

Web Services Partner API Specification

RemitONE Money Transfer Management Solution

**Version 1.69
December 2020**

Table of Contents

Introduction.....	4
Connection Parameters.....	4
Response Format.....	5
Sample Request/Response.....	5
Testing Web Services Functionality.....	7
Using Web Services in a Live Environment: Some suggestions.....	7
Data Integrity Hashing.....	8
Web Service Method Details.....	10
Sending Agent API.....	10
getAgentDetails.....	10
getDestinationCountries.....	10
getRates.....	11
getCurrentCredit.....	12
createRemitter.....	12
searchRemitter.....	15
updateRemitter.....	17
uploadRemitterKYCVideo.....	21
getRemitterKYCVideo.....	21
verifyRemitter.....	22
getBeneficiaryAccountTypes.....	22
createBeneficiary.....	23
searchBeneficiary.....	25
getRemitterBeneficiaries.....	27
updateBeneficiary.....	28
createTransaction.....	30
confirmTransaction.....	33
insertTransaction (legacy).....	34
updateErrorTransaction.....	38
getTransactionDetails.....	41
getRemitterTransactions.....	44
getTransactionStatus.....	45
getTransactionStatusByAgentTransRef.....	46
getCharges.....	46
getDeliveryBanks.....	47
getDeliveryBankBranches.....	48
getCollectionPoints.....	49
getMobileNetworkOperators.....	50
getMobileNetworkCreditTypes.....	51
getUtilityCompanies.....	52
transactionPaymentCleared.....	54
getCashCollectionTransactionDetails.....	54
processCashCollectionTransaction.....	57
getErrorTransactions.....	58
requestCancellation.....	59
addWalletCredit.....	59
Receiving Bank/Agent API.....	60
getPayoutTransactions.....	60

getPayoutTransactionDetails.....	64
acceptPayoutTransactions.....	67
getPendingPayoutTransactions.....	70
processPayoutTransaction.....	73
errorPayoutTransaction.....	75
getErrorTransactions.....	75
updateBuyRates.....	77
createDeliveryBank.....	78
updateDeliveryBank.....	79
getDeliveryBanks.....	81
getDeliveryBankBranches.....	82
createDeliveryBankBranch.....	82
updateDeliveryBankBranch.....	84
getCollectionPoints.....	85
createCollectionPoint.....	86
updateCollectionPoint.....	87
createUtilityCompany.....	88
updateUtilityCompany.....	90
getUtilityCompanies.....	92
getMobileNetworkOperators.....	94
createMobileNetworkOperators.....	95
updateMobileNetworkOperators.....	96
createMobileNetworkCreditType.....	97
updateMobileNetworkCreditType.....	98
getMobileNetworkCreditTypes.....	99
updatePayoutTransDetails.....	101
Appendix A : Destination Country ISO Codes.....	102
Appendix B : Remitter ID Type Codes.....	109
Appendix C : Beneficiary ID Type Codes.....	109
Appendix D : Payment Method Codes.....	109
Appendix E : Service Level (Delivery Speed) Codes.....	110
Appendix F : Source of Income Codes.....	110
Appendix G : Remittance Purpose Codes.....	110
Appendix H : Transaction Statuses.....	111
Appendix A : Destination Country ISO Codes.....	73

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.

This document contains 2 APIs:

1. Sending Agent API: This allows you to create transactions on the RemitONE system
2. Receiving Agent API: This allows you to receive transactions from the RemitONE system

They are collectively known as the **Partner API**.

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.

<BASE_URL>/<GROUP>/<METHOD>

For example,

BASE_URL = <https://test.remit.by/abctransfer/ws/>

GROUP = transaction

METHOD = createTransaction

This would give a complete URL of :

<https://test.remit.by/abctransfer/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>
```

Sample Request/Response

Sample Request (using curl):

```
curl -i -X POST \
  -H "Content-Type:application/x-www-form-urlencoded" \
  -d \
  'username=another_ws_agent&password=Apassword1&pin=12345' \
```

<https://test.remit.by/abctransfer/ws/country/getDestinationCountries>

Sample Response:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>1030111</responseId>
  <status>SUCCESS</status>
  <result>
    <countries>
      <country>
        <id>999</id>
        <name>Bangladesh</name>
        <iso_code>BD</iso_code>
        <currency>BDT</currency>
        <allowed_currencies>
          <currency>BDT</currency>
        </allowed_currencies>
      </country>
      <country>
        <id>986</id>
        <name>Brazil</name>
        <iso_code>BR</iso_code>
        <currency>BRL</currency>
        <allowed_currencies>
          <currency>BRL</currency>
          <currency>USD</currency>
          <currency>DZD</currency>
        </allowed_currencies>
      </country>
    </countries>
  </result>
</response>
```

Testing Web Services Functionality

It is possible to test the output of the WebServices.
The URL for accessing forms for each of the WebServices is :

BASE_URL/wstest/index.html

For example, <https://test.remit.by/abctransfer/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.

