



RSSOFTWARE

**Statement of Confidentiality**

*The information contained herein shall not be disclosed, duplicated, or used in whole or in part for any purpose other than to evaluate the proposal, provided that if a contract is awarded to this offer as a result of, or in connection with, the submission of these information, the recipient shall have the right to duplicate, use or disclose the information to the extent provided in the agreement. This restriction does not limit the right to use information contained in the data if it is obtained from another source without restriction.*

© 2018 RS Software (India) Ltd.

RS SOFTWARE

## **DOCUMENT REFERENCE**

A large, stylized watermark is printed across the page. The text "RS SOFTWARE" is written in a bold, sans-serif font, oriented diagonally from the bottom-left towards the top-right. The letters are a light gray color, matching the background grid.

**Table of Contents**

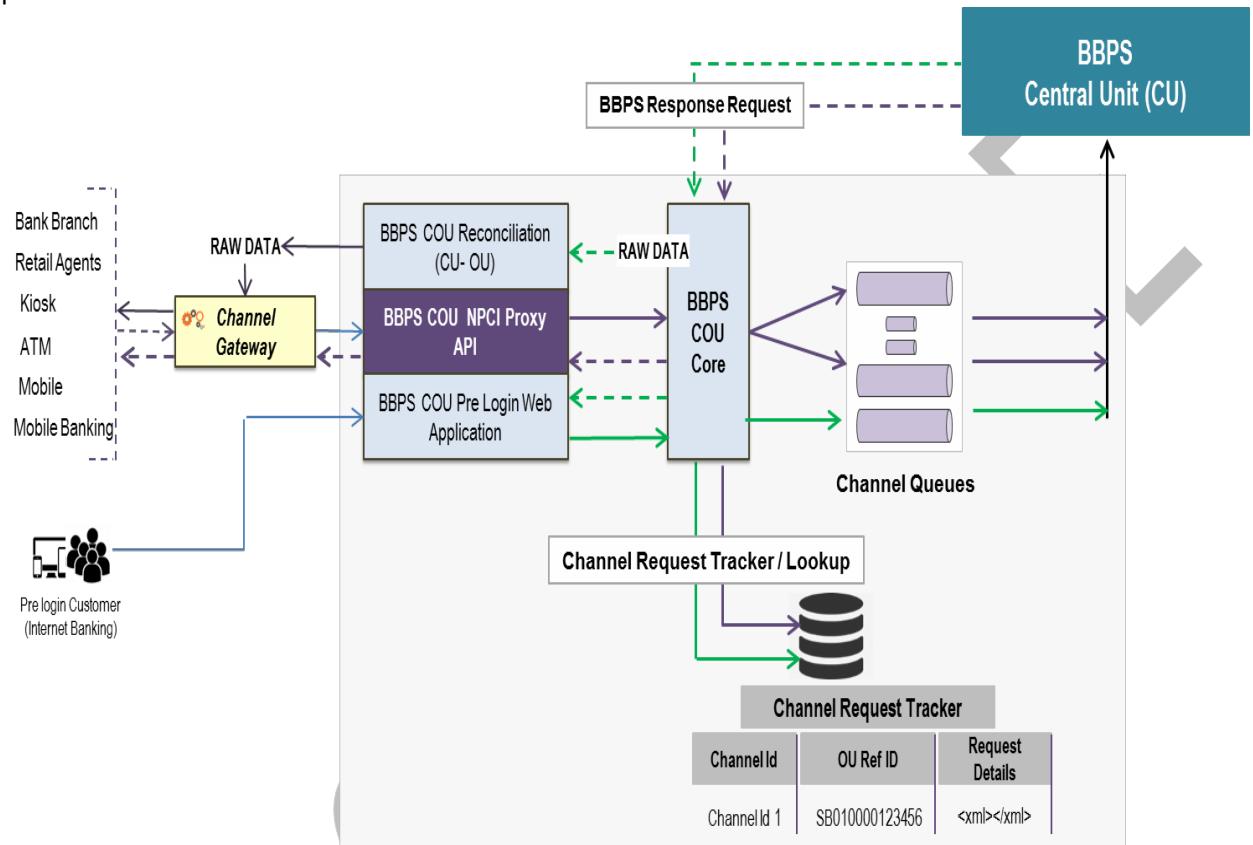
1	Introduction.....	4
1.1	NPCI Proxy - Router.....	5
	API Listing .....	5
2.1	Functional Overview .....	6
2.1.1	Get Category List .....	7
2.1.2	Get Biller List .....	7
2.1.3	Get Sub-Biller List .....	10
2.1.4	Get Biller Customer Parameters .....	12
2.1.5	Fetch Bill .....	13
2.1.6	Validate Customer Parameters (Pay-Only Biller) .....	17
2.1.7	Get Tax Amount.....	19
2.1.8	Pay Bill (Fetch & Pay Bill) ** .....	21
2.1.9	Pay Bill (Pay Only) ** .....	26
2.1.10	Get Customer Transactions.....	30
2.1.11	Get Txn Complaint Dispositions.....	32
2.1.12	Get Service Reasons .....	33
2.1.13	Get Complaint Status.....	35
2.1.14	Raise Complaint .....	36
2.1.15	Agent Login .....	39
2.1.16	Set Password .....	40
2.1.17	Forgot Password .....	41
2.1.18	Transaction Search .....	42
2.1.19	Get Duplicate Receipt .....	44

**1. Introduction**

The purpose of this document is how to integrate the other channels for BBPS solutions. The document defines general guidelines of expected results and formatting therein with sample examples for better understanding. There are 2 solution approaches were considered.

## 1.1 NPCI Proxy - Router

Develop a pure router, where the current deployment will only relay requests to CU (without any value add) and return the CU request. (Approach can be additional API on same deployment). Here is the process flow for the same.

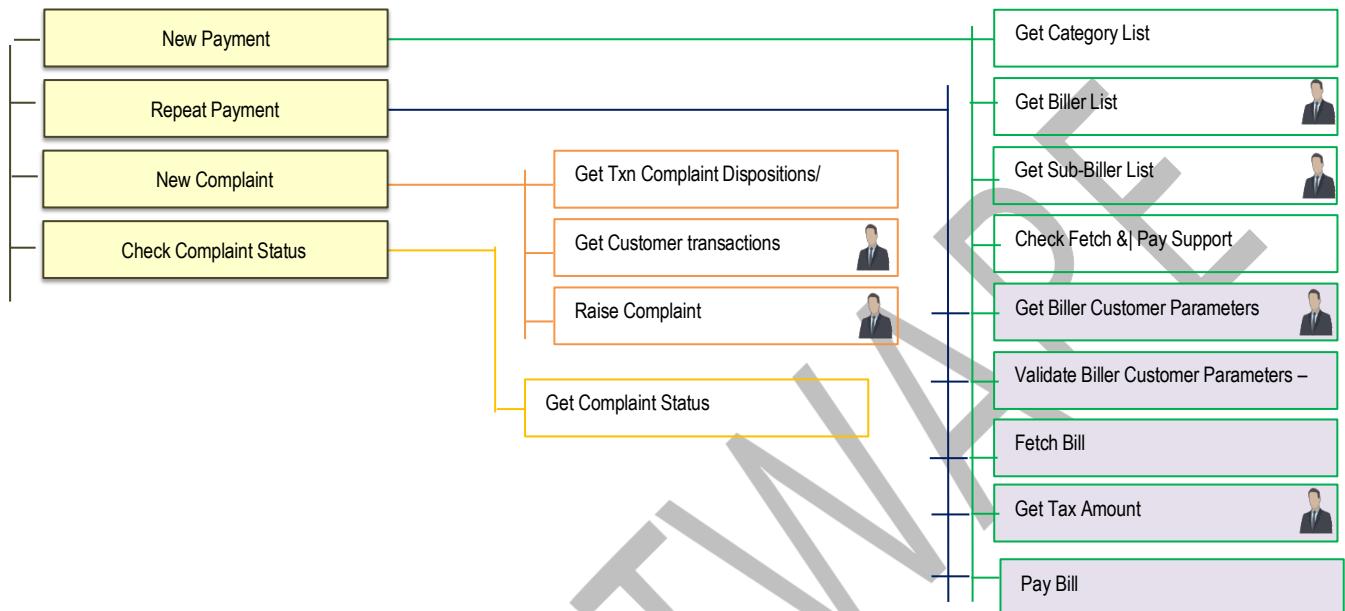


1. New COU NPCI Proxy APIs to be introduced
2. Channel Gateway interfacing various SBI Gateways will invoke these Proxy APIs.
3. Proxy API invocations to have a mandatory Channel Id identifier as part of "Request Header". This will be for NPCI request tracking and response re routing.
4. In case of Asynchronous Response Transactions of NPCI, Channels to opt for preconfigured modes API invocation or Response polling. (Channel URL for response posting or Channel can opt for polling approach)
5. RS BBPS system will handle Channel specific Request – Response routing based on configured domains and modes for the Channel.
6. For Reconciliation, NPCI Raw Data received by current system will be published as is to all Channels without any filtering.
7. Current Pre login deployment may need few changes for Channel Identifier conformity, especially for Reconciliation.

### API Listing

Here is the list of all the APIs available as part of RS BBPOU solution and how developers can use each API's for integrate the other channels.

## 2.1 Functional Overview



### a) Authorization & API tokens

#### I. Request:

All API access should mandatorily include 2 Headers

##### 1) The Basic Authorization Header value as

*Authorization Basic Base64.encode(<AgentId>:<Password>);*

*Example: ID76ID02BSC520098385:password*

*Eg. Authorization Basic T1U3MFNMMTJJTkI1MjA5MzY0NTE6cGFzc3dvcmQ=*

##### Example Request Header:

*Authorization:Basic T1U3MFNMMTJJTkI1MjA5MzY0NTE6cGFzc3dvcmQ=*

*Content-Type:application/xml*

*Accept:application/xml*

#### II. Response:

All API access response will include

##### 1) The RequestId Header value as (to uniquely identify the request)

*requestId b46dd68e-b4fc-4cc2-b37a-1b38ef4c4d6a*

Example Response Header:

*requestId : b46dd68e-b4fc-4cc2-b37a-1b38ef4c4d6a*

*Content-Type: application/xml; charset=UTF-8*

- b) Note: (\*) **asterisk** for mandatory fields

### 2.1.1 Get Category List

This api is used to fetch the categories of the **BBPS** billers. It will provide only category name.

