RBL Bank UPI Technical Specification Corporate Disbursement API (Version 1.1)

Document History

Revision	Date	Changes	
2	02-04-2018	Document Modified V1.1	Shashank Gupta

TABLE OF CONTENTS

I. **Introduction**

1. API Definition
2. About XML
3. UPI (Unified payment interface) Application

II. **RBL UPI APIs**

1. ChannelPartnerLogin
2. GenerateAuthToken
3. GetTransactionID
4. ValidateVPA
5. Disbursement/Payment
6. Transaction Re Query

III. **Error Handling**

IV. **Logic of HMAC Generation**

1. API Definition -

Application programming interface is a middleware that communicates between the host system and the front end application. This API connects the UPI 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.

3. RBL UPI API Specification

The below document provides the integration details for enabling the collections payments from the customers through UPI mode.

URL: https://apideveloper.rblbank.com/test/sb/rbl/api/v1/upi/payment

1. ChannelPartnerLogin - This is the XML service for Channel Partners authentication on RBL Bank UPI Platform. The Login API is designed to allow create login session of 1 hour. This API will provide session token for passing & using in other API Methods.

Request -

```
<channelpartnerloginreq>
                    <username></username>
                    <password></password>
                    <bcagent></bcagent>
</channelpartnerloginreq>
```

Response –

```
<channelpartnerloginres>
                    <sessiontoken></sessiontoken>
                    <timeout></timeout>
                    <status></status>
</channelpartnerloginres>
```

Request Fields			
Field Name	**Data Type**	**Length**	**Description**
Username*	String	1-30	Describes the Channel partner user name maintained by the RBL.
Password*	String	40	Describe the API password given by RBL
Bcagent*	String	1-50	Describes the BC Agent's 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

Description –

1. User name is BC user name of Channel Partner Company registered under UPI Engine.
2. Password is API password in format which will be shared by RBL Bank for UAT testing.
3. BC Agent is Registered & Approved CSP under Partner.
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 passing/using in other API Methods.

2. **GenerateAuthToken: -** This API is used to generate the token.

Request XML -

```
<generateauthtokenreq>
<header>
<sessiontoken> </sessiontoken>
<bcagent> </bcagent>
</header>
<mrchOrgId> </mrchOrgId>
<aggrOrgId> </aggrOrgId>
<note> </note>
<refId></refId>
<refUrl> </refUrl>
<mobile></mobile>
<geocode></geocode>
<location></location>
<ip></ip>
<type> </type>
<id></id>
<os> </os>
<app> </app>
<capability></capability>
<hmac> </hmac>
</generateauthtokenreq>
```

Response XML -

```
<generateauthtokenres>
  <status></status>
  <token> </token>
</generateauthtokenres>
```

Request Fields			
Field Name	**Data Type**	**Length**	**Description**
Session Token*	String	1-200	Session Token received in Login Response
Bcagent*	String	1-50	Describes the BC Agent's unique ID maintained by the Channel Partner
mrchOrgId *	String	1-50	Merchant id (bank will share)
aggrOrgId *	String	1-50	Aggregator id(bank will share)
Note	String	1-50	Remarks of the transaction
refId*	String	1-50	Random number
refUrl	String	1-50	URL
Mobile*	Integer	1-50	Mobile number of the Partner
Geocode	String	1-50	geocode of a customer
Location	String	1-50	Location of the customer
Ip	String	1-50	IP of the device
Type	String	1-50	Type
Id	String	1-50	Random number
OS	String	1-50	OS of the device (Andriod /IOS)
App	String	1-50	Application name
capability	String	1-50	NA
Hmac*	String	1-50	Key generated using mrchOrgId and aggrOrgId. Logic of HMAC creation at the end of document
Response Field			
Field Name	**Data Type**	**Length**	**Description**
Status	integer	1	Status (1- Success/0- Failed)
Token	String	1-100	Session Token. Unique value of alpha numeric

Description –

1. For Generate Token on RBL UPI Platform Agent will have to capture mrchOrgId, aggrOrgId, note, refId, refUrl, Mobile number, HMAC, and other fields in valid form.
2. Kindly find logic of HMAC creation at the end of document.
3. Merchant user has to generate Token which will be unique and valid for 60 minutes only.
4. Token received in response of GenerateAuthToken.
5. Once received, this Token can be used as HMAC need to pass/use in other API Methods.

