createMobileNetworkCreditType

Input fields :

Name	Type	Notes
name	Text	The name of the mobile network credit type.
code	Text	The code of the mobile network credit type.
mobile_network_id	Integer	The id of the Mobile Network Operator, as per the output of getMobileNetworkOperators .
type	Text	The code of the mobile network credit type. I.e. Fixed

		Open
currency	Text	The currency ISO code of the mobile network credit type.
amount	Float	The amount of the mobile network credit type (used when the type is set to Fixed)
minimum	Float	The minimum amount of the mobile network credit type (used when the type is set to Open).
maximum	Float	The maximum amount of the mobile network credit type (used when the type is set to Open).
enabled	Boolean	Enable/Disable the mobile network credit type.

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>42</responseId>
  <status>SUCCESS</status>
  <result>
    <credit_type_id>1</credit_type_id>
    <mobile_network_id>1</mobile_network_id>
    <name>My Credit Type</name>
    <code>CODE</code>
    <country_id>999</country_id>
    <type>Fixed</type>
    <currency>GBP</currency>
    <minimum></minimum>
    <maximum></maximum>
    <amount>13.37</amount>
    <enabled>t</enabled>
  </result>
</response>
```

updateMobileNetworkCreditType

Group : **mobileNetworkCreditType**
Method : **updateMobileNetworkCreditType**

Input fields :

Name	Type	Notes
credit_type_id	Integer	The id of the mobile network credit type.
name	Text	The name of the mobile network credit type.
code	Text	The code of the mobile network credit type.
mobile_network_id	Integer	The id of the Mobile Network Operator, as per

		the output of getMobileNetworkOperators .
type	Text	The code of the mobile network credit type. I.e. Fixed Open
currency	Text	The currency ISO code of the mobile network credit type.
amount	Float	The amount of the mobile network credit type (used when the type is set to Fixed)
minimum	Float	The minimum amount of the mobile network credit type (used when the type is set to Open).
maximum	Float	The maximum amount of the mobile network credit type (used when the type is set to Open).
enabled	Boolean	Enable/Disable the mobile network credit type.

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>42</responseId>
  <status>SUCCESS</status>
  <result>
    <credit_type_id>1</credit_type_id>
    <mobile_network_id>1</mobile_network_id>
    <name>My Credit Type</name>
    <code>CODE</code>
    <country_id>999</country_id>
    <type>Fixed</type>
    <currency>GBP</currency>
    <minimum></minimum>
    <maximum></maximum>
    <amount>13.37</amount>
    <enabled>t</enabled>
  </result>
</response>
```

getMobileNetworkCreditTypes

Group : mobileNetworkCreditType

Method : getMobileNetworkCreditTypes

Input fields :

Name	Type	Notes
username	Text	The username given to authenticate
password	Text	The password given for the user
PIN	Number	The 5-digits number given to authenticate along with username and password
country_id	Number	Optional. The ID of the country where to find credit types in. If it is empty and the destination country is empty, it will search for all the credit types. If the user has type WS_Bank, the country_id will be extracted from the user information, so this field can be empty.
dest_country	Text	Optional. The name of the country where to find credit types in. If it is empty and the country_id is empty, it will search for all the credit types. If the user has type WS_Bank, the country_id will be extracted from the user information, so this field can be empty.
country_code	Text	Option. The ISO code of the Destination Country.
mobile_network_operator_name	Text	Optional. The mobile network operator name.
enabled	Text	Optional. The enabled field. Supported values are true or false. If you send it empty it will return both the enabled and disabled mobile network credit types.

Example Output XML :

```

<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>121</responseId>
  <status>SUCCESS</status>
  <result>
    <count>1</count>
    <mncts>
      <mnct>
        <credit_type_id>1</credit_type_id>
        <mobile_network_id>1</mobile_network_id>
        <name>My First MNCT</name>
        <code>MNCTO</code>
        <country_id>999</country_id>
        <type>Open</type>
        <currency>EUR</currency>
        <minimum>12000.00</minimum>
        <maximum>85000.00</maximum>
        <amount>90000.00</amount>
      </mnct>
    </mncts>
  </result>
</response>

```

```

        <enabled>t</enabled>
      </mnet>
    </mnet>
  </result>
</response>

```

uploadCSVData

Group : CSV

Method : uploadCSVData

Input fields :

Name	Type	Notes
csv_type	Text	The type of the CSV file that is being uploaded. At the moment, we support the following types: collection_point commission_slab delivery_bank delivery_bank_branch sell_rate
csv_file	File	The CSV file to upload.

Based on the CSV file type, the file headers should be as follows:

1. collection_point

[name,code,bank,address,city,state,country,telephone,fax,email,working_hours,contact_person,default_in_country]

The 'name' field must be unique for all rows.

2. commission_slab

[SourceCountry,SourceCurrency,DestinationCountry,DestinationCurrency,CommissionSlabsCategory,TransferType,UpTo,FixedAmount,PercentageAmount>Delete]

In the last column, set Delete to 't' to delete that row. Any other value will result in the row being added.

3. delivery_bank

[name,country,bank_code,address,city,state,telephone]

4. delivery_bank_branch

[name,branch_code,enabled,delivery_bank,country,address,city,state,telephone,manager]

5. sell_rate

[SourceCountry,SourceCurrency,DestinationCountry,DestinationCurrency,IsHQAgent(T,F),CashCollectionRate,AccountTransferRate,CardTransferRate,HomeDeliveryRate,UtilityBillRate,MobileTransferRate,WalletTransferRate,Category]

Example Output XML for type=collection_point :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>42</responseId>
  <status>SUCCESS</status>
  <results>
    <total>5</total>
    <success>
      <total>1</total>
      <messages>
        <message>Line 3: My Collection Point inserted
successfully</message>
      </messages>
    </success>
    <failure>
      <total>1</total>
      <messages>
        <message>Line 5: Error performing operation, row not
inserted!</message>
      </messages>
    </failure>
    <missing_name>
      <total>1</total>
      <messages>
        <message>Line 2: Missing data in Name field, row not
inserted!</message>
      </messages>
    </missing_name>
    <missing_value>
      <total>1</total>
      <messages>
        <message>Line 4: The Bank XXX does not exist in
database, row not inserted!</message>
      </messages>
    </missing_value>
    <duplicate>
      <total>1</total>
      <messages>
        <message>Line 1: The Collection Point already exists in
the database, row not inserted!</message>
      </messages>
    </duplicate>
  </results>
</response>
```

updateSellRates

Group : rates

Method : updateSellRates()

This method updates the sell rates.

Input fields:

Name	Type	Notes
source_country ID*	Text	The ID of the Source Country, as per the output of getSourceCountries .
source_currency *	Text	The 3 letters ISO currency code, as listed in Appendix A
destination_country ID *	Text	The three digit country code
destination_currency *	Text	The 3 letters ISO currency code, as listed in Appendix A
Is HQ Agent (t,f)	Text	Either t or f for entries.
Cash Collection Rate	Text	Rate for the cash collection
Account Transfer Rate	Text	The amount to be transferred
Card Transfer Rate	Text	The 3 letters ISO currency code, as listed in Appendix A
Home Delivery Rate	Text	Rate for the home delivery
Utility Bill Rate	Text	Rate for the bill
Mobile Transfer Rate	Text	Rate for the mobile transfer
Wallet Transfer Rate	Text	Rate for the wallet transfer
Category*	Text	One letter character e.g. 'A', 'B' etc

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>SUCCESS</status>
</response>
```

updateConversionRates

Group : **rates**

Method : **updateConversionRates()**

This method updates the conversion rates.

Input fields:

Name	Type	Notes
------	------	-------

currency	Text	The 3 letters ISO currency code, as listed in Appendix A
type	Text	'source' or 'destination'. Source will change currency to system rate whereas destination will change system to currency rate
rate	float	Rate for the currency

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>SUCCESS</status>
</response>
```

addWalletCredit

Group : remitter

Method : addWalletCredit

This method adds credit to the remitter's wallet.

Input fields:

Name	Type	Notes
remitter_id *	Integer	The member ID of remitter
credit_amount *	Float	The credit amount to be added
credit_currency	Float	The 3 letter ISO currency code. If not provided the default currency of that Country will be used

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>105</responseId>
  <status>SUCCESS</status>
  <result>
    <remitter>
      <remitter_id>10001</remitter_id>
      <deposit_ref>RD0005614996</deposit_ref>
      <deposit_amount>GBP 50.00</deposit_amount>
      <balance_after>150.00</balance_after>
    </remitter>
  </result>
</response>
```

errorTransaction

Group : **admin**

Method : **errorTransaction**

This method errors transactions that are PROCESSED by the WS_Admin.

Input fields:

Name	Type	Notes
trans_ref	Integer	The transaction ref of the transaction
error_reason	Text	The reason for erroring the transaction
error_explanation	Text	Detailed explanation for erroring the transaction.