Action	Fetch all biller category list	
API	<domain>/APIService/biller-category-list/urn:tenantId:<OU Id>	
Request Header		
	Authorization	Basic Base64.encode(<AgentId>:<Password>); E.g. Basic T1U3MFNMMTJJTk1MjA5MzY0NTE6cGFzc3dvcmQ=
	Content-Type	application/xml   application/json
	Accept	application/xml   application/json
	Request Method	POST
Response Header		
	requestId	<i>b46dd68e-b4fc-4cc2-b37a-1b38ef4c4d6a</i>
	Content-Type	application/xml   application/json
Response	<b>Category</b> <b>Description:</b> Return the list of Biller category name.	
	JSON Format	XML Format
Request Example	No payload is required.	No payload is required.
Response Example	{ "catagories": [ "Bill Payment", "BillCat01", "BillerCategory1", "Broadband Postpaid", "Broadband Prepaid", "Cat_19_05", "DTH", "Electricity", "Electricity Prepaid", "Gas", "Landline Postpaid", "Landline Prepaid", ]	<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <ns2:BillerCatagory xmlns:ns2="http://bbps.rsssoftware.co.in/schema"> <category>BLRCategory3</category> <category>Bill Payment</category> <category>BillCat01</category> <category>BillerCategory1</category> <category>Broadband Postpaid</category> <category>Broadband Prepaid</category> <category>Cat_19_05</category> <category>Credit Card</category> <category>DTH</category> <category>Electricity</category> <category>Electricity Prepaid</category>

	<pre>         "MOB",         "MOBXX",         "Mobile",         "Mobile Postpaid",         "Mobile Prepaid",         "Recharge",         "School Fees Payment",         "Utility Bill Payment",         "Water",         "Water_Supply_B"     ],     "errorMessages": [] } </pre>	<pre> &lt;catagory&gt;Gas&lt;/catagory&gt; &lt;catagory&gt;Landline Postpaid&lt;/catagory&gt; &lt;catagory&gt;Landline Prepaid&lt;/catagory&gt; &lt;catagory&gt;MOB&lt;/catagory&gt; &lt;catagory&gt;MOBXX&lt;/catagory&gt; &lt;catagory&gt;Mobile&lt;/catagory&gt; &lt;catagory&gt;Mobile Postpaid&lt;/catagory&gt; &lt;catagory&gt;Mobile Prepaid&lt;/catagory&gt; &lt;catagory&gt;Municipal Taxes&lt;/catagory&gt; &lt;catagory&gt;Recharge&lt;/catagory&gt; &lt;catagory&gt;Registration Fee&lt;/catagory&gt; &lt;catagory&gt;School Fees Payment&lt;/catagory&gt; &lt;catagory&gt;Utility Bill Payment&lt;/catagory&gt; &lt;catagory&gt;Water&lt;/catagory&gt; &lt;catagory&gt;Water_Supply_B&lt;/catagory&gt; &lt;/ns2:BillerCatagory&gt; </pre>
--	---	--

### 2.1.2 Get Biller List

This api is used to fetch a biller from a selected category. Basic information about the billers is also provided.

Action	Fetch biller for the selected category	
API	<domain>/APIService/biller-list/urn:tenantId: <OU Id>	
Request Method		
Authorization	Basic Base64.encode(<AgentId>:<Password>); E.g. Basic T1U3MFNMMTJJTkI1MjA5MzY0NTExcGFzc3dvcmQ=	
Content-Type	application/xml   application/json	
Accept	application/xml   application/json	
Request Method	POST	
Request	<b>*Category</b> Description: Category name of the Billers	
Response Header		
requestId	b46dd68e-b4fc-4cc2-b37a-1b38ef4c4d6a	
Content-Type	application/xml   application/json	
Response	<b>biller</b> Description: Return the Parent Biller list and Biller list under the specific Biller Category.	
	<b>JSON Format</b>	<b>XML Format</b>

<b><i>RequestExample</i></b>	<pre>{"category":"Broadband Postpaid"}</pre>	<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <ns2:PaymentInformation xmlns:ns2="http://bbps.rsssoftware.co.in/schema"> <category>Broadband Postpaid</category> </ns2:PaymentInformation>
<b><i>Response Example</i></b>	<pre>{ "billers": [ { "id": "ICIC00000NATR6", "name": "ICICI_BillValidationUnitTest", "categoryName": "Broadband Postpaid", "mode": "ONLINE", "acceptsAdhoc": true, "isParent": false, "parentBillerId": null, "fetchRequirement": "NOT_SUPPORTED", "billerDescription": "BillValidation Description for UNIT TESTING..", "blrSupportBillValidation": "OPTIONAL", "subBillers": [] }, { "id": "MONL00000COOIP", "name": "monlinebiller", "categoryName": "Broadband Postpaid", "mode": "ONLINE", "acceptsAdhoc": true, "isParent": false, "parentBillerId": null, "fetchRequirement": "OPTIONAL", "billerDescription": "ABCD", "blrSupportBillValidation": "NOT_SUPPORTED", "subBillers": [] } ], "errorMessages": [] }</pre>	<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <ns2:billerList xmlns:ns2="http://bbps.rsssoftware.co.in/schema"> <biller> <id>ICIC00000NATR6</id> <name>ICICI_BillValidationUnitTest</name> <categoryName>Broadband Postpaid</categoryName> <mode>ONLINE</mode> <acceptsAdhoc>true</acceptsAdhoc> <isParent>false</isParent> <fetchRequirement>NOT_SUPPORTED</fetchRequirement> <billerDescription>BillValidation Description for UNIT TESTING..</billerDescription> <brSupportBillValidation>OPTIONAL</brSupportBillValidation> </biller> <biller> <id>CAN00000NATPI</id> <name>Canara Biller</name> <categoryName>Broadband Postpaid</categoryName> <mode>ONLINE</mode> <acceptsAdhoc>false</acceptsAdhoc> <isParent>false</isParent> <fetchRequirement>MANDATORY</fetchRequirement> <brSupportBillValidation>NOT_SUPPORTED</brSupportBillValidation> </biller> </ns2:billerList>

### 2.1.3 Get Sub-Biller List

This api is used to fetch the derived biller list. It can be fetched from it's parent biller.

Action	Fetch biller list under a Parent Biller	
API	<Domain>/ APIService/sub-biller-list/urn:tenantId: <OU Id>	
Request Header		
	Authorization	Basic Base64.encode(<AgentId>:<Password>); E.g. Basic T1U3MFNMMTJJTkI1MjA5MzY0NTE6cGFzc3dvcmQ=
	Content-Type	application/xml   application/json
	Accept	application/xml   application/json
	Request Method	POST
Request	<u>*billerd</u> Description: 14 digit ID of Parent Biller.	
Response Header		
	requestId	b46dd68e-b4fc-4cc2-b37a-1b38ef4c4d6a
	Content-Type	application/xml   application/json
Response	<u>Biller</u> Description: List of billers under the selected Parent Biller.	
	<u>ID</u> Description: 14digit Biller ID	
	<u>Name</u> Description: Biller Name	
	<u>categoryName</u> Description: Name of Biller Category	
	<u>Mode</u> Description: Mode of Biller i.e. either ONLINE, OFFLINEA or OFFLINEB.	
	<u>acceptsAdhoc</u> Description: Whether the biller accepts Adhoc payment or not?	
	<u>isParent</u> Description: Indicate whether the current biller is a parent biller or not. In this case 'isParent' tag always be 'false'.	
	<u>parentBillerId</u> Description: Parent biller id by which we are searching the subbiller list.	

## Biller MDM API

		<p><b><u>fetchRequirement</u></b></p> <p><b>Description:</b> indicate the biller which Fetch Requirement value is supported like OPTIONAL, MANDATORY, NOT_SUPPORTED</p>
		<p><b><u>billerDescription</u></b></p> <p><b>Description:</b> Biller BBPOUs can now pass any additional information for different Billers .this is the addition of billerDescription tag in Biller MDM Response.</p>
		<p><b><u>blrSupportBillValidation</u></b></p> <p>Validation API for Billers, supporting only payment will help validate a Customer's bill before payment initiation and thus reduces business declines.</p>
	<b>JSON Format</b>	<b>XML Format</b>
<b>Request Example</b>	{"billerId":"UCOP00000NATG8"}	<pre>&lt;?xml version="1.0" encoding="UTF-8" standalone="yes"?&gt; &lt;ns2:PaymentInformation xmlns:ns2="http://bbps.rssoftware.co.in/schema"&gt;     &lt;billerId&gt;UCOP00000NATG8&lt;/billerId&gt; &lt;/ns2:PaymentInformation&gt;</pre>
<b>Response Example</b>	<pre>{     "billers": [         {             "id": "UCOPBOUONSTA22",             "name": "BOUOnlineOptionalBiller",             "categoryName": "Water",             "mode": "ONLINE",             "acceptsAdhoc": true,             "isParent": false,             "parentBillerId": "UCOP00000NATG8",             "fetchRequirement": "OPTIONAL",             "billerDescription": "\"It takes 2 days to reflect the payment in your biller account\"",             "blrSupportBillValidation": "NOT_SUPPORTED",             "subBillers": []         },         {             "id": "UCOPBOUOFLHL8",             "name": "BOUOfflineNotSupportBiller",             "categoryName": "Water",             "mode": "OFFLINEA",             "acceptsAdhoc": true,             "isParent": false,             "parentBillerId": "UCOP00000NATG8",             "fetchRequirement": "NOT_SUPPORTED",             "billerDescription": "@ BOU Testing in PNB for OFFLINE A Biller@"         }     ] }</pre>	<pre>&lt;?xml version="1.0" encoding="UTF-8" standalone="yes"?&gt; &lt;ns2:billerList xmlns:ns2="http://bbps.rssoftware.co.in/schema"&gt;     &lt;biller&gt;         &lt;id&gt;UCOPBOUONSTA22&lt;/id&gt;         &lt;name&gt;BOUOnlineOptionalBiller&lt;/name&gt;         &lt;categoryName&gt;Water&lt;/categoryName&gt;         &lt;mode&gt;ONLINE&lt;/mode&gt;         &lt;acceptsAdhoc&gt;true&lt;/acceptsAdhoc&gt;         &lt;isParent&gt;false&lt;/isParent&gt;         &lt;parentBillerId&gt;UCOP00000NATG8&lt;/parentBillerId&gt;         &lt;fetchRequirement&gt;OPTIONAL&lt;/fetchRequirement&gt;         &lt;billerDescription&gt;It takes 2 days to reflect the payment in your biller account&lt;/billerDescription&gt;          &lt;blrSupportBillValidation&gt;NOT_SUPPORTED&lt;/blrSupportBillValidation&gt;     &lt;/biller&gt;     &lt;biller&gt;         &lt;id&gt;UCOPBOUOFLHL8&lt;/id&gt;         &lt;name&gt;BOUOfflineNotSupportBiller&lt;/name&gt;         &lt;categoryName&gt;Water&lt;/categoryName&gt;         &lt;mode&gt;OFFLINEA&lt;/mode&gt;         &lt;acceptsAdhoc&gt;true&lt;/acceptsAdhoc&gt;         &lt;isParent&gt;false&lt;/isParent&gt;         &lt;parentBillerId&gt;UCOP00000NATG8&lt;/parentBillerId&gt;         &lt;fetchRequirement&gt;NOT_SUPPORTED&lt;/fetchRequirement&gt;         &lt;billerDescription&gt;@ BOU Testing in PNB for OFFLINE A Biller@&lt;/billerDescription&gt;          &lt;blrSupportBillValidation&gt;MANDATORY&lt;/blrSupportBillValidation&gt;     &lt;/biller&gt; &lt;/ns2:billerList&gt;</pre>

	<pre>         "blrSupportBillValidation":          "MANDATORY",          "subBillers": []        }      ],      "errorMessages": []   } } </pre>	
--	--	--

#### 2.1.4 Get Biller Customer Parameters

This api is used to fetch the Customer parameter fields for a selected leaf node biller. If a biller has any derived biller then it's customer parameters can't be extracted.

