2021

API Specifications for Merchants for Integration with Timepay Merchant Switch - Evok

Innovation in every *byte…*

Product / Project Heading

API Specifications for Merchants for Integration with NPST’s Timepay Merchant Switch (Evok)

Description

This document consists of API specification for integration with merchant switch of NPST.

Nature of document

API Specifications

Date I Version

09/08/2022 I Version 1.6

REVISION HISTORY

It is the responsibility of the document owner to maintain and update this procedures standard. Affected departments and personnel are to be notified of changes to the process immediately. Questions or suggestions for improvement should be submitted to the document owner.

Legal Notices

Network People Services Technologies Ltd makes no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchant ability and fitness for a particular purpose. Network People Services Technologies Ltd shall not be held liable for errors contained herein or direct, indirect, special, incidental, or consequential damages in connection with the furnishing, performance, or use of this material.

Copyright Notice

Document History:

DD/MM/YYY	Version	Author	Reviewer	Remarks
21/08/2021	1.0	Jelish Gurvanshi, Mohd. Azad	Mukesh Chaurasia	Initial Version
15/09/2021	1.1	Jelish Gurvanshi, Mohd. Azad	Mukesh Chaurasia	Added provision for terminal IDs for merchant
23/10/2021	1.2	Mohd. Azad	Mukesh Chaurasia	Support for KYC flag for customers of merchants
22/11/2021	1.3	Mohd. Azad	Mukesh Chaurasia	Added submerchantid in verifyVPA, transfer; added merchant keyword in callback
01/12/2021	1.4	Mohd. Azad	Mukesh Chaurasia	Added RRN in Transaction Status, Transaction Report and Callback
04/01/2022	1.5	Mohd. Azad	Mukesh Chaurasia	Added QR related APIs. Changed endpoints of existing APIs to new version with enhanced engine
09/01/2022	1.6	Mohd. Azad	Mukesh Chaurasia	Sample code for encrption

Table of Contents

1. Introduction

UPI has become one of the most widely accepted payment modes in India. NPST is working extensively towards bringing more merchants into the UPI ecosystem. There are two types of merchant categories - one which does not have the technical setup and has to ride on the infrastructure of another technical solutions provider. The other category is merchants who have their internal technical teams and have the capability of adding UPI payments into their existing payment network.