3. GetTransactionID: - This API is used to generate the transaction ID, which is to be passed in Payment API.

Request XML -

```
<gettxnid>
<header>
<sessiontoken> </sessiontoken>
<bcagent> </bcagent>
</header>
<mrchOrgId> </mrchOrgId>
<aggrOrgId> </aggrOrgId>
<mobile></mobile>
<geocode></geocode>
<location></location>
<ip></ip>
<type> </type>
<id></id>
<os> </os>
<app> </app>
<capability></capability>
<hmac> </hmac>
</gettxnid>
```

Response XML -

```
<RespGetTxnId>
  <status></status>
  <description> </description>
</RespGetTxnId>
```

Request Fields			
Field Name	**Data Type**	**Length**	**Description**
Session Token*	String	1-200	Session Token received in Login Response
Bcagent*	String	1-50	Describes the BC Agent's unique ID maintained by the Channel Partner
mrchOrgId *	String	1-50	Merchant id (bank will share)
aggrOrgId *	String	1-50	Aggregator id(bank will share)
Mobile	Integer	1-12	Mobile number of the Partner
geocode	String	1-50	geocode of a customer
location	String	1-50	Location of the customer
ip	String	1-50	IP of the device
type	String	1-50	Type
id	String	1-50	Random number
OS	String	1-50	OS of the device
app	String	1-50	Application name
capability	String	1-50	NA
Hmac*	String	1-50	Key generated from GenerateAuthToken API
Response Field			
Field Name	**Data Type**	**Length**	**Description**
Status	integer	1	Status (1- Success/0- Failed)
Description	String	1-100	Transaction ID. This is unique value of alpha numeric. Pass this ID in the Disbursement request API.

Description –

1. For Get TxnID we have to pass Token generated from **GenerateAuthToken** as HMAC here in input.
2. Transaction Id will get in response of GetTxnID, which has to be passed in Disbursement API.

4. ValidateVPA - This API is used to validate the VPA with the name of the customer's vpa.

Request XML -

```
<validatevpa>
<header>
<sessiontoken> </sessiontoken>
<bcagent> </bcagent>
</header>
<mrchOrgId> </mrchOrgId>
<aggrOrgId> </aggrOrgId>
<note> </note>
<refId> </refId>
<orgTxnId> </orgTxnId>
<refUrl> </refUrl>
<mobile></mobile>
<geocode></geocode>
<location></location>
<ip></ip>
<type> </type>
<id></id>
<os> </os>
<app> </app>
<capability></capability>
<hmac> </hmac>
<addr> </addr>
</validatevpa>
```

Response XML -

```
<vpaResponse>
<status></status>
<description> </description>
</vpaResponse>
```

Request Fields			
Field Name	**Data Type**	**Length**	**Description**
Session Token*	String	1-200	Session Token received in Login Response
Bcagent*	String	1-50	Describes the BC Agent's unique ID maintained by the Channel Partner
mrchOrgId *	String	1-50	Merchant id (bank will share)
aggrOrgId *	String	1-50	Aggregator id(bank will share)
note	String	1-50	Remarks of the transaction
refId*	String	1-50	Unique ID for the Partner
orgTxnId*	String	1-50	Random Number
RefUrl	String	1-50	URL
Mobile	Integer	1-12	Mobile number of the Partner
geocode	String	1-50	geocode of a customer
location	String	1-50	Location of the customer
ip	String	1-50	IP of the device
type	String	1-50	Type
id	String	1-50	Random number
OS	String	1-50	OS of the device
app	String	1-50	Application name
capability	String	1-50	NA
Hmac*	String	1-50	Key generated from GenerateAuth Token API
Addr*	String	1-50	VPA of the customer.
Response Field			
Field Name	**Data Type**	**Length**	**Description**
Status	integer	1	Status (1- Success/0- Failed)
Description	String	1-100	Verified Name of customer/user behind the VPA.

Description –

1. For Validate VPA we have to pass virtual payment address (VPA) of the customer/user
2. Status received in response of Validate VPA.
3. In description tag, we will get the name of the customer whose VPA has passed.