Action	Fetch Biller Custom Parameter list	
<b>API</b>	<Domain>/ APIService/biller-fetch-custom-params/urn:tenantId:<OU Id>	
<b>Request Header</b>		
	Authorization	Basic Base64.encode(<AgentId>:<Password>); E.g. Basic T1U3MFNMMTJJTkI1MjA5MzY0NTEx6cGFzc3dvcmQ=
	Content-Type	application/xml   application/json
	Accept	application/xml   application/json
	Request Method	POST
<b>Request</b>	<u>*billId</u> <b>Description:</b> 14digit Biller id.	
<b>Response Header</b>		
	requestId	b46dd68e-b4fc-4cc2-b37a-1b38ef4c4d6a
	Content-Type	application/xml   application/json
<b>Response</b>	<u>customParams</u> <b>Description:</b> Contains the custom parameter details of the selected Biller.  <u>Name</u> <b>Description:</b> Contains the name of the custom parameter  <u>Type</u> <b>Description :</b> Contains the data type of the parameter which can be either Numeric or Alpha-Numeric.  <u>IsMandatory</u> <b>Description:</b> Contains the flag stating that whether the custom parameter is mandatory or not.  <u>Regex</u> <b>Description:</b> Regex (Validation Pattern for customer parameters across the billers)	

		as an additional yet optional tag in biller MDM to reduce business declines and refunds in the system which would considerably reduce overheads in the entire ecosystem.
	<b>Json Format</b>	<b>XML Format</b>
<b>Request Example</b>	{"billerId":"UCOPBOUOFLKM4N"}	<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <ns2:PaymentInformation xmlns:ns2="http://bbps.rsssoftware.co.in/schema"> <billerId>ONLI00000WBGQP</billerId> </ns2:PaymentInformation>

<b>Response Example</b>	<pre>{   "customParams": [     {       "name": "Account",       "type": "ALPHANUMERIC",       "isMandatory": true,       "minLength": 2,       "maxLength": 20,       "regex": "^[a-zA-Z0-9@#\$%]{4,10}"     },     {       "name": "Product Id",       "type": "ALPHANUMERIC",       "isMandatory": false,       "minLength": 1,       "maxLength": 25,       "regex": "^[a-zA-Z0-9@#\$%]{4,10}"     }   ],   "errorMessages": [] }</pre>	<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <ns2:BillFetchCustomParams xmlns:ns2="http://bbps.rsssoftware.co.in/schema"> <customParams> <name>Account</name> <type>ALPHANUMERIC</type> <isMandatory>true</isMandatory> <minLength>2</minLength> <maxLength>20</maxLength> <regex>^[a-zA-Z0-9@#\$%]{4,10}</regex> </customParams> <customParams> <name>Product Id</name> <type>ALPHANUMERIC</type> <isMandatory>false</isMandatory> <minLength>1</minLength> <maxLength>25</maxLength> <regex>^[a-zA-Z0-9@#\$%]{4,10}</regex> </customParams> </ns2:BillFetchCustomParams>
-------------------------	--	---

### 2.1.5 Fetch Bill

This api is used to fetch bill against a biller after providing the proper customer parameter fields. Providing any incorrect information the bill data can't be fetched.

Action	Fetch bill for the customer	
<b>API</b>	<Domain>/ APIService/bill-fetch-form-post/urn:tenantId:< OU Id >	
<b>Request Header</b>		
	Authorization	Basic Base64.encode(<AgentId>:<Password>); E.g. Basic T1U3MFNMMTJJTkI1MjA5MzY0NTEx6cGFzc3dvcmQ=

	Content-Type	application/xml   application/json
	Accept	application/xml   application/json
	Request Method	POST
		<p><b>*<u>origInst</u></b>  <b>Description:</b>          4digit OU ID</p> <p><b><u>riskScores</u></b>  <b>Description:</b>          risk evaluation associated with the transaction and the related parties in the transaction.</p> <p><b><u>Provider</u></b>  <b>Description:</b>          Entity providing the risk score.</p> <p><b><u>Type</u></b>  <b>Description:</b>          describes the type of risk</p> <p><b><u>Value</u></b>  <b>Description:</b>          Value of risk evaluation</p> <p><b>*<u>Customer</u></b>  <b>Description:</b>          Customer related details who initiates the transaction</p> <p><b><u>Tag</u></b>  <b>Description:</b>          Customer Tag Name &amp; Value</p> <p><b>*<u>Mobile</u></b>  <b>Description:</b>          10digit Mobile number of the customer starts with 7/8/9.</p> <p><b>*<u>agent</u></b>  <b>Description:</b>          detail information of Agent.</p> <p><b><u>agentId</u></b>  <b>Description:</b>          20digit unique Agent ID of the Agent involved in the transaction  <b>Data type:</b>          Alphanumeric          E.g: RB76RB12AGT527795726Agent Id</p> <p><b>*<u>Device</u></b>  <b>Description:</b>          Details of the device from which the transaction was initiated .</p> <p><b>*<u>billDetails</u></b>  <b>Description:</b>          Details of the Customer whose bill is generated for a Biller</p> <p><b>*<u>Biller Id</u></b>  <b>Description:</b>          14digit Biller ID</p> <p><b>*<u>CustomerParams</u></b>  <b>Description:</b>          It's required for Biller i.e. Name &amp; Value  <b>Data Type:</b></p>

		Text
<b>Response Header</b>		
	requestId	b46dd68e-b4fc-4cc2-b37a-1b38ef4c4d6a
	Content-Type	application/xml   application/json
<b>Response</b>		<p><b><u>refId</u></b>  <b>Description :</b>            Reference ID is a unique id to maintain the chain of request and response</p>
		<p><b><u>approvalRefNum</u></b>  <b>Description:</b>            Code for the Biller BBPOU to indicate acceptance of the request message if transaction is successful</p>
		<p><b><u>responseCode</u></b>  <b>Description:</b>            Carries the response code indicating success or failure of the transaction</p>
		<p><b><u>responseReason</u></b>  <b>Description:</b>            Description of the response code – possible values are “Successful” or “Failure”</p>
		<p><b><u>billResponse</u></b>  <b>Description</b>            Complete bill information provided by the biller           <ul style="list-style-type: none"> <li>• <b>customerName</b> - Customer Name</li> <li>• <b>amount</b> – Bill Amount</li> <li>• <b>dueDate</b> – Bill Due Date</li> <li>• <b>billDate</b> – Bill Date</li> <li>• <b>billNumber</b> – Bill Number</li> <li>• <b>billPeriod</b> – Bill Period</li> </ul> </p>
		<p><b><u>Tag</u></b>  <b>Description</b>            Biller Response Tag Name and Value</p>
		<p><b><u>additionalInfo</u></b>  <b>Description</b>            Additional Information of Biller.</p>
		<p><b><u>OtherPaymentFlag</u></b>  <b>Description:</b>            Other Payment Status i.e True/False</p>
	<b>JSON Format</b>	<b>XML Format</b>
<b>Request Example</b>	<pre>{   "billDetails": {     "biller": {       "id": "ONLI00000WBGSM"     },     "customerParams": {       "tags": [         "tag1",         "tag2"       ]     }   } }</pre>	<?xml version="1.0" encoding="UTF-8"?> <ns2:FetchRequest   xmlns:ns2="http://bbps.rssoftware.co.in/schema">   <origInst>RB09</origInst>   <riskScores>     <Score provider="RB09" type="TXNRISK" value="030"/>   </riskScores>   <customer mobile="9332417385">



	<pre>         },         {           "name":         "30.00 BillerResponse2",           "value": "BillerResponse2 (30.00)=30.00"         }       ],       "customerName": "Arpita Mukherjee",       "amount": "6000",       "dueDate": "2019-12-31",       "custConvFee": null,       "custConvDesc": null,       "billDate": "2018-02-08",       "billNumber": "ASDF11111111",       "billPeriod": "September"     },     "additionalInfo": {       "tags": [         {           "name": "Energy Charges",           "value": "Electricity"         },         {           "name": "MVCA",           "value": "AAAA"         }       ]     },     "errorMessages": [],     "agentChannelDetails": null,     "otherPaymentFlag": true   } } </pre>	
--	---	--

## 2.1.6 Validate Customer Parameters (Pay-Only Biller)

This api is used to determine whether the customer parameters entered by user are correct. This is to prevent unwanted reversal and incorrect payments. It returns a REF ID and Additonal info. The REFID should be used in subsequent Payment Request, the additional Information should be shown to customer for confirmation.

Action	Validate Customer parameter from Biller	
API	<domain>/APIService/ customer-validation-form-post /urn:tenantId:<OU Id>	
Request Header		
	Authorization	Basic Base64.encode(<AgentId>:<Password>); E.g. Basic T1U3MFNMMTJJTk1MjA5MzY0NTExcGFzc3dvcmQ=
	Content-Type	application/xml   application/json

	Accept	application/xml   application/json
	Request Method	POST
<b>Request</b>		<p><b>*origInst</b>            Description:            4digit Customer OU Id.</p>
		<p><b>*Agent</b>            Description:            Detail Agent Information.            • <b>id:</b> 20digit agent id</p>
		<p><b>Device:</b>            Description:            Detail Information of Agent Device in Tag name value format.</p>
		<p><b>*BillDetails:</b>            Description:            Details of the Customer whose bill is generated for a Biller</p> <p><b>*Biller</b>            Description:            Details information of Biller            • <b>*id:</b> 14digit biller id.</p>
		<p><b>*CustomerParams</b>            Description:            It's required for Biller i.e. Name &amp; Value</p>
<b>Response Header</b>		
	requestId	b46dd68e-b4fc-4cc2-b37a-1b38ef4c4d6a
	Content-Type	application/xml   application/json
<b>Response</b>		<p><b>origInst</b>            Description:            Customer OU Id (4 chars)</p> <p><b>billDetails –</b>            Description:            Details of Biller            • <b>Biller id–</b> 14digit Biller Id            • <b>CustomerParams –</b> Biller specific customparam information in Tag name / value format.</p>
	<b>JSON Format</b>	<b>XML Format</b>
<b>Request Example</b>	<pre>{   "origInst": "RB09",   "billDetails": {     "biller": {       "id": "ONLI00000WBGQP"     },     "customerParams": {       "tags": [         ...       ]     } }</pre>	<pre>&lt;?xml version="1.0" encoding="UTF-8" standalone="yes"?&gt; &lt;ns2:FetchValidationRequest   xmlns:ns2="http://bbps.rsssoftware.co.in/schema"&gt;   &lt;origInst&gt;RB09&lt;/origInst&gt;   &lt;BillDetails&gt;     &lt;Biller id="ONLI00000WBGQP"/&gt;     &lt;CustomerParams&gt;       &lt;Tag name="Consumer No." value="100"/&gt;       &lt;Tag name='Account1' value='200' /&gt;     &lt;/CustomerParams&gt;   &lt;/BillDetails&gt; &lt;/ns2:FetchValidationRequest&gt;</pre>

## 2.1.7 Get Tax Amount

This api is used to calculate all kind of taxes.

