**Application programming interface**

**RBL Bank money transfer for NON KYC**

**(Version 1.1)**

**Document History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Changes** | |
| 1 | 28/05/2014 | Document Modified | Sandeep Reddy K |
| 2 | 22/03/2015 | Document Modified | Sandeep Reddy K |
| 3 | 11/04/2017 | Addition of Bene account validation API | Stalin Babu |

Contents

[I. INTRODUCTION 3](#_Toc514776246)

[1. API Definition 3](#_Toc514776247)

[2. About XML 3](#_Toc514776248)

[II. RBL MONEY TRANSFER API SPECIFICATION 4](#_Toc514776249)

[1. Channel Partner Login 5](#_Toc514776250)

[2. Remitter Registration 7](#_Toc514776251)

[3. Remitter Validation 11](#_Toc514776252)

[4. Remitter Resend OTP 12](#_Toc514776253)

[5. Beneficiary Registration 13](#_Toc514776254)

[6. Beneficiary Validation 17](#_Toc514776255)

[7. Beneficiary Resend OTP 18](#_Toc514776256)

[8. Transaction 18](#_Toc514776257)

[9. Refund OTP 23](#_Toc514776258)

[10. Refund 23](#_Toc514776259)

[11. Remitter Details 25](#_Toc514776260)

[12. Delete Beneficiary 30](#_Toc514776261)

[13. Delete Beneficiary Validation 31](#_Toc514776262)

[14. Transaction Re-Query 32](#_Toc514776263)

[15. Get Service Charge 37](#_Toc514776264)

[16. Beneficiary Account Validation 39](#_Toc514776265)

[17. Common Error 41](#_Toc514776266)

# INTRODUCTION

# 1. API Definition

Application programming interface is a middleware that communicates between the host system and the front end application. This API connects the Bank Wallet Platform with the Application front end channel. This API is built with standard XML protocol for portability and easy integration.

# 2. About XML

XML provides a Rich text-based to describe and also apply a tree-based structure for information processing. At its base level, all information manifests as text, interspersed with markup that indicates the information's separation into a hierarchy of character data, container-like elements, and attributes of those elements. In this respect, it is similar to the LISP programming language's S-expressions, which describe tree structures wherein each node may have its own property list.

The fundamental unit in XML is the character, as defined by the Universal Character Set. Characters are combined to form an XML document. The document consists of one or more entities, each of which is typically some portion of the document's characters, stored in a text file.

XML files may be served with a variety of Media types. RFC 3023 defines the types "application/xml" and "text/xml", which say only that the data is in XML, and nothing about its semantics. The use of "text/xml" has been criticized as a potential source of encoding problems but now is in the process of being deprecated RFC 3023 also recommends that XML-based languages be given media types beginning in "application/" and ending in "+xml"; for example "application/atom+xml" for Atom. This page discusses further XML and MIME.

The ubiquity of text file authoring software (basic text editors such as Notepad and Text Edit as well as word processors) facilitates rapid XML document authoring and maintenance. Prior to the advent of XML, there were very few data description languages that were general-purpose, Internet protocol-friendly, and very easy to learn and author. In fact, most data interchange formats were proprietary, special-purpose, "binary" formats (based foremost on bit sequences rather than characters) that could not be easily shared by different software applications or across different computing platforms, much less authored and maintained in common text editors.

# RBL MONEY TRANSFER API SPECIFICATION

The below document provides the integration details for enabling the onlne BC Agent service to the third party portal :

**Please note that the given all request parameters are mandatory and at here to the XSD schema attached.**

**RBL UAT test URL-** <https://apideveloper.rblbank.com/test/sb/rbl/v1/dmt/non-kyc>

# [Channel](#CHANNELPARTNERLOGIN) Partner Login

This XML service is used for channelpartner to create login session with the validity of 1 hour. Every 1 hour partners needs to regenerate the session and the same session token can be used in all other API requests.

**Request -**

<channelpartnerloginreq>

<username></username>

<password></password>

<bcagent></bcagent>

</channelpartnerloginreq>

**Response –**

<channelpartnerloginres>

<sessiontoken></sessiontoken>

<timeout></timeout>

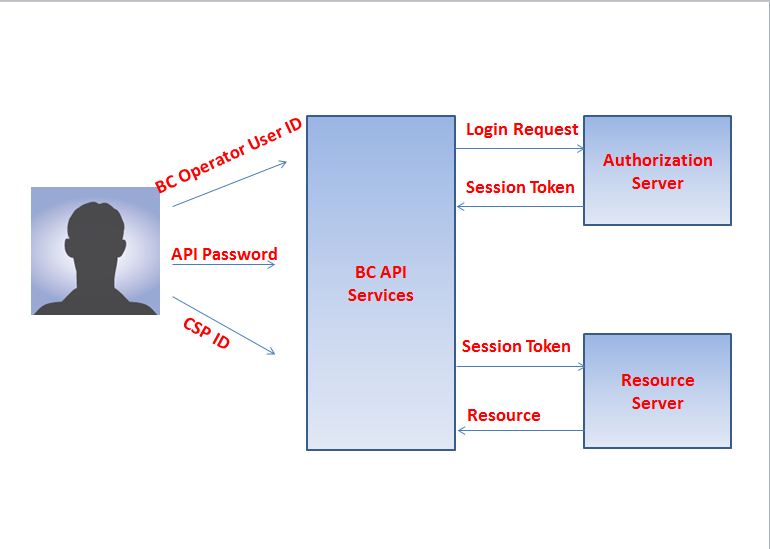
<status></status>

</channelpartnerloginres>

XSD -



|  |  |  |  |
| --- | --- | --- | --- |
| **Request Fields** | | | |
| **Field Name** | **Data Type** | **Length** | **Description** |
| Username\* | String | 5-29 | Describes the Channel partner user name maintained by the RBL. |
| Password\* | String | 5-40 | Describe the API password given by RBL |
| Bcagent\* | String | 5-45 | Describes the agent unique ID maintained by the Channel Partner. |
| **Response Fields** | | | |
| **Field Name** | **Data Type** | **Length** | ***Description*** |
| sessiontoken | string | 1-200 | Describes the session token created for Channel partner in encrypted format and it will be used in other API Methods and session token will valid till 1 hour. |
| timeout | datetime | 1-50 | Describes the channel partner login timeout .Format(mm/dd/yyyy h:mm:ss Am/pm) |
| Status | int | 1 | 0-failure  1-success |

**Architecture of Login API -**  


***Important Points:-***

*1.* ***Username*** *is BC username of Channel Partner Company registered under Remittance Engine.   
2.* ***Password*** *is API password in encrypted format which will be shared by RBL Bank.  
3.* ***BC Agent*** *is Registered & Approved CSP under Channel Partner Company  
4. There can be multiple sessions with different credentials  
5. Session Token will be created as per Username, API Password, time and Agent ID  
6. Session Token received in response of login request will be active for next 60 minutes only.  
7. Once received, this token can be used in other API Methods.*

# Remitter Registration

This API is used to create/register the customers for remittance transactions.