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>105</responseId>
  <status>SUCCESS</status>
  <result>
    <remitter>
      <transaction_id>10001</remitter_id>
      <trans_ref>RD0005614996</deposit_ref>
      <deposit_amount>GBP 50.00</deposit_amount>
      <balance_after>150.00</balance_after>
    </remitter>
  </result>
</response>
```

createRemitter

Group : **remitter**

Method : **createRemitter**

Input fields :

Name	Type	Notes
agent_name *	Text	Agent code to assign remitter to

firstname *	Text	
middlename	Text	
lastname *	Text	
type *	Text	registered / nonregistered
nationality	Text	2 letter country ISO code. Please refer to Appendix A.
gender	Text	Male / Female
status *	Text	valid / expired / blocked
comments	Text	
address1 *	Text	
address2	Text	
city *	Text	
state	Text	
postcode	Text	
email	Text	
telephone	Text	
mobile	Text	
dob	Date	YYYY-MM-DD. Required if type is 'registered'.
place_of_birth	Text	
country_of_birth	Text	2 letter country ISO code. Please refer to Appendix A.
fathers_name	Text	
mothers_name	Text	
national_id_number	Text	
id_type	Code	See Appendix B for codes. Required if type is 'registered'.
id_details	Text	Required if type is 'registered'.
id_issued_by	Text	Required if type is 'registered'.
id_issue_place	Text	Required if type is 'registered'.
id_start	Date	YYYY-MM-DD. Required if type is 'registered'.
id_expiry	Date	YYYY-MM-DD. Required if type is 'registered'.
id2_type	Code	See Appendix B for codes.
id2_details	Text	
id2_issued_by	Text	
id2_issue_place	Text	

id2_start	Date	YYYY-MM-DD
id2_expiry	Date	YYYY-MM-DD
account_number	Text	
sort_code	Text	
groups	Text	Comma separate string e.g Default,Group2,Group3
credit_balance	Text	
credit_currency	Text	
ld1_scan	File	png, gif, jpg, jpeg, bmp and pdf
ld2_scan	File	png, gif, jpg, jpeg, bmp and pdf
orgtype	Text	'Individual' or 'Corporate'
company_name	Text	If orgtype is 'Individual' it will be set to empty
company_type	Text	If orgtype is 'Individual' it will be set to empty
company_reg_no	Text	If orgtype is 'Individual' it will be set to empty
purpose	Text	
source_of_income	Text	
employer	Text	
business_address	Text	The address of the employer
annual_income	Text	
annual_remittance	Text	
hear_about_us	Text	as per the output of getRemitterUISettings.
hear_about_us_others	Text	If hear_about_us is "Other" then specify, otherwise it will be saved as empty anyway
referral_code	Text	Customer referral code
residency_status	Text	Values can be "resident" or "nonresident"
nearest_airport	Text	The name of the nearest airport

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>SUCCESS</status>
  <new_remitter_id>10024</new_remitter_id>
</response>
```

searchRemitter

Group : remitter

Method : searchRemitter

Input fields :

Name	Type	Notes
agent_name *	Text	Agent code of the remitter's Agent
remitter_id	Text	Required if the full name is not provided.
firstname	Text	Required if remitter_id is not provided.
middlename	Text	
lastname	Text	Required if remitter_id is not provided.
dob	Date	YYYY-MM-DD
address_line1	Text	
city	Text	
postcode	Text	
telephone	Text	
mobile	Text	
email	Text	
id1_type	Text	See Appendix C for codes
id1_details	Text	
remitter_type	Text	registered / nonregistered
show_scans	Boolean	Show the Base64 encoded scans
referral_code	Text	Customer referral code

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>

<responseid>440</responseid>
  <status>SUCCESS</status>
  <result>
    <remitters>
      <remitter>
        <remitter_id>10001</remitter_id>
        <status>valid</status>
        <firstname>Rifter</firstname>
        <middlename></middlename>
        <lastname>Saleem</lastname>
        <address1>100 Some Street</address1>
        <address2></address2>
        <address3></address3>
      </remitter>
    </remitters>
  </result>
</response>
```

```
<city>N/A</city>
<state>N/A</state>
<postcode></postcode>
<country>United Kingdom</country>
<telephone></telephone>
<mobile></mobile>
<email></email>
<dob></dob>
<place_of_birth></place_of_birth>
<country_of_birth></country_of_birth>
<nationality></nationality>
<fathers_name></fathers_name>
<mothers_name></mothers_name>
<national_id_number></national_id_number>
<id_type>Passport</id_type>
<id_details>
    <!--[CDATA['01283018320ojahsdjkhasd']]-->
</id_details>
<id_issued_by>Me</id_issued_by>
<id_issue_place>ZA</id_issue_place>
<id_start>2012-12-12</id_start>
<id_expiry>9999-12-11</id_expiry>
<id_scan1>10001_3167915_201908011715.jpg</id_scan1>
<id_scan1_content></id_scan1_content>
<id_scan2></id_scan2>
<id_scan2_content></id_scan2_content>
<id_scan3></id_scan3>
<id_scan3_content></id_scan3_content>
<id2_type></id2_type>
<id2_details></id2_details>
<id2_issued_by></id2_issued_by>
<id2_issue_place></id2_issue_place>
<id2_start></id2_start>
<id2_expiry></id2_expiry>
<id2_scan1></id2_scan1>
<id2_scan1_content></id2_scan1_content>
<id2_scan2></id2_scan2>
<id2_scan2_content></id2_scan2_content>
<id2_scan3></id2_scan3>
<id2_scan3_content></id2_scan3_content>
<id3_type></id3_type>
<id3_details></id3_details>
<id3_issued_by></id3_issued_by>
<id3_issue_place></id3_issue_place>
<id3_start></id3_start>
<id3_expiry></id3_expiry>
<id3_scan1></id3_scan1>
<id3_scan1_content></id3_scan1_content>
<id3_scan2></id3_scan2>
<id3_scan2_content></id3_scan2_content>
<id3_scan3></id3_scan3>
<id3_scan3_content></id3_scan3_content>
<id4_type></id4_type>
<id4_details></id4_details>
<id4_issued_by></id4_issued_by>
<id4_issue_place></id4_issue_place>
<id4_start></id4_start>
<id4_expiry></id4_expiry>
<id4_scan1></id4_scan1>
<id4_scan1_content></id4_scan1_content>
<id4_scan2></id4_scan2>
```

```

<id4_scan2_content></id4_scan2_content>
<id4_scan3></id4_scan3>
<id4_scan3_content></id4_scan3_content>
<kyc_video>https://.../video.mp4</kyc_video>
<comments></comments>
<groups>Default</groups>
<credit_balance>0.00</credit_balance>
<credit_currency>GBP</credit_currency>
<bank_name></bank_name>
<bank_branch_name></bank_branch_name>
<bank_account_number></bank_account_number>
<bank_iban></bank_iban>
<bank_swift_code></bank_swift_code>
<bank_bic_code></bank_bic_code>
<orgtype>Individual</orgtype>
<annual_income></annual_income>
<annual_remittance></annual_remittance>
<purpose></purpose>
<source_of_income></source_of_income>
<hear_about_us></hear_about_us>
<hear_about_us_other></hear_about_us_other>
<residency_status></residency_status>
<nearest_airport></nearest_airport>
</remitter>
</remitters>
</result>
</response>

```

updateRemitter

Group : remitter

Method : updateRemitter

Input fields :

Name	Type	Notes
agent_name *	Text	Agent code of the remitter's Agent
remitter_id *	Text	
firstname	Text	
middlename	Text	
lastname	Text	
type	Text	registered / nonregistered
nationality	Text	2 letter country ISO code. Please refer to Appendix A.
status	Text	valid / expired / blocked
comments	Text	
address1	Text	
address2	Text	

city	Text	
state	Text	
postcode	Text	
email	Text	
telephone	Text	
mobile	Text	
Occupation	Text	
dob	Date	YYYY-MM-DD
place_of_birth	Text	
country_of_birth	Text	2 letter country ISO code. Please refer to Appendix A.
fathers_name	Text	
mothers_name	Text	
national_id_number	Text	
id_type	Code	See Appendix B for codes
id_details	Text	
id_issued_by	Text	
id_issue_place	Text	
id_start	Date	YYYY-MM-DD
id_expiry	Date	YYYY-MM-DD
id_verified	Text	VERIFIED, UNVERIFIED, FAIL
id2_type	Code	See Appendix B for codes
id2_details	Text	
id2_issued_by	Text	
id2_issue_place	Text	
id2_start	Date	YYYY-MM-DD
id2_expiry	Date	YYYY-MM-DD
id2_verified	Text	VERIFIED, UNVERIFIED, FAIL
id3_type	Code	See Appendix B for codes
id3_details	Text	
id3_issued_by	Text	
id3_issue_place	Text	
id3_start	Date	YYYY-MM-DD
id3_expiry	Date	YYYY-MM-DD

id3_verified	Text	VERIFIED, UNVERIFIED, FAIL
id4_type	Code	See Appendix B for codes
id4_details	Text	
id4_issued_by	Text	
id4_issue_place	Text	
id4_start	Date	YYYY-MM-DD
id4_expiry	Date	YYYY-MM-DD
id4_verified	Text	VERIFIED, UNVERIFIED, FAIL
annual amount	Text	
account_number	Text	
sort_code	Text	
groups	Text	Comma separate string e.g Default,Group2,Group3
ld1_scan	File	png, gif, jpg, jpeg, bmp and pdf
ld2_scan	File	png, gif, jpg, jpeg, bmp and pdf
ld3_scan	File	png, gif, jpg, jpeg, bmp and pdf
ld4_scan	File	png, gif, jpg, jpeg, bmp and pdf
orgtype	Text	'Individual' or 'Corporate'
company_name	Text	If orgtype is 'Individual' it will be set to empty
company_type	Text	If orgtype is 'Individual' it will be set to empty
company_reg_no	Text	If orgtype is 'Individual' it will be set to empty
purpose	Text	
source_of_income	Text	
employer	Text	
business_address	Text	The address of the employer
annual_income	Text	
annual_remittance	Text	
hear_about_us	Text	as per the output of getRemitterUISettings.
hear_about_us_other	Text	If hear_about_us is "Other" then specify, otherwise it will be saved as empty anyway
remitter_username	Text	The new email (username) for the remitter
referral_code	Text	Customer referral code

residency_status	Text	Values can be "resident" or "nonresident"
nearest_airport	Text	The name of the nearest airport

NOTE: All valid fields must be provided on update, otherwise existing fields may be blanked out.

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseid>445</responseid>
  <status>SUCCESS</status>
  <result>
    <remitter>
      <remitter_id>10001</remitter_id>
      <firstname>Rifter</firstname>
      <middlename>Arif</middlename>
      <lastname>Saleem</lastname>
      <address1>100 Some Street</address1>
      <address2></address2>
      <address3></address3>
      <city>N/A</city>
      <state>N/A</state>
      <postcode></postcode>
      <country>United Kingdom</country>
      <telephone></telephone>
      <mobile></mobile>
      <email></email>
      <dob></dob>
      <gender>UNSPECIFIED</gender>
      <place_of_birth></place_of_birth>
      <country_of_birth></country_of_birth>
      <nationality></nationality>
      <fathers_name></fathers_name>
      <mothers_name></mothers_name>
      <national_id_number></national_id_number>
      <id_type>Passport</id_type>
      <id_details>asd</id_details>
      <id_issued_by>Me</id_issued_by>
      <id_issue_place>ZA</id_issue_place>
      <id_issue_country></id_issue_country>
      <id_start>2018-01-01</id_start>
      <id_expiry>2022-01-01</id_expiry>
      <id_scan>10001_9978098_201908021047.jpg</id_scan>
      <id_scan1_content></id_scan1_content>
      <id_scan2></id_scan2>
      <id_scan2_content></id_scan2_content>
      <id_scan3></id_scan3>
      <id_scan3_content></id_scan3_content>
      <orgtype>Individual</orgtype>
      <purpose></purpose>
      <source_of_income></source_of_income>
      <employer></employer>
      <business_address></business_address>
      <annual_income></annual_income>
      <annual_remittance></annual_remittance>
      <hear_about_us></hear_about_us>
      <hear_about_us_other></hear_about_us_other>
    </remitter>
  </result>
</response>
```

```

        <residency_status></residency_status>
        <nearest_airport></nearest_airport>
        <remitter_username></remitter_username>
    </remitter>
</result>
</response>

