**Application Programming Interface**

**RBL – OXIGEN Collection API Integration**

**(Version 1.2)**

**Document History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Changes** | |
| 1 | 12/09/2017 | Document Created (First Draft) | Stalin Babu P |
| 2 | 13/09/2017 | Document Modified   1. Changing of start tag of transaction API(**corpPaymentReq***)*   *Removed below tags in request*  <TransactionNumber>  <Paymentlocation>  Added tag in response  <TransactionNumber> | Stalin Babu P |
| 3 | 26/09/2017 | UAT URL addition and changes in the all API request and responses | Stalin Babu |

## TABLE OF CONTENTS

1. **Introduction**
   1. API Definition
   2. About XML
2. [**RBL**](#APISPECIFICATION) **Money Transfer APIs**
   1. [Channel Partner Login](#CHANNELPARTNERLOGIN)
   2. [Loan Enquiry](#LoanEnquiry)
   3. [Transaction](#Transaction)
   4. [Transaction Re-Query](#TransactionReQuery)
   5. [Common Error](#Commonerror)

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 COLLECTION 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/BC/collection>

**1.** **[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 |

***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.

**2.** **Enquiry API: -** This is the XML service exposed by the RBL bank to the Channel Partners to check the loan details of a customer

**Request XML -**

<getcustdtlsreq>

<header>

<sessiontoken></sessiontoken>

</header>

<bcagent></ bcagent>

<inputparamcode></inputparamcode>

<paramvalue></paramvalue>

<payeename></payeename>

<payeemobileno></payeemobileno>

<corpid></corpid>

<freetext1></freetext1>

<freetext2></freetext2>

<freetext3></freetext3>

<freetext4></freetext4>

<freetext5></freetext5>

<channelpartnerrefno></channelpartnerrefno>

</getcustdtlsreq>

**Response XML -**

<getcustdtlsres>

<status></status >

<channelpartnerrefno></channelpartnerrefno>

<corpid></corpid>

<lannumber></lannumber>

<applicationid></applicationid>

<customername></customername>

<customermobileno></customermobileno>

<customeremail></customeremail>

<emiamount></emiamount>

<installmentdue></installmentdue>

<lppdue></lppdue>

<bccdue></bccdue>

<otherdue></otherdue>

<currentdues></currentdues>

<rblrefno></rblrefno>

<fetchrefno></fetchrefno>

</getcustdtlsres>

**XSD**



|  |  |  |  |
| --- | --- | --- | --- |
| **Request Fields** | | | |
| **Field Name** | **Data Type** | **Length** | **Description** |
| Session Token\* | String | 2-199 | Session Token received in Login Response |
| InputParamCode\* | String | 1-4 | Type of input  1-LAN No 2- Application ID |
| ParamValue\* | String | 5-30 | LAN No or Application ID value |
| PayeeName\* | String | 5-49 | Describes the customer’s Name. |
| PayeeMobileNo\* | Int | 1-10 | Describes the customer’s Mobile Number. |
| Bcagent\* | String | 5-30 | Describes the CSP unique id maintained by the Channel Partner. |
| CorpID\* | String | 5-30 | Describes the corpid (HERO) |
| FreeText1 | String | 5-30 | Optional (future use) |
| FreeText2 | String | 5-30 | Optional (future use) |
| FreeText3 | String | 5-30 | Optional (future use) |
| FreeText4 | String | 5-30 | Optional (future use) |
| FreeText5 | String | 5-30 | Optional (future use) |
| ChannelPartnerRefno\* | String | 5-30 | Describes the unique transaction ref no for each payment. |
| **Response Field** | | | |
| **Field Name** | **Data Type** | **Length** | **Description** |
| Status | String | 5-30 | 0-failure  1-success |
| ChannelPartnerRefno | String | 5-30 | Describes the unique transaction ref no of BC partner for each payment. |
| LANNumber | String | 5-30 | Describes the Loan account Number of the customer |
| ApplicationID | String | 5-30 | Describes the Application ID of the customer |
| CustomerName | String | 5-49 | Describes the Name of the customer |
| CustomerMobileNo | Int | 10 | Describes the Mobile Number of the customer |
| CustomerEmail | String | 5-49 | Describes the email id of the customer |
| EMIAmount | Money | 1-10 | Describes the EMI amount of the customer |
| InstallmentDue | Int | 1-10 | Describes the no of Installments Due for the customer |
| LPPDue | Money | 1-10 | Describes the Late payment due of the customer |
| BCCDue | Money | 1-10 | Describes the cheque bouncing charges of the customer |
| OtherDue | Money | 1-10 | Describes the other dues of the customer |
| CurrentDues | Money | 1-10 | Describes the current dues of the customer |
| Rblrefno | String | 5-30 | Describes the RBL track no |
| Fetchrefno | String | 5-30 | Describes the unique no used for payments. The same ref no needs to be passed in Payment API. |

***Important Points –***

* Channelpartner ref no should always be unique; this check has to be maintained at BC Partner system level.
* No restriction on no of enquiries in a day
* Fetchrefno is the unique ref no which will come in response and the same needs to be passed in payment request.
* If Transcation gets failed, you can pass the same fetch ref no in payment request by any no of times until it gets success. Once payment success it will fails the next request at API level.