**Request XML -**

<remitterregistrationrmreq>

<header>

<sessiontoken></sessiontoken>

</header>

<bcagent></bcagent>

<remittermobilenumber></remittermobilenumber>

<remittername></remittername>

<remitteraddress1></remitteraddress1>

<remitteraddress2></remitteraddress2>

<pincode></pincode>

<cityname></cityname>

<statename></statename>

<alternatenumber></alternatenumber>

<idproof></idproof>

<idproofnumber></idproofnumber>

<idproofissuedate></idproofissuedate>

<idproofexpirydate></idproofexpirydate>

<idproofissueplace></idproofissueplace>

<lremitteraddress></lremitteraddress>

<lpincode></lpincode>

<lstatename></lstatename>

<lcityname></lcityname>

</remitterregistrationrmreq>

**Response XML -**

<remitterregistrationrmres>

<bcagent></bcagent>

<remitterid></remitterid>

<status></status>

</remitterregistrationrmres>

**XSD**



|  |  |  |  |
| --- | --- | --- | --- |
| **Request Fields** | | | |
| **Field Name** | **Data Type** | **Length** | **Description** |
| Session Token\* | String | 2-199 | Session Token received in Login Response |
| bcagent\* | String | 5-49 | Describes the BC Agent’s unique ID maintained by the Channel Partner. |
| remittermobilenumber\* | int | 10 | Describes the Remitter’s Registered Mobile Number. |
| remittername\* | String | 5-49 | Describes the Remitter’s Name. |
| remitteraddress1\* | String | 5-199 | Describes the Remitter’s Address. |
| remitteraddress2 | String | 5-199 | Describes the Remitter’s Address2. |
| Pincode\* | int | 1-10 | Describes the Remitter’s Pincode. |
| Cityname\* | String | 5-49 | Describes the Remitter’s City. |
| Statename\* | String | 5-49 | Describes the Remitter’s State. |
| Alternatenumber | int | 10 | Describes the Remitter’s Other Mobile Number. (OPTIONAL) |
| Lremitteraddress\* | String | 5-199 | Describes the Remitter’s local Address. |
| Lpincode\* | int | 1-10 | Describes the Remitter’s local Pincode number. |
| Lstatename\* | String | 5-49 | Describes the Remitter’s local Statename. |
| Lcityname\* | String | 5-49 | Describes the Remitter’s local Cityname. |
| addressproof | string | 5-199 | Describes the Remitter’s address proof Scan Copy Name   |  | | --- | | 1-Voters’ PHOTOID Card of same address(valid for both Address and Identity) | | 2-Valid Driving License of Same Address(valid for both Address and Identity) | | 3-Valid Passport of Same Address(valid for both Address and Identity) | | 4-Family Ration Card | | 5-Utility bill(less than 3 months old) | | 6-Letter of Introduction /Confirmation from Post Office(valid for both Address and Identity) | | 7-Letter of Introduction by Local Self Government Official(valid for both Address and Identity) | |
| idproof | string | 5-199 | Describes Name of the id proof Scan Copy   |  | | --- | | 8-PAN Card | | 9-Voters’ PHOTOID Card of different address | | 10-Valid Driving License of different Address | | 11-Valid Passport of different Address | | 12-Aadhaar Card | |
| idproofnumber | String | 5-49 | Describes the remitter id proof number.[M] |
| idproofissuedate | datetime | - | Describes the remitter id proof issue date.  Only for Driving License(02,10) and Passport(03,11)  Format Should be MM/DD/YYYY |
| idproofexpirydate | datetime | - | Describes the remitter id proof expiry date. Only for Driving License(02,10) and Passport(03,11)  Format Should be MM/DD/YYYY |
| idproofissueplace | String | 5-49 | Describes the remitter id proof issue place. |
| **Response Field** | | | |
| **Field Name** | **Data Type** | **Length** | **Description** |
| bcagent | string | 5-49 | Describes the BC Agent’s unique ID maintained by the Channel Partner. |
| remitterid | int | 1-10 | Describes the Remitter’s unique ID. |
| status | int | 1 | 0- failure  1-success |

***Important Points –***

* *Remitter Mobile Number should be unique & correct for each request.*
* *OTP Message for Remitter Validation will be communicated to remitter mobile number.*
* *Remitter has to share the OTP code with BC Agent for verification of registration.*

Remitter will receive following messages from RBL Bank on Registered Mobile Number

|  |  |
| --- | --- |
| 1 | Remitter Registration / Validation |
| 2 | Beneficiary Registration / Validation |
| 3 | NEFT transaction- Pass/Fail |
| 4 | IMPS Transaction- Pass/Fail |
| 5 | Beneficiary Deletion |
| 6 | OTP Code for failure transaction for refund |

# Remitter Validation

This API is used to validate the remitter with OTP Code. Remitter has to input verification code sent on registered mobile number.

**Request XML -**

<remitterregistrationvalidatereq>

<header>

<sessiontoken></sessiontoken>

</header>

<remitterid></remitterid>

<mobilenumber></mobilenumber>

<verficationcode></verficationcode>

<channelpartnerrefno></channelpartnerrefno>

</remitterregistrationvalidatereq>

**Response XML -**

<remitterregistrationvalidateres>

<remitterid></remitterid>

<status></ status >

<Enrollmentfee></Enrollmentfee>

</remitterregistrationvalidatereq>

**XSD**



|  |  |  |  |
| --- | --- | --- | --- |
| **Request Fields** | | | |
| **Field Name** | **Data Type** | **Length** | **Description** |
| Session Token\* | String | 1-200 | Session Token received in Login Response |
| remitterid | Int | 1-10 | Describes the Remitter’s unique ID. |
| mobilenumber | int | 10 | Describes the BC Agent Remitter’s Registered Mobile Number. |
| verficationcode | int | 1-10 | Describes code send to Remitter’s Registered Mobile Number |
| channelpartnerrefno | String | 5-29 | Describes that Channel Partner Reference id. |
| **Response Field** | | | |
| **Field Name** | **Data Type** | **Length** | **Description** |
| remitterid | int | 1-10 | Describes the Remitter’s unique ID. |
| status | int | 1 | 0-failure 1-success |
| Enrollmentfee | Money |  | The amount to be collected from remitter. |

***Important Points***

* *Remitter has to share the OTP code send by RBL Bank to BC Agent for Remitter Verification*
* *After remitter Validation, Remitter will be able to register beneficiary*

# Remitter Resend OTP

This API will be used to resend the OTP code in case customer has not received OTP.

**Request XML -**

<remitterregistrationresendotpreq>

<header>

<sessiontoken></sessiontoken>

</header>

<remitterid></remitterid>

</remitterregistrationresendotpreq>

**Response XML -**

<remitterregistrationresendotpres>

<status></ status>

</remitterregistrationresendotpres>

**XSD**

****

|  |  |  |  |
| --- | --- | --- | --- |
| **Request Fields** | | | |
| **Field Name** | **Data Type** | **Length** | **Description** |
| Session Token\* | String | 1-200 | Session Token received in Login Response |
| Remitterid\* | Int | 1-10 | Describes the Remitter’s unique ID. |
| **Response Field** | | | |
| **Field Name** | **Data Type** | **Length** | **Description** |
| Status | integer | 1 | Status (1- Success/0- Failed) |

***Important Points*** *1.If Remitter has not received OTP message then agent can use resend API to get the OTP message again*

# Beneficiary Registration

This API will be used for Beneficiary Registration.