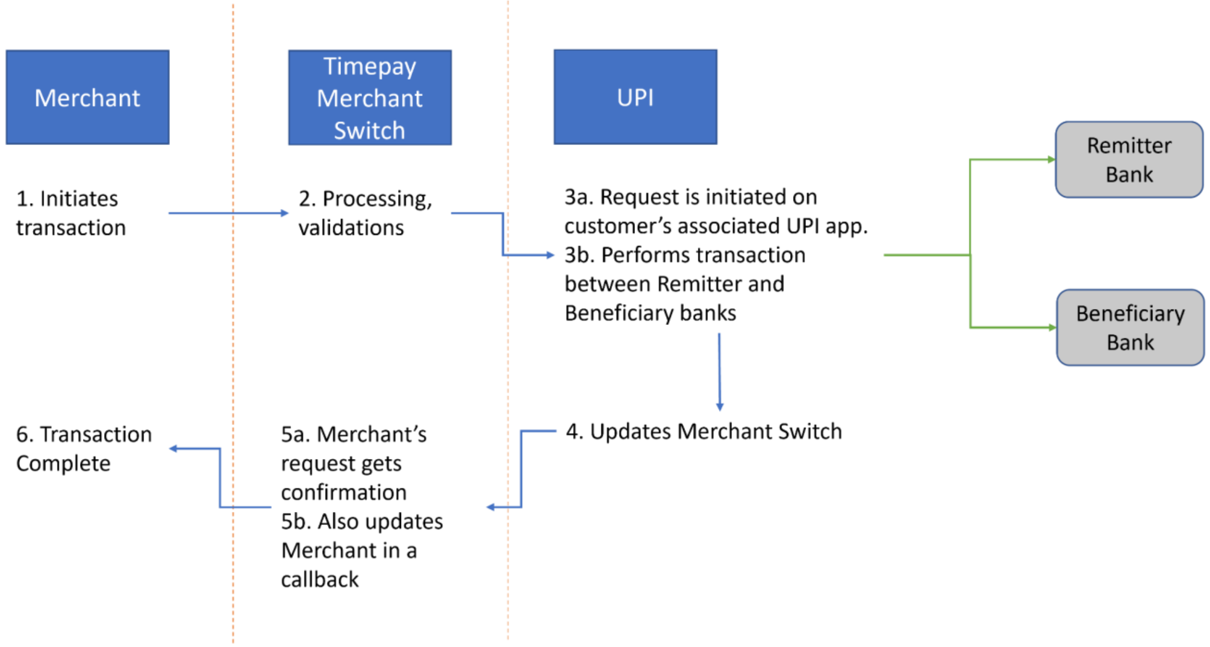
1.1 Purpose

This document focuses on integration of Merchant with UPI payment switch of NPST for enabling real-time transactions. The APIs shared in this document should be consumed by the merchant for enabling UPI in their application.
The merchant switch of NPST is named as “Evok”. The same is mentioned hereafter in the document.

1.2 Assumptions, Constraints and Dependencies

1. Merchants will be using their own infrastructure to connect with TMS
2. All the transactions are executed in the Bank’s environment under the regulator’s governance.
3. Customers will be responsible for the transaction against the UPI Id shared to the switch.
4. Merchants will be responsible for the KYC of their customers.
5. All the transactions will be performed in rea-time. There is no usage a nodal/pool account on behalf of NPST.
6. All guidelines as laid out by the regulator will be final.

2. Process Flow


Merchant
Timepay Merchant Switch
UPI
1. Initiates transaction
2. Processing, validations
3a. Request is initiated on customer's associated UPI app.
3b. Performs transaction between Remitter and Beneficiary banks
Remitter Bank
Beneficiary Bank
4. Updates Merchant Switch
5a. Merchant's request gets confirmation
5b. Also updates Merchant in a callback
6. Transaction Complete

3. Integration API

3.1 Security - Common to all APIs

All the APIs shared in this document will have a common security as mentioned below.

1. All requests are on HTTPS.
2. Only POST is allowed.
3. API header should have below information:
    i. Basic authentication
4. AES Encryption of the entire request/response should be performed between both the parties. Key will be shared by NPST.

3.2 Verify VPA

This API will be used to verify the UPI id entered by the customer/payer making payment or funding the wallet. Response will return whether the UPI is valid or not, customer banking name and MCC code in case of merchant (mcc code can be ignored if not needed).

3.2.1 Request

Endpoint: /evok/cm/v2/verifyVPA

Key	Description	Possible values/Example
source	merchantid/merchantname. This will be assigned by NPST at the time of onboarding the merchant.	Merchant123
channel	For server to server communication, value of this parameter will be "api" and web portal access it will be "portal"	api/sms/portal
extTransactionId	Unique id prefix with merchant assigned 3-4 character. Prefix will be assigned by NPST	XYZ211001adkhff123233adnfddadf
upiId	Customer payer upi/VPA	akash@upi
terminalId	Merchant terminal id in cases of sub merchant	merchant123-001
sid	Sub merchant Id This will be issued by NPST at the time of onboarding the merchant/submerchant	submerchant-001
param_1	Custom info or any other information can be put and same will be return in response	Optional
param_2	Custom info or any other information can be put and same will be return in response	Optional
param_3	Custom info or any other information can be put and same	Optional

	will be return in response	
checksum	Concatenated encrypted string in the order as mentioned above	Checksum key will be shared separately along with logic and algo