```

uploadRemitterKYCVideo

Group : remitter

Method : uploadRemitterKYCVideo

Input fields :

Name	Type	Notes
remitter_id *	Text	
agent_name *	Text	Agent code of the remitter's Agent
filename	Text	The filename to get the extension from (at least the basename)
kyc_video	Text	Base64 encoding of the video content

Example Output XML :

```

<?xml version="1.0" encoding="utf-8"?>
<response>
    <status>SUCCESS</status>
    <result>
        <kyc_video>https://.../video.mp4</kyc_video>
    </result>
</response>

```

getRemitterKYCVideo

Group : remitter

Method : getRemitterKYCVideo

Input fields :

Name	Type	Notes
remitter_id *	Text	

agent_name *	Text	Agent code of the remitter's Agent
--------------	------	------------------------------------

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>SUCCESS</status>
  <result>
    <kyc_video>
      <data>...base64...</data>
      <mimetype>video/mp4</mimetype>
    </result>
  </response>
```

verifyRemitter

Group : remitter
Method : verifyRemitter

Input fields :

Name	Type	Notes
remitter_id *	Text	
agent_name *	Text	Agent code of the remitter's Agent

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>SUCCESS</status>
  <result>
    <remitter>
      <remitter_id>10018</remitter_id>
      <firstname>Joe</firstname>
      <lastname>Blogger</lastname>
      <verified>t</verified>
    </remitter>
  </result>
</response>
```

createBeneficiary

Group : beneficiary
Method : createBeneficiary

Input fields :

Name	Type	Notes
name *	Text	
fname	Text	
mname	Text	
lname	Text	
address1 *	Text	
address2	Text	
address3	Text	
city *	Text	
state	Text	
postcode	Text	
country *	Text	Country Name, as per output of getDestinationCountries
nationality	Text	Nationality Name, See Appendix A for the names
dob	Date	YYYY-MM-DD
gender	Text	Allowed values: UNSPECIFIED, MALE, FEMALE
telephone	Text	
mobile	Text	
email	Text	
fathers_name	Text	
mothers_name	Text	
national_id_number	Text	
id_type	Code	See Appendix C for codes
id_details	Text	
id_start	Date	When the document has been released, format yyyy-mm-dd or dd/mm/yyyy
id_expiry	Date	When the document is going to expire, format yyyy-mm-dd or dd/mm/yyyy
id_issued_by	Text	
id_issue_place	Text	
id_issue_country	Text	

id_scan	File	png, gif, jpg, jpeg, bmp and pdf
benef_employer_id_details	Text	
benef_taxpayer_reg	Text	
benef_occupation	Text	
card_type	Text	Allowed values are from the variable BENEFICIARY_CARD_TYPES from ARM
card_number	Text	The actual number if known, or REQUESTED if the bank needs to create a new card
account_number	Text	
bank	Text	
bank_branch	Text	
bank_branch_city	Text	
bank_branch_state	Text	
bank_branch_telephone	Text	
bank_branch_manager	Text	
benef_bank_swift_code	Text	
benef_bank_ifsc_code	Text	
benef_bank_account_name	Text	
benef_ac_type	Text	benef_ac_type retrieved from calling getBeneficiaryAccountTypes
homedelivery_notes	Text	
enabled	Text	t=true, f=false (By default if left blank, beneficiary will be enabled)
suspicious	Text	t=true, f=false (By default if left blank, beneficiary is NOT suspicious)
suspicion_reason	Text	
linked_member_id *	Number	The member ID this beneficiary is linked to

NOTE: If the Bank needs to create a new Card for the beneficiary, then specify the Card Number as 'REQUESTED'. This data will then be fed to the Bank dealing with that country for creation of the ATM Card.

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>SUCCESS</status>
```

```
<new_beneficiary_id>7</new_beneficiary_id>
</response>
```

searchBeneficiary

Group : beneficiary

Method : searchBeneficiary

Input fields :

Name	Type	Notes
beneficiary_id	Number	
name *	Text	
city	Text	
state	Text	
postcode	Text	
country *	Text	Required, as per output of getDestinationCountries
dob	Date	YYYY-MM-DD
telephone	Text	
mobile	Text	
id_details	Text	
account_number	Text	
card_number	Text	
enabled	Text	

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>SUCCESS</status>
  <result>
    <beneficiaries>
      <beneficiary>
        <beneficiary_id>4</beneficiary_id>
        <name>Jahangir Khan</name>
        <fname>Jahangir</fname>
        <mname></mname>
        <lname>Khan</lname>
        <address1>34 Small Lane</address1>
        <address2></address2>
        <address3></address3>
        <city>Mymensingh</city>
        <state></state>
```

```

    <postcode></postcode>
    <country>Bangladesh</country>
    <dob></dob>
    <fathers_name></fathers_name>
    <mothers_name></mothers_name>
    <national_id_number></national_id_number>
    <telephone>38254235</telephone>
    <mobile>7367365</mobile>
    <email>jk@work.com</email>
    <status>t</status>
    <id_type>NATIONALID</id_type>
    <id_details>35345345435</id_details>
    <account_number>4534534534534</account_number>
    <bank>Agrani Bank</bank>
    <bank_branch>Agrani Chittagong</bank_branch>
    <bank_branch_city>Chittagong</bank_branch_city>
    <bank_branch_state>Chittagong</bank_branch_state>
    <bank_branch_telephone>87787876</bank_branch_telephone>
    <bank_branch_manager>Mr Manager</bank_branch_manager>
    <benef_bank_swift_code>SW736435</benef_bank_swift_code>
    <benef_bank_ifsc_code>IFSC435</benef_bank_ifsc_code>
    <benef_bank_account_name>Savings AC</benef_bank_account_name>
    <card_number>345345435</card_number>
    <homedelivery_notes></homedelivery_notes>
    <suspicious>f</suspicious>
    <suspicion_reason></suspicion_reason>
  </beneficiary>
</beneficiaries>
</result>
</response>

```

updateBeneficiary

Group : beneficiary

Method : updateBeneficiary

Input fields :

Name	Type	Notes
beneficiary_id *	Number	
name	Text	
fname	Text	
mname	Text	
lname	Text	
address1	Text	
address2	Text	
address3	Text	
city	Text	

state	Text	
postcode	Text	
country	Text	As per output of getDestinationCountries
nationality	Text	Nationality Name, See Appendix A for the names
dob	Date	YYYY-MM-DD
fathers_name	Text	
mothers_name	Text	
national_id_number	Text	
gender	Text	Allowed values: UNSPECIFIED, MALE, FEMALE
telephone	Text	
mobile	Text	
email	Text	
id_type	Code	See Appendix C for codes
id_details	Text	
id_start	Date	When the document has been released, format yyyy-mm-dd or dd/mm/yyyy
id_expiry	Date	When the document is going to expire, format yyyy-mm-dd or dd/ mm/yyyy
id_issued_by	Text	
id_issue_place	Text	
id_issue_country	Text	
id_scan	File	png, gif, jpg, jpeg, bmp and pdf
card_type	Text	Allowed values are from the variable BENEFICIARY_CARD_TYPES from ARM
card_number	Text	
account_number	Text	
bank	Text	
bank_branch	Text	
bank_branch_city	Text	
bank_branch_state	Text	

bank_branch_telephone	Text	
bank_branch_manager	Text	
benef_bank_swift_code	Text	
benef_bank_ifsc_code	Text	
benef_bank_account_name	Text	
benef_ac_type	Text	benef_ac_types retrieved from calling getBeneficiaryAccountTypes
homedelivery_notes	Text	
enabled	Text	t=true, f=false
suspicious	Text	t=true, f=false
suspicion_reason	Text	

NOTE: All valid fields must be provided on update, otherwise existing fields may be blanked out.

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>SUCCESS</status>
  <result>
    <beneficiary>
      <beneficiary_id>26</beneficiary_id>
      <name></name>
      <fname></fname>
      <mname></mname>
      <lname></lname>
      <alias></alias>
      <type></type>
      <tel></tel>
      <mobile></mobile>
      <email></email>
      <gender></gender>
      <address1></address1>
      <address2></address2>
      <address3></address3>
      <city></city>
      <state></state>
      <postcode></postcode>
      <country_id></country_id>
      <nationality></nationality>
      <dob></dob>
      <fathers_name></fathers_name>
      <mothers_name></mothers_name>
      <national_id_number></national_id_number>
      <id_type></id_type>
      <id_details></id_details>
      <id_issued_by></id_issued_by>
      <id_issue_place></id_issue_place>
      <id_issue_country></id_issue_country>
      <id_start></id_start>
    </beneficiary>
  </result>
</response>
```

```

        <id_expiry></id_expiry>
        <linked_member_id></linked_member_id>
        <enabled></enabled>
        <suspicious></suspicious>
        <suspicious_reason></suspicious_reason>
    </beneficiary>
</result>
</response>

```

getPayoutTransactions

Group: **transaction**

Method: **getPayoutTransactions**

Get all transactions that have been assigned to receiving Bank/Agent for payout. Note: Calling this method does not assign the transactions to the Bank/Agent. The transaction(s) must be explicitly accepted by calling the method `acceptPayoutTransactions` for the transaction(s) to be accepted so that the Bank/Agent can payout the transaction.

Input fields:

Name	Type	Notes
bank_name *	Text	Abbreviated Bank name
is_bank_name_prefix	Boolean	t=true, f=false

Example Output XML :

