E-Collect Standard API
Implementation Specifications

Document Version: 2.9

EXECUTIVE SUMMARY

E-Collect Service is used by Customers to streamline processing of their Transactions emanating from various sources. With it Bank's Customers can provide a prompt, high quality and on-demand service mechanism to their valued Patrons. The E-Collect solution caters to their requirement of receiving on-line payments (NEFT/RTGS/IMPS transfers) for processing.

This guideline document depicts the **E-Collect Standard API** implementation for *Customer*.

Table of Contents

1 E-Collect Standard API

The **E-Collect Standard Application is** deployed and running on Bank's Server. Customer should build their Service as per the specifications shared with him. He then gets on-boarded by Bank's team by setting up the Configuration as described below.

I. **App Code:** One of the 2 values can be selected

- ECSTDX01 – (XML)
- ECSTDJ01 – (JSON for REST Service)

a) The **App Codes** for Customers opting for *Standard E-Collect API* will be entered using the above Screen.

b) The **App Codes** for existing E-Collect Customers will be uploaded into the ECOL_APPS table using a DB Script.

c) Any subsequent Customers that are NOT opting for the E-Collect Standard API will be uploaded in the same manner. (i.e., using a DB Script)

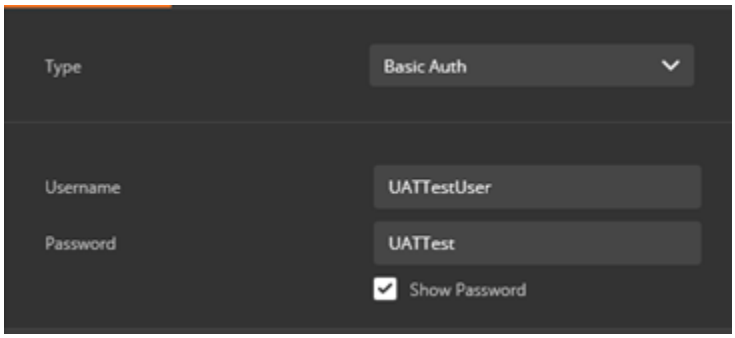
Customer Code: Customer ***must*** exist before setting up the E-Collect Apps.

II. **Validate URL:** In case that **Validation Method** in Customer Setup is specified as the *Web Service*, then *Validation URL* **must** be entered. HTTPS service needs TLS V1.2

III. **Notify URL:** If the value specified in **Customer Alert-On** in the Customer Setup is Always, On Credit, then *Notification URL* **must** be entered. HTTPS service needs TLS V1.2

IV. **HTTP Username:** *Username* as specified for Basic Authentication by the Customer.

HTTP Password: *Password* as specified for Basic Authentication by the Customer.



V. **Panel for key-entry of UDF Settings:** Customer's tags/parameters can be specified using this entry panel in the screen. It allows entry of up to five (5) Key-Value pairs that can be of types: Text, Number Date or Password (encrypted). These are UDF's defined for passing fixed environment *values* to the Service. They are **optional**.

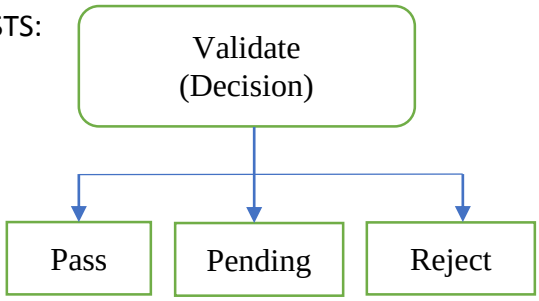
A IP Whitelisting: Please whitelist these IP's for UAT and Production environments.