3.2.1.1 Sample

```
{
  "source":"Merchant123",
  "channel":"api",
  "extTransactionId":"XYZ211001adkhff123233adnfddadf",
  "upiId":"akash@upi",
  "terminalId":"Merchant123-001",
  "sid":"submerchant-001",
  "checksum":"askjfafasdifasidfhsdfh=="
}
```

3.2.2 Response

Key	Description	Possible values/Example
source	Same as received in request	Merchant123
channel	Same as received in request	api/sms/portal
extTransactionId	Same as received in request	XYZ211001adkhff123233adnfddadf
upiId	Same as received in request	akash@upi
status	SUCCESS/FAILURE	

customerName	Banking name of the customer	“Akash Kumar”
mcc	Merchant code of the payer	“0000”
respCode	0 for success response from UPI HOST	0/UM2/U16/U29
respMessge	Error message description	success/Invalid request
param_1	Custom info or any other information can be put and same will be return in response	Optional
param_2	Custom info or any other information can be put and same will be return in response	Optional
param_3	Custom info or any other information can be put and same will be return in response	Optional
checksum	concatenated encrypted string in order as shown as above	Checksum key will be shared separately along with logic and algo

3.2.2.1 Sample

```
{
 "source":"Merchant123",
 "channel":"api",
  "terminalId":"Merchant123-001",
 "extTransactionId":"XYZ211001adkhff123233adnfddadf",
 "upiId":"akash@upi",
 "data":[
   {
       "customerName":"Akash Kumar",

       "mccCode":"0000",

       “respMessge”:”SUCCESS”,

       “respCode”:”0”

   }

 ],

 "status":"SUCCESS",

 "checksum":"wiowejrioweirweri=="

}
```

3.3 Raise Collect by Payee or Merchant Transfers

This API will be used by merchants to facilitate users to put valid upi id so that a request for money transfer can be raised through UPI channel. Before initiating this request, merchants have the option to verify the UPI Id entered by the customer.

3.3.1 Request

Endpoint: /evok/cm/v2/transfer

Key	Description	Possible values/Example
source	merchantid/merchantname. This will be assigned by NPST at the time of onboarding	Merchant123
channel	For server to server communication, value of this parameter will be “api” and web portal access it will be “portal”	api/sms/portal
extTransactionId	Unique id prefix with merchant assigned 3-4 characters. Prefix will be assigned by NPST	XYZ211001adkhff123233adnfddadf

upiId	customer payer upi/VPA	akash@upi
terminalId	Merchant terminal id in cases of sub merchant	merchant123-001
amount	Amount value upto 2 decimal	2000.00
customerName	Banking name of the customer	"Akash Kumar"
infoKYC	Optional; KYC details masked with visible few character	Optional; "PAN******273G"
statusKYC	KYC status as captured by the merchant. If 'N', then requests will NOT be processed for certain types of merchants.	Y/N
remark	Optional parameter: purpose of transaction or any other txn note. Not more that 50 char	For school fees
requestTime	Optional parameter, put the request initiation time stamp in 24 hour format	2021-01-01 14:12:00
sid	Sub merchant Id This will be issued by NPST at the time of onboarding the merchant/submerchant	submerchant-001
param_1	Custom info or any other information can be put and same will be returned in response	Optional
param_2	Custom info or any other information can be put and same will be returned in response	Optional
param_3	Custom info or any other information can be put and same will be returned in response	Optional

checksum	concatenated encrypted string in order as shown as above	Checksum key will be shared separately along with logic and algo

3.3.1.1 *Sample*

```
{
  "source":"merchant123",
  "channel":"api",
  "terminalId":"merchant123-001",
  "extTransactionId":"XYZ211001adkhff123233adnfddadf",
  "upiId":"akash@upi",
  "customerName":"akash kumar",
  "amount":"3000.00",
  "statusKYC":"Y",
  "infoKYC":"PAN*******273G",
  "Remark":"payment for wallet topup",
  "requestTime":"2021-01-01 10:12:35",
  "sid":"submerchant-001",
  "checksum":"adifaopdfioadfiasdifsf=="
}
```