**Request XML -**

<beneficiaryregistrationreq>

<header>

<sessiontoken></sessiontoken>

</header>

<bcagent></bcagent>

<remitterid></remitterid>

<beneficiaryname></beneficiaryname>

<beneficiarymobilenumber></beneficiarymobilenumber>

<beneficiaryemailid></beneficiaryemailid>

<relationshipid></relationshipid>

<ifscode></ifscode>

<accountnumber></accountnumber>

<mmid></mmid>

<flag></flag>

</beneficiaryregistrationreq>

**Response XML -**

<beneficiaryregistrationres>

<bcagent></bcagent>

<remitterid></remitterid>

<beneficiaryid></beneficiaryid>

<status></status>

</beneficiaryregistrationres>

**XSD**

****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Request Fields** | | | | |
| **Field Name** | **Data Type** | | **Length** | **Description** |
| sessiontoken | | string | 5-199 | Describes the session token created by login API. |
| bcagent | | String | 5-49 | Describes the BC Agent’s unique ID maintained by the Channelpartner |
| remitterid | | int | 1-10 | Describes the Remitter’s unique ID. |
| beneficiaryname | | String | 5-49 | Describes Name of Receiver. |
| Beneficiarymobilenumber | | int | 10 | Describes Mobile Number of Receiver. |
| beneficiaryemailid | | String | 5-49 | Describes Receiver emailed. |
| relationshipid | | Int | 1-2 | Describes the Relationship for Receiver.   |  |  | | --- | --- | | 0 | Other | | 1 | Father | | 2 | Mother | | 3 | Husband | | 4 | Wife | | 5 | Brother | | 6 | Sister | | 7 | Son | | 8 | Daughter | | 9 | Uncle | | 10 | Aunty | | 11 | Father in law | | 12 | Mother in law | | 13 | Sister in law | | 14 | Brother in law | | 15 | Daughter in law | | 16 | Grandfather | | 17 | Grandmother | | 18 | Grandson | | 19 | Grand daughter | | 20 | Supplier | | 21 | Customer | | 22 | Retail Buyer | | 23 | Aggregator | | 24 | Employee | | 25 | Employer | |
| ifscode | | String | 5-29 | Describes the Beneficiary’s Bank IFSC code. |
| bank | | String | 5-49 | Describes the Beneficiary’s Bank Name.(optional) |
| state | | String | 5-49 | Describes the Beneficiary’s Bank State name.(optional) |
| city | | String | 5-49 | Describes the Beneficiary’s Bank City Name.(optional) |
| branch | | String | 5-49 | Describes the Beneficiary’s Bank Branch Name.(optional) |
| address | | String | 5-199 | Describes the Beneficiary’s Bank Address Details.(optional) |
| accountnumber | | int | 1-20 | Describes Beneficiary’s Bank Account Number. |
| mmid | | Int | 1-10 | Describes Beneficiary’s Mobile Money Identifier number (optional). |
| flag | | int | 1 | 2 – Will get registered for NEFT/IMPS |
| **Response Fields** | | | | |
| **Field Name** | **Data Type** | | **Length** | **Description** |
| bcagent | String | | 5-49 | Describes the BC Agent’s unique ID maintained by the channelpartner |
| remitterid | int | | 1-10 | Describes the Remitter’s unique ID. |
| beneficiaryid | int | | 1-10 | Describes the Beneficiary’s unique ID. |
| status | int | | 1 | 0-failure 1- Success |

***Important Point:-***

1. *Remitter has to provide the details of Beneficiary account to BC Agent for registration*
2. *Remitter will have to provide the Receiver Name, Account Number, IFSC Code, Relationship ID*
3. *Remitter will receive the OTP message on his registered mobile number for validation.*
4. *Remitter has to share the OTP code to BC Agent for beneficiary validation.*
5. *After validation of beneficiary remitter can transfer the money to receiver.*
6. *This API lets you register the maximum 10 beneficiary. If remitter required registering more than 10 beneficiaries, remitter has to delete one out of 10.*
7. *MMID Tag is optional. Please keep this tag as empty or pass unique value always.*

# Beneficiary Validation

This API will be used to validate the beneficiary with OTP.

**Request XML -**

<beneficiaryregistrationvalidatereq>

<header>

<sessiontoken></sessiontoken>

</header>

<beneficiaryid></beneficiaryid>

<verficationcode></verficationcode>

</beneficiaryregistrationvalidatereq>

**Response XML -**

<beneficiaryregistrationvalidateres>

<beneficiaryid></beneficiaryid>

<status></status>

</beneficiaryregistrationvalidateres>

**XSD**

****

|  |  |  |  |
| --- | --- | --- | --- |
| **Request Fields** | | | |
| **Field Name** | **Data Type** | **Length** | **Description** |
| sessiontoken | string | 5-199 | Describes the session token created by login API |
| beneficiaryid | int | 1-10 | Describes the Beneficiary’s unique ID. |
| verficationcode | int | 1-10 | Describes code send to Beneficiary’s Registered Mobile Number |
| **Response Fields** | | | |
| **Field Name** | **Data Type** | **Length** | **Description** |
| beneficiaryid | String | 1-10 | Describes the Beneficiary’s unique ID. |
| Status | integer | 1 | Status (1- Success/0- Failed) |

***Important Points:-***

* *Remitter has to share the OTP shared by RBL bank for beneficiary validation.*
* *After that remitter can transfer money to receiver*

# Beneficiary Resend OTP

This API will be used to get the OTP ,if remitter not received at first time for beneficiary registration.

**Request XML -**

<beneficiaryresendotpreq>

<header>

<sessiontoken></sessiontoken>

</header>

<remitterid></remitterid>

<beneficiaryid></beneficiaryid>

</beneficiaryresendotpreq>

**Response XML -**

<beneficiaryresendotpres>

<status></status>

</beneficiaryresendotpres>

**XSD**

****

|  |  |  |  |
| --- | --- | --- | --- |
| **Request Fields** | | | |
| **Field Name** | **Data Type** | **Length** | **Description** |
| sessiontoken | string | 5-199 | Describes the sessionid created with login API |
| remitterid | int | 1-10 | Describes the Remitter’s unique ID. |
| beneficiaryid | int | 1-10 | Describes the Beneficiary’s unique ID. |
| **Response Fields** | | | |
| **Field Name** | **Data Type** | **Length** | **Description** |
| Status | integer | 1 | Status (1- Success/0- Failed) |

# Transaction

This API is used to Transfer the money to the benificiary and here possible transfers are (IMPS and NEFT).