- UAT IP's:

  125.99.57.154
  115.96.19.123
  54.84.112.66
  54.209.113.91

- Live IP: **123.136.18.22**

B VALIDATE REQUESTS:

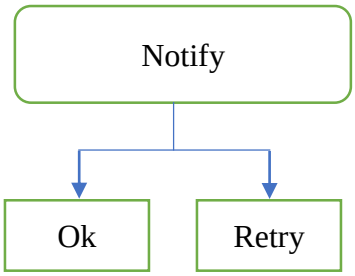


Fresh or Duplicate request can be identified by checking the Transaction Logs.
i.e., in ECOL_TRANSACTIONS Table

- For NEFT Transactions: UTR + RMTR_ACCOUNT_IFSC will have multiple entries in the table.
- For Other Transactions: UTR can be checked for having multiple tuples in the table.

Note: If the response for *Validate* has a credit_account_no tag, then value of Cr Acct# will be taken from it, otherwise **Credit Account Number** specified in the Customer Setup will take effect.

C NOTIFY REQUESTS:



2 SECTIONS TO READ BASED ON: XML / JSON; VALIDATE / NOTIFY

Serial#	Perspective	Section(s)
1	XML	3
2	JSON	4
3	VALIDATE	3.1, 4.1
4	NOTIFY	3.2, 4.2
5	Unit Testing	5
6	Test Scenarios	6
7	Frequently Asked Questions	7
8	Optional Fields & Fields Mapping **(Appendix-A)**	8
9	SCHEMA Reference **(Appendix-B)**	9

3 XML

3.1 VALIDATE

3.1.1 Optional Fields: Refer to the APPENDIX-A at end of this document

3.1.2 Schema: Refer please to the APPENDIX-B at end of this document

3.1.3 Purpose: Service Validates that source of transaction is authentic.

3.1.3.1 The Validate request is for ensuring that money is coming into the Customer's account from an authentic source. i.e., remitter of the money is appropriate and authentic.

Should not have the SOAP Envelope should be built as an XML service

3.1.3 Sample Request:

```
<validate>
  <customer_code>TESTX1</ customer_code >
  <bene_account_no>TESTX176235233</bene_account_no>
  <bene_account_ifsc>YESB0CMSNOC</bene_account_ifsc>
  <bene_full_name>reds</bene_full_name>
  <transfer_type>NEFT</transfer_type>
  <transfer_unique_no>7346tyeghd</transfer_unique_no>
  <transfer_timestamp>2018-03-27 17:29:17</transfer_timestamp>
  <transfer_ccy>INR</transfer_ccy>
  <transfer_amt>43.00</transfer_amt>
  <rmtr_account_no>87654356789</rmtr_account_no>
  <rmtr_account_ifsc>HDFC0123456</rmtr_account_ifsc>
  <rmtr_account_type>ca</rmtr_account_type>
  <rmtr_full_name>refd</rmtr_full_name>
  <rmtr_address>refdx</rmtr_address>
  <attempt_no> 9</attempt_no>
  <rmtr_to_bene_note>pass</rmtr_to_bene_note>
</validate>
```

Response:

```
<validateResponse>
  <decision>pass</decision>
  </validateResponse>
```

Response Codes:

Serial No.	Result	HTTP Status
1	pass	200
2	reject	200
3	pending	200
4	Unauthorized	401
5	Internal Server Error	500
6	Bad Request	400

3.1.4 SAMPLE RESPONSES FOR: PASS / RETURN / FAILURE / RETRY

'pass' response:

```
<validateResponse>
  <decision>pass</decision>
  </validateResponse>
```

[Note: If the *Validate response* has a credit_account_no tag, this will take precedence over the **Credit Account Number** specified in the Customer Setup.]

'reject' response:

```
<validateResponse>
  <decision>reject</decision>
  <reject_reason>invalid request </reject_reason>
</validateResponse>
```

'pending' response:

```
<validateResponse>
  <decision>pending</decision>
</validateResponse>
```

Note: Response codes are case sensitive

3.2 NOTIFY

3.2.1 Optional Fields: Refer to the APPENDIX-A at end of this document

3.2.2 Schema: Refer please to the APPENDIX-B at end of this document

3.2.3 Purpose: Sends Notification to the Customer per the Configuration set in the system.

3.2.3.1 The Notify request is for sending an appropriate Notification to the Customer when the specific event occurs. i.e., Account is Credited, Money Returned to the remitter, etc.