Action	Calculate CCF
<b>API</b>	<Domain/APIService/calculate-taxes/urn:tenantId:<Ouid>
<b>Request Header</b>	
Authorization	Basic Base64.encode(<AgentId>:<Password>); E.g. Basic T1U3MFNMMTJJTkI1MjA5MzY0NTE6cGFzc3dvcmQ=
	Content-Type application/xml   application/json

	Accept	application/xml   application/json
	Request Method	POST
<b>Request</b>		<p><b><u>billAmount</u></b>  <b>Description:</b>            Bill Amount in Paisa</p> <p><b><u>billId</u></b>  <b>Description:</b>            14digit Biller Id</p> <p><b><u>paymentMethod</u></b>  <b>Description:</b>            Payment Method Details:            quickPay value ' Yes'/'No'            splitPay value always 'No'</p> <p><b><u>paymentMode</u></b>  <b>Description</b>            Agent supported payment Mode .</p>
<b>Response Header</b>	requestId	b46dd68e-b4fc-4cc2-b37a-1b38ef4c4d6a
	Content-Type	application/xml   application/json
<b>Response</b>		<p><b><u>amount</u></b>  <b>Description</b>            payable customer Convenience fee amount</p> <ul style="list-style-type: none"> <li>• <b>igst</b>- Integrated GST               <ul style="list-style-type: none"> <li>○ flatfee- Flat tax amount</li> <li>○ percentfee- Percentage tax amount</li> </ul> </li> <li>• <b>cgst</b>- Central GST               <ul style="list-style-type: none"> <li>○ flatfee- Flat tax amount</li> <li>○ percentfee- Percentage tax amount</li> </ul> </li> <li>• <b>sutgst</b> – State or Union Territory GST               <ul style="list-style-type: none"> <li>○ flatfee- Flat tax amount</li> <li>○ percentfee- Percentage tax amount</li> </ul> </li> </ul>
	<b>Json Format</b>	<b>XML Format</b>
<b>Request Example</b>	<pre>{   "billAmount": "15000",   "billId": "RELI00000NAT4I",   "agent": null,   "paymentMethod": {     "quickPay": "NO",     "splitPay": "NO",     "paymentMode": "Internet Banking"   } }</pre>	<?xml version="1.0" encoding="UTF-8"?> <bbps:CalTaxReq xmlns:bbps="http://bbps.rssoftware.co.in/schema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://bbps.rssoftware.co.in/schema CalTaxReq.xsd ">   <billAmount>15200</billAmount>   <billerId>RELI00000NAT4I</billerId>   <paymentMethod OFFUSPay="Yes" paymentMode="Internier_Banking" quickPay="Yes" splitPay="No"/> </bbps:CalTaxReq>

<b>Response Example</b>	<pre>{   "ccf": {     "amount": "250",     "igst": {       "value": "45",       "flatfee": "0",       "percentfee": "18.00"     },     "cgst": {       "value": "0",       "flatfee": "0",       "percentfee": "0"     },     "sutgst": {       "value": "0",       "flatfee": "0",       "percentfee": "0"     }   },   "couccf": {     "amount": "0",     "igst": {       "value": "0",       "flatfee": "0",       "percentfee": "18.00"     },     "cgst": {       "value": "0",       "flatfee": "0",       "percentfee": "0"     },     "sutgst": {       "value": "0",       "flatfee": "0",       "percentfee": "0"     }   },   "errors": [] } </pre>	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;bbps:CalTaxResp xmlns:bbps="http://bbps.rssoftware.co.in/schema"   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"   xsi:schemaLocation="http://bbps.rssoftware.co.in/schema   CalTaxResp.xsd "&gt;  &lt;ccf&gt;   &lt;amount&gt;1200&lt;/amount&gt;   &lt;igst flatfee="0" percentfee="0"&gt;0&lt;/igst&gt;   &lt;cgst flatfee="0" percentfee="18"&gt;216&lt;/cgst&gt;   &lt;sutgst flatfee="0" percentfee="0"&gt;0&lt;/sutgst&gt; &lt;/ccf&gt; &lt;couccf&gt;   &lt;amount&gt;200&lt;/amount&gt;   &lt;igst flatfee="0" percentfee="9"&gt;24&lt;/igst&gt;   &lt;cgst flatfee="0" percentfee="0"&gt;0&lt;/cgst&gt;   &lt;sutgst flatfee="0" percentfee="9"&gt;24&lt;/sutgst&gt; &lt;/couccf&gt; &lt;/bbps:CalTaxResp&gt;</pre>
-------------------------	--	---

### 2.1.8 Pay Bill (Fetch & Pay Bill) \*\*

This api is used to Fetch and Pay. After bill fetch this api is used for payment. It required the unique refId which is provided in "bill-fetch-form-post" api.

Action	Submit Payment Request	
<b>API</b>	<Domain> /APIService/bill-payment-form-post/urn:tenantId:<OU Id>	
<b>Request Header</b>		
	Authorization	Basic Base64.encode(<AgentId>:<Password>);

	E.g. Basic T1U3MFNMMTJJTkI1MjA5MzY0NTE6cGFzc3dvcnQ=
Content-Type	application/xml   application/json
Accept	application/xml   application/json
Request Method	POST
<b>Request</b>	<p><b>*<u>refId</u></b>            Description:            OU ref Id returned in the Bill Fetch Response (mandatory)  <b>Data Type:</b>            Alphanumeric Fixed Length – 35</p> <p><b>*<u>customer</u></b>            Description:            Customer related details who initiates the transaction</p> <p><b>*<u>mobile</u></b>            Description:            Mobile number of the customer</p> <p><b><u>Tag</u></b>            Description:            Customer Tag Name &amp; Value</p> <p><b>*<u>billDetails</u></b>            Description:            Details of the Customer whose bill is generated for a Biller</p> <p><b>*<u>id</u></b>            Description:            14digit Biller ID</p> <p><b>*<u>CustomerParams</u></b>            Description:            It's required for Biller i.e. Name &amp; Value</p> <p><b>*<u>paymentMethod</u></b>            Description            Payment method of the customer</p> <p><b>*<u>quickPay</u></b>            Description:            Flag indicating if the payment is initiated without a fetch or not. For this API the value should be 'NO'</p> <p><b>*<u>splitPay</u></b>            Description            Flag indicating if the bill is paid using two different payment modes . For this API the value should be 'NO'</p> <p><b>*<u>paymentMode</u></b>            Description            The payment mode which is accepted from the Customer.</p> <p><b>*<u>amount</u></b>            Description            Amount to be paid           <ul style="list-style-type: none"> <li>• amount – Amount should be in Paisa format.</li> <li>• splitPayAmount - Split pay amount accept '0' only.</li> </ul>           customerConvinienceFee – Customer Convenience Fee amount should be in         </p>

		paisa format.
		<p><b><u>*paymentInformation</u></b></p> <p>Description:</p> <p>Payment information of the instrument which is used for making the bill payment</p> <p><b><u>*Name</u></b></p> <p>Description</p> <p>Payment Information Tag Name</p> <p><b><u>*Value</u></b></p> <p>Description</p> <p>Payment Information Tag Value</p>
		<p><b><u>*amountOptions</u></b></p> <p>Description:</p> <p>The selected Bill Payment amount Option</p> <p><b><u>paymentStatus:</u></b></p> <p>Description</p> <p>Payment status should be RESPONSE_SUCCESS</p>
		<p><b><u>finTxnDetails&gt;txnRefId:</u></b></p> <p>Description</p> <p>user specific client reference Id.</p>
		<p><b><u>finTxnDetails&gt;txnRefId:</u></b></p> <p>Description</p> <p>user specific client reference Id.</p>
<b><i>Response Header</i></b>		b46
	requestId	application/xml   application/json
	Content-Type	
<b><i>Response</i></b>		<p><b><u>customerName</u></b></p> <p>Description</p> <p>Customer name who initiates the transaction</p> <p><b><u>transactionRefId</u></b></p> <p>Description</p> <p>Transaction reference id of the transaction</p> <p><b><u>transactionDateTime</u></b></p> <p>Description</p> <p>Date and time of the transaction in YYYY-MM-DD HH:MM format</p> <p><b><u>billDate</u></b></p> <p>Description</p> <p>Generation date of the bill</p> <p><b><u>billerId</u></b></p> <p>Description</p> <p>Biller Id Details</p> <p><b><u>billNumber</u></b></p> <p>Description</p> <p>Unique identifier of the bill</p> <p><b><u>billPeriod</u></b></p> <p>Description</p> <p>Billing period of the bill</p>

	<p><b><u>billerName</u></b> Description Biller Name</p> <p><b><u>Amount</u></b> Description Bill payment amount</p> <p><b><u>Ccf</u></b> Description Customer Convenience Fee amount</p> <p><b><u>paymentChannel</u></b> Description Payment Channel Details</p> <p><b><u>paymentMode</u></b> Description Payment Mode Details</p> <p><b><u>customerMobile</u></b> Description 10 digit Customer Mobile Number starts with 5/6/7/8/9</p> <p><b><u>refId</u></b> Description 35 digit unique Reference Id</p> <p><b><u>clientRefId</u></b> Description Client Ref Id Details</p> <p><b><u>transactionRefId:</u></b> Description Transaction reference id will generate automatically..</p> <p><b><u>authCode</u></b> authorization code of a particular transaction</p>
	<p><b><u>Json Format</u></b></p> <pre>{   "amount": {     "amount": "5000",     "splitPayAmount": "0",     "tag": {       "name": "Energy Charges",       "value": "0"     },     "customerConvinienceFee": "61"   },   "billDetails": {     "biller": {       "id": "ONLI00000NATWL"     },     "customerParams": {       "tags": [         {           "name": "Consumer No.",           "value": "#ABCD--/123"         }       ]     }   },   "customer": {} }</pre> <p><b><u>XML Format</u></b></p> <pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;ns2:PaymentRequest xmlns:ns2="http://bbps.rsssoftware.co.in/schema"&gt;   &lt;refId&gt;RB092019070300000000000000001004&lt;/refId&gt;   &lt;riskScores&gt;     &lt;Score provider="RB09" type="TXNRISK" value="030" /&gt;   &lt;/riskScores&gt;   &lt;customer mobile="9879879870"&gt;     &lt;Tag name="EMAIL" value="arpita@yahoo.com" /&gt;     &lt;Tag name="AADHAAR" value="499118665246" /&gt;     &lt;Tag name="PAN" value="BHKPS4288L" /&gt;   &lt;/customer&gt;   &lt;billDetails&gt;     &lt;Biller id="ONLI00000WBGSM"/&gt;     &lt;CustomerParams&gt;       &lt;Tag name="Account" value="1000" /&gt;     &lt;/CustomerParams&gt;   &lt;/billDetails&gt;   &lt;paymentMethod quickPay="No" splitPay="No" paymentMode="Cash" OFFUSPay="Yes" /&gt;   &lt;amount&gt;     &lt;amount&gt;12000&lt;/amount&gt;   &lt;/amount&gt; </pre>