Using Web Services in a Live Environment: Some suggestions

Sending Agent API

Although it is possible to use the WebServices in any way that seems fit, we would suggest the following procedure.

1. Obtain a list of valid Destination Countries using **getDestinationCountries**. This can be done on a weekly or monthly basis.
2. Obtain the current Agent Credit level using **getCurrentCredit**. The credit response will also indicate the *source currency* that the system is expecting transactions to be created in.
3. Obtain delivery entity information by calling **getDeliveryBanks**, **getDeliveryBankBranches** and **getCollectionPoints**. This information will be needed when creating beneficiaries and transactions.
4. Create a new remitter using **createRemitter**, or search for an existing one using **searchRemitter**.
5. Create a new beneficiary using **createBeneficiary**, or search for an existing one using **searchBeneficiary**. Provide the `remitter_id` from the previous step to ensure the correct beneficiary is found. Also, the **updateBeneficiary** method can be used at this point to set transfer details required in the next step.
6. Create a transaction between this remitter and beneficiary by calling **createTransaction**. *NOTE:* When creating a Cash Collection transaction the collection point details are specified in the **createTransaction** call. For all other transfer types (Account Transfer, Utility Bill, Mobile Transfer, Home Delivery) the details are retrieved from the beneficiary profile, so it

is necessary to add those details using `createBeneficiary` or `updateBeneficiary` beforehand.

7. Receive a transaction creation response containing the actual charges as calculated by the system, compliance check results, and a temporary transaction session id. It is the responsibility of the user to examine these results.
8. Once the user is satisfied with the results, call **`confirmTransaction`** and provide the temporary transaction session id from step 6 above to actually create the transaction in the system.
9. At some later time, **`getTransactionStatus`** or **`getTransactionStatusByAgentTranRef`** can be used to determine whether the transaction has been processed.

Receiving Agent API

In general, we suggest the following procedure :

1. Call **`getPayoutTransactions`** to get a list of transactions that need to be paid out by your Processing Bank.
2. Call **`acceptPayoutTransactions`** to 'accept' the transactions that you agree to payout. This will bring the status of the transactions to `SENT_FOR_PAY` or `SENT_FOR_DELIVERY`.
3. From your own systems, payout the transactions to the beneficiaries.
4. For each transaction that is paid out, call **`processPayoutTransaction`**.
5. If there is an error in processing a transaction call **`errorPayoutTransaction`**. These transactions will then need to be corrected by Admin and then passed back for payout. These error corrected transactions can then be obtained using the **`getPendingPayoutTransactions`** method (as long as the system state machine puts them into `SENT_FOR_PAY` or `SENT_FOR_DELIVERY` status).

Data Integrity Hashing

If you enabled Data Integrity Hashing for a WS user and set a key, for all requests done on the API by this user, the field **`data_integrity_hash`** will need to be added.

The data integrity hash can be created by doing the following:

1. Take all the POST parameters except for **`username`**, **`password`**, **`pin`**, **`id1_scan`** and **`id2_scan`** and concatenate them into a single string.
2. Append the "Data integrity key" value set on the WS users profile, to the end of the concatenated string.
3. Then generate a SHA256 Hash of the this string, and send that in the new POST variable `data_integrity_hash`.

The hash will be checked and validated on the server side.

The response XML from the server will also include a <data_integrity_hash> generated by taking all the string data from <result> open and close tags, add the "Data integrity key" to the end on the XML string and generate a SHA256 hash of that data. This can be used by the partner to verify data integrity on their end by regenerating the same has and comparing.

The hash can be generated by doing the following:

Given a response XML take the string with the opening and closing <result> tags. E.g.

```
"<result>  
    <trans_ref>12345</trans_ref>  
    <fname>Adam</fname>  
    <lname>West</lname>  
</result>"
```

Append the key to the end of the result string and hash the result. This will produce the hash returned in the response <data_integrity_hash> field.

Web Service Method Details

Sending Agent API

getAgentDetails

Group: **agent**

Method : **getAgentDetails**

This method provides the agent credentials including the agent name, country, currency and the destination countries.

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>
    <agent>
      <id>13</id>
      <username>agent_bidi</username>
      <name>Agnets</name>
      <country>United Kingdom-USD</country>
      <currency>USD</currency>
      <destination_countries>
        <country>
          <name>Lebanon</name>
          <code>LB</code>
          <currency>LBP</currency>
        </country>
        <country>
          <name>United Kingdom-USD</name>
          <code>UK-USD</code>
          <currency>USD</currency>
        </country>
      </destination_countries>
    </agent>
  </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>
```

getRates

Group : **rates**

Method : **getRates**

This method provides current rates from the system.

Input fields:

Name	Type	Notes
source_currency	Text	Optional
dest_country *	Text	The name of the Destination Country, as per the output of getDestinationCountries .
dest_currency	Text	Optional
Delivery Bank	Text	Optional

Name		
------	--	--

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>
```