3.2.4 Sample Request (Credit):

```
<notify>
  <customer_code>TESTX1</ customer_code >
  <bene_account_no>TESTX176235233</bene_account_no>
  <bene_account_ifsc>YESB0CMSNOC</bene_account_ifsc>
  <bene_full_name>reds</bene_full_name>
  <transfer_type>NEFT</transfer_type>
  <transfer_unique_no>7346tyeghd</transfer_unique_no>
  <transfer_timestamp>2018-03-27 17:29:17</transfer_timestamp>
  <transfer_ccy>INR</transfer_ccy>
  <transfer_amt>43.00</transfer_amt>
  <rmtr_account_no>87654356789</rmtr_account_no>
  <rmtr_account_ifsc>hdfc0123456</rmtr_account_ifsc>
  <rmtr_account_type>ca</rmtr_account_type>
  <rmtr_full_name>refd</rmtr_full_name>
  <rmtr_address></rmtr_address>
  <rmtr_to_bene_note>pass</rmtr_to_bene_note>
  <attempt_no> 9</attempt_no>
  <status>CREDITED</status>
  <credit_acct_no>765433456789</credit_acct_no>
  <credited_at>2017-07-27 11:48:02</credited_at>
</notify>
```

Response:

```
<notifyResult>
  <result>ok</result>
</notifyResult>
```

3.2.5 Sample Request (Return):

```
<notify>
   <customer_code>TESTX1</customerCode>
   <bene_account_no>TESTX176235233</bene_account_no>
   <bene_account_ifsc>YESB0CMSNOC</bene_account_ifsc>
   <bene_full_name>reds</bene_full_name>
   <transfer_type>NEFT</transfer_type>
   <transfer_unique_no>7346tyeghd</transfer_unique_no>
   <transfer_timestamp>2018-03-27 17:29:17</transfer_timestamp>
   <transfer_ccy>INR</transfer_ccy>
   <transfer_amt>43.00</transfer_amt>
   <rmtr_account_no>87654356789</rmtr_account_no>
   <rmtr_account_ifsc>hdfc0123456</rmtr_account_ifsc>
   <rmtr_account_type>ca</rmtr_account_type>
   <rmtr_full_name>refd</rmtr_full_name>
   <rmtr_address></rmtr_address>
   <rmtr_to_bene_note>pass</rmtr_to_bene_note>
   <attempt_no> 9</attempt_no>
   <status>RETURNED</status>
   <credit_acct_no>765433456789</credit_acct_no>
   <returned_at>2017-07-27 11:48:02</returned_at >
</notify>
```

Response:

```
<notifyResult>
   <result>ok</result>
</notifyResult>
```

3.2.6 Response Codes:

Serial No.	Result	HTTP Status
1	ok (Credited)	200
2	ok (Returned)	200
3	retry	200
4	Unauthorized	401
5	Internal Server Error	500
6	Bad Request	400

3.2.7 SAMPLE RESPONSES FOR: PASS / RETURN / FAILURE / RETRY

'retry' response:

```
<notifyResult>
   <result>retry</result>
</notifyResult>
```

'ok' response:

```
<notifyResult>
   <result>ok</result>
</notifyResult>
```

Note: Response codes are case sensitive

4 JSON

4.1 VALIDATE

4.1.1 Optional Fields: Refer to the APPENDIX-A at end of this document

4.1.2 Schema: Refer please to the APPENDIX-B at end of this document

4.1.3 Purpose: Service Validates that source of transaction is authentic.

4.1.3.1 The Validate request is for ensuring that money is coming the Customer's account from an authentic source. i.e., remitter of the money is appropriate and authentic.