```

<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>SUCCESS</status>
  <result>
    <count>1</count>
    <transactions>
      <transaction>
        <trans_ref>RA20079112</trans_ref>
        <status>HQ_OK</status>
        <creation_date>2010-05-07 16:25:12.26266</creation_date>
        <bank_branch_name></bank_branch_name>
        <bank_branch_code></bank_branch_code>
        <bank_accept_date>2010-05-07
16:25:12.26266</bank_accept_date>
        <bank_branch_accept_date></bank_branch_accept_date>
        <delivery_date>2010-05-14 00:00:00</delivery_date>
        <processed_date></processed_date>
        <processed_by></processed_by>
        <delivery_date>2010-05-14 00:00:00</delivery_date>
        <trans_type>Account</trans_type>
        <benef_name>
          <![CDATA[HASSAN SYFULLAH]]>
        </benef_name>
      </transaction>
    </transactions>
  </result>
</response>

```

```
<remitt_benef_relation><![CDATA[]]>
</remitt_benef_relation>
<benef_tel>
  <![CDATA[]]>
</benef_tel>
<benef_mobile>
  <![CDATA[]]>
</benef_mobile>
<benef_email>
  <![CDATA[]]>
</benef_email>
<collection_point>
  <![CDATA[BOGRA]]>
</collection_point>
<collection_point_code>
  <![CDATA[]]>
</collection_point_code>
<collection_pin>
  <![CDATA[]]>
</collection_pin>
<benef_id_type>
  <![CDATA[NONE]]>
</benef_id_type>
<benef_id_detail>
  <![CDATA[]]>
</benef_id_detail>
<benef_ac>
  <![CDATA[1234567890]]>
</benef_ac>
<benef_bank>
  <![CDATA[PRIME BANK LIMITED]]>
</benef_bank>
<benef_bank_swift_code>
  <![CDATA[ABC123]]>
</benef_bank_swift_code>
<benef_branch>
  <![CDATA[AGRABAD]]>
</benef_branch>
<benef_bank_city>
  <![CDATA[CHITTAGONG]]>
</benef_bank_city>
<benef_bank_state>
  <![CDATA[CHITTAGONG]]>
</benef_bank_state>
<benef_card_number>
  <![CDATA[]]>
</benef_card_number>
<benef_address1>
  <![CDATA[BOGRA]]>
</benef_address1>
<benef_address2>
  <![CDATA[]]>
</benef_address2>
<benef_address3>
  <![CDATA[]]>
</benef_address3>
<benef_city>
  <![CDATA[BOGRA]]>
</benef_city>
<benef_state>
  <![CDATA[]]>
```

```

</benef_state>
<benef_postcode>
  <![CDATA[]]>
</benef_postcode>
<benef_mobiletransfer_number>
  <![CDATA[01234567]]>
</benef_mobiletransfer_number>
<benef_mobiletransfer_network>
  <![CDATA[mobileNetwork]]>
</benef_mobiletransfer_network>
<delivery_notes>
  <![CDATA[]]>
</delivery_notes>
<utility_company><![CDATA[]]></utility_company>
<utility_account_no><![CDATA[]]></utility_account_no>
<utility_invoice><![CDATA[]]></utility_invoice>
<utility_bank><![CDATA[]]></utility_bank>
<utility_bank_code><![CDATA[]]></utility_bank_code>
<utility_description><![CDATA[]]></utility_description>
<remitter_name>
  <![CDATA[ALI IMAN]]>
</remitter_name>
<remitter_id>1210</remitter_id>
<remitter_address>
  <![CDATA[#123, 10 TEMASEK STREET, SINGAPORE,
SINGAPORE]]>
</remitter_address>
<remitter_city><![CDATA[]]></remitter_city>
<remitter_state><![CDATA[]]></remitter_state>
<remitter_type><![CDATA[]]></remitter_type>
<remitter_orgtype><![CDATA[]]></remitter_orgtype>
<remitter_gender><![CDATA[]]></remitter_gender>
<remitter_occupation><![CDATA[]]></remitter_occupation>
<remitter_id1_type><![CDATA[]]></remitter_id1_type>

<remitter_id1_details><![CDATA[]]></remitter_id1_details>

<remitter_id1_issue_place><![CDATA[]]></remitter_id1_issue_place>
  <remitter_id1_expiry><![CDATA[]]></remitter_id1_expiry>

<remitter_nationality><![CDATA[]]></remitter_nationality>
  <remitter_dob></remitter_dob>
  <remitter_place_of_birth></remitter_place_of_birth>
  <remitter_tel>
    <![CDATA[]]>
  </remitter_tel>
  <remitter_mobile>
    <![CDATA[]]>
  </remitter_mobile>
  <send_country>SG</send_country>
  <send_currency>SGD</send_currency>
  <send_amount>85.00</send_amount>
  <rate>49.2800</rate>
  <commission>5.00</commission>
  <fees>0</fees>
  <paid>90.00</paid>
  <receive_country>BD</receive_country>
  <receive_currency>BDT</receive_currency>
  <receive_amount>4188.80</receive_amount>
  <bank_sequence>82</bank_sequence>
  <pay_method></pay_method>

```

```

        <issue_date></issue_date>
        <bank_ref></bank_ref>
        <bank_comments></bank_comments>
        <bank_credit_date></bank_credit_date>
        <bank_clear_date></bank_clear_date>
        <deleted_reason></deleted_reason>
        <admin_comments><![CDATA[]]></admin_comments>
        <agent_comments><![CDATA[]]></agent_comments>
        <payout_comments><![CDATA[]]></payout_comments>
    </transaction>
</transactions>
</result>
</response>

```

getPayoutTransactionDetails

Group: **transaction**

Method: **getPayoutTransactionDetails**

Get details of a single transaction that has been allocated to Bank/Agent for payout.

Input fields:

Name	Type	Notes
bank_name *	Text	Abbreviated Bank name
trans_ref *	Text	

Example Output XML :

```

<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>SUCCESS</status>
  <result>
    <transaction>
      <trans_ref>RA20079127</trans_ref>
      <status>HQ_OK</status>
      <creation_date>2010-05-11 10:28:02.244553</creation_date>
      <purpose></purpose>
      <source_of_income></source_of_income>
      <bank_branch_name></bank_branch_name>
      <bank_branch_code></bank_branch_code>
      <bank_accept_date>2010-05-11
10:28:35.435216</bank_accept_date>
      <bank_branch_accept_date></bank_branch_accept_date>
      <delivery_date>2010-05-18 00:00:00</delivery_date>
      <processed_date></processed_date>
      <processed_by></processed_by>
      <delivery_date>2010-05-18 00:00:00</delivery_date>
      <trans_type>Account</trans_type>
      <benef_name>
        <![CDATA[ZIAD HASSAN SYFULLAH]]>
      </benef_name>
      <remitt_benef_relation><![CDATA[]]>
    </transaction>
  </result>
</response>

```

```
</remitt_benef_relation>
<benef_tel>
  <![CDATA[]]>
</benef_tel>
<benef_email>
  <![CDATA[]]>
</benef_email>
<collection_point>
  <![CDATA[BOGRA]]>
</collection_point>
<collection_point_code>
  <![CDATA[]]>
</collection_point_code>
<collection_pin>
  <![CDATA[]]>
</collection_pin>
<benef_id_type>
  <![CDATA[NONE]]>
</benef_id_type>
<benef_id_detail>
  <![CDATA[]]>
</benef_id_detail>
<benef_ac>
  <![CDATA[1234567890]]>
</benef_ac>
<benef_bank>
  <![CDATA[PRIME BANK LIMITED]]>
</benef_bank>
<benef_bank_swift_code>
  <![CDATA[ABC123]]>
</benef_bank_swift_code>
<benef_bank_iban></benef_bank_iban>
<benef_bank_bsb_code><![CDATA[ABC123]]></benef_bank_bsb_code>
```

```
<benef_bank_routing_transit_number><![CDATA[ABC123]]></benef_bank_routing_transi
t_number>
```

```
  <benef_bank_ifsc_code></benef_bank_ifsc_code>
```

```
  <benef_bank_account_name><![CDATA[ABC123]]></benef_bank_account_name>
```

```
  <benef_branch>
    <![CDATA[AGRABAD]]>
  </benef_branch>
```

```
  <benef_bank_city>
    <![CDATA[CHITTAGONG]]>
  </benef_bank_city>
```

```
  <benef_bank_state>
    <![CDATA[CHITTAGONG]]>
  </benef_bank_state>
```

```
  <benef_card_number>
    <![CDATA[]]>
  </benef_card_number>
```

```
  <benef_address1>
    <![CDATA[BOGRA]]>
  </benef_address1>
```

```
  <benef_address2>
    <![CDATA[]]>
  </benef_address2>
```

```
  <benef_address3>
    <![CDATA[]]>
  </benef_address3>
```

```

<benef_city>
  <![CDATA[BOGRA]]>
</benef_city>
<benef_state>
  <![CDATA[]]>
</benef_state>
<benef_postcode>
  <![CDATA[]]>
</benef_postcode>
<benef_mobiletransfer_number>
  <![CDATA[01234567]]>
</benef_mobiletransfer_number>
<benef_mobiletransfer_network>
  <![CDATA[mobileNetwork]]>
</benef_mobiletransfer_network>
<benef_mobiletransfer_network_credit_type>
  <![CDATA[]]>
</benef_mobiletransfer_network_credit_type>
<benef_mobiletransfer_network_credit_type_code>
  <![CDATA[]]>
</benef_mobiletransfer_network_credit_type_code>
<delivery_notes>
  <![CDATA[]]>
</delivery_notes>
<utility_company><![CDATA[]]></utility_company>
<utility_account_no><![CDATA[]]></utility_account_no>
<utility_invoice><![CDATA[]]></utility_invoice>
<utility_bank><![CDATA[]]></utility_bank>
<utility_bank_code><![CDATA[]]></utility_bank_code>
<utility_description><![CDATA[]]></utility_description>
<remitter_name>
  <![CDATA[ALI IMAN]]>
</remitter_name>
<remitter_id>1210</remitter_id>
<remitter_address>
  <![CDATA[#123, 10 TEMASEK STREET, SINGAPORE,
SINGAPORE]]>
</remitter_address>
<remitter_city><![CDATA[]]></remitter_city>
<remitter_state><![CDATA[]]></remitter_state>
<remitter_type><![CDATA[]]></remitter_type>
<remitter_orgtype><![CDATA[]]></remitter_orgtype>
<remitter_gender><![CDATA[]]></remitter_gender>
<remitter_occupation><![CDATA[]]></remitter_occupation>
<remitter_id1_type><![CDATA[]]></remitter_id1_type>
<remitter_id1_details><![CDATA[]]></remitter_id1_details>
<remitter_id1_issue_place><![CDATA[]]></remitter_id1_issue_place>
<remitter_id1_expiry><![CDATA[]]></remitter_id1_expiry>
<remitter_nationality><![CDATA[]]></remitter_nationality>
<remitter_dob></remitter_dob>
<remitter_tel>
  <![CDATA[]]>
</remitter_tel>
<remitter_mobile>
  <![CDATA[]]>
</remitter_mobile>
<send_country>SG</send_country>
<send_currency>SGD</send_currency>
<send_amount>157.57</send_amount>
<rate>49.2800</rate>

```