getCurrentCredit

Group : **agent**

Method : **getCurrentCredit**

Input fields :

none

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>SUCCESS</status>
  <result>
    <credit>
      <currency>CAD</currency>
      <available_credit>1952</available_credit>
      <date_last_updated>2009-06-19 17:03:15</date_last_updated>
    </credit>
  </result>
</response>
```

createRemitter

Group : **remitter**

Method : **createRemitter**

Input fields :

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

firstname *	Text	
middlename	Text	
lastname *	Text	
type *	Text	registered / basicregistered
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	
Id1_scan	File	png, gif, jpg, jpeg, bmp and pdf
Id2_scan	File	png, gif, jpg, jpeg, bmp and pdf
receive_marketing	Boolean	t / f
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
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
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 / basicregistered
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>
  <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
remitter_id *	Text	
firstname	Text	
middlename	Text	
lastname	Text	
type	Text	registered / basicregistered
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
Annual income	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
receive_marketing	Boolean	t / f
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) of 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>
  <status>SUCCESS</status>
  <result>
    <remitter>
      <remitter_id></remitter_id>
```

```
<firstname></firstname>
<middlename></middlename>
<lastname></lastname>
<address1></address1>
<address2></address2>
<address3></address3>
<city></city>
<state></state>
<postcode></postcode>
<country></country>
<telephone></telephone>
<mobile></mobile>
<email></email>
<dob></dob>
<fathers_name></fathers_name>
<mothers_name></mothers_name>
<national_id_number></national_id_number>
<gender></gender>
<place_of_birth></place_of_birth>
<country_of_birth></country_of_birth>
<nationality></nationality>
<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>
<id_expiry></id_expiry>
<id_scan></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></orgtype>
<company_name></company_name>
<company_type></company_type>
<company_reg_no></company_reg_no>
<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>
<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	
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	

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	

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>
```

getBeneficiaryAccountTypes

Group : **beneficiary**

Method : **getBeneficiaryAccountTypes**

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>792</responseId>
  <status>SUCCESS</status>
  <result>
    <benef_ac_types>
      <benef_ac_type>INDIVIDUAL</benef_ac_type>
      <benef_ac_type>CORPORATE</benef_ac_type>
    </benef_ac_types>
  </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
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
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	
linked_remitter_id	Number	member_id of the linked remitter. This will filter the result set
name *	Text	
fname	Text	Beneficiary's first name
mname	Text	Beneficiary's middle name
lname	Text	Beneficiary's last name
address_line1	Text	
city	Text	
postcode	Text	
country *	Text	Required, as per output of getDestinationCountries
dob	Date	YYYY-MM-DD
email	Text	
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_iban>IBN123456789</benef_bank_iban>
        <benef_bank_bsb_code></benef_bank_bsb_code>
        <benef_bank_routing_transit_number>
          </benef_bank_routing_transit_number>
        <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>
        <linked_remitter_ids>
          <remitter_id>10007</remitter_id>
        </linked_remitter_ids>
      </beneficiary>
    </beneficiaries>
  </result>
</response>
```

getRemitterBeneficiaries

Group : **beneficiary**

Method : **getRemitterBeneficiaries**

Input fields :

Name	Type	Notes
remitter_id	Number	

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>
        <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_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_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>
    </beneficiary>
  </result>
</response>
```

```

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

```

createTransaction

Group: **transaction**

Method: **createTransaction**

Input fields:

Name	Type	Notes
remitter_id *	Number	
beneficiary_id *	Number	
destination_country *	Text	Name of country retrieved from calling getDestinationCountries
agent_trans_ref	Text	Agent's reference number for transaction
trans_type *	Text	Account / Cash Collection / Card Transfer / Home Delivery / Mobile

		Transfer
purpose *	Code	See Appendix G for codes
source_of_income *	Code	See Appendix F for codes
collection_point	Text	Name of collection point where money will be collected by beneficiary. Required if transaction is of type Cash Collection. All the fields starting with 'collection_' below only apply to Cash Collection transactions.
collection_point_code	Text	Collection point code.
collection_point_bank	Text	Name of bank or organisation that the collection point is under. Required if transaction is of type Cash Collection.
collection_point_address	Text	Collection point address.
collection_point_city	Text	Collection \$transactionpoint city.
collection_point_state	Text	Collection point state.
collection_point_telephone	Text	Collection point telephone number.
collection_pin	Numeric	A PIN number which the beneficiary needs to provide on cash pick-up. This may be required, please check with your partner.
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 to Pay.
benef_utilitybillbankcode	Text	Bank IBAN/SWIFT Code.
benef_utilitybilldescription	Text	Bill Description.
benef_mobiletransfer_number	Text	The mobile number of the beneficiary for mobile transfer.
benef_mobiletransfer_network	Text	The mobile phone network of the beneficiary for mobile transfer.
payment_method *	Code	See Appendix D for codes.
amount_deposited *	Numeric	Only required when the payment method is "BankTransfer". This is the amount that has been deposited by the remitter.
date_deposited *	Date	YYYY-MM-DD

