**RBL MONEY TRANSFER APPLICATION PROGRAMMING INTERFACE**

**FOR CP API-Limit Enhancement**

**REFERENCE MANUAL**

**VERSION 1.0**

**Document History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Changes** | |
| 1 | 06/05/2016 | Document Modified | P Stalin Babu |
| 2 | 01/11/2016 | NEFT Store and Forward | P Stalin Babu |

## TABLE OF CONTENTS

1. **INTRODUCTION**
   1. API DEFINITION
   2. ABOUT XML
2. [**RBL MONEY TRANSFER API SPECIFICATION**](#APISPECIFICATION)
   1. [CHANNEL PARTNER LOGIN](#CHANNELPARTNERLOGIN)
   2. [NETBANKING TRANSACTION](#NETBANKINGIMPSTRANSACTION)
   3. [TRANSACTION RE-QUERY](#TRANSACTIONREQUERY)

**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 DMR platform with the banking 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.**

**Sample URL:**

<https://apideveloper.rblbank.com/test/sb/rbl/v1/limit/enhance>

1. **CHANNEL PARTNER LOGIN:-**

This is the XML service exposed by the Channel Partners authentication to the RBL.

**Request:**

<channelpartnerloginreq>

<username></username>

<password></password>

<bcagent></bcagent>

</channelpartnerloginreq>

**XSD**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Mandatory** | **Length** | **Description** |
| username | String | Yes | 1-30 | Describes the Channel partner user name |
| password | String | Yes | 40 | Describes the Channel partner API password |
| bcagent | String | Yes | 1-50 | Describes the System BC Agent ID provided by RBL to the Channel Partner. |

**Response:**

<channelpartnerloginres>

<sessiontoken></sessiontoken>

<timeout></timeout>

<status></status>

</channelpartnerloginres>

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Length** | **Description** |
| sessiontoken | String | 1-200 | Session token generated with login API and can be used in all other API’s till its life time |
| 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 |

**Important Points:-**

* User name is BC user name of Channel Partner Company.
* Password is API password in format which will be shared by RBL Bank for UAT testing.
* BC Agent is Registered & Approved CSP under Partner and this will be provided by RBL bank
* There can be multiple sessions with different credentials
* Session Token will be created as per Username, API Password, time and Agent ID
* This session token will get expire in 1 hour?
* This Session token will be used by other API method to continue the task.

1. **NETBANKING TRANSACTION:-**

This is the XML service exposed for Channel partner to initiate the IMPS Transactions to customers. Here all transactions will be done basis on Beneficiary Account number and IFSCode .

**Request:**

<netbankingneft>

<header><sessiontoken></sessiontoken></header>

<remittermmid></remittermmid>

<remittername></remittername>

<remittermobilenumber></remittermobilenumber>

<remitteraddress1></remitteraddress1>

<remitteraddress2></remitteraddress2>

<pincode></pincode>

<city></city>

<state></state>

<beneficiaryaccountnumber></beneficiaryaccountnumber>

<beneficiaryifsc></beneficiaryifsc>

<beneficiarymobilenumber></beneficiarymobilenumber>

<beneficiarymailid></beneficiarymailid>

<beneficiarybank></beneficiarybank>

<beneficiaryname></beneficiaryname>

<amount></amount>

<remarks></remarks>

<channelpartnerrefno></channelpartnerrefno>

<parameter1></parameter1>

<parameter2></parameter2>

<flag></flag>

</netbankingneft>

**Xsd:**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Mandatory** | **Length** | **Description** |
| sessiontoken | String | Yes | 1-200 | Describes the Channel partner usertrack id in encrypted format created by Admin. |
| remittermmid | String | No | 1-20 | Describes the Remitter’s MMID |
| remittername | String | Yes | 5-29 | Describes the Remitter’s name. |
| remittermobilenumber | String | Yes | 5-15 | Describes the Remitter’s mobile number |
| remitteraddress1 | String | No | 1-200 | Describes the Address of Remitter. |
| remitteraddress2 | String | No | 1-200 | Describes the Address of Remitter. |
| pincode | String | No | 1-50 | Describes the Pincode of Remitter’s Area |
| city | String | No | 1-50 | Describes the City of Remitter. |
| state | String | No | 1-50 | Describes the State of Remitter. |
| beneficiaryaccountnumber | String | Yes | 5-29 | Describes the Beneficiary’s Account number |
| beneficiaryifsc | String | Yes | 11 | Describes the Beneficiary’s IFSC Code |
| beneficiarymobilenumber | String | Yes | 1-10 | Describes the Beneficiary’s mobile number |
| beneficiarymailid | String | No | 1-50 | Describes the Beneficiary’s Email Id |
| beneficiarybank | String | Yes | 5-29 | Describes the Beneficiary’s Bank |
| beneficiaryname | String | Yes | 5-49 | Describes the Beneficiary’s name |
| amount | Money | Yes | 1-10 | Describes amount of Transaction. |
| remarks | String | Yes | 1-200 | Describes any Remarks about  remitter Registration Transaction |
| channelpartnerrefno | String | Yes | 5-30 | Unique reference ID created by API Integrator for future requery reference. If transaction is timed out at API integrator based on this ID can request Transaction Requery for the same transaction. |
| parameter1 | String | No | 1-100 | This parameter for future use |
| parameter2 | String | No | 1-100 | his parameter for future use |
| flag | Int | Yes | 1 | 2-IMPS(IFSC) |