**Request XML -**

<transactionreq> <header>

<sessiontoken></sessiontoken>

</header>

<bcagent> </bcagent>

<remitterid></remitterid>

<beneficiaryid></beneficiaryid><amount></amount>

<remarks> </remarks>

<cpid></cpid>

<channelpartnerrefno></channelpartnerrefno>

<flag></flag>

</transactionreq>

**Response XML –**

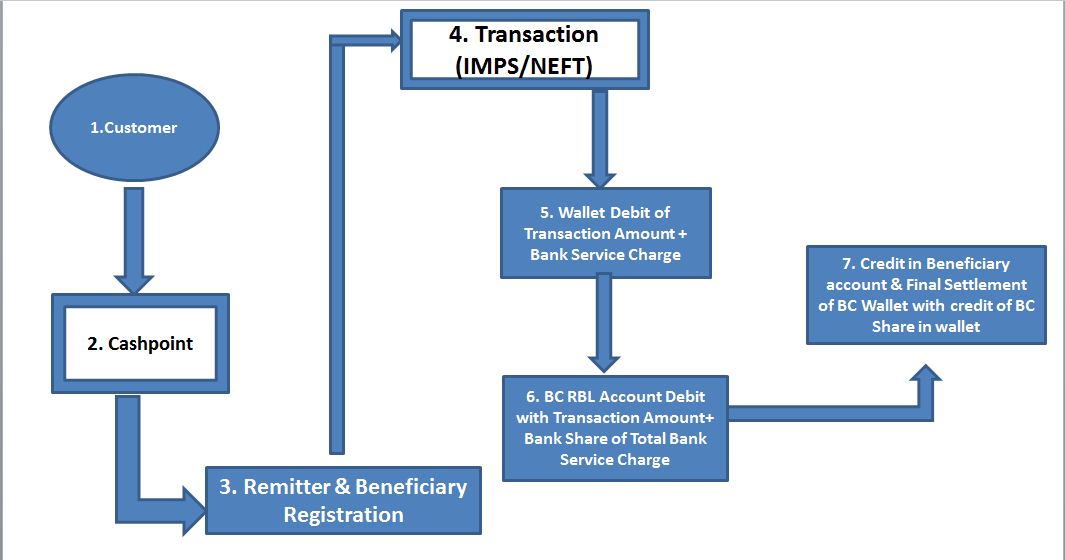
|  |  |
| --- | --- |
| IMPS:-  <transactionres> <channelpartnerrefno></channelpartnerrefno>  <RBLtransactionid> </RBLtransactionid>  <status></status>  <amount></amount>  <servicecharge><servicecharge>  <grossamount></grossamount>  <kycstatus></kycstatus>  <remarks></remarks>  <bankrefno></bankrefno> <NPCIResponsecode></NPCIResponsecode> </transactionres> | NEFT:-  <transactionres>  <channelpartnerrefno> </channelpartnerrefno>  <RBLtransactionid> </RBLtransactionid>  <status></status>  <amount></amount>  <servicecharge></servicecharge>  <grossamount></grossamount>  <kycstatus></kycstatus>  <remarks> </remarks>  <bankrefno> </bankrefno>  <UTRNo></UTRNo> </transactionres> |
|
|
|
|
|
|
|
|
|

**XSD**



|  |  |  |  |
| --- | --- | --- | --- |
| **Request Fields** | | | |
| **Field Name** | **Data Type** | **Length** | **Description** |
| sessiontoken | string | 5-199 | Describes the session token created with login API. |
| bcagent | int | 5-49 | Describes the name of BC agent. |
| remitterid | int | 1-10 | Describes the Remitter’s unique ID. |
| beneficiaryid | int | 1-10 | Describes the Beneficiary’s unique ID. |
| amount | int | 1-10 | Describes amount of Transaction. |
| remarks | string | 5-199 | Describes any Remarks about Transaction |
| channelpartnerrefno | string | 5-29 | Describes the channel partner unique transaction reference id. |
| flag | int | 1 | 2 - IMPS (IFSC) 3 - NEFT |
| **Response Field** | | | |
| **Field Name** | **Data Type** | **Length** | **Description** |
| channelpartnerrefno | string | 5-29 | Describes the channel partner unique transaction reference id. |
| rbltransactionid | int | 1-10 | Unique ID Generated by Admin. |
| Status | int | 1 | 0-failure  1-success  -1-This is not Success, This is not failure. If -1 received need to invoke the Transaction requery method. |
| amount | Money |  | Describes the remittance amount |
| servicecharge | Money |  | Describes the service change for the remittance amount |
| grossamount | Money |  | Describes the gross amount to be collected from the sender |
| kycstatus | Int | 1 | 0-Customer not Verified/Pending  1-KYC is under process  3-KYC Reject by bank  4-KYC Approved by BC Partner  5-KYC Rejected by BC Partner  7- kyc verified& Approved by bank ,KYC Process Completed |
| remarks | String | 1-50 | Describes the Remarks about the Txn |
| bankrefno | String | 1-50 | Describes the RRN of the Transaction given by bank |
| NPCIResponsecode | String | 1-50 | Response code given by NPCI for IMPS transactions |
| UTRNo | String | 1-50 | Used for Reconciliation purpose (NEFT) |

**Transaction Flow**

****

***Important Points:-***

*1. Make sure that Session token is same as login session.*

*2. Make sure that you are passing the correct remitter & Beneficiary ID which is validated under RBL*

*3. KYC Customer can transfer up to 50000 & Non KYC customer can transfer up to 5000 Rs.*

*4. CPID will be provided by RBL bank for testing*

*5. Channel Partner reference should be unique, later this number will be required to know the status of transaction*

*6. RBL transaction ID will be unique for transaction*

*7. Status “1” indicates that transaction is success, 0-failue,-1 –Unknown (Exception/Connectivity issues)*

*8. Gross amount will be sum of transaction amount & bank service charges*

*9. Bank Reference is RRN*

*10. NPCI Response code TAG will let you know the Response code sent by NPCI for IMPS Transaction.*

*-Kycstatus tag will give the customer kyc status.*

*11.* ***00 Response code with remarks SUCCESSSuccess indicates that transaction is completely success & credited to beneficiary account.***

***12. 91 Response code with remarks SUCCESS Time Out indicates that transaction is Success & credited but confirmation of beneficiary bank is pending for amount credit.***

***13. IN case of 91 Transaction, RBL clearing team will update the status of 91 transactions in T+2.***

***13.1 If IMPS transaction with response code 91 becomes success, reports will get automatically update with status CREDITED & remarks SUCCESS.***

***13.2 If IMPS transaction with response code 91 becomes fail, reports will get automatically update with status rejected & remarks Failed.***

***• IF RBL Clearing team has marked the transaction 91 to rejected. RBL Will triggers a SMS for failure to remitter mobile number with OTP code, which will be used for wallet reversal.***

***• Transaction amount will be reversed after marking transaction 91 to reject.***

***14. If “–1” status received in response (Due to web exception or connectivity issues) need to invoke the Transaction requery method. If Transaction requery method responded with failure status (-1) to do the reconciliation of this transaction.***

***15. NEFT Transactions happened in NEFT hours, first the transactions will be in NEFT Initialized status (Payment status 1) and later the same transaction will gets updated to NEFT Credited (Payment status 2) or NEFT Rejected (Payment status 3)***

*16. NEFT Transactions happened in Non NEFT hours or In Holidays, first the transactions will be in NEFT Pending status (Payment status 0) and the same will be stored in RBL system.*

*When the next NEFT batch starts the same transaction will be updated to Initialized status and debit will happen to the BC account (Payment status 1) and later the same transaction will gets updated to NEFT Credited (Payment status 2) or NEFT Rejected (Payment status 3).—****As of now this functionality is available in UAT only***

*17. NEFT transaction status will be changed from*

*• NEFT Initialized status (Payment status 1) to NEFT Credited (Payment status 2)*

*• NEFT Initialized status (Payment status 1) to NEFT Rejected (Payment status 3)*

*• NEFT Credited (Payment status 2) to NEFT Rejected (Payment status 3)*

*18. IMPS IFT transactions are RBL to RBL transactions*

*19.* ***There will be 3 debits for each transaction (IMPS/NEFT/IMPS IFT)***

***• Transaction amount***

***• Bank commission***

***• Bank service tax***

**20.** As per NPCI mandate, the max response time for a single IMPS transaction is 120 sec. Please change the logic accordingly.