		Only required when the payment method is "BankTransfer". This is the date on which the remitter has deposited the money.
bank_branch_deposited *	Text	Only required when the payment method is "BankTransfer". This is the name of the branch at which the remitter has deposited the money.
service_level *	Code	See Appendix E for codes.
sms_confirmation *	Boolean	t / f
sms_notification *	Boolean	t / f
sms_mobile	Text	Mobile number for Remitter to send SMS to
sms_benef_confirmation	Boolean	t / f
sms_benef_mobile	Text	Mobile number for Beneficiary to send SMS to
source_currency	Text	Source currency, if blank will use default
dest_currency	Text	Dest currency, if blank will use default
amount_type *		Whether the amount is specified in the source currency or the destination currency. Possible values : SOURCE / DESTINATION.
amount_to_send *	Numeric	
agent_comments	Text	Remitter/Agent Comments
rate	Numeric	Updated rate for the single transactions. It works only if ALLOW_EDITABLE_RATE is true.
commission	Numeric	Updated commission for the single transaction. It works only if EDITABLE_COMMISSION is true or agent can edit the commissions

NOTE: After invoking this method, confirmTransaction must be called with the trans_session_id to successfully create the transaction.

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>SUCCESS</status>
  <result>
    <trans_session_id>245861246457499</trans_session_id>
    <remitter_id>10019</remitter_id>
    <remitter_name>MUHAMMAD KHAN</remitter_name>
  </result>
</response>
```



```

    <beneficiary_id>5</beneficiary_id>
    <beneficiary_name>BENEF TEST</beneficiary_name>
    <trans_type>Card Transfer</trans_type>
    <destination_country>Philippines</destination_country>
    <source_currency>CAD</source_currency>
    <source_transfer_amount>100.00</source_transfer_amount>
    <rate>37.5300</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>
    <tax>3.40</tax>
    <remitter_pay_amount>114.40</remitter_pay_amount>
    <agent_deduction>4.20</agent_deduction>
    <agent_to_pay_hq>110.20</agent_to_pay_hq>
    <delivery_date>2009-07-02 00:00:00</delivery_date>
    <payment_token>2fe1ebbf197c7d4e4762d154602762ef941c110</payment_token>
  </result>
</response>

```

confirmTransaction

Group: **transaction**

Method: **confirmTransaction**

Input fields:

Name	Type	Notes
trans_session_id *	Numeric	Temporary ID generated by the system and returned as result of createTransaction

Example Output XML :

```

<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>SUCCESS</status>
  <result>
    <reference_number>RB0003113</reference_number>
    <remitter_id>10019</remitter_id>
    <remitter_name>MUHAMMAD KHAN</remitter_name>
    <beneficiary_id>5</beneficiary_id>
    <beneficiary_name>BENEF TEST</beneficiary_name>
    <status>{$trans->status}</status>
    <trans_type>Card Transfer</trans_type>
    <destination_country>Philippines</destination_country>
    <source_currency>CAD</source_currency>
    <source_transfer_amount>100.00</source_transfer_amount>
    <rate>37.5300</rate>
    <destination_currency>PHP</destination_currency>
    <destination_amount>3753.00</destination_amount>
  </result>
</response>

```