```

    <commission>5.00</commission>
    <fees>0</fees>
    <paid>162.57</paid>
    <receive_country>BD</receive_country>
    <receive_currency>BDT</receive_currency>
    <receive_amount>7765.05</receive_amount>
    <bank_sequence>2929</bank_sequence>
    <pay_method></pay_method>
    <issue_date></issue_date>
    <bank_ref></bank_ref>
    <benef_trans_ref>GHT33762634</benef_trans_ref>
    <bank_comments></bank_comments>
    <bank_credit_date></bank_credit_date>
    <bank_clear_date></bank_clear_date>
    <deleted_reason></deleted_reason>
    <admin_comments><![CDATA[]]></admin_comments>
    <agent_comments><![CDATA[]]></agent_comments>
    <payout_comments><![CDATA[]]></payout_comments>
  </transaction>
</result>
</response>

```

acceptPayoutTransactions

Group: **transaction**

Method: **acceptPayoutTransactions**

Accept transaction(s) so that the Bank/Agent can pay them out to beneficiaries.

Input fields:

Name	Type	Notes
bank_name *	Text	Abbreviated Bank name
trans_ref *	Text	Multiple transactions can be accepted in one call by providing the transaction references of all the transactions separated by commas.

Example Output XML :

```

<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>SUCCESS</status>
  <result>
    <total_count>2</total_count>
    <success_count>2</success_count>
    <failed_count>0</failed_count>
  </result>
</response>

```

```

    <transactions>
      <transaction>
        <trans_ref>RA20079112</trans_ref>
        <operation_result>SUCCESS</operation_result>
        <message></message>
      </transaction>
      <transaction>
        <trans_ref>RA20079121</trans_ref>
        <operation_result>SUCCESS</operation_result>
        <message></message>
      </transaction>
    </transactions>
  </result>
</response>

```

getPendingPayoutTransactions

Group: **transaction**

Method: **getPendingPayoutTransactions**

Get all transactions that are to be paid to beneficiaries by the Bank/Agent. So these are transactions that were allocated to the Bank/Agent and the Bank/Agent explicitly accepted them.

Input fields:

Name	Type	Notes
bank_name *	Text	Abbreviated Bank name

Example Output XML :

```

<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>SUCCESS</status>
  <result>
    <count>1</count>
    <transactions>
      <transaction>
        <trans_ref>RT20278150</trans_ref>
        <status>SENT_FOR_DELIVERY</status>
        <creation_date>2010-03-21
17:09:13.622815</creation_date>
        <purpose></purpose>
        <source_of_income></source_of_income>
        <bank_branch_name>DAULATPUR</bank_branch_name>
        <bank_branch_code>47-1170</bank_branch_code>
        <bank_accept_date>2010-03-21
17:39:48.760175</bank_accept_date>
        <bank_branch_accept_date></bank_branch_accept_date>

```

```
<delivery_date>2010-03-21 00:00:00</delivery_date>
<processed_date></processed_date>
<processed_by></processed_by>
<delivery_date>2010-03-21 00:00:00</delivery_date>
<trans_type>Cash Collection</trans_type>
<benef_name>
  <![CDATA[MD SAIFUL ISLAM]]>
</benef_name>
<remitt_benef_relation><![CDATA[]]>
</remitt_benef_relation>
<benef_tel>
  <![CDATA[08215112686]]>
</benef_tel>
<benef_email>
  <![CDATA[]]>
</benef_email>
<collection_point>
  <![CDATA[DAULATPUR, PUBALI]]>
</collection_point>
<collection_point_code>
  <![CDATA[47-1170]]>
</collection_point_code>
<collection_pin>
  <![CDATA[12345678]]>
</collection_pin>
<benef_id_type>
  <![CDATA[NATIONAL_ID]]>
</benef_id_type>
<benef_id_detail>
  <![CDATA[]]>
</benef_id_detail>
<benef_ac>
  <![CDATA[]]>
</benef_ac>
<benef_bank>
  <![CDATA[PUBALI BANK LIMITED]]>
</benef_bank>
<benef_bank_swift_code>
  <![CDATA[ABC123]]>
</benef_bank_swift_code>
<benef_bank_iban></benef_bank_iban>
<benef_bank_bsb_code>
  <![CDATA[ABC123]]>
</benef_bank_bsb_code>
<benef_bank_routing_transit_number>
  <![CDATA[ABC123]]>
</benef_bank_routing_transit_number>
<benef_bank_ifsc_code></benef_bank_ifsc_code>
<benef_bank_account_name>
  <![CDATA[ABC123]]>
</benef_bank_account_name>
<benef_ac_type>
  <![CDATA[INDIVIDUAL]]>
</benef_ac_type>
<benef_branch>
  <![CDATA[DAULATPUR]]>
</benef_branch>
<benef_bank_city>
  <![CDATA[MANIKGONJ]]>
</benef_bank_city>
<benef_bank_state>
```

```

        <![CDATA[MANIKGONJ]]>
    </benef_bank_state>
    <benef_card_number>
        <![CDATA[]]>
    </benef_card_number>
    <benef_address1>
        <![CDATA[MANIKGONJ]]>
    </benef_address1>
    <benef_address2>
        <![CDATA[]]>
    </benef_address2>
    <benef_address3>
        <![CDATA[]]>
    </benef_address3>
    <benef_city>
        <![CDATA[MANIKGONJ]]>
    </benef_city>
    <benef_state>
        <![CDATA[]]>
    </benef_state>
    <benef_postcode>
        <![CDATA[]]>
    </benef_postcode>
    <benef_mobiletransfer_number>
        <![CDATA[01234567]]>
    </benef_mobiletransfer_number>
    <benef_mobiletransfer_network>
        <![CDATA[mobileNetwork]]>
    </benef_mobiletransfer_network>
    <delivery_notes>
        <![CDATA[]]>
    </delivery_notes>
<utility_company><![CDATA[]]></utility_company>
    <utility_account_no><![CDATA[]]></utility_account_no>
    <utility_invoice><![CDATA[]]></utility_invoice>
    <utility_bank><![CDATA[]]></utility_bank>
    <utility_bank_code><![CDATA[]]></utility_bank_code>
    <utility_description><![CDATA[]]></utility_description>
    <remitter_name>
        <![CDATA[ALI IMAN]]>
    </remitter_name>
    <remitter_id>1210</remitter_id>
    <remitter_address>
        <![CDATA[#123, 10 TEMASEK STREET, SINGAPORE,
SINGAPORE]]>
    </remitter_address>
    <remitter_city><![CDATA[]]></remitter_city>
    <remitter_state><![CDATA[]]></remitter_state>
    <remitter_type><![CDATA[]]></remitter_type>
    <remitter_orgtype><![CDATA[]]></remitter_orgtype>
    <remitter_gender><![CDATA[]]></remitter_gender>
    <remitter_occupation><![CDATA[]]></remitter_occupation>
    <remitter_id1_type><![CDATA[]]></remitter_id1_type>

    <remitter_id1_details><![CDATA[]]></remitter_id1_details>

    <remitter_id1_issue_place><![CDATA[]]></remitter_id1_issue_place>
        <remitter_id1_expiry><![CDATA[]]></remitter_id1_expiry>

    <remitter_nationality><![CDATA[]]></remitter_nationality>
        <remitter_dob></remitter_dob>

```

```

    <remitter_tel>
      <![CDATA[98012322]]>
    </remitter_tel>
    <remitter_mobile>
      <![CDATA[]]>
    </remitter_mobile>
    <send_country>SG</send_country>
    <send_currency>SGD</send_currency>
    <send_amount>345.00</send_amount>
    <rate>49.4000</rate>
    <commission>5.00</commission>
    <fees>0</fees>
    <paid>350.00</paid>
    <receive_country>BD</receive_country>
    <receive_currency>BDT</receive_currency>
    <receive_amount>17043.00</receive_amount>
    <bank_sequence>65</bank_sequence>
    <pay_method></pay_method>
    <issue_date></issue_date>
    <bank_ref></bank_ref>
    <bank_comments></bank_comments>
    <bank_credit_date></bank_credit_date>
    <bank_clear_date></bank_clear_date>
    <deleted_reason></deleted_reason>
    <admin_comments><![CDATA[]]></admin_comments>
    <agent_comments><![CDATA[]]></agent_comments>
    <payout_comments><![CDATA[]]></payout_comments>
  </transaction>
</transactions>
</result>
</response>

```

processPayoutTransaction

Group: transaction

Method: processPayoutTransaction

Inform the RemitONE system that a transaction that was accepted by the Bank/Agent for payout has paid out and should be marked as processed with the details provided.

Input fields:

Name	Type	Notes
bank_name *	Text	Abbreviated Bank name
trans_ref *	Text	
pay_method *	Text	How the transaction was paid out.
bank_credit_date	Date	YYYY-MM-DD
bank_clear_date	Date	YYYY-MM-DD

po_clear_date	Date	YYYY-MM-DD
benef_id_type	Text	The type of ID the beneficiary provided at payout (if applicable)
benef_id_details	Text	The details of the ID provided at payout.
issue_date	Date	YYYY-MM-DD
bank_ref	Text	
bank_comments	Text	
sms_benef_payout	Boolean	t=true, f=false true if the beneficiary has to be notified by sms when the transaction has been processed
sms_benef_payout_mobile	Text	The mobile number where to notify the beneficiary by sms

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>SUCCESS</status>
  <result>
    <trans_ref>RA20079121</trans_ref>
    <sms_benef_payout>True</sms_benef_payout>
    <sms_benef_payout_mobile>+447767123456</sms_benef_payout_mobile>
    <process_result>SUCCESS</process_result>
    <message></message>
  </result>
</response>
```

errorPayoutTransaction

Group: **transaction**

Method: **errorPayoutTransaction**

Inform the RemitONE system that there is something wrong with this transaction that prevents the Bank/Agent being able to payout the transaction so it should be marked as ERROR.