You will get response within 40 to 60 secs or less for a Normal IMPS transactions case, but when timeout happens at bene bank/NPCI you will get the response max by 120 secs.

# Refund OTP

This API is used to resend the transaction OTP if remitter is not received it first time.

**Request XML -**

<refundotpreq>

<header>

<sessiontoken></sessiontoken>

</header>

< RBLtransactionid ></ RBLtransactionid >

</refundotpreq>

**Response XML -**

< refundotpres >

<status></status>

</refundotpres>

XSD



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Request Fields | | | | |
| Field Name | | Data Type | Length | Description |
| sessiontoken | | string | 5-199 | Describes the session token created with login API |
| rbltransactionid | | int | 5-9 | Unique ID Generated by Admin. |
| **Response Field** | | | | |
| **Field Name** | **Data Type** | | **Length** | **Description** |
| status | int | | 1 | 0- failure  1-success |

***Important Points:-***

* *Refund Request API is to generate refund OTP for remitter*
* *Remitter will receive OTP for Failed transaction that will be used for remitter refund.*

# Refund

This API will be used to refund the money to the remitter**.**

**Request XML -**

<refundreq>

<header>

<sessiontoken></sessiontoken>

</header>

<bcagent></bcagent>

<channelpartnerrefno></channelpartnerrefno>

<verficationcode></verficationcode >

<flag></flag>

</refundreq>

**Response XML -**

<refundres>

<bcagent></bcagent>

<amount></amount>

<tamount></tamount>

<servicecharge></servicecharge>

<status></status>

</refundres>

**XSD**

****

|  |  |  |  |
| --- | --- | --- | --- |
| **Request Fields** | | | |
| **Field Name** | **Data Type** | **Length** | **Description** |
| sessiontoken | string | 5-199 | Describes the session token created with login API. |
| bcagent | string | 5-49 | Describes the name of BC agent. |
| channelpartnerrefno | string | 5-49 | Describes the channel partner reference id. |
| verificationcode | int | 1-10 | Describes code send to Remitter’s Registered Mobile Number |
| amount | int | 1-10 | Describes the transaction amount. |
| flag | int | 1 | 1-Remittance |
| **Response Fields** | | | |
| **Field Name** | **Data Type** | **Length** | **Description** |
| bcagent | string | 5-49 | Describes the name of BC agent. |
| amount | int | 1-10 | Describes the transaction amount. |
| tamount | int | 1-10 | Describes the transaction total amount. |
| servicecharge | int | 1-10 | Describes the transaction service  charge amount. |
| status | int | 1 | 0-failure 1-success |
| description | string | 5-199 | Status Description. |

***Important Points –***

* *This method will initiate refund process for remitter.*
* *Only remitter has to share the OTP with BC agent.*

# Remitter Details

This API is used to get the registered remitter and beneficiary details of same remitter.