<p><b>Request Example</b></p>	<pre> "mobile":"9830535956", "tags":[ ], }, "finTxnDetails":{ "paymentDetails":"5000 BASE_BILL_AMOUNT", "paymentStatus":"RESPONSE_SUCCESS", "txnRefId":"ClientRefID1234", "authCode":"authcode1234" }, "paymentInformation": { "tags": [ { "name": "IFSC AccountNo", "value": "765765675 1111" } ] }, "paymentMethod": { "paymentMode": "Internet Banking", "quickPay": "NO", "splitPay": "NO" }, "refId": "RB092020070900000000000000000000000000445" } } </pre>	<pre> &lt;splitPayAmount&gt;0&lt;/splitPayAmount&gt; &lt;customerConvinienceFee&gt;143&lt;/customerConvinienceFee&gt; &lt;COUcustConvFee&gt;354&lt;/COUcustConvFee&gt; &lt;/amount&gt; &lt;paymentInformation&gt; &lt;Tag name="Remarks" value="IDBI Payment with Cash" /&gt; &lt;/paymentInformation&gt; &lt;finTxnDetails&gt; &lt;txnRefId&gt;ClientRefID1234&lt;/txnRefId&gt; &lt;authCode&gt;1112345678&lt;/authCode&gt; &lt;otherPaymentFlag&gt;false&lt;/otherPaymentFlag&gt; &lt;paymentDetails&gt;12000 BillerResponse1 BillerResponse2 BASE_BILL_AMOUNT&lt;/paymentDetails&gt; &lt;paymentStatus&gt;RESPONSE_SUCCESS&lt;/paymentStatus&gt; &lt;/finTxnDetails&gt; &lt;/ns2:PaymentRequest&gt; </pre>
		<pre> &lt;?xml version="1.0" encoding="UTF-8" standalone="yes"?&gt; &lt;ns2:PaymentReceipt xmlns:ns2="http://bbps.rssoftware.co.in/schema"&gt; &lt;customerName&gt;Arpita Mukherjee&lt;/customerName&gt; &lt;transactionRefId&gt;RB0900067294&lt;/transactionRefId&gt; &lt;transactionDateTime&gt;2018-07-09T07:24:00Z&lt;/transactionDateTime&gt; &lt;billDate&gt;2019-02-18&lt;/billDate&gt; &lt;billerId&gt;ONLI00000NATWL&lt;/billerId&gt; &lt;billNumber&gt;ASD222&lt;/billNumber&gt; &lt;billPeriod&gt;August&lt;/billPeriod&gt; &lt;billerName&gt;OnlineMandatoryBiller&lt;/billerName&gt; &lt;approvalRefNum&gt;&lt;/approvalRefNum&gt; &lt;amount&gt;5000&lt;/amount&gt; &lt;ccf&gt;61&lt;/ccf&gt; &lt;coupCcf&gt;&lt;/coupCcf&gt; &lt;paymentChannel&gt;Internet&lt;/paymentChannel&gt; &lt;paymentMode&gt;Internet Banking&lt;/paymentMode&gt; &lt;customerMobile&gt;2203456789&lt;/customerMobile&gt; &lt;refId&gt;RB092020070900000000000000000000000000445&lt;/refId&gt; &lt;clientRefId&gt;ClientRefID1234&lt;/clientRefId&gt; &lt;customerParams&gt;{ "tags": [ { "name": "Consumer No.", "value": "#ABCD--/123" } ] }, "dueDate": "2023-12-28", "errorMessages": [], "authCode": null } </pre>

## 2.1.9 Pay Bill (Pay Only) \*\*

This api is used to payment for Pay only amount. In this case no need to fetch the bill. The attribute **quickPay** should be **Yes** for this payment. If the Biller supports the' Validate Customer Parameters' API in that case **quickPay** should be **No** and RefId should be passing in the Payment Request which is generated from 'Validate Customer Parameters' API.

Action	Submit Payment Request (Quick Pay)	
<b>API</b>	<Domain> /APIService/bill-payment-form-post/urn:tenantId:<OU Id>	
<b>Request Header</b>		
	Authorization	Basic Base64.encode(<AgentId>:<Password>); E.g. Basic T1U3MFNMMTJJTkI1MjA5MzY0NTE6cGFzc3dvcmQ=
	Content-Type	application/xml   application/json
	Accept	application/xml   application/json
	Request Method	Post
<b>Request</b>		<p><b><u>refId</u></b>            Description:            If the Biller supports the' Validate Customer Parameters' API in that case RefId should be pass in the Payment Request which is generated from 'Validate Customer Parameters' API.</p> <p><b><u>*customer</u></b>            Description:            Customer related details who initiates the transaction</p> <p><b><u>*mobile</u></b>            Description:            10digit Mobile number of the customer starts with 7/8/9.</p> <p><b><u>*Tag</u></b>            Description:            Customer Tag Name &amp; Value</p> <p><b><u>*billDetails</u></b>            Description:            Details of the Customer whose bill is generated for a Biller</p> <p><b><u>*Id</u></b>            Description:            14digit Biller ID</p> <p><b><u>*CustomerParams</u></b>            Description:            It's required for Biller i.e. Name &amp; Value</p> <p><b><u>*paymentMethod</u></b>            Description            Payment method of the customer</p> <p><b><u>*quickPay</u></b></p>

		<p>Description:</p> <p>Flag indicating if the payment is initiated without a fetch or not. For this API the value should be 'YES' in json format request.</p> <p>'Yes' in xml format request</p> <p>If the Biller supports the' Validate Customer Parameters' API in that case quickPay should be "NO" in json format request..</p> <p>'No' in xml format request.</p> <p><b>*splitPays</b></p> <p>Description</p> <p>Flag indicating if the bill is paid using two different payment modes . For this API the value should be 'NO' in json format request</p> <p>For this API the value should be 'No' in xmIs format request</p> <p><b>*paymentMode</b></p> <p>Description</p> <p>The payment mode which is accepted from the Customer.</p>
		<p><b>*amount</b></p> <p>Description</p> <p>Amount to be paid</p> <ul style="list-style-type: none"> <li>• amount – Amount should be in Paisa format.</li> <li>• splitPayAmount - Split pay amount accept '0' only.</li> </ul> <p>customerConvinieneFee – Customer Convenience Fee amount should be in paisa format.</p>
		<p><b>*paymentInformation</b></p> <p>Description:</p> <p>Payment information of the instrument which is used for making the bill payment</p> <p><b>*Name</b></p> <p>Description</p> <p>Payment Information Tag Name</p> <p><b>*Value</b></p> <p>Description</p> <p>Payment Information Tag Value</p>
		<p><b>finTxnDetails&gt;txnRefId:</b></p> <p>Description</p> <p>user specific client reference Id.</p>
		<p><b>authCode:</b></p> <p>authorization code of particular transaction</p>
<b>Response Header</b>		
	requestId	b46dd68e-b4fc-4cc2-b37a-1b38ef4c4d6a
	Content-Type	application/xml   application/json
<b>Response</b>		
		<p><b>customerName</b></p> <p>Description</p> <p>Customer name who initiates the transaction</p>
		<p><b>transactionRefId</b></p> <p>Description</p>

		<p>Transaction reference id of the transaction</p> <p><b><u>transactionDateTime</u></b></p> <p>Description</p> <p>Date and time of the transaction in YYYY-MM-DD HH:MM format</p> <p><b><u>billDate</u></b></p> <p>Description</p> <p>Generation date of the bill</p> <p><b><u>billerId</u></b></p> <p>Description</p> <p>Biller Id Details</p> <p><b><u>billNumber</u></b></p> <p>Description</p> <p>Unique identifier of the bill</p> <p><b><u>billPeriod</u></b></p> <p>Description</p> <p>Billing period of the bill</p> <p><b><u>billerName</u></b></p> <p>Description</p> <p>Biller Name</p> <p><b><u>Amount</u></b></p> <p>Description</p> <p>Bill payment amount</p> <p><b><u>Ccf</u></b></p> <p>Description</p> <p>Customer Convenience Fee amount</p> <p><b><u>paymentChannel</u></b></p> <p>Description</p> <p>Payment Channel Details</p> <p><b><u>paymentMode</u></b></p> <p>Description</p> <p>Payment Mode Details</p> <p><b><u>customerMobile</u></b></p> <p>Description</p> <p>10 digit Customer Mobile Number starts with 7/8/9</p> <p><b><u>refId</u></b></p> <p>Description</p> <p>35 digit unique Reference Id</p> <p><b><u>clientRefId</u></b></p> <p>Description</p> <p>Client Ref Id Details</p>
	<b><u>Json Format</u></b>	<b><u>XML Format</u></b>
<b><u>Request Example</u></b>	<pre>{   "amount": {     "amount": "6000",     "splitPayAmount": "0",     "tag": {       "name": "Energy Charges",       "value": "0"     },     "customerConvinienceFee": "73" }</pre>	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;ns2:PaymentRequest xmlns:ns2="http://bbps.rsssoftware.co.in/schema"&gt;   &lt;!--refId--&gt;RB092019070300000000000000000681&lt;/refId--&gt;   &lt;txRefId&gt;RB09105D1705&lt;/txRefId&gt;   &lt;riskScores&gt;     &lt;Score provider="RB09" type="TXNRISK" value="030" /&gt;   &lt;/riskScores&gt;   &lt;customer mobile="9879879870"&gt;     &lt;Tag name="EMAIL" value="arpita@yahoo.com" /&gt;</pre>



### 2.1.10 Get Customer Transactions

This api is used to search previous transaction. This api will provide data either by transaction ref id directly or by mobile number. Both the sample has attached below.