Input fields:

Name	Type	Notes
bank_name *	Text	Abbreviated Bank name

trans_ref *	Text	
error_reason *	Text	The overall reason for error.
error_details	Text	Details of the error.

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>SUCCESS</status>
  <result>
    <trans_ref>RA20079084</trans_ref>
    <error_result>SUCCESS</error_result>
  </result>
</response>
```

getRemitterWhitelist

Group : remitter

Method : **getRemitterWhitelist**

This method can be used to check if the remitter has been whitelisted.

Input fields:

Name	Type	Notes
remitter_id *	Integer	The member ID of remitter

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>105</responseId>
  <status>SUCCESS</status>
  <result>
    <remitter_id>10001</remitter_id>
    <compliance_whitelist>>true</compliance_whitelist>
  </result>
</response>
```

setRemitterWhitelist

Group : remitter

Method : **setRemitterWhitelist**

This method can be used to set/unset a remitter as whitelisted.

Input fields:

Name	Type	Notes
remitter_id *	Integer	The member ID of remitter
whitelist	Boolean	Set or unset a remitter as whitelisted.

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>105</responseId>
  <status>SUCCESS</status>
  <result>
    <remitter_id>10001</remitter_id>
    <compliance_whitelist>true</compliance_whitelist>
  </result>
</response>
```

getBeneficiaryWhitelist

Group : **beneficiary**

Method : **getBeneficiaryWhitelist**

This method can be used to check if the beneficiary has been whitelisted.

Input fields:

Name	Type	Notes
beneficiary_id *	Integer	The benef ID of beneficiary

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>105</responseId>
  <status>SUCCESS</status>
  <result>
    <beneficiary_id>300</beneficiary_id>
    <compliance_whitelist>true</compliance_whitelist>
  </result>
</response>
```

setBeneficiaryWhitelist

Group : **beneficiary**

Method : **setBeneficiaryWhitelist**

This method can be used to set/unset a beneficiary as whitelisted.

Input fields:

Name	Type	Notes
beneficiary_id *	Integer	The benef ID of beneficiary
whitelist	Boolean	Set or unset a beneficiary as whitelisted.

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>105</responseId>
  <status>SUCCESS</status>
  <result>
    <beneficiary_id>300</beneficiary_id>
    <compliance_whitelist>true</compliance_whitelist>
  </result>
</response>
```

transactionPaymentCleared

Group : **transaction**

Method : **transactionPaymentCleared**

This method can be used to mark a PENDING_CLEARANCE transaction as payment cleared.

Input fields:

Name	Type	Notes
trans_ref *	Text	Transaction reference number
update_rate	Boolean	t or f

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>184</responseId>
  <status>SUCCESS</status>
  <result>
```

```

    <trans_ref>RA000503000269</trans_ref>
    <message>Transaction payment cleared successfully</message>
  </result>
</response>

```

createPromotionRule

Group : **admin**

Method : **createPromotionRule**

This method can be used to create a promotion rule.

Input fields:

Name	Type	Notes
name	Text	Name of promotion
enabled	Boolean	t or f
from_date	Date	YYYY-MM-DD
to_date	Date	YYYY-MM-DD
remitter_id	Text	Remitter ID (remitter specific promotion code)
source_country	Text	The name of the Source Country
destination_country	Text	The name of the Destination Country
agency	Text	The Agency ID (which agency the code can be used for)
agent	Text	The Agent ID (which agent the code can be used for)
hq_agents	Boolean	Leave empty for 'All'. 't' for HQ-Agents only and 'f' for Non-HQ Agents only.
transfer_type	Text	Which transaction type the code is available for. Leave empty for all. See Appendix I for more details.
from_trans_no	Integer	
to_trans_no	Integer	
period_type	Text	How long a period should be
period	Integer	Period Number
single_promotion	Boolean	One time promotion

creation_channel	Text	Which application the promotion applies to. See Appendix J.
discount_code	Text	What the code should be called
discount_type	Text	Whether it is a PERCENTAGE promotion or FIXED promotion
discount_value	Numeric(23,2)	Value of discount
description	Text	Anymore details of promotion

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>SUCCESS</status>
</response>
```

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>INVALID CREDENTIALS</status>
</response>
```

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>EXCEPTION ERROR</status>
</response>
```

nameScreenCheck

Group : Admin

Method : nameScreenCheck

This method can be used to check a name against the blacklist.

Input fields:

Name	Type	Notes
name	Text	Name of person to check. Must provide 2 names.
dob	Date	Date of Birth. Optional.

Example Output XML :

```
<response>
  <responseid>356</responseid>
  <status>SUCCESS</status>
  <search_name>Barack Obama</search_name>
  <search_dob></search_dob>
  <response>
    <compliance_results>
      <entity>
        <ent_num>every_politician de09bde1-80a2-48ee-b03a-76ca71a0bb0d</ent_num>
        <name>
<!--[CDATA[barack obama]]-->
          </name>
          <address>
<!--[CDATA[]]-->
          </address>
          <city>
<!--[CDATA[]]-->
          </city>
          <country>
<!--[CDATA[]]-->
          </country>
          <dob>
<!--[CDATA[]]-->
          </dob>
          <place_of_birth>
<!--[CDATA[]]-->
          </place_of_birth>
          <remarks>
<!--[CDATA[]]-->
          </remarks>
          <aliases>
<!--[CDATA[]]-->
          </aliases>
          <nationality>
<!--[CDATA[United States of America]]-->
          </nationality>
          <type>
<!--[CDATA[individual]]-->
          </type>
        </entity>
      </compliance_results>
    </response>
  </response>
```

```

        <programme>
<!--[CDATA[Senate]]-->
        </programme>
        <matchcount>2</matchcount>
        <relevance>10</relevance>
    </entity>
</compliance_results>
</response>
</response>

```

updateBuyRates

Group : rates

Method : updateBuyRates

Input fields :

Name	Type	Notes
username	Text	The username given to authenticate
password	Text	The password given for the user
PIN	Number	The 5-digits number given to authenticate along with username and password
bank_name	Text	Optional. If the user has type WS_Admin then the bank is required, otherwise is automatically detected from the user's bank.
buy_rate	Number	The new buy rate to set for the bank; it can accept till 8 decimal digits

Example Output XML :

```

<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>326</responseId>
  <status>SUCCESS</status>
</response>

```

```

<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>326</responseId>
  <status>FAIL</status>
  <result>
    <message>VALIDATION FAILED</message>
    <errors><error>Processing Bank is required</error></errors>
  </result>
</response>

```

createDeliveryBank

Group : banks

Method : createDeliveryBank

Input fields :

Name	Type	Notes
username	Text	The username given to authenticate
password	Text	The password given for the user
PIN	Number	The 5-digits number given to authenticate along with username and password
name	Text	The name to give to the delivery bank.
bank_code	Text	Optional. If the user has type WS_Admin then the bank_code is required, otherwise is automatically detected from the user's bank.
country_id	Number	Needed to check if the delivery bank exists. If the user has type WS_Bank the field will be detected from the user's information.
address	Text	Optional. The address of the delivery bank
city	Text	Optional. The city of the delivery bank
state	Text	Optional. The country of the delivery bank
telephone	Text	Optional. The phone number of the delivery bank
swift_code	Text	Optional. The swift code of the delivery bank
Enabled	Boolean	Default Yes

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>62</responseId>
  <status>SUCCESS</status>
  <result>
    <id>12</id>
    <name>My Test Bank</name>
    <bank_code>BL</bank_code>
    <country_id>999</country_id>
    <address>Piazza Grande, 12</address>
    <city>Lonigo</city>
    <state>Italy</state>
    <telephone>0444123456</telephone>
    <swift_code></swift_code>
    <enabled>t</enabled>
  </result>
</response>
```

```

<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>63</responseId>
  <status>FAIL</status>
  <result>
    <message>Bank Code is missing</message>
  </result>
</response>

<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>64</responseId>
  <status>FAIL</status>
  <result>
    <message>Delivery Bank already exists</message>
  </result>
</response>

```

updateDeliveryBank

Group : **banks**

Method : **updateDeliveryBank**

Input fields :

Name	Type	Notes
username	Text	The username given to authenticate
password	Text	The password given for the user
PIN	Number	The 5-digits number given to authenticate along with username and password
name	Text	The name to give to the delivery bank.
bank_code	Text	Optional. If the user has type WS_Admin then the bank_code is required, otherwise is automatically detected from the user's bank.
country_id	Number	Needed to check if the delivery bank exists. If the user has type WS_Bank the field will be detected from the user's information.
address	Text	Optional. The address of the delivery bank
city	Text	Optional. The city of the delivery bank
state	Text	Optional. The country of the delivery bank
telephone	Text	Optional. The phone number of the delivery

		bank
swift_code	Text	Optional. The swift code of the delivery bank
Enabled	Boolean	Default Yes

Example Output XML :

```

<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>65</responseId>
  <status>SUCCESS</status>
  <result>
    <id>12</id>
    <name>My Test Bank</name>
    <bank_code>BL</bank_code>
    <country_id>999</country_id>
    <address>Piazza Grande, 12</address>
    <city>Lonigo</city>
    <state>Italy</state>
    <telephone>0444123456</telephone>
    <swift_code></swift_code>
    <enabled>t</enabled>
  </result>
</response>

<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>66</responseId>
  <status>FAIL</status>
  <result>
    <message>Bank Code is missing</message>
  </result>
</response>

<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>67</responseId>
  <status>FAIL</status>
  <result>
    <message>Delivery Bank not found</message>
  </result>
</response>

```

getDeliveryBanks

Group : banks

Method : getDeliveryBanks

Input fields :

Name	Type	Notes
username	Text	The username given to authenticate
password	Text	The password given for the user
PIN	Number	The 5-digits number given to authenticate along with username and password
name	Text	The name to give to the delivery bank.
dest_country	Text	Optional. The country where to search the banks in. If the user has type WS_Bank, the country will be detected in the user's data. If the user has type WS_Admin it is necessary to insert at least dest_country or country_code.
country_code	Number	Optional. The code of the country. If the user has type WS_Admin it is necessary to insert at least dest_country or country_code.
bank_code	Text	Optional. If the user has type WS_Admin and then code is not given, then all the delivery banks for that country will be returned, otherwise the search will be restricted to the bank_code in that country.