**Request XML -**

<remitterdetailsreq>

<header>

<sessiontoken></sessiontoken>

</header>

<bcagent></bcagent>

<mobilenumber></mobilenumber>

<flag></flag>

</remitterdetailsreq>

**Response XML -**

<remitterdetailsres>

<bcagent></bcagent>

<status></status>

<remitterdetail>

<remitterid></remitterid>

<remittername></remittername>

<remitteraddress1></remitteraddress1>

<remitteraddress2></remitteraddress2>

<pincode></pincode>

<cityid></cityid>

<stateid></stateid>

<alternatenumber></alternatenumber>

<idproof></idproof>

<idproofnumber></idproofnumber>

<idproofissuedate></idproofissuedate>

<idproofexpirydate></idproofexpirydate>

<idproofissueplace></idproofissueplace>

<laddress></laddress>

<lpincode></lpincode>

<lcity></lcity>

<lstate></lstate>

<remitterstatus></remitterstatus>

<kycstatus></kycstatus>

<consumedlimit></consumedlimit>

<remaininglimit></remaininglimit>

<kycremarks></kycremarks>

<Title></Title>

<Gender></Gender>

<Religion></Religion>

<Category></Category>

<Education></Education>

<MiddleName></MiddleName>

<LastName></LastName>

<NameofNominee></NameofNominee>

<Relationship></Relationship>

<NomineeAge></NomineeAge>

<NomineeDateofBirth></NomineeDateofBirth>

<MaidenName></MaidenName>

<FatherOrSpouseName></FatherOrSpouseName>

<MotherName></MotherName>

<DateofBirth></DateofBirth>

<MaritalStatus></MaritalStatus>

<Nationality></Nationality>

<ResidentialStatus></ResidentialStatus>

<Emailid></Emailid>

</remitterdetail>

<beneficiarydetail>

<beneficiary>

<beneficiaryid></beneficiaryid>

<beneficiaryname></beneficiaryname>

<beneficiarymobilenumber></beneficiarymobilenumber>

<beneficiaryemailid></beneficiaryemailid>

<relationshipid></relationshipid>

<bank></bank>

<state></state>

<city></city>

<branch></branch>

<address></address>

<ifscode></ifscode>

<accountnumber></accountnumber>

<mmid></mmid>

<beneficiarystatus></beneficiarystatus>

<impsstatus></impsstatus>

</beneficiary>

</beneficiarydetail>

</remitterdetailsres>

**XSD**

****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Request Fields** | | | | |
| **Field Name** | | **Data Type** | **Length** | **Description** |
| sessiontoken | | string | 5-199 | Describes the Channel partner usertrack id in encrypted format created by Admin. |
| bcagent | | string | 5-49 | Describes the name of BC agent. |
| mobilenumber | | int | 1-10 | Describes the mobilenumber of remitter. |
| flag | | int | 1 | 1-Remittance |
| **Response Field** | | | | |
| **Field Name** | **Data Type** | | **Length** | **Description** |
| bcagent | String | | 5-49 | Describes the BC Agent’s unique ID maintained by the Channel Partner. |
| remitterid | String | | 5-49 | Describes the BC Agent Remitter’s ID. |
| remittername | String | | 5-49 | Describes the BC Agent Remitter’s Name. |
| remitteraddress1 | String | | 5-199 | Describes the BC Agent Remitter’s Address. |
| remitteraddress2 | String | | 5-199 | Describes the BC Agent Remitter’s Address. |
| pincode | int | | 1-10 | Describes the BC Agent Remitter’s Pincode. |
| city | String | | 5-49 | Describes the BC Agent Remitter’s City. |
| state | String | | 5-49 | Describes the BC Agent Remitter’s State. |
| alternatenumber | int | | 1-10 | Describes the BC Agent Remitter’s Other Number. |
| idproof | String | | 1-30 | Describes the BC Agent Remitter’s Idproof |
| idproofnumber | String | | 1-30 | Describes the BC Agent Remitter’s idproof Number |
| idproofissuedate | Date time | |  | Describes the BC Agent Remitter’s idproof Date of Issue.(format:mm/dd/yyyy) |
| idproofexpirydate | Date time | |  | Describes the BC Agent Remitter’s idproof Date of Expiry.(format:mm/dd/yyyy) |
| idproofissueplace | String | | 5-49 | Describes the BC Agent Remitter’s idproof Place of Issue. |
| laddress | String | | 5-199 | Describes the BC Agent Remitter’s local Address. |
| lpincode | int | | 1-10 | Describes the BC Agent Remitter’s local Pincode number. |
| lstate | String | | 5-49 | Describes the BC Agent Remitter’s local statename. |
| lcity | String | | 5-49 | Describes the BC Agent Remitter’s local cityname. |
| remitterstatus | int | | 1 | 0-pending  1-Active  2-reject |
| kycstatus | int | | 1 | 0-Customer not Verified/Pending  1-KYC is under process  3-KYC Reject by bank  4-KYC Approved by BC Partner  5-KYC Rejected by BC Partner  7-kyc verified& Approved by bank ,KYC Process Completed |
| beneficiaryname | String | | 5-49 | Describes Name of Receiver. |
| consumedlimit | Money | |  | Describes the remitter consumed amount |
| remaininglimit | Money | |  | Describes the remaining limit |
| kycremarks | String | | 5-49 | Describes the KYC approval status |
| Title | String | | 5-20 | Describes the title |
| Gender | String | | 5-20 | Describes the gender of remitter |
| Religion | String | | 5-20 | Describes the gender of remitter |
| Category | String | | 5-20 | Describes the Category of remitter |
| Education | String | |  | Describes the Education status of remitter |
| MiddleName | String | | 5-49 | Describes the Middle name of remitter |
| LastName | String | | 5-49 | Describes the Last Name of remitter |
| NameofNominee | String | | 5-49 | Describes the Nominee name of remitter |
| Relationship | String | | 5-20 | Describes the nominee relationship with remitter |
| NomineeAge | int | | 1-10 | Describes the age of remitter |
| NomineeDateofBirth | Date time | |  | Describes the Nominee dateofbirth |
| MaidenName | String | | 5-49 | Describes the Maiden name of remitter |
| FatherOrSpouseName | String | | 5-49 | Describes the Father/Spouse of remitter |
| MotherName | String | | 5-49 | Describes the Mother name of remitter |
| DateofBirth | Date time | |  | Describes the Date of birth of remitter |
| MaritalStatus | String | | 10-30 | Describes the Marital of remitter |
| Nationality | String | | 5-30 | Describes the Nationality of remitter |
| ResidentialStatus | String | | 5-30 | Describes the Residential status of remitter |
| Emailid | String | | 5-49 | Describes Sender email id. |
| beneficiarymobilenumber | int | | 1-10 | Describes Mobile Number of Receiver. |
| beneficiaryemailid | String | | 5-49 | Describes Receiver email id. |
| relationshipid | String | | 5-49 | Describes the Relationship for Receiver. |
| bank | String | | 5-49 | Describes the Beneficiary’s Bank Name. |
| state | String | | 5-49 | Describes the Beneficiary’s Bank State name. |
| city | String | | 5-49 | Describes the Beneficiary’s Bank City Name. |
| branch | String | | 5-49 | Describes the Beneficiary’s Bank Branch Name. |
| address | String | | 5-199 | Describes the Beneficiary’s Bank Address Details. |
| ifscode | String | | 1-30 | Describes the Beneficiary’s Bank IFSC code. |
| accountnumber | int | | 1-20 | Describes Beneficiary’s Bank Account Number. |
| mmid | Int | | 1-10 | Describes Military Person ID number. |
| beneficiarystatus | int | | 1 | 0-pending  1-active  2-reject |
| impsstatus | int | | 1 | 0-pending  1-active |
| status | int | | 1 | 0- failure  1-success |

***Important Points-***

* *This method will be used to get complete remitter detail along with registered & active beneficiary detail.*
* *Kycstatus tag will gives the customer kyc status*

# Delete Beneficiary

This API will be used to delete the beneficiary.

**Request XML -**

<beneficairydeletereq>

<header>

<sessiontoken></sessiontoken>

</header>

<remitterid></remitterid>

<beneficairyid></beneficairyid>

</beneficairydeletereq>

**Response XML -**

<beneficiarydeleteres>

<beneficiaryid></beneficiaryid>

<status></status>

</beneficiarydeleteres>

***XSD***

******

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Request Fields | | | | |
| Field Name | | Data Type | Length | Description |
| sessiontoken | | string | 5-199 | Describes the session token created with login API. |
| remitterid | | int | 1-10 | Describes the Remitter’s unique ID. |
| beneficiaryid | | int | 1-10 | Describes the Beneficiary’s unique ID. |
| **Response Field** | | | | |
| **Field Name** | **Data Type** | | **Length** | **Description** |
| beneficiaryid | Int | | 1-10 | Describes the Beneficiary’s unique ID. |
| Status | Int | | 1 | 0- failure 1-success |

***Important Points:-***

* *Remitter can ask BC Agent to delete the any beneficiary registered under RBL System.*
* *Remitter has to share the OTP for beneficiary deletion with BC Agent*
* *Maximum count of active beneficiary can be 10.*

# Delete Beneficiary Validation

This API is used to validate the beneficiary with OTP before deletion.

**Request XML -**

<beneficairydeletevalidationreq>

<header>

<sessiontoken> </sessiontoken>

</header>

< beneficiaryid></beneficiaryid>

<verficationcode></verficationcode>

</beneficairydeletevalidationreq>

**Response XML -**

< beneficiarydeletevalidationres>

<status><status>

</ beneficiarydeletevalidationres>

***XSD***

******

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Request Fields | | | | |
| Field Name | | Data Type | Length | Description |
| sessiontoken | | string | 5-199 | Describes the session token created with login API. |
| beneficiaryid | | Int | 1-10 | Describes the Beneficiary’s Unique ID. |
| verficationcode | | Int | 1-10 | Describes code send to Remitter’s Registered Mobile Number. |
| **Response Field** | | | | |
| **Field Name** | **Data Type** | | **Length** | **Description** |
| Status | Int | | 1 | 0- failure 1-success |

***Important Points:-***

* *Beneficiary validation will be done along with validation.*
* *Remitter has to share the OTP message with BC agent for deletion of beneficiary*.