Action	Search Transactions	
API	<domain>/APIService/complaint/searchtransaction/urn:tenantId:<OU Id>	
Request Header		
Request	Authorization	Basic Base64.encode(<AgentId>:<Password>); E.g. Basic T1U3MFNMMTJJTkI1MjA5MzYONTE6cGFzc3dvcmQ=
	Content-Type	application/xml   application/json
	Accept	application/xml   application/json
	Request Method	Post
Request		<b>*<u>txnrreferenceld</u></b> Description Transaction reference number used by the Customer for referring to a Payment transaction <b>Data Type:</b> Alphanumeric Fixed Length – 12
		<b>*<u>fromDate</u></b> Description start date of the transaction search. Date format: YYYY-MM-DD
		<b>*<u>toDate</u></b> Description end date of the transaction search. Date format: YYYY-MM-DD
Response Header		
	requestId	b46dd68e-b4fc-4cc2-b37a-1b38ef4c4d6a
	Content-Type	application/xml   application/json
Response		
		<b><u>txndetails</u></b> Description Transaction details <b><u>txnrreferenceld</u></b> Description Transaction ref id <b><u>amount</u></b> Description Amount paid in the transaction <b><u>txndate</u></b> Description

		<p>Date of the transaction</p> <p><b><u>agentId</u></b></p> <p>Description:</p> <p>20digit unique Agent ID of the Agent involved in the transaction</p> <p>Data type:</p> <p>Alphanumeric</p> <p>E.g: RB76RB12AGT527795726</p> <p><b><u>billerId</u></b></p> <p>Description</p> <p>14 digit Biller id</p> <p><b><u>txnStatus</u></b></p> <p>Description</p> <p>Transaction status</p>
		<p><b><u>mobile-</u></b></p> <p>Description</p> <p>Mobile number used for the transaction</p>
	<b><i>Json Format</i></b>	<b><i>XML Format</i></b>
<b><i>Request Example1</i></b>	{"txnid": "RB7600013601"}	<ns2:TransactionSearchRequest xmlns:ns2="http://bbps.rsssoftware.co.in/schema"> <txnid>OU2100029748</txnid> </ns2:TransactionSearchRequest>
<b><i>Response Example1</i></b>	{     "errors": [],     "txndetails": [       {         "txnid": "RB7600013601",         "amount": "6000",         "txndate": "11-12-2017 06:07 PM",         "agentid": "RB76RB12AGT528495783",         "billerid": "ONLI00000WBGSMS",         "txnstatus": "SUCCESS"       }     ],     "mobile": "9999999999",     "errormessages": []   }	<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <ns2:TxnSearchResponse xmlns:ns2="http://bbps.rsssoftware.co.in/schema"> <txndetails txnid="OU2100029748" amount="6000" txndate="05-03-2018 06:52 PM" agentid="OU21AD12BNK519157639" billerid="RELI0000NAT3I" txnstatus="SUCCESS"/> <mobile>9811111111</mobile> </ns2:TxnSearchResponse>
<b><i>Request Example2</i></b>	{"fromDate": "2017-12-06", "mobile": "9999999999", "toDate": "2017-12-06"}	<ns2:TransactionSearchRequest xmlns:ns2="http://bbps.rsssoftware.co.in/schema"> <fromDate>2018-03-01</fromDate> <toDate>2018-03-05</toDate> <mobile>9811111111</mobile> </ns2:TransactionSearchRequest>
<b><i>Response Example2</i></b>	{     "errors": [],     "txndetails": [       {         "txnid": "RS3500059096",         "amount": "1000",         "txndate": "06-12-2017 11:27 AM",         "agentid": "MN10MN02BNK519348138"       }     ]   }	<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <ns2:TxnSearchResponse xmlns:ns2="http://bbps.rsssoftware.co.in/schema"> <txndetails txnid="MN1000000080" amount="100" txndate="05-03-2018 07:08 PM" agentid="MN10MN02BNK519348138" billerid="RELI0000NAT3I" txnstatus="SUCCESS"/> <txndetails txnid="OU2100029748" amount="6000"

	<pre>     "RS35AT03AGT512738960",       "billerId": "DEMO00000WBGAY",       "txnStatus": "SUCCESS"     },     {       "txnReferenceId": "RS3500059094",       "amount": "1000",       "txnDate": "06-12-2017 10:52 AM",       "agentId":         "RS35AT03AGT512738960",         "billerId": "DEMO00000WBGAY",         "txnStatus": "SUCCESS"     }   ],   "mobile": "9999999999",   "errorMessages": [] } </pre>	<pre>   txnDate="05-03-2018 06:52 PM"   agentId="OU21AD12BNK519157639" billerId="RELI00000NAT3I"   txnStatus="SUCCESS"/&gt;   &lt;txnDetails txnReferenceId="OU2100029747" amount="10000"   txnDate="02-03-2018 02:31 PM"   agentId="OU21AD12BNK519157639" billerId="RELI00000NAT4I"   txnStatus="SUCCESS"/&gt;   &lt;txnDetails txnReferenceId="OU2100029746" amount="6000"   txnDate="02-03-2018 02:30 PM"   agentId="OU21AD12BNK519157639" billerId="RELI00000NAT3I"   txnStatus="SUCCESS"/&gt;   &lt;mobile&gt;9811111111&lt;/mobile&gt; &lt;/ns2:TxnSearchResponse&gt; </pre>
--	---	--

### 2.1.11 Get Txn Complaint Dispositions

This api is used to fetch disposition list. These dispositions are used to raise transaction type complaint.

Action	Get Txn Complaint Dispositions	
API	<domain>/APIService/complaint/urn:tenantId:<OU Id>/dispositionlist	
Request Header		
	Authorization	Basic Base64.encode(<AgentId>:<Password>); E.g. Basic T1U3MFNMMTJJTk1MjA5MzY0NTE6cGFzc3dvcmQ=
	Content-Type	application/xml   application/json
	Accept	application/xml   application/json
	Request Method	GET
Request	No Payload required	
Response Header		
	Content-Type	application/xml   application/json
	requestId	b46dd68e-b4fc-4cc2-b37a-1b38ef4c4d6a
Response		<p><b><u>disposition</u></b></p> <p>Description:</p> <ul style="list-style-type: none"> <li>• <b>id</b>- Id of the disposition</li> <li>• <b>name</b>- Name of the disposition.</li> </ul>
	<b>Json Format</b>	
Response Example	{ "errors": [], "dispositions": [	
	<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <ns2:dispositionList	

	<pre>{     "id": "1",     "name": "Transaction Succesful, account not updated" }, {     "id": "2",     "name": "Amount deducted, biller account credited but transaction ID not received" }, {     "id": "3",     "name": "Amount deducted, biller account not credited &amp; ID not received" }, {     "id": "4",     "name": "Amount deducted multiple times" }, {     "id": "5",     "name": "Double payment updated" }, {     "id": "6",     "name": "Erroneously paid in wrong account" }, {     "id": "7",     "name": "Others, provide details in description" } ]</pre>	<pre>xmlns:ns2="http://bbps.rsssoftware.co.in/schema"&gt; &lt;disposition&gt; &lt;id&gt;1&lt;/id&gt; &lt;name&gt;Transaction Succesful, account not updated&lt;/name&gt; &lt;/disposition&gt; &lt;disposition&gt; &lt;id&gt;2&lt;/id&gt; &lt;name&gt;Amount deducted, biller account credited but transaction ID not received&lt;/name&gt; &lt;/disposition&gt; &lt;disposition&gt; &lt;id&gt;3&lt;/id&gt; &lt;name&gt;Amount deducted, biller account not credited &amp; transaction ID not received&lt;/name&gt; &lt;/disposition&gt; &lt;disposition&gt; &lt;id&gt;4&lt;/id&gt; &lt;name&gt;Amount deducted multiple times&lt;/name&gt; &lt;/disposition&gt; &lt;disposition&gt; &lt;id&gt;5&lt;/id&gt; &lt;name&gt;Double payment updated&lt;/name&gt; &lt;/disposition&gt; &lt;disposition&gt; &lt;id&gt;6&lt;/id&gt; &lt;name&gt;Erroneously paid in wrong account&lt;/name&gt; &lt;/disposition&gt; &lt;disposition&gt; &lt;id&gt;7&lt;/id&gt; &lt;name&gt;Others, provide details in description&lt;/name&gt; &lt;/disposition&gt; &lt;/ns2:dispositionList&gt;</pre>
--	---	---

### 2.1.12 Get Service Reasons

This api is used to fetch service reasons. These services are used to raise service type complaint.

Action	Get Service Reasons	
API	<domain>/APIService/complaint/getAllServiceReason/urn:tenantId:<OU Id>	
Request Header		
	Authorization	Basic Base64.encode(<AgentId>:<Password>);

		E.g. Basic T1U3MFNMMTJJTkI1MjA5MzY0NTE6cGFzc3dvcnQ=
	Content-Type	application/xml   application/json
	Accept	application/xml   application/json
	Request Method	POST
<b>Request</b>		
		<b>*participationType</b> <b>Description:</b> Participation types are agent/biller/system
<b>Response Header</b>		
	Content-Type	application/xml   application/json
<b>Response</b>		<b>serviceReason</b> <b>Description:</b> <b>id</b> <b>Description:</b> <i>ID</i> of the service reason <b>name</b> <b>Description:</b> Name of the service reason <b>reasonType</b> <b>Description:</b> Service reason type(Agent, Biller, System) <b>reasonCode</b> <b>Description:</b> <ul style="list-style-type: none"> <li>• Service reason code</li> </ul>
	<b>Json Format</b>	<b>XML Format</b>
<b>Request Example</b>	{"participationType":"biller"}	<ns2:ParticipationData xmlns:ns2="http://bbps.rsssoftware.co.in/schema"> <participationType>Biller</participationType> </ns2:ParticipationData>
<b>Response Example</b>	<pre>{   "errors": [],   "serviceReasons": [     {       "id": "12",       "name": "Biller available. Unable to transact",       "reasonType": "SYSTEM",       "reasonCode": "A12"     },     {       "id": "13",       "name": "Multiple failure for same biller",       "reasonType": "SYSTEM",       "reasonCode": "A13"     }   ] }</pre>	<?xml version="1.0" encoding="UTF-8"?> <standalone="yes"?> <ns2:ServiceReasonList xmlns:ns2="http://bbps.rsssoftware.co.in/schema">   <serviceReason>     <id>7</id>     <name>Biller available. Unable to transact</name>     <reasonType>BILLER</reasonType>     <reasonCode>A7</reasonCode>   </serviceReason>   <serviceReason>     <id>8</id>     <name>Multiple failure for same biller</name>     <reasonType>BILLER</reasonType>     <reasonCode>A8</reasonCode>   </serviceReason> </serviceReasonList>

	<pre>     }   </pre>	<pre> &lt;id&gt;9&lt;/id&gt; &lt;name&gt;Denomination not available&lt;/name&gt; &lt;reasonType&gt;BILLER&lt;/reasonType&gt; &lt;reasonCode&gt;A9&lt;/reasonCode&gt; &lt;/serviceReason&gt; &lt;serviceReason&gt; &lt;id&gt;10&lt;/id&gt; &lt;name&gt;Incorrect bill details displayed&lt;/name&gt; &lt;reasonType&gt;BILLER&lt;/reasonType&gt; &lt;reasonCode&gt;A10&lt;/reasonCode&gt; &lt;/serviceReason&gt; &lt;serviceReason&gt; &lt;id&gt;11&lt;/id&gt; &lt;name&gt;Incomplete / No details reflecting&lt;/name&gt; &lt;reasonType&gt;BILLER&lt;/reasonType&gt; &lt;reasonCode&gt;A11&lt;/reasonCode&gt; &lt;/serviceReason&gt; &lt;/ns2:ServiceReasonList&gt;   </pre>
--	----------------------	---

### 2.1.13 Get Complaint Status

This api is used to fetch complaint status which already raised. Both the **SERVICE** and **TRANSACTION** type complaint can be searched using this api.