```

    <commission>10.50</commission>
    <agent_fee>0.00</agent_fee>
    <hq_fee>0.50</hq_fee>
    <tax>3.40</tax>
    <remitter_pay_amount>114.40</remitter_pay_amount>
    <agent_deduction>4.20</agent_deduction>
    <agent_to_pay_hq>110.20</agent_to_pay_hq>
    <delivery_date>2009-07-02 00:00:00</delivery_date>
    <payment_token>2fe1ebbf197c7d4e4762d154602762ef941c110</payment_token>
    <compliance_check_required>f</compliance_check_required>
    <collection_pin>762634</collection_pin>
    <agent_trans_ref>GHT33762634</agent_trans_ref>
    <benef_trans_ref>GHT33762634</benef_trans_ref>
  </result>
</response>

```

insertTransaction (legacy)

Group: **transaction**

Method: **insertTransaction**

This method should not be used anymore.
Please use createTransaction and confirmTransaction methods instead.

Input fields:

Name	Type	Notes
agent_trans_ref *	Text	Transaction reference from the sending system.
trans_type*		Account / Cash Collection / Card Transfer / Home Delivery / Mobile Transfer
purpose	Text	See 'Description' on Appendix G
source_of_income	Text	See 'Description' on Appendix F
member_id	Number	Optional, remitter's member id
occupation	Text	
remitt_name	Text	
remitt_fname	Text	
remitt_mname	Text	
remitt_lname	Text	
remitt_addr	Text	
remitter_address2	Text	
remitt_city	Text	
remitt_state	Text	

remitt_postcode	Text	
remitt_tel	Text	
remitt_id_type	Text	
remitt_id_details	Text	
id1_issued_by	Text	
id1_issue_place	Text	
remitt_nationality	Text	2 letter country ISO code. Please refer to Appendix A.
remitt_dob	Date	YYYY-MM-DD
remitt_mobile	Text	
remitt_orgtype	Text	Individual or Corporate
remitt_company_name	Text	Optional if Individual
remitt_company_type	Text	Optional if Individual
remitt_company_reg_no	Text	Optional if Individual
remitt_business_type	Text	Optional if Individual
benef_id	Number	Optional, beneficiary's member id
benef_name	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_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_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	
benef_utilitybillcompany	Text	Name of the Utility Company where money will be paid. Required if transaction is of type Utility Bill.
benef_utilitybillcompanycode	Text	The Utility Company Code. It is used for routing the money.
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	
payment_method*	Code	See Appendix D for codes.
source_currency*	Text	

source_amount*	Numeric	
dest_currency*	Text	
dest_amount*	Numeric	
rate*	Numeric	
commission*	Numeric	
source_tax	Numeric	
originating_country*	Text	2 letter country ISO code. Please refer to Appendix A.
destination_country*	Text	2 letter country ISO code. Please refer to Appendix A.
service_level	Code	See Appendix E for codes.
sms_notification*	Boolean	t / f
sms_confirmation*	Boolean	t / f
sms_benef_confirmation*	Boolean	t / f
agent_fee	Numeric	
agent_comments	Text	
sms_remitter_mobile	Text	
sms_benef_mobile	Text	
agent_settlement_exchange_rate	Numeric	
agent_comments	Text	Remitter/Agent Comments

NOTE: Required fields will vary depending on the Transaction Type and configurations of the receiving system. Currently this methods is only used for importing Transactions from 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>
  </result>
</response>
```

```

    <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>

```

Comparison of insertTransaction with createTransaction/confirmTransaction:

create/confirmTransaction:

- This is a two step process, so there is explicit confirmation from the partner creating the transaction
- The fees, rates etc are always calculated on the R1 MTMS, thus ensuring that the settings made in the R1 system are enforced on the transaction
- Remitters and beneficiaries have to be created and updated explicitly before being used in transactions. This ensures data is maintained explicitly and helps to prevent duplicates

insertTransaction:

- This is a one-step transaction creation, so there is no explicit confirmation from the partner
- In some configurations the fees and rates can be provided by the partner and this will override the R1 system settings
- Remitter and Beneficiary data is provided on every call, and the R1 system is then left to try to match this to existing entities

Overall, we urge you to use the create/confirm sequence rather than insertTransaction for obvious data integrity reasons.

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_address	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>

```

getTransactionDetails

Group: **transaction**

Method: **getTransactionDetails**

Get details of a single transaction that has been created by the Agent.

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>
    </transaction>
  </result>
</response>

```

```
</benef_name>
<benef_tel>
  <![CDATA[]]>
</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>
</transaction>

```

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

getRemitterTransactions

Group : **transaction**

Method : **getTransactions**

Get details of a remitter's transactions for the past three months

Input fields :

Name	Type	Notes
remitter_id	Number	

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>
        <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_address>123 New Street</remitter_address>
        <remitter_city>London</remitter_city>
        <remitter_tel>012345678</remitter_tel>
        <remitter_mobile>012345678</remitter_mobile>
        <remitter_id_type>passport</remitter_id_type>
        <remitter_id_details>54321</remitter_id_details>
        <remitter_id>11217</remitter_id>
        <payment_method>1</payment_method>
        <benef_name>Kevin Smith</benef_name>
        <benef_mobile>0987654321</benef_mobile>
        <benef_ac>87654321</benef_ac>
        <benef_bank_swift_code></benef_bank_swift_code>
        <benef_bank_ifsc_code></benef_bank_ifsc_code>
        <benef_branch>Main Branch</benef_branch>
        <collection_point_bank></collection_point_bank>
```

```

        <account_transfer_bank>United
Bank</account_transfer_bank>

        <benef_mobiletransfer_number><![CDATA[]]></benef_mobiletransfer_number>

        <benef_mobiletransfer_network><![CDATA[]]></benef_mobiletransfer_network>
        <benef_mobiletransfer_network_credit_type><![CDATA[]]></
benef_mobiletransfer_network_credit_type>
        <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>
        <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>
        <admin_comments><![CDATA[]]></admin_comments>
        <agent_comments><![CDATA[]]></agent_comments>
    </transaction>
</transactions>
</result>
</response>

```

getTransactionStatus

Group: **transaction**

Method: **getTransactionStatus**

Input fields:

Name	Type	Notes
trans_ref *	Text	Transaction reference from the RemitONE system. See getTransactionStatusByAgentTransRef below to request using the agent's transaction reference.

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>

getTransactionStatusByAgentTransRef

Group: **transaction**

Method: **getTransactionStatusByAgentTransRef**

Input fields:

Name	Type	Notes
trans_ref *	Text	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.

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>
```

getCharges

Group: **transaction**

Method: **getCharges**

Input fields:

Name	Type	Notes
destination_country *	Text	Name of country retrieved from calling getDestinationCountries
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.