# Transaction Re-Query

This is the XML service used to get the updated status of the Transaction

**Request XML -**

<transactionrequeryreq>

<header>

<sessiontoken></sessiontoken>

</header>

<bcagent></bcagent>

<channelpartnerrefno></channelpartnerrefno>

</transactionrequeryreq>

**Response XML -**

**IMPS -**

<transactionrequeryres>

<bcagent></bcagent>

<bcagentname></bcagentname>

<transactiondt></transactiondt>

<transactionid></transactionid>

<amount></amount>

<servicechrg></servicechrg>

<tamount></tamout>

<paymentstatus></paymentstatus>

<remittername></remittername>

<remittermblno></remittermblno>

<beneficiaryname></beneficiaryname>

< relationship></relationship>

< relationshiptype></ relationshiptype>

<bank></bank>

<ifsccode></ifsccode>

<accountnumber></accountnumber>

<branch></branch>

<address1></address1>

<address2></address2>

<idproof></idproof>

<idproofnumber></idproofnumber>

<idproofissuedate></idproofissuedate>

<idproofexpdate></idproofexpdate>

<status><status>

<BankReferenceNo></BankReferenceNo>

<TranType> </TranType>

<BankRemarks> </BankRemarks>

</transactionrequeryres>

**NEFT –**

<transactionrequeryres>

<bcagent></bcagent>

<bcagentname></bcagentname>

<transactiondt></transactiondt>

<transactionid></transactionid>

<amount></amount>

<servicechrg></servicechrg>

<tamount></tamout>

<paymentstatus></paymentstatus>

<remittername></remittername>

<remittermblno></remittermblno>

<beneficiaryname></beneficiaryname>

< relationship></relationship>

< relationshiptype></ relationshiptype>

<bank></bank>

<ifsccode></ifsccode>

<accountnumber></accountnumber>

<branch></branch>

<address1></address1>

<address2></address2>

<idproof></idproof>

<idproofnumber></idproofnumber>

<idproofissuedate></idproofissuedate>

<idproofexpdate></idproofexpdate>

<status><status>

<UTRNo> </UTRNo>

<PONum></PONum>

<BankReferenceNo> </BankReferenceNo>

<TranType> </TranType>

<BankRemarks> </BankRemarks>

</transactionrequeryres>

***XSD***

******

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Request Fields | | | | |
| Field Name | | Data Type | Length | Description |
| sessiontoken | | string | 5-199 | Describes the session token created with login API. |
| Bcagent | | String | 5-29 | Describes the agent id |
| channelpartnerrefno | | String | 5-29 | Describes that Channel Partner Reference id. [M] |
| **Response Field** | | | | |
| **Field Name** | **Data Type** | | **Length** | **Description** |
| bcagentid | string | | 5-49 | Describes the id of BC agent. |
| bcagentname | string | | 5-49 | Describes the Name of BC agent. |
| transactiondt | Date time | | 1-10 | Describes the Date of Transaction.(format:mm/dd/yyyy) |
| transactionid | int | | 1-10 | Describes the ID of Transaction. |
| amount | int | | 1-10 | Describes the Amount of Transaction. |
| servicechrg | int | | 1-10 | Describes the Service Charge of Transaction. |
| tamount | int | | 1-10 | Describes the Total Amount of Transaction. |
| paymentstatus | int | | 1 | 0-pending  1-processing  2-credited  3-refund  4-refund process |
| remittername | String | | 5-49 | Describes the BC Agent Remitter’s Name. |
| remittermobilenumber | int | | 1-10 | Describes the BC Agent Mobile Number. |
| beneficiaryname | string | | 5-49 | Describes the Name of Receiver. |
| relationship | String | | 1-30 | Describes the relationship id. |
| relationshiptype | string | | 1-30 | Describes the type of relationship. |
| bank | String | | 5-49 | Describes the Beneficiary’s Bank Name. |
| ifscode | String | | 1-30 | Describes the Beneficiary’s Bank IFSC code. |
| accountnumber | int | | 5-49 | Describes Beneficiary’s Bank Account Number. |
| branch | String | | 5-49 | Describes the Beneficiary’s Bank Branch Name. |
| address1 | string | | 5-199 | Describes Beneficiary’s Address. |
| address2 | string | | 5-199 | Describes Beneficiary’s Address. |
| idproof | String | | 1-30 | Describes the BC Agent Remitter’s Idproof |
| idproofnumber | String | | 1-30 | Describes the BC Agent Remitter’s idproof Number |
| idproofissuedate | Date time | | 1-10 | Describes the BC Agent Remitter’s idproof Date of Issue.(format:mm/dd/yyyy) |
| idproofexpdate | Date time | | 1-10 | Describes the BC Agent Remitter’s idproof Date of Expiry.(format:mm/dd/yyyy) |
| status | int | | 1 | 0- failure  1-success  91-Unknown cases(Operational reconciliation taken place) |
| PONum | string | | 1-50 | Used for Reconciliation purpose |
| BankReferenceNo | string | | 1-50 | Describes the RRN of the Transaction given by bank |
| UTRNO | string | | 1-50 | Used for Reconciliation purpose |
| TRAN TYPE | string | | 1-50 | NEFT/IMPS |
| BANK REMARKS | string | | 1-200 | Describes the Transaction Remarks |
| bcagentid | string | | 5-49 | Describes the id of BC agent. |
| bcagentname | string | | 5-49 | Describes the Name of BC agent. |
| transactiondt | Date time | | 1-10 | Describes the Date of Transaction.(format:mm/dd/yyyy) |
| transactionid | int | | 1-10 | Describes the ID of Transaction. |
| amount | int | | 1-10 | Describes the Amount of Transaction. |
| servicechrg | int | | 1-10 | Describes the Service Charge of Transaction. |
| tamount | int | | 1-10 | Describes the Total Amount of Transaction. |
| paymentstatus | int | | 1 | 0-pending  1-processing  2-credited  3-refund  4-refund process |
| remittername | String | | 5-49 | Describes the BC Agent Remitter’s Name. |
| remittermobilenumber | int | | 1-10 | Describes the BC Agent Mobile Number. |
| beneficiaryname | string | | 5-49 | Describes the Name of Receiver. |
| relationship | String | | 1-30 | Describes the relationship id. |
| relationshiptype | string | | 1-30 | Describes the type of relationship. |
| bank | String | | 5-49 | Describes the Beneficiary’s Bank Name. |
| ifscode | String | | 1-30 | Describes the Beneficiary’s Bank IFSC code. |
| accountnumber | int | | 5-49 | Describes Beneficiary’s Bank Account Number. |
| branch | String | | 5-49 | Describes the Beneficiary’s Bank Branch Name. |
| address1 | string | | 5-199 | Describes Beneficiary’s Address. |
| address2 | string | | 5-199 | Describes Beneficiary’s Address. |
| idproof | String | | 1-30 | Describes the BC Agent Remitter’s Idproof |
| idproofnumber | String | | 1-30 | Describes the BC Agent Remitter’s idproof Number |
| idproofissuedate | Date time | | 1-10 | Describes the BC Agent Remitter’s idproof Date of Issue.(format:mm/dd/yyyy) |
| idproofexpdate | Date time | | 1-10 | Describes the BC Agent Remitter’s idproof Date of Expiry.(format:mm/dd/yyyy) |
| status | int | | 1 | 0- failure  1-success  91-Unknown cases(Operational reconciliation taken place) |
| PONum | string | | 1-50 | Used for Reconciliation purpose |
| BankReferenceNo | string | | 1-50 | Describes the RRN of the Transaction given by bank |
| UTRNO | string | | 1-50 | Used for Reconciliation purpose |
| TRAN TYPE | string | | 1-50 | NEFT/IMPS |
| BANK REMARKS | string | | 1-200 | Describes the Transaction Remarks |