Action	Get Complaint Status	
API	<domain>/APIService/complaint/searchcomplaintstatus/urn:tenantId:<OU Id>	
Request Header		
	Authorization	Basic Base64.encode(<AgentId>:<Password>); E.g. Basic T1U3MFNMMTJJTkI1MjA5MzY0NTExcGFzc3dvcmQ=
	Content-Type	application/xml   application/json
	Accept	application/xml   application/json
	Request Method	POST
Request		
	<b>*complaintId</b> <b>Description:</b> Unique complaint ID. <b>Data Type:</b> Fixed length 15, Alphanumeric Complaint ID to check the status of a complaint subsequently	
	<b>*complaintType</b> <b>Description:</b> Type of complaint – Transaction or Service based complaint	
	Response	Content-Type
		application/xml   application/json

<b>Header</b>	requestId	b46dd68e-b4fc-4cc2-b37a-1b38ef4c4d6a
<b>Response</b>		<p><b><u>complaintId</u></b></p> <p><b>Description:</b> Data Type: Fixed length 15, Alphanumeric Complaint ID to check the status of a complaint subsequently</p>
		<p><b><u>Assigned</u></b></p> <p><b>Description:</b> Assigned users like Customer OU/Biller OU.</p>
		<p><b><u>responseCode</u></b></p> <p><b>Description:</b> Carries the response code indicating success or failure of the transaction</p>
		<p><b><u>responseReason</u></b></p> <p><b>Description</b> Response reason should be sent per the below response reason</p> <ul style="list-style-type: none"> <li>• SUCCESS</li> <li>• No Transaction found</li> </ul>
	<b>Json Format</b>	<b>XML Format</b>
<b>Request Example</b>	{"complaintId": "UO1543567660682", "complaintType": "SERVICE"}	<ns2:ComplainStatusRequest xmlns:ns2="http://bbps.rsssoftware.co.in/schema"> <complaintId>UO1543567660682</complaintId> <complaintType>SERVICE</complaintType> </ns2:ComplainStatusRequest>
<b>Response Example</b>	{     "complaintId": "UO1543567660682",     "openComplaint": null,     "complaintStatus": "ASSIGNED",     "assigned": "BBPOUQA",     "responseCode": "000",     "responseReason": "SUCCESS",     "remarks": "SERVICE COMPLAINT.. SERVICE COMPLAINT",     "errorMessages": []   }	<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <ns2:ComplaintRaisedResponse xmlns:ns2="http://bbps.rsssoftware.co.in/schema"> <complaintId>UO1543567660682</complaintId> <complaintStatus>ASSIGNED</complaintStatus> <assigned>BBPOUQA</assigned> <responseCode>000</responseCode> <responseReason>SUCCESS</responseReason> <remarks>SERVICE COMPLAINT.. SERVICE COMPLAINT </remarks> </ns2:ComplaintRaisedResponse>

### 2.1.14 Raise Complaint

This api is used to raise complaint. Both type of complaint (SERVICE & TRANSACTION) can be raised by this single api.

<b>Action</b>	Raise Complaint	
<b>API</b>	<domain>/APIService/complaint/raisecomplaintreq/urn:tenantId:<OU Id>	
<b>Request Method</b>	POST	
	Authorization	Basic Base64.encode(<AgentId>:<Password>); E.g. Basic T1U3MFNMMTJJTkI1MjA5MzY0NTEx6cGFzc3dvcmQ=

	Content-Type	application/xml   application/json
	Accept	application/xml   application/json
	Request Method	POST
<b>Request</b>	<b>*<i>txnid</i></b>	<b>Description</b> Transaction ref id (TRANSACTION Only) <b>Data Type:</b> Alphanumeric, Fixed Length – 12
	<b>*<i>complaintType</i></b>	<b>Description</b> Complaint type(TRANSACTION/SERVICE)
	<b>*<i>disposition</i></b>	<b>Description</b> Disposition type (TRANSACTION Only)
	<b>*<i>description</i></b>	<b>Description</b> Description of the complaint (SERVICE + TRANSACTION)
	<b>*<i>billerId</i></b>	<b>Description</b> 14digit Biller id (SERVICE only)
	<b>*<i>servReason</i></b>	<b>Description</b> Service Reason (SERVICE only)
	<b>*<i>participationType</i></b>	<b>Description</b> participation type (SERVICE only). Values are : AGENT/ BILLER/ SYSTEM
<b>Response Header</b>		
	Content-Type	application/xml   application/json
	requestId	b46dd68e-b4fc-4cc2-b37a-1b38ef4c4d6a
<b>Response</b>	<b><i>complaintId</i></b>	<b>Description</b> Complaint id (SERVICE + TRANSACTION)
	<b><i>openComplaint</i></b>	<b>Description</b> Whether this complaint is open (TRANSACTION Only)
	<b><i>complaintStatus</i></b>	<b>Description</b> Complaint status (TRANSACTION Only)
	<b><i>assigned</i></b>	<b>Description</b> Assigned id (SERVICE + TRANSACTION)
	<b><i>responseCode</i></b>	<b>Description</b> Response code (SERVICE + TRANSACTION)
	<b><i>responseReason</i></b>	<b>Description</b>

		Response reason (SERVICE + TRANSACTION)
	<b>Json Format</b>	<b>XML format</b>
<b>Request Example1</b>	{"complaintType":"Transaction","description":"hfjfj","disposition":"Tr ansaction Succesful, account not updated","txnid":"RS350 0059096"}	<ns2:ComplainRequest xmlns:ns2="http://bbps.rsssoftware.co.in/schema">     <complaintType>TRANSACTION</complaintType>     <txnid>OU2100029754</txnid>     <disposition>Transaction Succesful, account not updated</disposition>     <description>Test1234</description> </ns2:ComplainRequest>
<b>Response Example1</b>	{         "complaintId": "AP1512540407809",         "openComplaint": "Y",         "complaintStatus": "ASSIGNED",         "assigned": "AndTest Bank",         "responseCode": "000",         "responseReason": "SUCCESS",         "errorMessages": []     }	<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <ns2:ComplaintRaisedResponse xmlns:ns2="http://bbps.rsssoftware.co.in/schema">     <complaintId>AP1520345797467</complaintId>     <openComplaint>Y</openComplaint>     <complaintStatus>ASSIGNED</complaintStatus>     <assigned>CustAD</assigned>     <responseCode>000</responseCode>     <responseReason>SUCCESS</responseReason> </ns2:ComplaintRaisedResponse>
<b>Request Example2</b>	{"agentId":"SB23MN09INT5212723 42","complaintType":"SERVICE","d escription":"ggg","participationTyp e":"AGENT","servReason":"Agent not willing to print receipt"}	<ns2:ComplainRequest xmlns:ns2="http://bbps.rsssoftware.co.in/schema">     <billerId>NADH00000BLR2L</billerId>     <complaintType>SERVICE</complaintType>     <servReason>Biller available. Unable to transact</servReason>     <description>Test1234</description>     <participationType>BILLER</participationType> </ns2:ComplainRequest>
<b>Response Example2</b>	{         "complaintId": "UC1517988391363",         "openComplaint": null,         "complaintStatus": null,         "assigned": "SBICustomerOU",         "responseCode": "000",         "responseReason": "SUCCESS",         "errorMessages": []     }	<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <ns2:ComplaintRaisedResponse xmlns:ns2="http://bbps.rsssoftware.co.in/schema">     <complaintId>YH1520348040647</complaintId>     <assigned>BBPOUQA</assigned>     <responseCode>000</responseCode>     <responseReason>SUCCESS</responseReason> </ns2:ComplaintRaisedResponse>

### 2.1.15 Agent Login

This is the first api which is used by an agent for login. No authorization header is required to invoke this api. In case the user password has not been reset at least once, this api will provide agent data with **ISNEWUSER** flag as true ,**RESET\_PASSWORD\_FLAG** as **true** to indicate reset password required. The integrating client should make provisions to invoke setPassword API.

Action	Agent Login	
API	<domain>/APIService/agents/urn:tenantId:<OU Id>/login	
Request Header		
	Content-Type	application/xml   application/json
	Accept	application/xml   application/json
	ChannelId	iMobile
	Request Method	POST
Request	loginHash- Basic Base64.encode(<AgentId>)	
Response Header		
	ChannelId	iMobile
	requestId	b46dd68e-b4fc-4cc2-b37a-1b38ef4c4d6a
	Content-Type	application/xml   application/json
Response		<p><b><u>token:</u></b></p> <p><b>Description:</b> OTP will be generated and will be sending to the specific user mobile number.</p>
	Json Format	XML Format
Request Example	{"loginHash":"REsOMEIDMjVJTkI1MTUzODQyMjE6"}	<ns:RegisterLoginRequest xmlns:ns="http://bbps.rssoftware.co.in/schema"> <loginHash>REsOMEIDMjVBVE01MTA4MjUxOTU6</loginHash> </ns:RegisterLoginRequest>
Response Example	{     "token": "OTP Generated Successfully",     "errorMessages": []   }	<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <ns2:RegisterLoginResponse xmlns:ns2="http://bbps.rssoftware.co.in/schema"> <token>OTP Generated Successfully</token> </ns2:RegisterLoginResponse>

### 2.1.16 Set Password

This api is used to set the new password.