Example Output XML :

```

<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>70</responseId>
  <status>SUCCESS</status>
  <result>
    <delivery_banks>
      <delivery_bank>
        <bank_id>1</bank_id>
        <name>Banca di Lonigo</name>
        <country_id>999</country_id>
        <bank_code>BL</bank_code>
        <swift_code></swift_code>
        <address>Piazza dei Signori, 22</address>
        <city>Lonigo</city>
        <state>Vicenza</state>
        <telephone>+39444747485</telephone>
        <account_number_mask></account_number_mask>
        <enabled>t</enabled>
      </delivery_bank>
      <delivery_bank>
        <bank_id>12</bank_id>
        <name>My Test Bank</name>
        <country_id>999</country_id>
        <bank_code>BL</bank_code>
        <swift_code></swift_code>
        <address>Piazza Grande, 12</address>
        <city>Lonigo</city>
        <state>Italy</state>
      </delivery_bank>
    </delivery_banks>
  </result>
</response>

```

```

        <telephone>0444123456</telephone>
        <account_number_mask></account_number_mask>
        <enabled>t</enabled>
    </delivery_bank>
</delivery_banks>
</result>
</response>

<?xml version="1.0" encoding="utf-8"?>
<response>
    <responseId>66</responseId>
    <status>FAIL</status>
    <result>
        <message>Destination Country and Country Code are both
missing</message>
    </result>
</response>

```

createUtilityCompany

Group : utilityCompany

Method : createUtilityCompany

Input fields :

Name	Type	Notes
username	Text	The username given to authenticate
password	Text	The password given for the user
PIN	Number	The 5-digits number given to authenticate along with username and password
company_name	Text	The name to give to the utility company.
company_code	Text	The code to give to the utility company.
address1	Text	Optional. First part of the address of the utility company
address2	Text	Optional. Second part of the address of the utility company.
address3	Text	Optional. Third part of the address of the utility company.
city	Text	Optional. The city of the utility company
state	Text	Optional. The country of the utility company
postcode	Text	Optional. The postcode of the utility company

bank_account_no	Text	The bank account number of the utility company
iban_no	Text	Optional. The IBAN number of the utility company
bic_no	Text	Optional. The BIC number of the utility company
enabled	Boolean	Default Yes
country_id	Number	Optional. The ID of the country. If the user has type WS_Admin it is required to provide at least country_id or dest_country field.
dest_country	Text	Optional. The name of the country. If the user has type WS_Admin it is required to provide at least country_id or dest_country field.

Example Output XML :

```

<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>84</responseId>
  <status>SUCCESS</status>
  <result>
    <utility_company_id>2</utility_company_id>
    <company_name>My Second Utility Company</company_name>
    <company_code>MSUC</company_code>
    <address1>Viale Trieste, 32</address1>
    <address2></address2>
    <address3></address3>
    <city>Lonigo</city>
    <state>Italy</state>
    <postcode>36045</postcode>
    <bank_account_no>000000123456</bank_account_no>
    <iban_no>IT60 X054 2811 010 0000 0123 456</iban_no>
    <bic_no>AAGSIT21XXX</bic_no>
    <country_id>999</country_id>
    <enabled>t</enabled>
  </result>
</response>

<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>85</responseId>
  <status>FAIL</status>
  <result>
    <message>No Bank Account Number has been given</message>
  </result>
</response>

<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>85</responseId>
  <status>FAIL</status>
  <result>

```

```

        <message>Utility Company Name must be unique.</message>
    </result>
</response>

<?xml version="1.0" encoding="utf-8"?>
<response>
    <responseId>85</responseId>
    <status>FAIL</status>
    <result>
        <message>Cannot add another Utility Company with the same pair
(Bank Account Number, IBAN)</message>
    </result>
</response>

```

updateUtilityCompany

Group : utilityCompany

Method : updateUtilityCompany

Input fields :

Name	Type	Notes
username	Text	The username given to authenticate
password	Text	The password given for the user
PIN	Number	The 5-digits number given to authenticate along with username and password
company_name	Text	The name to give to the utility company.
company_code	Text	The code to give to the utility company.
address1	Text	Optional. First part of the address of the utility company
address2	Text	Optional. Second part of the address of the utility company.
address3	Text	Optional. Third part of the address of the utility company.
city	Text	Optional. The city of the utility company
state	Text	Optional. The country of the utility company
postcode	Text	Optional. The postcode of the utility company
bank_account_no	Text	The bank account number of the utility company

iban_no	Text	Optional. The IBAN number of the utility company
bic_no	Text	Optional. The BIC number of the utility company
enabled	Boolean	Default Yes
country_id	Number	Optional. The ID of the country. If the user has type WS_Admin it is required to provide at least country_id or dest_country field.
dest_country	Text	Optional. The name of the country. If the user has type WS_Admin it is required to provide at least country_id or dest_country field.

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>84</responseId>
  <status>SUCCESS</status>
  <result>
    <utility_company_id>2</utility_company_id>
    <company_name>My Second Utility Company</company_name>
    <company_code>MSUC</company_code>
    <address1>Viale Trieste, 32</address1>
    <address2></address2>
    <address3></address3>
    <city>Lonigo</city>
    <state>Italy</state>
    <postcode>36045</postcode>
    <bank_account_no>000000123456</bank_account_no>
    <iban_no>IT60 X054 2811 010 0000 0123 456</iban_no>
    <bic_no>AAGSIT21XXX</bic_no>
    <country_id>999</country_id>
    <enabled>t</enabled>
  </result>
</response>
```

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>85</responseId>
  <status>FAIL</status>
  <result>
    <message>No Bank Account Number has been given</message>
  </result>
</response>
```

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>85</responseId>
  <status>FAIL</status>
  <result>
    <message>Cannot update the Utility Company as there is another
one with the same pair (Bank Account Number, IBAN)</message>
  </result>
```

</response>

getUtilityCompanies

Group : utilityCompany

Method : getUtilityCompanies

Input fields :

Name	Type	Notes
username	Text	The username given to authenticate
password	Text	The password given for the user
PIN	Number	The 5-digits number given to authenticate along with username and password
country_id	Number	Optional. The ID of the country where to find utility companies in. If it is empty and the destination country is empty, it will search for all the utility companies. If the user has type WS_Bank, the country_id will be extracted from the user information, so this field can be empty.
dest_country	Text	Optional. The name of the country where to find utility companies in. If it is empty and the country_id is empty, it will search for all the utility companies. If the user has type WS_Bank, the country_id will be extracted from the user information, so this field can be empty.
country_code	Text	Optional. The ISO code for the Destination Country.

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>95</responseId>
  <status>SUCCESS</status>
  <result>
    <utility_companies>
      <utility_company>
        <utility_company_id>1</utility_company_id>
```

```

    <company_name>My First Utility Company</company_name>
    <company_code>MFUC</company_code>
    <address1>Viale Venezia, 22</address1>
    <address2></address2>
    <address3></address3>
    <city>Lonigo</city>
    <state>Italy</state>
    <postcode>36045</postcode>
    <bank_account_no>5454651205656556</bank_account_no>
    <iban_no>IT60 X054 2811 1010 0000 0123 456</iban_no>
    <bic_no>AAGSIT21XXX</bic_no>
    <country_id>999</country_id>
    <enabled>t</enabled>
  </utility_company>
  <utility_company>
    <utility_company_id>2</utility_company_id>
    <company_name>My Second Utility Company</company_name>
    <company_code>MSUC</company_code>
    <address1>Viale Trieste, 32</address1>
    <address2></address2>
    <address3></address3>
    <city>Lonigo</city>
    <state>Italy</state>
    <postcode>36045</postcode>
    <bank_account_no>000000123456</bank_account_no>
    <iban_no>IT60 X054 2811 010 0000 0123 456</iban_no>
    <bic_no>AAGSIT21XXX</bic_no>
    <country_id>999</country_id>
    <enabled>t</enabled>
  </utility_company>
</utility_companies>
</result>
</response>

```

updateTransactionDetails

Group : transaction

Method : updateTransactionDetails

Input fields :

Name	Type	Notes
username	Text	The username given to authenticate
password	Text	The password given for the user
PIN	Number	The 5-digits number given to authenticate along with username and password
trans_ref	Text	The transaction reference number of the transaction you wish to add comments to
admin_comments	Text	Optional. The Admin Comments which you wish to add to the transaction.

agent_comments	Text	Optional. The Agent Comments which you wish to add to the transaction
payout_comments	Text	Optional. The Payout Comments which you wish to add to the transaction
comments_to_benef	Text	Optional. The Comments to the Beneficiary which you wish to add to the transaction
bank_comments	Text	Optional. The Bank Comments which you wish to add to the transaction
benef_trans_ref	Text	Optional. The payout system transaction reference number.

```
<?xml version="1.0" encoding="utf-8"?>
  <response>
    <responseId>85</responseId>
    <status>SUCCESS</status>
    <result>
      <trans_id>1</trans_id>
      <trans_ref>RC000224000100</trans_ref>
      <admin_comments>Some admin comments</admin_comments>
      <agent_comments>Some agent comments</agent_comments>
      <payout_comments>Some payout comments</payout_comments>
      <comments_to_benef>Some comments to
beneficiary</comments_to_benef>
      <bank_comments>Some bank comments</bank_comments>
      <benef_trans_ref>XM234001212</benef_trans_ref>
    </result>
  </response>
```

checkCredentials

Group : **admin**

Method : **checkCredentials**

Input fields :

Name	Type	Notes
username	Text	The username given to authenticate
password	Text	The password given for the user
PIN	Number	The 5-digits number given to authenticate along with username and password
check_username	Text	The system username whose password needs to be checked
check_password	Text	The password of the user to be checked

```
<?xml version="1.0" encoding="utf-8"?>
```



```
<response>
  <responseId>150</responseId>
  <status></status>
  <result>
    <valid_credentials>true</valid_credentials>
    <username>ukfasttransfer</username>
    <user_type>Agent</user_type>
  </result>
</response>
```

Web Services Functionality

It is possible to test the output of the WebServices if the system is in debugging mode. The URL for accessing forms for each of the WebServices is :

SYSTEM_URL/wstest/index.html

For example, <https://www.remitone.com/wstest/index.html>

You can enter values for each of the WebService Forms and the result will be presented back as neatly formatted HTML.