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	
Source Currency	Text	Same code as you get when calling getSourceCountries
Destination currency	Text	Same code as you get when calling getDestinationCountries

If source and destination currencies are not entered, then the default source and destination currencies will be used instead.

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>
```

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>
```

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>

```

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>

```

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>
        <enabled>t</enabled>
      </mnct>
    </mncts>
  </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>
      </utility_company>
    </utility_companies>
  </result>
</response>

```

```

        <country_id>999</country_id>
        <enabled>t</enabled>
    </utility_company>
</utility_companies>
</result>
</response>

```

transactionPaymentCleared

Group: **transaction**

Method: **transactionPaymentCleared**

Marks a transaction which is in status PENDING_CLEARANCE (i.e still requires payment) as payment cleared to say that the payment has been made.

Input fields:

Name	Type	Notes
trans_ref	Text	Enter either trans_ref and agent_trans_ref
agent_trans_ref	Text	

```

<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>45</responseId>
  <status>SUCCESS</status>
  <result>
    <trans_ref>RC12321321321</trans_ref>
    <message>Transaction payment cleared successfully</message>
  </result>
</response>

```

getCashCollectionTransactionDetails

Group: **transaction**

Method: **getCashCollectionTransactionDetails**

Look up for cash collection transaction as a **BIDI** agent.

Name	Type	Notes
trans_ref	Text	Transaction reference number
collection_pin_number	Text	Collection pin number
security_question	Text	Security question

security_answer	Text	Security answer
-----------------	------	-----------------

Input fields:

Need to input the trans_ref or/and collection_pin_number depending on the configuration setting.

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>SUCCESS</status>
  <result>
    <transaction>
      <trans_ref>RC000224000133</trans_ref>
      <status>HQ_OK</status>
      <creation_date>2014-09-19 10:41:40.755687+00</creation_date>
      <purpose>FAMILY SUPPORT</purpose>
      <bank_branch_name></bank_branch_name>
      <bank_branch_code></bank_branch_code>
      <bank_accept_date></bank_accept_date>
      <bank_branch_accept_date></bank_branch_accept_date>
      <delivery_date>2014-09-22 10:41:00+00</delivery_date>
      <processed_date></processed_date>
      <processed_by></processed_by>
      <delivery_date>2014-09-22 10:41:00+00</delivery_date>
      <trans_type>Cash Collection</trans_type>
      <benef_name>
        <![CDATA[MUM AND DAD]]>
      </benef_name>
      <benef_tel>
        <![CDATA[45124897217421]]>
      </benef_tel>
      <benef_mobile>
        <![CDATA[]]>
      </benef_mobile>
      <benef_email>
        <![CDATA[hani.safa@remitone.com]]>
      </benef_email>
      <collection_point>
        <![CDATA[COLLECTION POINT UK]]>
      </collection_point>
      <collection_point_bank>
        <![CDATA[DELIVERY BANK UK]]>
      </collection_point_bank>
      <collection_point_code>
        <![CDATA[]]>
      </collection_point_code>
      <collection_pin>
        <![CDATA[1497974]]>
      </collection_pin>
      <benef_id_type>
        <![CDATA[NONE]]>
      </benef_id_type>
      <benef_id_detail>
        <![CDATA[]]>
      </benef_id_detail>
      <benef_ac>
```

```

                <![CDATA[]]>
</benef_ac>
        <benef_bank_swift_code>
                <![CDATA[]]>
</benef_bank_swift_code>
        <benef_bank>
                <![CDATA[]]>
</benef_bank>
        <benef_branch>
                <![CDATA[]]>
</benef_branch>
        <benef_branch_code>
                <![CDATA[]]>
</benef_branch_code>
        <benef_bank_city>
                <![CDATA[]]>
</benef_bank_city>
        <benef_bank_state>
                <![CDATA[]]>
</benef_bank_state>
        <benef_card_number>
                <![CDATA[]]>
</benef_card_number>
        <benef_address1>
                <![CDATA[]]>
</benef_address1>
        <benef_address2>
                <![CDATA[]]>
</benef_address2>
        <benef_address3>
                <![CDATA[]]>
</benef_address3>
        <benef_city>
                <![CDATA[]]>
</benef_city>
        <benef_state>
                <![CDATA[WALTHAMSTOW]]>
</benef_state>
        <benef_postcode>
                <![CDATA[E17]]>
</benef_postcode>
        <benef_mobiletransfer_number>
                <![CDATA[]]>
</benef_mobiletransfer_number>
        <benef_mobiletransfer_network>
                <![CDATA[]]>
</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[HANI SAFA]]>
</remitter_name>
    <send_country>UK-USD</send_country>
    <send_currency>USD</send_currency>
    <send_amount>10.00</send_amount>
    <rate>1.00000000</rate>
    <receive_country>UK-USD</receive_country>
    <receive_currency>USD</receive_currency>
    <receive_amount>10.00</receive_amount>
    <bank_sequence></bank_sequence>
    <pay_method></pay_method>
    <issue_date></issue_date>
    <bank_ref></bank_ref>
    <benef_trans_ref>RC000224000133</benef_trans_ref>
    <bank_comments></bank_comments>
    <bank_credit_date></bank_credit_date>
    <bank_clear_date></bank_clear_date>
  </transaction>
</result>
</response>

```

processCashCollectionTransaction

Group: **transaction**

Method: **processCashCollectionTransaction**

Process cash collection transaction as a **BIDI** agent.