4.1.4 Sample Request:


```
{
 "validate": {
   "customer_code": "TESTX1",
  "bene_account_no": "TESTX176235233",
  "bene_account_ifsc": "YESB0CMSNOC",
  "bene_full_name": "reds",
  "transfer_type": "NEFT",
  "transfer_unique_no": "7346tyeghd",
  "transfer_timestamp": "2018-03-27 17:29:17",
  "transfer_ccy": "INR",
  "transfer_amt": "43.00",
  "rmtr_account_no": "87654356789",
  "rmtr_account_ifsc": "hdfc0123456",
  "rmtr_account_type": "ca",
  "rmtr_full_name": "refd",
  "rmtr_address": "refdx",
  "rmtr_to_bene_note": "pass",
  "attempt_no": 1,
  }
}
```


Response:


```
{
"validateResponse": {
"decision": "pass"
}
}
```


4.1.5 Response Codes:

Serial No.	Result	HTTP Status
1	pass	200
2	reject	200
3	pending	200
4	Unauthorized	401
5	Internal Server Error	500
6	Bad Request	400

4.1.6 SAMPLE RESPONSES FOR: PASS / RETURN / FAILURE / RETRY

'pass' response:


```
{
"validateResponse": {
"decision": "pass"
}
}
```


[Note: If the *Validate response* has a credit_account_no tag, this will take precedence over the **Credit Account Number** specified in the Customer Setup.]

'reject' response:


```
{
"validateResponse": {
"decision": "reject",
"reject_reason": "invalid request "
}
}
```


'pending' response:


```
{
"validateResponse": {
"decision": "pending"
}
}
```


Note: Response codes are case sensitive

4.2 NOTIFY

4.2.1 Optional Fields: Refer to the APPENDIX-A at end of this document

4.2.2 Schema: Refer please to the APPENDIX-B at end of this document

4.2.3 Purpose: Sends Notification to the Customer per the Configuration set in the system.

4.2.3.1 The Notify request is for sending an appropriate Notification to the Customer when the specific event occurs. i.e., Account is Credited, Money Returned to the remitter, etc.

4.2.4 Sample Request (Credit):


```
{
 "notify": {
  "customer_code": "TESTX1",
  "bene_account_no": "TESTX176235233",
  "bene_account_ifsc": "YESB0CMSNOC",
  "bene_full_name": "reds",
  "transfer_type": "NEFT",
  "transfer_unique_no": "7346tyeghd",
  "transfer_timestamp": "2018-03-27 17:29:17",
  "transfer_ccy": "INR",
  "transfer_amt": "43.00",
  "rmtr_account_no": "87654356789",
  "rmtr_account_ifsc": "hdfc0123456",
  "rmtr_account_type": "ca",
  "rmtr_full_name": "refd",
  "rmtr_to_bene_note": "pass",
  "attempt_no": 1,
  "status": "CREDITED",
  "credit_acct_no": "765433456789",
  "credited_at": "2017-07-27 11:48:02",
 }
}
```


Response:


```
{
"notifyResult": {
"result": "ok"
}
}
```

4.2.5 Sample Request (Return):


```
{
 "notify": {
  "customer_code": "TESTX1",
  "bene_account_no": "TESTX176235233",
  "bene_account_ifsc": "YESB0CMSNOC",
  "bene_full_name": "reds",
  "transfer_type": "NEFT",
  "transfer_unique_no": "7346tyeghd",
  "transfer_timestamp": "2018-03-27 17:29:17",
  "transfer_ccy": "INR",
  "transfer_amt": "43.00",
  "rmtr_account_no": "87654356789",
  "rmtr_account_ifsc": "hdfc0123456",
  "rmtr_account_type": "ca",
  "rmtr_full_name": "refd",
  "rmtr_to_bene_note": "pass",
  "attempt_no": 1,
  "status": "RETURNED",
  "credit_acct_no": "765433456789",
  "returned_at": "2017-07-27 11:48:02.009928",
 }
}
```


Response:


```
{
"notifyResult": {
"result": "ok"
}
}
```


4.2.6 Response Codes:

Serial No.	Result	HTTP Status
1	ok (Credited)	200
2	ok (Returned)	200
3	retry	200
4	Unauthorized	401
5	Internal Server Error	500
6	Bad Request	400