**3. Collection API:** This API will be used for BC Partner to make the EMI payment of a customer.

**Request XML -**

<corppaymentreq>

<header>

<sessiontoken></sessiontoken>

</header>

<bcagent></bcagent>

<customernewemail></customernewemail>

<transactionamount></transactionamount>

<channelpartnerrefno></channelpartnerrefno>

<fetchrefno></fetchrefno>

</corppaymentreq>

**Response XML -**

<corppaymentres>

<status></status>

<channelpartnerrefno></channelpartnerrefno>

<paymentstatus> </paymentstatus>

</corppaymentres>

**XSD**

****

|  |  |  |  |
| --- | --- | --- | --- |
| **Request Fields** | | | |
| **Field Name** | **Data Type** | **Length** | **Description** |
| Session Token\* | String | 2-199 | Session Token received in Login Response |
| Bcagent | String | 5-30 | Describes the agent unique id |
| CustomerNewEmail | String | 5-49 | Describes the email id of the customer |
| TransactionAmount\* | Money | 5-20 | Transaction amount |
| ChannelPartnerRefno\* | String | 5-30 | Describes the unique transaction ref no for each payment. |
| Fetchrefno | String | 5-30 | Describes the unique no used for payments. The same ref no needs to be passed in Payment API. |
| **Response Field** | | | |
| **Field Name** | **Data Type** | **Length** | **Description** |
| Status | String | 5-30 | 0-failure  1-success  -1- **exception** |
| ChannelPartnerRefno | String | 5-30 | Describes the unique transaction ref no of BC partner for each payment. |
| PaymentStatus | String | 5-30 | Success or failed with remarks |

***Important Point:-***

* Channelpartner ref no should always be unique
* Max transaction amount for single transaction is 49999
* No restriction on no of transactions in a day
* Same Fetchrefno needs to be passed in the collection request for making payment. you can pass the same multiple time until it gets success.
* Payment status will be success or failed with remarks.
* Updation of payment of EMI against any LAN or APP ID will be done in 2 days at HERO end, so customer will continue to see the same outstanding payment as before even if he has made the payment.
* Payment will always success, in cases of timeout….Requery needs to be done (Wait time before enquiry is 3 minutes)

1. **Collection Payment Re-Query:** This is the XML service used to get the updated status of the Transaction

**Request XML -**

<herocorpenquiryreq>

<header>

<sessiontoken></sessiontoken>

</header>

<bcagent></bcagent>

<channelpartnerrefno></channelpartnerrefno>

<corpid></corpid>

</herocorpenquiryreq>

**Response XML -**

<herocorpenquiryres>

<status></status>

<channelpartnerrefno></channelpartnerrefno>

<paymentstatus></paymentstatus>

</herocorpenquiryres>

***XSD***

******

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Request Fields | | | | |
| Field Name | | Data Type | Length | Description |
| SessionToken\* | | String | 5-199 | Describes the session token created with login API. |
| ChannelPartnerRefno \* | | String | 5-30 | Describes the unique transaction ref no of BC partner for each payment. |
| Bcagent | | String | 5-30 | Describes unique agent id |
| Corpid | | String | 3-30 | HERO for both UAT and PROD |
| **Response Field** | | | | |
| **Field Name** | **Data Type** | | **Length** | **Description** |
| Status | String | | 5-30 | 0-Failure  1-Success  -1- exception |
| ChannelPartnerRefno | String | | 5-30 | Describes the unique transaction ref no of BC partner for each payment. |
| PaymentStatus | String | | 5-30 | Success or failed with remarks |

***Important points:-***

1. Requery can be done for unknown transactions by passing channelpartner ref no in API
2. Requery can be done for max 7 days.
3. Transaction should be kept on hold if improper details from API. Manual recon needs to be done for those cases with RBL.
4. **Common error:** You will get below response when request fails at API level and even for any exceptions while doing the payment response will come in below format. For exceptions do the requery to know the Transcation status **.**

**Response:**

<errorres>

<status></status>

<description></description>

</errorres>

Status 0 or -1

**Important point**:

1. Anything which is not mentioned in the document is assumed and needs to kept on hold till further discussion and confirmation from RBL.

2. In future if RBL adds any new tag in response xml to give more info, partner system should be always in a position to accept those.

3.updated xsd’s will be shared later.

The End