Name	Type	Notes
trans_ref	Text	Transaction reference number
collection_pin_number	Text	Collection pin number
benef_id_type	Text	Beneficiary ID Type
beneficiary_id_no	Text	Beneficiary ID number
security_question	Text	Security question
security_answer	Text	Security answer
payout_comments	Text	Payout Comments

Input fields:

Need to input the trans_ref or/and collection_pin_number depending on the configuration setting.

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>SUCCESS</status>
  <result>
    <trans_ref>RC000224000133</trans_ref>
    <process_result>SUCCESS</process_result>
    <message></message>
  </result>
</response>
```

getErrorTransactions

Group : **transaction**

Method : **getErrorTransactions**

Input fields :

Name	Type	Notes
from_date	Date	YYYY-MM-DD
to_date	Date	YYYY-MM-DD

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>
      </transaction>
    </transactions>
  </result>
  <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>
</response>
```

```

        <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>
    </transaction>
</transactions>
</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.

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>
```

Receiving Bank/Agent API

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:

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

This method can be used by WS_Bank_Super users. In this case, an optional parameter "bank_name" can be used to filter the results by the specified bank name (as long as this Processing bank belongs to this WS_Bank_Super user).

Name	Type	Notes
bank_name	Text	<p>Optional. This is only considered when the user is a WS_Bank_Super. This would be a Processing Bank Name that belongs to this WS_Bank_Super user.</p> <p>If the current user is simply a WS_Bank, this field is not necessary</p>

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>
        <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
trans_ref *	Text	
use_agent_trans_ref	Text	Send value "true" to search by Agent Trans Ref instead of TransRef

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[]]>
        </remitt_benef_relation>
      <benef_tel>
        <![CDATA[]]>
      </benef_tel>
    </transaction>
  </result>
</response>

```



```
<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>
    <send_country>SG</send_country>
    <send_country_iso>SG</send_country_iso>
    <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_country_iso>BD</receive_country_iso>
    <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>
    <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
trans_ref *	Text	Multiple transactions can be accepted in one call by providing the transaction references of all the transactions separated by commas.
bank_name*	Text	This is only required when the user is a WS_Bank_Super. This would be a Processing Bank Name that belongs to this WS_Bank_Super user. If the current user is simply a WS_Bank, this field is not necessary