Action	Reset Password	
<b>API</b>	<domain>/APIService/agents/urn:tenantId:<OU Id>/setPassword	
<b>Request Header</b>	Content-Type	application/xml   application/json
	Accept	application/xml   application/json
	ChannelId	iMobile
	Request Method	POST
<b>Request</b>	<b>newLoginHash</b> - Hash code of the new password. Basic Base64.encode of agentid with newPassword . E.g: Basic Base64.encode(RB76RB12AGT527795726:password1)	
	<b>otp</b> - unique <b>OTP</b> which was sent to agent's mobile after <b>/login</b> api invoke.	
<b>Response Header</b>		
	requestId	b46dd68e-b4fc-4cc2-b37a-1b38ef4c4d6a
	Content-Type	application/xml   application/json
	ChannelId	iMobile
<b>Response</b>	<b>status</b> – Whether password successfully reset or not. The status value will be ‘true’ or ‘false’	
	<b>Json Format</b>	
<b>Request Example</b>	<pre>{ "newLoginHash":"REs0MEIDMjVJTkI1MTUzODQyMjE6CGFzc3dvcmQ=", "otp":"123456" }</pre>	
	<pre> &lt;ns:PasswordResetRequest xmlns:ns="http://bbps.rsssoftware.co.in/schema"&gt;  &lt;newLoginHash&gt;REs0MEIDMjVBVE01MTA4MjUxOTU6cGFzc3dvcmQ=&lt;/newLoginHash&gt; &lt;otp&gt;987654&lt;/otp&gt; &lt;/ns:PasswordResetRequest&gt;</pre>	
<b>Response Example</b>	<pre>{ "status": true, "errorMessages": [], "token": "REs0MEIDMjVJTkI1MTUzODQyMjE6JDjhJDEwJGZ ZRDNIz1IHZHJNV3NjL09hdk56RnVmc1U3a3JFSGJ2 U0plRHZzU0Yva1NDZFRRckZ6bjky" }</pre>	<pre> &lt;?xml version="1.0" encoding="UTF-8" standalone="yes"?&gt; &lt;ns2:PasswordResetResponse xmlns:ns2="http://bbps.rsssoftware.co.in/schema"&gt; &lt;status&gt;true&lt;/status&gt;  &lt;token&gt;REs0MEIDMjVBVE01MTA4MjUxOTU6JDjhJDEwJ HhSbhkbC54enpDNUZGVG1leUNwak95T0NBLmpvTVpX QlQ4WERjcUJIOFB0c3VMemk5cHQy&lt;/token&gt; &lt;/ns2:PasswordResetResponse&gt;</pre>

### 2.1.17 Forgot Password

Action	Reset Password	
<b>API</b>	<domain>/APIService/agents/urn:tenantId:<OU Id>/ forgotPassword	
<b>Request Header</b>	Content-Type	application/xml   application/json
	Accept	application/xml   application/json
	ChannelId	iMobile
	Request Method	POST
<b>Request</b>	<b>LoginHash</b> - Basic Base64.encode(<AgentId> E.g: Basic Base64.encode(RB76RB12AGT527795726)	
	<b>mobile</b> - Agent Mobile Number	
<b>Response Header</b>		
	requestId	b46dd68e-b4fc-4cc2-b37a-1b38ef4c4d6a
	Content-Type	application/xml   application/json
	ChannelId	iMobile
<b>Response</b>	<b>status</b> – Whether password successfully reset or not. The status value will be ‘true’ or ‘false’ <b>message</b> - OTP generated successfully and send to agent’s mobile number.	
	<b>Json Format</b>	
<b>Request Example</b>	<pre>{   "loginHash": "REs0MEIDMjVCU0M1MjkwMTY1MTA6",   "mobile": "7666666666" }</pre>	
	<b>XML Format</b>	
	<pre>&lt;ns:PasswordResetInitiationRequest   xmlns:ns="http://bbps.rsssoftware.co.in/schema"&gt;   &lt;loginHash&gt;REs0MEIDMjVCU0M1MjkwMTY1MTA6&lt;/loginHash&gt;   &lt;mobile&gt;7666666666&lt;/mobile&gt; &lt;/ns:PasswordResetInitiationRequest&gt;</pre>	
<b>Response Example</b>	<pre>{   "status": true,   "errorMessages": [],   "message": "OTP generated successfully" }</pre>	
	<b>XML Format</b>	
	<pre>&lt;?xml version="1.0" encoding="UTF-8" standalone="yes"?&gt; &lt;ns2:PasswordResetInitiationResponse   xmlns:ns2="http://bbps.rsssoftware.co.in/schema"&gt;   &lt;status&gt;true&lt;/status&gt;   &lt;message&gt;OTP generated successfully&lt;/message&gt; &lt;/ns2:PasswordResetInitiationResponse&gt;</pre>	

### 2.1.18 Transaction Search

Action	Transaction Search	
API	<domainUrl>/APIService/urn:tenantId:{OUID}/allTransactions	
Request Method	POST	
Request		<ul style="list-style-type: none"> <li>• fromDate – user can search the transactions using fromdate</li> <li>• toDate- user can search the transactions using date range fromdate to toDate.</li> <li>• txnRefIdList: user can search the transaction using the more than one transaction refid under one customerID.</li> <li>• *customerId – customerId is required for searching the transactions</li> <li>• txnStatus- user can search the transaction providing the status as Success, Failure, In Progress.</li> <li>• billerCategory- user can search the transaction providing the biller category</li> <li>• billerId- user can search the transaction providing the biller Id</li> <li>• amountFrom - user can search the transaction providing the transaction amount range</li> <li>• amountTo - user can search the transaction providing the transaction amount range</li> <li>• accountNo - user can search the transaction providing the customer accountNo</li> </ul>
Response		<ul style="list-style-type: none"> <li>• display the transaction details</li> </ul>
Request Example	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;ns2:AllTransactionDetailsRequest xmlns:ns2="http://bbps.rssoftware.co.in/schema"&gt;     &lt;fromDate&gt;2019-03-12&lt;/fromDate&gt;     &lt;toDate&gt;2019-03-29&lt;/toDate&gt;     &lt;txnRefIdList&gt;</pre>	

	<pre> &lt;txnRefIds&gt;&lt;/txnRefIds&gt; &lt;/txnRefIdList&gt; &lt;customerId&gt;47601505604&lt;/customerId&gt; &lt;txnStatus&gt;&lt;/txnStatus&gt; &lt;billerCategory&gt;&lt;/billerCategory&gt; &lt;billerId&gt;&lt;/billerId&gt; &lt;amountFrom&gt;&lt;/amountFrom&gt; &lt;amountTo&gt;&lt;/amountTo&gt; &lt;accountNo&gt;&lt;/accountNo&gt; &lt;/ns2&gt;AllTransactionDetailsRequest&gt; </pre>
Response Example	<pre> &lt;?xml version="1.0" encoding="UTF-8" standalone="yes"?&gt; &lt;ns2:AllTransactionDetailsResponse xmlns:ns2="http://bbps.rsssoftware.co.in/schema"&gt; &lt;totCount&gt;2&lt;/totCount&gt; &lt;allTransactionDetailsList&gt; &lt;allTransactions&gt; &lt;txRefId&gt;DK40A0510138&lt;/txRefId&gt; &lt;customerId&gt;47601505604&lt;/customerId&gt; &lt;billerCategory&gt;Broadband Postpaid&lt;/billerCategory&gt; &lt;billerId&gt;ONL100000WBGSM&lt;/billerId&gt; &lt;billerName&gt;Broadband Postpaid&lt;/billerName&gt; &lt;custParams&gt;Account:100&lt;/custParams&gt; &lt;paidAmount&gt;6000&lt;/paidAmount&gt; &lt;pmtDate&gt;2019-03-29T14:52:58+05:30&lt;/pmtDate&gt; &lt;txnStatus&gt;Success&lt;/txnStatus&gt; &lt;accountType&gt;Wallet&lt;/accountType&gt; &lt;accountNo&gt;ICI 0053 icici1111225&lt;/accountNo&gt; &lt;pmtMode&gt;Wallet&lt;/pmtMode&gt; &lt;chnlId&gt;C1&lt;/chnlId&gt; &lt;reconStatus&gt;REFUNDED&lt;/reconStatus&gt; &lt;reconCrtnTs&gt;2019-03-2915:41:53.926&lt;/reconCrtnTs&gt; &lt;txndate&gt;2019-03-29 14:52:58.386&lt;/txndate&gt; &lt;refundStatus&gt;REFUNDED&lt;/refundStatus&gt; &lt;refundDate&gt;2019-03-29 15:41:53.926&lt;/refundDate&gt; &lt;/allTransactions&gt; &lt;allTransactions&gt; &lt;txRefId&gt;DK40A0510138&lt;/txRefId&gt; &lt;customerId&gt;47601505604&lt;/customerId&gt; &lt;billerCategory&gt;Broadband Postpaid&lt;/billerCategory&gt; &lt;billerId&gt;ONL100000WBGSM&lt;/billerId&gt; &lt;billerName&gt;Broadband Postpaid&lt;/billerName&gt; &lt;custParams&gt;Account:100&lt;/custParams&gt; &lt;paidAmount&gt;6000&lt;/paidAmount&gt; &lt;pmtDate&gt;2019-03-29T14:52:58+05:30&lt;/pmtDate&gt; &lt;txnStatus&gt;Success&lt;/txnStatus&gt; &lt;accountType&gt;Wallet&lt;/accountType&gt; &lt;accountNo&gt;ICI 0053 icici1111225&lt;/accountNo&gt; &lt;pmtMode&gt;Wallet&lt;/pmtMode&gt; &lt;chnlId&gt;C1&lt;/chnlId&gt; &lt;reconStatus&gt;RESPONSE_SUCCESS&lt;/reconStatus&gt; &lt;reconCrtnTs&gt;2019-03-2915:03:59.176&lt;/reconCrtnTs&gt; &lt;txndate&gt;2019-03-29 14:52:58.386&lt;/txndate&gt; &lt;/allTransactions&gt; </pre>

	</allTransactionDetailsList> </ns2:AllTransactionDetailsResponse>
--	--

### 2.1.19 Get Duplicate Receipt

Action	Get Duplicate Receipt	
API	<domainUrl>/APIService/duplicatePaymentReceipt/urn:tenantId:{OUID}	
Request Method	POST	
Request		<ul style="list-style-type: none"> <li>• id –12digit transaction RefId</li> </ul>
Response		display the duplicate receipt for successful transaction
Request Example	<?xml version="1.0" encoding="UTF-8"?> <ns2:TransactionRefId xmlns:ns2="http://bbps.rsssoftware.co.in/schema"> <id>DK40A0510124</id> </ns2:TransactionRefId>	
Response Example	<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <ns2:DuplicatePaymentReceipt xmlns:ns2="http://bbps.rsssoftware.co.in/schema"> <txnRefId>DK40A0510124</txnRefId> <customerId>AM26thMar1</customerId> <customerMobileNo>5555555555</customerMobileNo> <billerId>ONLI00000WBGSMS</billerId> <billerName>Broadband Postpaid</billerName> <CustomerParams> <Tag name="Account" value="100"/> <Tag name="Product Id" value="200"/> </CustomerParams> <billNumber>88888888883017072517</billNumber> <paidAmount>6000</paidAmount> <paymentDate>2019-03-27</paymentDate> <accountType>Savings</accountType> <accountNumber>1234567890123</accountNumber> <paymentStatus>Success</paymentStatus> </ns2:DuplicatePaymentReceipt>	