4.2.7 SAMPLE RESPONSES FOR: PASS / RETURN / FAILURE / RETRY

'retry' response:


```
{
"notifyResult": {
"result": "retry"
}
}
```

'ok' response:


```
{
"notifyResult": {
"result": "ok"
}
}
```


5 Unit Testing

Serial #	Test Scenario	Expected Result
1	VALIDATE request as per the Schema	SUCCESS
2	VALIDATE request NOT as per the Schema	FAILURE
3	NOTIFY request as per the Schema	SUCCESS
4	NOTIFY request NOT as per the Schema	FAILURE

6 Test-Scenarios (Postman Collections)

Serial#	Test Scenarios	Link for Postman Collection
1	XML Validate	https://github.com/apibanking/e-collect/blob/master/ecollect-xml-validate.postman_collection
2	XML Notify	https://github.com/apibanking/e-collect/blob/master/ecollect-xml-notify.postman_collection
3	JSON Validate	https://github.com/apibanking/e-collect/blob/master/ecollect-json-validate.postman_collection
4	JSON Notify	https://github.com/apibanking/e-collect/blob/master/ecollect-json-notify.postman_collection

7. Frequently Asked Questions:

1. What is the significance of attempt_no field?

✓ The **attempt_no** is used internally and this tag can be ignored by the service. If the customers service is unavailable then E-collect does 3 more attempts in an interval of 30 minutes each for both the Validation and Notification service

2. Why are we receiving the Schema error?

✓ Please compare the response that is being sent by the Service with that specified in the document. There will be a difference between the two. You are required to rectify this at the Service end.

3. Why there is no *Duplicate* case for Notify API call?

✓ For Validate: There is no *Duplicate* case for Validate API. ('pass', 'reject', 'pending'…)

✓ For Notify : Service should either return the **ok** or **retry** for NOTIFICATION API call. In real scenario, seldom there are chances that the same request will be fired twice. To avoid **retry** in such cases, Service should return **'ok'**

4. The duplication Logic can be set basis the Transaction reference number and the IFSC Code

5. The service can be hosted at the clients end in HTTP or HTTPS protocol. If the service is hosted in HTTPS then TLS Version 1.2 needs to be used

8 APPENDIX-A: Optional Fields in *Request & Response* and the Fields Mapping

Following fields are **non-mandatory** in the Validate and Notify request and response and will be dropped if there are no values passed by the remitting bank

Request Fields:

1. bene_full_name
2. rmtr_account_type (If this parameter is blank the Tag is dropped)
3. rmtr_address
4. rmtr_to_bene_note (If this parameter is blank the Tag is dropped)
5. attempt_no
6. credit_acct_no
7. credited_at
8. returned_at
9. Custom: Setting1 through Setting5
   (User Defined Fields; i.e., Fixed values that are passed to the Service)

Response Fields:

10. credit_account_no
11. reject_reason

	TagName	Mandatory <Y/N>	DataType (Length)	Source/Value	Format	Sample Value
1	customer_code	Y	String (15)	$ecol_transactions.CUSTOMER_CODE		NEWCUST
2	bene_account_no	Y	String (64)	$ecol_transactions.BENE_ACCOUNT_NO		ABCDEF23456789
3	bene_account_ifsc	Y	String (20)	$ecol_transactions.BENE_ACCOUNT_IFSC		0987654321
4	bene_full_name	N	String (255)	$ecol_transactions.BENE_FULL_NAME		ASHOKA KING
5	transfer_type	Y	String (04)	$ecol_transactions.TRANSFER_TYPE		NEFT | RTGS | IMPS | UPI
6	transfer_unique_no	Y	String (64)	$ecol_transactions.TRNSFER_UNIQUE_NO		PQRS39856574893
7	transfer_timestamp	Y	Date	$ecol_transactions.CREATED_AT	YYYY-MM-DDT24HH:MM:SS	2017-06-30T07:59:02
8	transfer_ccy	Y	String (05)	$ecol_transactions.TRANSFER_CCY		INR
9	transfer_amt	Y	Number	$ecol_transactions.TRNSFER_AMOUNT	+ve	1400.00
10	rmtr_account_no	Y	String (64)	$ecol_transactions.RMTR_ACCOUNT_NO		8291800010588
11	rmtr_account_ifsc	Y	String (20)	$ecol_transactions.RMTR_ACCOUNT_IFSC		1234567890
12	rmtr_account_type	N	String (10)	$ecol_transactions.RMTR_ACCOUNT_TYPE		SB
13	rmtr_full_name	Y	String (255)	$ecol_transactions.RMTR_FULL_NAME		CHINESE CHECKERS || Can have special characters
14	rmtr_address	N	String (255)	$ecol_transactions.RMTR_ADDRESS		BANARASI HERITAGE || Can have special characters
15	rmtr_to_bene_note	N	String (255)	$ecol_transactions.REMITTER_TO_BENE_NOTE		Xgdgjhf 6576789988 ADDC || Can have special characters
16	attempt_no	N	String (38)	$ecol_transactions.CREDIT_ATTEMPT_NO		1
17	status	Y	String (20)	$ecol_transactions.STATUS		NEW
18	credit_acct_no	N	String (25)	$ecol_transactions.CREDIT_ACCOUNT_NO		1234567890123
19	credited_at	N	Date	$ecol_transactions.CREDITED_AT	YYYY-MM-DD 24HH:MM:SS	2017-06-30 13:15:00
20	returned_at	N	Date	$ecol_transactions.RETURNED_AT	YYYY-MM-DD 24HH:MM:SS.SSS	2018-01-24 15:27:46.009928