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>
    <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>
```

getCashCollectionTransactionDetails

Group: **transaction**

Method: **getCashCollectionTransactionDetails**

Look up for cash collection transaction as a **BIDI** agent.

Name	Type	Notes
trans_ref	Text	Transaction reference number
collection_pin_number	Text	Collection pin number
security_question	Text	Security question
security_answer	Text	Security answer

Input fields:

Need to input the trans_ref or/and collection_pin_number depending on the configuration setting.

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <status>SUCCESS</status>
  <result>
    <transaction>
      <trans_ref>RC000224000133</trans_ref>
      <status>HQ_OK</status>
      <creation_date>2014-09-19 10:41:40.755687+00</creation_date>
      <purpose>FAMILY SUPPORT</purpose>
      <bank_branch_name></bank_branch_name>
      <bank_branch_code></bank_branch_code>
      <bank_accept_date></bank_accept_date>
      <bank_branch_accept_date></bank_branch_accept_date>
      <delivery_date>2014-09-22 10:41:00+00</delivery_date>
      <processed_date></processed_date>
      <processed_by></processed_by>
      <delivery_date>2014-09-22 10:41:00+00</delivery_date>
      <trans_type>Cash Collection</trans_type>
      <benef_name>
        <![CDATA[MUM AND DAD]]>
      </benef_name>
      <benef_tel>
        <![CDATA[45124897217421]]>
      </benef_tel>
      <benef_mobile>
        <![CDATA[]]>
      </benef_mobile>
      <benef_email>
        <![CDATA[hani.safa@remitone.com]]>
      </benef_email>
      <collection_point>
        <![CDATA[COLLECTION POINT UK]]>
      </collection_point>
      <collection_point_bank>
        <![CDATA[DELIVERY BANK UK]]>
      </collection_point_bank>
      <collection_point_code>
        <![CDATA[]]>
      </collection_point_code>
    </transaction>
  </result>
</response>
```

```

</collection_point_code>
  <collection_pin>
    <![CDATA[1497974]]>
  </collection_pin>
  <benef_id_type>
    <![CDATA[NONE]]>
  </benef_id_type>
  <benef_id_detail>
    <![CDATA[]]>
  </benef_id_detail>
  <benef_ac>
    <![CDATA[]]>
  </benef_ac>
  <benef_bank_swift_code>
    <![CDATA[]]>
  </benef_bank_swift_code>
  <benef_bank>
    <![CDATA[]]>
  </benef_bank>
  <benef_branch>
    <![CDATA[]]>
  </benef_branch>
  <benef_branch_code>
    <![CDATA[]]>
  </benef_branch_code>
  <benef_bank_city>
    <![CDATA[]]>
  </benef_bank_city>
  <benef_bank_state>
    <![CDATA[]]>
  </benef_bank_state>
  <benef_card_number>
    <![CDATA[]]>
  </benef_card_number>
  <benef_address1>
    <![CDATA[]]>
  </benef_address1>
  <benef_address2>
    <![CDATA[]]>
  </benef_address2>
  <benef_address3>
    <![CDATA[]]>
  </benef_address3>
  <benef_city>
    <![CDATA[]]>
  </benef_city>
  <benef_state>
    <![CDATA[WALTHAMSTOW]]>
  </benef_state>
  <benef_postcode>
    <![CDATA[E17]]>
  </benef_postcode>
  <benef_mobiletransfer_number>
    <![CDATA[]]>
  </benef_mobiletransfer_number>
  <benef_mobiletransfer_network>
    <![CDATA[]]>
  </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[HANI SAFA]]>
</remitter_name>
    <send_country>UK-USD</send_country>
    <send_currency>USD</send_currency>
    <send_amount>10.00</send_amount>
    <rate>1.00000000</rate>
    <receive_country>UK-USD</receive_country>
    <receive_currency>USD</receive_currency>
    <receive_amount>10.00</receive_amount>
    <bank_sequence></bank_sequence>
    <pay_method></pay_method>
    <issue_date></issue_date>
    <bank_ref></bank_ref>
    <benef_trans_ref>RC000224000133</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>
    <payout_comments><![CDATA[]]></payout_comments>
  </transaction>
</result>
</response>

```

NOTES: Please note that this method is not available to WS_Bank_Super users.

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:

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

This method can be used by WS_Bank_Super users. In this case, an optional parameter "bank_name" can be used to filter the results by the specified bank name (as long as this Processing bank belongs to this WS_Bank_Super user).

Name	Type	Notes
bank_name	Text	Optional. This is only considered when the user is a WS_Bank_Super. This would be a Processing Bank Name that belongs to this WS_Bank_Super user. If the current user is simply a WS_Bank, this field is not necessary

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>
    <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>
    <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
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	Bank Comments
payout_comments	Text	Payout Comments
bank_name*	Text	This is only required when the user is a WS_Bank_Super and the transaction is a Cash Collection. This would be a Processing Bank Name that belongs to this WS_Bank_Super user. If the current user is simply a WS_Bank, this field is not necessary
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
trans_ref *	Text	
error_reason *	Text	The overall reason for error.
error_details	Text	Details of the error.
bank_name *	Text	This is only required when the user is a WS_Bank_Super and the transaction is a Cash Collection. This would be a Processing Bank Name that belongs to this WS_Bank_Super user. If the current user is simply a WS_Bank, this field is not necessary

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>
```

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	Optional. This is only considered when the user is a WS_Bank_Super. This would be a Processing Bank Name that belongs to this WS_Bank_Super user. If the current user is simply a WS_Bank, this field is not necessary

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>
        <payout_comments><![CDATA[]]></payout_comments>
      </transaction>
    </transactions>
  </result>
</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**

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>
        <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>
```

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 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>
```

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>
```

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>
      </utility_company>
    </utility_companies>
  </result>
</response>
```

```

        <bic_no>AAGSIT21XXX</bic_no>
        <country_id>999</country_id>
        <enabled>t</enabled>
    </utility_company>
</utility_companies>
</result>
</response>

```

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>
        <enabled>t</enabled>
      </mnct>
    </mncts>
  </result>
</response>

```

updatePayoutTransDetails

Group : **transaction**

Method : **updatePayoutTransDetails**

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 code of the transaction to update
benef_trans_ref	Text	The new code for the beneficiary transaction reference
bank_comments	Text	Optional. The bank comments. The will be appended to the previous ones.
payout_comments	Text	Optional. The payout comments. The will be appended to the previous ones.
collection_pin	Text	Optional. The collection PIN for the transaction.

Example Output XML :

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <responseId>135</responseId>
  <status>SUCCESS</status>
  <result>
    <trans_id>1</trans_id>
    <trans_ref>RC000224000100</trans_ref>
    <benef_trans_ref>RC000224000112</benef_trans_ref>
    <bank_comments>Some bank comments</bank_comments>
    <payout_comments>Some payout comments</payout_comments>
    <collection_pin>123456789</collection_pin>
  </result>
</response>
```

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.