5. Disbursement/Payment: - This API is used to do disbursements/payment on the customers VPA.

Request XML -

```
<upidisbursement>
<header>
<sessiontoken> </sessiontoken>
<bcagent> </bcagent>
</header>
<mrchOrgId> </mrchOrgId>
<aggrOrgId> </aggrOrgId>
<note></note>
<refId></refId>
<refUrl></refUrl>
<orgTxnId></orgTxnId>
<txnId></txnId>
<mobile></mobile>
<geocode></geocode>
<location></location>
<ip></ip>
<type> </type>
<id></id>
<os> </os>
<app> </app>
<capability></capability>
<hmac> </hmac>
<payeraddress> </payeraddress>
<payername> </payername>
<payeeaddress> </payeeaddress>
<payeename> </payeename>
<amount></amount>
</upidisbursement>
```

Response XML -

```
<?xml version="1.0" encoding="utf-16" standalone="no"?>
<upidisbursementres>
  <responseId></responseId>
  <status> </status>
  <refid> </refid>
</upidisbursementres>
```

Request Fields			
Field Name	**length**	**Data Type**	**Description**
Sessiontoken *	1-200	String	Describe session token received by RBL
Bcagent*	1-50	String	Describes the BC Agent's unique ID maintained by the Channel Partner
MrchOrgId *	1-50	String	Describe MrchOrgId received by RBL
AggrOrgId *	1-50	String	Describe AggrOrgId received by RBL
Note	1-50	String	Describe Remarks
RefId *	1-50	String	Describe reference id
RefUrl	1-50	String	Describe reference url
orgTxnId*	3-50	String	orgtxnid (For Future use)
txnId	**1-50**	String	RBL Transaction ID (From GetTxn ID API)
mobile *	**1-12**	Integer	Describe mobile number
geocode	1-50	String	Describe geocode
location	1-50	String	Describe location
ip	1-50	String	Describe ip address of client
type	1-50	String	Describe the type
id	1-50	String	Describe id
os	1-50	String	Describe operating system
app	1-50	String	Describe app id
capability	1-50	String	NA
hmac *	1-200	String	Describe Encrypted HMAC
payeraddress *	1-50	String	Describe payer address (Partner's VPA)
payername *	1-50	String	Describe payer name (Partner Name)
Payeeaddress *	1-50	String	Describe payee address (Customer's VPA)
payeename *	1-50	String	Describe payee name (Customer name)
amount *	1-50	Float	Describe amount
Response Field			
Field Name	**Data Type**	**Length**	**Description**
responseId	integer	1	(1- Success/0- Failed)
Status	String	1-100	Status of the payment Transaction done on customer vpa: Success /Failure
refid	String	1-100	UPI Transaction ID.

Description –

1. This API is used to do the payment to customer VPA by the Partner.
2. Status tag denoted final status of the transaction: SUCCESS/FAILURE.

6. Transaction Re Query: - This API is used to check the transaction status.

Request XML -

```
<searchrequest>
<header>
<sessiontoken> </sessiontoken>
<bcagent> </bcagent>
</header>
<mrchOrgId> </mrchOrgId>
<aggrOrgId> </aggrOrgId>
<mobile></mobile>
<geocode></geocode>
<location></location>
<ip></ip>
<type> </type>
<id></id>
<os> </os>
<app> </app>
<capability></capability>
<hmac> </hmac>
<orgTxnIdorrefId> </orgTxnIdorrefId>
<flag></flag>
</searchrequest>
```

Response XML -

```
<searchrequest>
  <status></status>
  <txnstatus> </ txnstatus >
  <payeraddr> </ payeraddr >
  <payermobileno></ payermobileno >
  <amount></amount>
  <payeraccno></ payeraccno >
  <payerifsc> </payerifsc>
  <custref></custref>
</searchrequest>
```