If the system is in live mode, then the result will be returned as pure XML, and hence you can still use the 'View Source' option in the Web Browser to see the XML that is returned.

Appendix A : Destination Country ISO Codes

Name	ISO code	Currency
Afghanistan	AF	AFA
Albania	AL	ALL
Algeria	DZ	DZD
American Samoa	AS	
Andorra	AD	ADP
Angola	AO	AON
Anguilla	AI	
Antarctica	AQ	
Antigua and Barbuda	AG	
Argentina	AR	ARS
Armenia	AM	AMD
Aruba	AW	AWG
Australia	AU	AUD
Austria	AT	EUR
Azerbaijan	AZ	AZN
Bahamas	BS	BSD
Bahrain	BH	BHD
Bangladesh	BD	BDT
Barbados	BB	BBD
Belarus	BY	BYR
Belgium	BE	EUR
Belize	BZ	BZD
Benin	BJ	
Bermuda	BM	BMD
Bhutan	BT	BTN
Bolivia	BO	BOB
Bosnia and Herzegovina	BA	BAM
Botswana	BW	BWP
Bouvet Island	BV	
Brazil	BR	BRE
British Indian Ocean Territory	IO	
Brunei Darussalam	BN	BND
Bulgaria	BG	BGL

Burkina Faso	BF	
Burundi	BI	BIF
Cambodia	KH	KHR
Cameroon	CM	XAF
Canada	CA	CAD
Cape Verde	CV	CVE
Cayman Islands	KY	KYD
Central African Republic	CF	XAF
Chad	TD	XAF
Chile	CL	CLF
China	CN	CNY
Christmas Island	CX	
Cocos (Keeling) Islands	CC	
Colombia	CO	COP
Comoros	KM	
Congo	CG	
Congo, the Democratic Republic of the	CD	
Cook Islands	CK	
Costa Rica	CR	CRC
Cote D_Ivoire	CI	
Croatia	HR	HRK
Cuba	CU	CUP
Cyprus	CY	CYP
Czech Republic	CZ	CZK
Denmark	DK	DKK
Djibouti	DJ	DJF
Dominica	DM	
Dominican Republic	DO	DOP
Ecuador	EC	ECS
Egypt	EG	EGP
El Salvador	SV	SVC
Equatorial Guinea	GQ	
Eritrea	ER	
Estonia	EE	EEK
Ethiopia	ET	ETB
Falkland Islands (Malvinas)	FK	
Faroe Islands	FO	

Fiji	FJ	FJD
Finland	FI	EUR
France	FR	EUR
French Guiana	GF	
French Polynesia	PF	
French Southern Territories	TF	
Gabon	GA	
Gambia	GM	GMD
Georgia	GE	
Germany	DE	EUR
Ghana	GH	GHC
Gibraltar	GI	GIP
Greece	GR	GRD
Greenland	GL	
Grenada	GD	
Guadeloupe	GP	
Guam	GU	
Guatemala	GT	GTQ
Guinea	GN	GNF
Guinea-Bissau	GW	GWP
Guyana	GY	GYD
Haiti	HT	HTG
Heard Island and Mcdonald Islands	HM	
Holy See (Vatican City State)	VA	
Honduras	HN	HNL
Hong Kong	HK	HKD
Hungary	HU	HUF
Iceland	IS	ISK
India	IN	INR
Indonesia	ID	IDR
Iran, Islamic Republic of	IR	
Iraq	IQ	IQD
Ireland	IE	EUR
Israel	IL	ILS
Italy	IT	EUR
Jamaica	JM	JMD
Japan	JP	JPY
Jordan	JO	JOD

Kazakhstan	KZ	
Kenya	KE	KES
Kiribati	KI	
Korea, Democratic People's Republic of	KP	
Korea, Republic of	KR	
Kuwait	KW	KWD
Kyrgyzstan	KG	
Lao People's Democratic Republic	LA	
Latvia	LV	
Lebanon	LB	LBP
Lesotho	LS	LSL
Liberia	LR	LRD
Libyan Arab Jamahiriya	LY	LYD
Liechtenstein	LI	
Lithuania	LT	
Luxembourg	LU	EUR
Macao	MO	
Macedonia, the Former Yugoslav Republic of	MK	
Madagascar	MG	MGF
Malawi	MW	MWK
Malaysia	MY	MYR
Maldives	MV	
Mali	ML	
Malta	MT	MTL
Marshall Islands	MH	
Martinique	MQ	
Mauritania	MR	MRO
Mauritius	MU	MUR
Mayotte	YT	
Mexico	MX	MXP
Micronesia, Federated States of	FM	
Moldova, Republic of	MD	
Monaco	MC	
Mongolia	MN	MNT
Montserrat	MS	
Morocco	MA	MAD
Mozambique	MZ	MZN

Myanmar	MM	
Namibia	NA	
Nauru	NR	
Nepal	NP	NPR
Netherlands	NL	EUR
Netherlands Antilles	AN	ANG
New Caledonia	NC	
New Zealand	NZ	NZD
Nicaragua	NI	NIO
Niger	NE	
Nigeria	NG	NGN
Niue	NU	
Norfolk Island	NF	
Northern Mariana Islands	MP	
Norway	NO	NOK
Oman	OM	OMR
Pakistan	PK	PKR
Palau	PW	
Palestinian Territory, Occupied	PS	
Panama	PA	PAB
Papua New Guinea	PG	PGK
Paraguay	PY	PYG
Peru	PE	PEN
Philippines	PH	PHP
Pitcairn	PN	
Poland	PL	PLZ
Portugal	PT	EUR
Puerto Rico	PR	
Qatar	QA	QAR
Reunion	RE	
Romania	RO	ROL
Russian Federation	RU	
Rwanda	RW	RWF
Saint Helena	SH	
Saint Kitts and Nevis	KN	
Saint Lucia	LC	
Saint Pierre and Miquelon	PM	
Saint Vincent and the Grenadines	VC	

Samoa	WS	WST
San Marino	SM	
Sao Tome and Principe	ST	
Saudi Arabia	SA	SAR
Senegal	SN	
Serbia and Montenegro	CS	
Seychelles	SC	SCR
Sierra Leone	SL	SLL
Singapore	SG	SGD
Slovakia	SK	
Slovenia	SI	
Solomon Islands	SB	SBD
Somalia	SO	SOS
South Africa	ZA	ZAR
South Georgia and the South Sandwich Islands	GS	
Spain	ES	EUR
Sri Lanka	LK	LKR
Sudan	SD	SDP
Suriname	SR	
Svalbard and Jan Mayen	SJ	
Swaziland	SZ	SZL
Sweden	SE	SEK
Switzerland	CH	CHF
Syrian Arab Republic	SY	
Taiwan, Province of China	TW	
Tajikistan	TJ	
Tanzania, United Republic of	TZ	
Thailand	TH	THB
Timor-Leste	TL	
Togo	TG	
Tokelau	TK	
Tonga	TO	TOP
Trinidad and Tobago	TT	TTD
Tunisia	TN	TND
Turkey	TR	TRL
Turkmenistan	TM	
Turks and Caicos Islands	TC	

Tuvalu	TV	
Uganda	UG	UGX
Ukraine	UA	
United Arab Emirates	AE	AED
United Kingdom	GB	GBP
United States	US	USD
United States Minor Outlying Islands	UM	
Uruguay	UY	UYP
Uzbekistan	UZ	
Vanuatu	VU	VUV
Venezuela	VE	VEB
Vietnam	VN	
Virgin Islands, British	VG	
Virgin Islands, U.s.	VI	
Wallis and Futuna	WF	
Western Sahara	EH	
Yemen	YE	YER
Zambia	ZM	ZMK
Zimbabwe	ZW	ZWD

Appendix B : Remitter ID Type Codes

Code	Description
Passport	Passport
Driving_License	Driving License
National_Insurance	National Insurance No.
National_ID	National ID
Utility_Bill	Utility Bill
Other	Other

Appendix C : Beneficiary ID Type Codes

Code	Description
PASSPORT	Passport
CHEQUEBOOK	Cheque Book
DRIVINGLICENSE	Driving License
NATIONALID	National ID
VOTINGCARD	Voting Card
OTHER	Other form of ID

Appendix D : Payment Method Codes

Code	Description
1	Cash
2	Cheque
3	Bank Transfer

Note: These codes may vary – please confirm with your Administrator

Appendix E : Service Level (Delivery Speed) Codes

Code	Description
1	5 Days
2	Next Day
3	Immediate

Note: These codes may vary – please confirm with your Administrator

Appendix F : Source of Income Codes

Code	Description
1	Salary
2	Business
3	Return on Investment
4	Gift
5	Pension

Note: These codes may vary – please confirm with your Administrator

Appendix G : Remittance Purpose Codes

Code	Description
1	Family Support
2	Medical Support
3	Education Support
4	Investment
5	Charity

Appendix H : Transaction Statuses

Status	Description
PENDING_CLEARANCE	Transaction has been created in the system but the remitter has not paid for transaction. Awaiting payment before transaction will be ready for processing.
ENTERED	Transaction awaiting approval by sending agent before processing.
AGENT_OK	Transaction awaiting HQ approval or routing before processing.
HQ_READY	Transaction has been approved by HQ and will be sent to processing bank/agent in the next batch. This only applies if you are using batch processing.
HQ_OK	Transaction awaiting processing bank/agent approval before ready for processing.
HQ_OK_PAID	Transaction awaiting processing bank/agent approval before ready for processing but transaction has been credited to processing bank/agent already. This status only applies in error correction scenarios.
SENT_FOR_PAY	Transaction ready to be paid out by processing bank/agent.
SENT_FOR_DELIVERY	Transaction ready to be paid out by processing bank/agent branch.
PROCESSED	Transaction has been processed.
ABORTED	Transaction has been accepted by processing bank/agent but now it has been requested to be cancelled. Awaiting processing bank/agent approval for cancellation or rejection of cancellation request.
ERROR	Transaction has some problem with it and is going through error correction cycle.
DELETED	Transaction has been cancelled.

Appendix I : Transfer Type*

Transfer Type
Account
Cash Collection
Card Transfer
Home Delivery
Utility Bill
Mobile Transfer
Wallet Transfer

* May differ depending on what transaction types you have available on your system.

Appendix J : Creation Channel

Creation Channel
Frontend
Frontend Offline
File Upload
Web Service
ORM