***Important points:-***

1. *Timeout is occurred at RBL end, API responded with failure status (-1).If Transaction requery method responded with failure status (-1) to do the reconciliation of this transaction.*
2. *Re-Query API method will not require the agent ID of the original transaction to be sent in the message.*
3. *Any BC Agent Can do Requery of any transaction but make sure that session created by BC Agent should be same.*
4. *In XML we are validating the only relation between Agent & Channel Partner reference numbers.*
5. *Payment status TAG will lets you know the exact status of transaction*

* *0 transaction is in NEFT pending status*
* *1- Transaction is in processing (I.e. Neft Initialized)*
* *2- Credited if transaction is credited to beneficiary*
* *3- refund status stated that transaction was failed & BC Account has got the credit of failed amount*
* *4-Refund Process is the final status of failed transaction if remitter has received the failed money after credit of BC Account.*

# Get Service Charge

This API is used to get the service charge for the transaction.

**Request XML -**

<getservicechargereq>

<header>

<sessiontoken></sessiontoken>

</header>

<bcagent></bcagent>

<remitterid></remitterid>

<beneficiaryid></beneficiaryid>

<amount></amount>

</getservicechargereq>

**Response XML -**

<getservicechargeres>

<servicecharge></servicecharge>

</getservicechargeres>

***XSD***

******

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Request Fields | | | | |
| Field Name | | Data Type | Length | Description |
| sessiontoken | | string | 5-199 | Describes the session token created with login API. |
| bcagent | int | | 5-49 | Describes the name of BC agent. |
| remitterid | int | | 1-10 | Describes the Remitter’s unique ID. |
| beneficiaryid | int | | 1-10 | Describes the Beneficiary’s unique ID. |
| amount | int | | 1-10 | Describes amount of Transaction. |
| beneficiaryid | | Int | 1-10 | Describes the Beneficiary’s Unique ID. |
| verficationcode | | Int | 1-10 | Describes code send to Remitter’s Registered Mobile Number. |
| **Response Field** | | | | |
| **Field Name** | **Data Type** | | **Length** | **Description** |
| servicecharge | string | | 1-10 | Service charge is collected from the customer at the time of domestic remittance. |

**Important Points-**

* *This method will let you know the exact service charge on Transaction amount*.

# Beneficiary Account Validation

This is the XML service exposed by the Channel Partners to check the account Name of beneficiary whether it is correct or not. This method will let you know the beneficiary name associated with beneficiary account.

**Request XML -**

<beneficiaryaccvalidationreq>

<header>

<sessiontoken></sessiontoken>

</header>

<bcagent> </bcagent>

<remitterid></remitterid>

<beneficiaryname> </beneficiaryname>

<beneficiarymobilenumber></beneficiarymobilenumber>

<accountnumber></accountnumber>

<ifscode> </ifscode>

<channelpartnerrefno></channelpartnerrefno>

</beneficiaryaccvalidationreq>

**Response XML -**

<beneficiaryaccvalidationres>

<status></status>

<benename> </benename>

<channelpartnerrefno></channelpartnerrefno>

<amount></amount>

<remarks> </remarks>

<bankrefno></bankrefno>

<NPCIResponsecode></NPCIResponsecode>

</beneficiaryaccvalidationres>

***XSD***



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Request Fields | | | | |
| Field Name | | Data Type | Length | Description |
| sessiontoken | | string | 5-199 | Describes the session token created with login API. |
| bcagent | | int | 5-49 | Describes the name of BC agent. |
| remitterid | | int | 1-10 | Describes the Remitter’s unique ID. |
| Beneficiary name | | String | 1-50 | Describes Name of Receiver. |
| Mobile Number | | int | 1-10 | Describes Mobile Number of Receiver. |
| Account Number | | int | 1-20 | Describes Beneficiary’s Bank Account Number. |
| IFSC Code | | string | 1-30 | Describes the Beneficiary’s Bank IFSC code. |
| Channel Partner Ref No | | string | 1-50 | Describes the channel partner reference id. |
| **Response Field** | | | | |
| **Field Name** | **Data Type** | | **Length** | **Description** |
| Status | int | | 1 | 0-failure  1-success |
| benename | String | | 1-30 | Describe Bene name |
| Channel Partner Ref no | string | | 1-50 | Describes the channel partner reference id. |
| Amount | money | |  | Describe the remittance amount for bene validation |
| Remarks | String | | 1-200 | Describe Remarks |
| Bank Ref No | int | | 0-20 | Describe bank reference number |
| NPCI response code | int | | 0-3 | Response code given by NPCI |

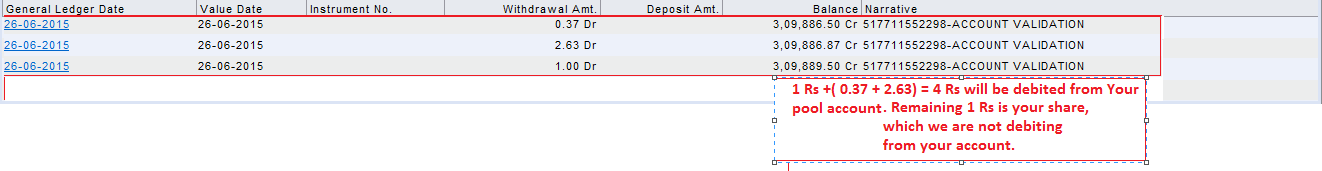
Please note if BC partner wants to use this service –

First they need to call the Bene account validation API after that they need to call Beneficiary Registration API.

Account validation pricing/process.

|  |  |  |
| --- | --- | --- |
| Structure of 5 Rs Debit | | |
| 1 | Out of 5 Rs.  4 Rs will get debited from your pool account. | Total 5 Rs |
| 2 | Your share of Re 1 will remain in your account. |
| Structure of 4 Rs Debit from your pool account | | |
| 1 | 1 Rs will get credited to Beneficiary account | Total 4 Rs |
| 2 | 3 Rs Bank Share (Service Tax + Commission amount) |
|  |  |

If Beneficiary account validation is failed, Your BC Account will get credited back by 1 Rs only (IMPS Trans Amt initiated). Bank will not refund Service tax & Commission amount in case of validation failed from NPCI. Actually there is no refund to end customer for failed transaction or validation. Beneficiary name availability is completely dependent on beneficiary bank and RBL Bank do not have any control on same. There are many banks that do not provide beneficiary name on validation. This point needs to be educated to CSPs so that they can communicate same to customer before initiating a/c validation.

******

# Common Error

**Response:**

<errorres>

<status> </status>

<description></description>

</errorres>

Status 0 or -1

The End