7 APPENDIX-B: SCHEMA for REQUEST / RESPONSE

```
<?xml version="1.0" encoding="UTF-8"?><xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified"  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:ibmSchExtn="http://www.ibm.com/schema/extensions" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:tns="http://tempuri.org/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">

<xsd:element  name="validate">
      <xsd:complexType>
       <xsd:all>
         <xsd:element name="customer_code"  type="customerCodeType"/>
         <xsd:element name="bene_account_no" type="beneficiaryAccountNoType"/>
         <xsd:element name="bene_account_ifsc" type="beneficiaryAccountIFSCType"/>
         <xsd:element name="bene_full_name" type="benificiaryfullNameType" maxOccurs="1" minOccurs="0"/>
         <xsd:element name="transfer_type"  type="transferType"/>
         <xsd:element name="transfer_unique_no" type="transferUniqueNoType"/>
         <xsd:element  name="transfer_timestamp" type="dateTimeType"/>
         <xsd:element  name="transfer_ccy" type="transferCcyType"/>
         <xsd:element  name="transfer_amt" type="transferAmountType"/>
         <xsd:element  name="rmtr_account_no" type="remitterAccountNoType"/>
         <xsd:element  name="rmtr_account_ifsc" type="remitterIFSCType"/>
         <xsd:element  name="rmtr_account_type" type="remitterAccountType"/>
         <xsd:element  name="rmtr_full_name" type="remitterNameType" maxOccurs="1" minOccurs="0"/>
         <xsd:element  name="rmtr_address" type="remitterAddressType" maxOccurs="1" minOccurs="0"/>
         <xsd:element  name="rmtr_to_bene_note" type="remitterToBeneficiaryNoteType" maxOccurs="1" minOccurs="0"/>
         <xsd:element  name="attempt_no" type="attempt_noType" maxOccurs="1" minOccurs="1"/>
         <xsd:element name="settings" type="settingsType" maxOccurs="1" minOccurs="0"></xsd:element>
       </xsd:all>
```

```
 </xsd:complexType>
</xsd:element>

 <xsd:element name="validateResponse">
  <xsd:complexType>
   <xsd:all>
    <xsd:element  name="decision" type="decisionType"/>
    <xsd:element  name="credit_account_no" type="xsd:string" maxOccurs="1" minOccurs="0"/>
    <xsd:element  name="reject_reason" type="xsd:string" maxOccurs="1" minOccurs="0"/>
   </xsd:all>
  </xsd:complexType>
</xsd:element>

<xsd:element name="notify">
  <xsd:complexType>
   <xsd:all>
    <xsd:element name="customer_code"  type="customerCodeType"/>
    <xsd:element name="bene_account_no" type="beneficiaryAccountNoType"/>
    <xsd:element name="bene_account_ifsc" type="beneficiaryAccountIFSCType"/>
    <xsd:element name="bene_full_name" type="benificiaryfullNameType" maxOccurs="1" minOccurs="0"/>
    <xsd:element name="transfer_type"  type="transferType"/>
    <xsd:element name="transfer_unique_no" type="transferUniqueNoType"/>
    <xsd:element  name="transfer_timestamp" type="dateTimeType"/>
    <xsd:element  name="transfer_ccy" type="transferCcyType"/>
    <xsd:element  name="transfer_amt" type="transferAmountType"/>
    <xsd:element  name="rmtr_account_no" type="remitterAccountNoType"/>
    <xsd:element  name="rmtr_account_ifsc" type="remitterIFSCType"/>
    <xsd:element  name="rmtr_account_type" type="remitterAccountType"/>
    <xsd:element  name="rmtr_full_name" type="remitterNameType" maxOccurs="1" minOccurs="0"/>
    <xsd:element  name="rmtr_address" type="remitterAddressType" maxOccurs="1" minOccurs="0"/>
    <xsd:element  name="rmtr_to_bene_note" type="remitterToBeneficiaryNoteType" maxOccurs="1" minOccurs="0"/>
```

```
    <xsd:element  name="attempt_no" type="attempt_noType" maxOccurs="1" minOccurs="1"/>
    <xsd:element  name="status" type="xsd:string">
      <xsd:annotation>
     <xsd:documentation>The status of the transaction</xsd:documentation>
     </xsd:annotation></xsd:element>
    <xsd:element  name="credit_acct_no" type="xsd:string" maxOccurs="1" minOccurs="0">
      <xsd:annotation>
     <xsd:documentation>The credit account number which is credited after validation transaction success</xsd:documentation>
     </xsd:annotation></xsd:element>
    <xsd:element  name="credited_at" type="xsd:dateTime" maxOccurs="1" minOccurs="0">
          <xsd:annotation>
     <xsd:documentation>The time when credit happened</xsd:documentation>
     </xsd:annotation></xsd:element>
   <xsd:element  name="returned_at" type="xsd:dateTime" maxOccurs="1" minOccurs="0">
          <xsd:annotation>
     <xsd:documentation>The time when return happened</xsd:documentation>
     </xsd:annotation></xsd:element>
 <xsd:element name="settings" type="settingsType" maxOccurs="1" minOccurs="0"></xsd:element>
   </xsd:all>
  </xsd:complexType>
</xsd:element>

 <xsd:element name="notifyResult">
  <xsd:complexType>
   <xsd:all>
    <xsd:element  name="result" type="resultType"/>
   </xsd:all>
  </xsd:complexType>
</xsd:element>

 <xsd:simpleType name="customerCodeType">
```