**IMPS Response**

<netbankingneftres>

<channelpartnerrefno></channelpartnerrefno>

<RBLtransactionid></RBLtransactionid>

<status></status>

<amount></amount>

<servicecharge></servicecharge>

<totalamount></totalamount>

<remarks></remarks>

<bankrefno></bankrefno>

<NPCIResponsecode></NPCIResponsecode>

</netbankingneftres>

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Length** | **Description** |
| channelpartnerrefno | String | 1-10 | Describes the channel partner reference information. |
| RBLtransactionid | String | 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 remittance service charge |
| totalamount | Money |  | Describes the remittance total amount |
| Remarks | String | 1-20 | Describes the remarks given by Bank |
| Bankrefno | String | 1-20 | Describes the Ref Number given by Bank |
| NPCIResponsecode | Int | 1-5 | Describes the Responsecode given by NPCI |

**Note: Timeout is occurred at RBL end, API responded with failure status (-1).** **If –1 received need to invoke the Transaction requery method.If Transaction requery method responded with failure status (-1) to do the manual reconciliation.**

**Sample Failure status Response:**

<errorres>

<status>-1</status>

<description>Transaction Failed</description>

</errorres>

**Important Points**

1. Channel Partner reference should be unique, later this no will be required to know the status of transaction by transaction requery API.

2. RBL transaction ID will be unique for transaction

3. Status “1” indicates that Response of API Request is success

4. Bank Reference is RRN for future for communication purpose

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

6. 00 Response code with remarks SuccessSUCCESS indicates that transaction is completely success & credited to beneficiary account.

7. 91 Response code with remarks SUCCESSTime Out indicates that transaction is deemed approved but confirmation of beneficiary bank is pending for amount credit.

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

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

8.2 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.

• Transaction amount will be reversed to partner settlement account after marking transaction 91 to reject.

9. 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.

10.IMPS IFT transactions are RBL to RBL transactions

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

* Transaction amount
* Bank commission
* Bank service tax

12. Transaction limits for IMPS/IFT is 200000 per transaction

**3.** **TRANSACTION RE-QUERY:-**

This is the XML service exposed to produce the Re-Query of the Transaction.

**Request:**

<transactionrequeryreq>

<header>

<sessiontoken></sessiontoken>

</header>

<channelpartnerrefno></channelpartnerrefno>

</transactionrequeryreq>

**Xsd:**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Mandatory** | **Length** | **Description** |
| sessiontoken | String | Yes | 1-200 | Describes the Channel partner usertrack id in encrypted format created by Admin. |
| channelpartnerrefno | String | Yes | 1-30 | Describes that Channel Partner Reference id. |

**IMPS Response:**

<transactionrequeryres>

<channelpartnerrefno></channelpartnerrefno>

<transactondt></transactondt>

<transactionid ></transactionid >

<amount></amount>

<servicechrg></servicechrg>

<tamount></tamount>

<paymentstatus></paymentstatus>

<remittername></remittername>

<remittermblno></remittermblno>

<beneficiaryname></beneficiaryname>

<bank></bank>

<ifsccode></ifsccode>

<accountnumber></accountnumber>

<status></status>

<bankRefNo></bankRefNo>

<BankReferenceNo></BankReferenceNo>

<TranType></TranType>

<BankRemarks></BankRemarks>

</transactionrequeryres>

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Length** | **Description** |
| channelpartnerrefno | String | 1-30 | Describes that Channel Partner Reference id. |
| 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. |
| servicecharge | Money |  | Describes the remittance service charge |
| totalamount | Money |  | Describes the remittance total amount |
| paymentstatus | Int | 1 | 2- Credited to Beneficiary  3- Rejected & Refunded to BC Account  6- Failure/Exceptions |
| remittername | String | 1-50 | Describes the sender Name. |
| remittermobilenumber | Int | 1-10 | Describes the sender Mobile Number. |
| beneficiaryname | String | 1-50 | Describes the Name of Receiver. |
| bank | String | 1-50 | Describes the Beneficiary’s Bank Name. |
| ifscode | String | 1-30 | Describes the Beneficiary’s Bank IFSC code. |
| accountnumber | Int | 1-50 | Describes Beneficiary’s Bank Account Number. |
| status | Int | 1 | 0- failure  1-success |
| bankRefNo | String | 1-50 | Blank (no data) |
| BankReferenceNo | String | 1-20 | Describes the Ref Number given by Bank |
| TranType | String | 1-40 | Describes the type of transaction(IMPS) |
| BankRemarks | String | 1-40 | 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. Transaction Requery needs to be fire for unknown IMPS transaction when customer approach for refund to check the whether payment status is 3 or not.
3. Status 2 is credited to beneficiary
4. Status 3 is rejected & amount refunded to BC account.
5. Status 6 is exception & failure. [in this case there is no debit and credit to bc account]

**====== ================= ===========The End ====================================**