Request Fields			
Field Name	**length**	**Data Type**	**Description**
Sessiontoken *	1-200	String	Describe session token received by RBL
Bcagent*	1-50	String	Describes the BC Agent's unique ID maintained by the Channel Partner
MrchOrgId *	1-50	String	Describe MrchOrgId received by RBL
AggrOrgId *	1-50	String	Describe AggrOrgId received by RBL
Note	1-50	String	Describe Remarks
RefId *	1-50	String	Describe reference id
RefUrl	1-50	String	Describe reference url
mobile *	**1-12**	Integer	Describe mobile number
geocode	1-50	String	Describe geocode
location	1-50	String	Describe location
ip	1-50	String	Describe ip address of client
type	1-50	String	Describe the type
id	1-50	String	Describe id
os	1-50	String	Describe operating system
app	1-50	String	Describe app id
capability	1-50	String	NA
hmac *	1-200	String	Describe Encrypted HMAC
orgTxnIdorrefId *	1-50	String	Describe the ID of Originating transaction . If flag is 1 , the pass Partner Ref ID, If flag is 0, then pass UPI Txn Dd
flag *	1	String	Describe flag 1- With partner ref id 0- With upi txn id
Response Field			
Field Name	**Data Type**	**Length**	**Description**
Status	integer	1	Status (1- Success/0- Failed)
txnstatus	String	1-100	Status of the Collect Transaction raised on customer vpa: Success /Failure/ In Progress
Payeraddr	String	1-100	Payer/Customer's VPA
PayermobileNumber	String	1-12	Payer/Customer's Mobile Number
amount	float	1-50	Amount of the transaction
PayeraccountNumber	Integer	1-100	Payer/Customer's Account Number
Payerifsc	String	1-11	Payer/Customer's IFSC
Custref	Integer	1-12	Customer Reference number (RRN).

Description –

1. For Transaction Re query we have to pass Txn ID of UPI or Ref ID of partner. We can do query in both ways.
2. txnstatus tag denoted Status of the transaction: SUCCESS/FAILURE/IN PROGRESS

Error Responses:

1) <errorres>
<status>0</status>
<description>Your Session has been Expired.Please Relogin the Application</description>
</errorres>

2) <errorres>
<status>0</status>
<description>E001:Invalid Auth token</description>
</errorres>

3) <errorres>
<status>0</status>
<description>ERR010:Duplicate Ref Id</description>
</errorres>

4) <errorres>
<status>0</status>
<description>MRER005:txn:orgTxnId:MIN_LENGTH_REQUIRED</description>
</errorres>

5) < errorres >
<status>0</status>
< description >MRER005:txn:refId:MIN_LENGTH_REQUIRED</ description >
</ errorres >

Error Code Description

Sr No	Error Description	Error Code
1.	Invalid Auth token	E001
2.	Invalid MCC	E003
3	No channel found or channel is not active	E004
4.	System Error	E005
5.	Aggregator Id, Merchant Id and hmac are Required	E006
6.	Invalid Search Request!	E007
7.	Invalid Payment Request!	E008
8.	Invalid Id Type!	E009
9.	Invalid TxnId!	ERR00094
10.	Invalid RefId!	ERR010
11.	SYSTEM_ERROR	MRER000
12.	NULL_VALUE	MRER002
13.	EMPTY_STRING	MRER003
14.	INVALID_FORMAT	MRER004
15.	MIN_LENGTH_REQUIRED	MRER005
16.	MAX_LENGTH_EXCEED	MRER006
17.	MIN_VALUE_REQUIRED	MRER007
18.	MAX_VALUE_EXCEED	MRER008
19.	NOT_NUMERIC	MRER009
20.	INVALID_ID_NUMBER	MRER010

Logic of HMAC Generation

```
package com.rbl.util;
import javax.crypto.Cipher;
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;

import org.apache.commons.codec.binary.Base64;
import org.apache.commons.codec.binary.Hex;
public class hmac_generation {

public static void main(String[] args) throws Exception {

          String KEY =
"7b4df22b948d19c64cc3382d90b7d0eab43b028a6400cf473e8b0177a2e5dc41";

          String hmacForAuthToken = new
String(encryptUsingAes256Key("AGGREGATOR101|MC234", KEY));

           /*Here AGGREGATOR101 is the aggregator code and MC234 is the merchant code as
registered in UPI Manager. Please pass your value shared by bank*/


        }
private static String encryptUsingAes256Key(String message, String key) throws Exception {
                byte[] messageBytes = message.getBytes();
                byte[] keyBytes = Hex.decodeHex(key.toCharArray());
                SecretKey secretKey = new SecretKeySpec(keyBytes, "AES");
                Cipher cipher = Cipher.getInstance("AES");
                cipher.init(Cipher.ENCRYPT_MODE, secretKey);
                byte[] encryptedByte = cipher.doFinal(messageBytes);
                String encryptedText = Base64.encodeBase64String(encryptedByte);
                System.out.println("in security encrypted byte is" + encryptedByte.toString());
                System.out.println("in security encrypted text is" + encryptedText.toString());
                return encryptedText;
        }


}
```

The End