<xsd:annotation>
<xsd:documentation>The customer ID that is allocated to the corporate, accessing the service.</xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
<xsd:maxLength value="15" />
<xsd:whiteSpace value="collapse"></xsd:whiteSpace>
<xsd:minLength value="4"></xsd:minLength>
</xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="beneficiaryAccountNoType">
<xsd:annotation>
<xsd:documentation>The account number of the beneficiary.</xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
<xsd:maxLength value="64"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="beneficiaryAccountIFSCType">
<xsd:annotation>
<xsd:documentation>The IFSC code of the beneficiary</xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
<xsd:maxLength value="20"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="benificiaryfullNameType">
<xsd:annotation>
<xsd:documentation>The full name of the beneficiary</xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">

```
        <xsd:maxLength value="255"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="transferUniqueNoType">
<xsd:annotation>
    <xsd:documentation>The unique number of the transactions</xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
        <xsd:maxLength value="64"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="transferType">
<xsd:annotation>
   <xsd:documentation>The transfer type of the transactions</xsd:documentation>
   </xsd:annotation>
     <xsd:restriction base="xsd:string">
     <xsd:enumeration value="NEFT"></xsd:enumeration>
     <xsd:enumeration value="IMPS"></xsd:enumeration>
     <xsd:enumeration value="RTGS"></xsd:enumeration>
     <xsd:enumeration value="UPI"></xsd:enumeration>
     </xsd:restriction>
    </xsd:simpleType>

<xsd:simpleType name="dateTimeType">
<xsd:annotation>
   <xsd:documentation>The  timestamp when the transaction was happened</xsd:documentation>
   </xsd:annotation>
    <xsd:restriction base="xsd:dateTime">
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="transferCcyType">
```

```
<xsd:annotation>
   <xsd:documentation>The currency code for the transaction</xsd:documentation>
   </xsd:annotation>
    <xsd:restriction base="xsd:string">
       <xsd:maxLength value="5"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="transferAmountType">
<xsd:annotation>
   <xsd:documentation>The amount which was transferred in the transaction</xsd:documentation>
   </xsd:annotation>
    <xsd:restriction base="xsd:decimal">
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="remitterAccountNoType">
<xsd:annotation>
      <xsd:documentation> The account number of the remitter</xsd:documentation>
      </xsd:annotation>
    <xsd:restriction base="xsd:string">
       <xsd:maxLength value="64"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="remitterIFSCType">
<xsd:annotation>
      <xsd:documentation>The IFSC code of the remitter.</xsd:documentation>
      </xsd:annotation>
    <xsd:restriction base="xsd:string">
       <xsd:maxLength value="20"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="remitterAccountType">
```

```
<xsd:annotation>
      <xsd:documentation> The account type of the remitter</xsd:documentation>
      </xsd:annotation>
   <xsd:restriction base="xsd:string">
      <xsd:maxLength value="10"/>
   </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="remitterNameType">
<xsd:annotation>
      <xsd:documentation>The full name of the remitter,</xsd:documentation>
      </xsd:annotation>
   <xsd:restriction base="xsd:string">
      <xsd:maxLength value="255"/>
   </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="remitterAddressType">
<xsd:annotation>
      <xsd:documentation>The address of the remitter</xsd:documentation>
      </xsd:annotation>
   <xsd:restriction base="xsd:string">
      <xsd:maxLength value="255"/>
   </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="remitterToBeneficiaryNoteType">
<xsd:annotation>
      <xsd:documentation>The friendly note from the remitter to beneficiary</xsd:documentation>
      </xsd:annotation>
   <xsd:restriction base="xsd:string">
      <xsd:maxLength value="255"/>
   </xsd:restriction>
</xsd:simpleType>
```

```
<xsd:simpleType name="decisionType">
<xsd:annotation>
   <xsd:documentation>The transfer type of the transactions</xsd:documentation>
   </xsd:annotation>
     <xsd:restriction base="xsd:string">
     <xsd:enumeration value="pass"></xsd:enumeration>
     <xsd:enumeration value="reject"></xsd:enumeration>
     <xsd:enumeration value="pending"></xsd:enumeration>
     </xsd:restriction>
    </xsd:simpleType>

    <xsd:simpleType name="resultType">
<xsd:annotation>
   <xsd:documentation>The transfer type of the transactions</xsd:documentation>
   </xsd:annotation>
     <xsd:restriction base="xsd:string">
     <xsd:enumeration value="ok"></xsd:enumeration>
     <xsd:enumeration value="retry"></xsd:enumeration>
     </xsd:restriction>
    </xsd:simpleType>

    <xsd:simpleType name="attempt_noType">
<xsd:annotation>
      <xsd:documentation>The attempt for the transaction</xsd:documentation>
      </xsd:annotation>
    <xsd:restriction base="xsd:integer">
    </xsd:restriction>
</xsd:simpleType>

   <xsd:complexType name="settingsType">
    <xsd:sequence>
```

<xsd:any/>
</xsd:sequence>
</xsd:complexType>
</xsd:schema>