3.3.2 Response

Key	**Description**	**Possible values/Example**
source	Same as received in request	Merchant123
channel	Same as received in request	api/sms/portal
extTransactionId	Same as received in request	XYZ211001adkhff123233adnfddadf

upiId	Same as received in request	akash@upi
terminalId	Same as received in request	merchant123-001
amount	Amount value upto 2 decimal	2000.00
customerName	Banking name of the customer	“Akash Kumar”
respCode	Response of transaction submission for collect	0 for success in case of successful submission
respMessage	Error code description or message	
upiTxnId	Upi transaction id, please associate with your order id / extTransactionId	
txnTime	Transaction submission time	
status	Status of API SUCCESS/FAILURE/ PENDING - in case of response awaited	SUCCESS/FAILURE/ PENDING
remark	Same as received in request	
responseTime	Optional parameter, put the response initiation time stamp	2021-01-01 14:12:00
param_1	Custom info or any other information can be put and same will be returned in response	Optional
param_2	Custom info or any other information can be put and same will be returned in response	Optional
param_3	Custom info or any other information can be put and same will be returned in response	Optional
checksum	concatenated encrypted string in	Checksum key will be shared separately along with logic and

	order as shown as above	algo

3.3.2.1 Sample

```
{
 "source": "merchant123",
 "channel": "api",
 "terminalId":"merchant123-001",
 "extTransactionId": "XYZ211001adkhff123233adnfddadf",
 "upiId":"akash@upi",
 "amount":"3000.00",
 "customerName":"akash kumar",
 "data":[
 {"upiTxnId":"COSB122321323123ADNDASFDSF",
  "respCode":"0",
  "respMessage":"SUCCESS",
  "txnTime":"2021-02-01 20:20:18"
 }],
 "status":"SUCCESS",
 "responseTime":"",
 "checksum":"wioejriq23223njknfafsdfsdjabfb=="
}
```

3.4 Transaction Status

Transaction status of request money submission can be pulled by this API.

3.4.1 Request

Endpoint: /evok/cm/v2/status

Key	Description	Possible values/Example

source	merchantid/merchantname. This will be assigned by NPST at the time of onboarding	Merchant123
channel	For server to server communication, value of this parameter will be “api” and web portal access it will be “portal”	api/sms/portal
extTransactionId	Unique id prefix with merchant assigned 3-4 characters. Prefix will be assigned by NPST	XYZ211001adkhff123233adnfddadf
terminalId	Merchant terminal id in cases of sub merchant	merchant123-001
checksum	concatenated encrypted string in order as shown as above	Checksum key will be shared separately along with logic and algo

3.4.1.1 Sample

```
{

"source": "merchant123",

"channel": "api",

"terminalId":"merchant123-001",

"extTransactionId": "XYZ211001adkhff123233adnfddadf",
  "checksum":""
}
```

3.4.2 Response

Key	Description	Possible values/Example
source	Same as received in request	Merchant123
channel	Same as received in request	api/sms/portal
extTransactionId	Same as received in request	XYZ211001adkhff123233adnfddadf
upiId	customer payer upi/VPA	akash@upi
terminalId	Merchant terminal id in cases of sub merchant	merchant123-001
amount	Amount value upto 2 decimal	2000.00
customerName	Banking name of the customer	"Akash Kumar"
respCode	Response of transaction submission for collect	0 for success in case of successful submission
respMessage	Error code description or message	"Insufficient balance"
upiTxnId	Upi transaction id, please associate with your order id / extTransactionId	COSB12NJEEWQJRBEWRE EERBWER
status	Status of API SUCCESS/FAILURE PENDING - in case of response awaited	
custRefNo	Transaction RRN	134312519111
checksum	concatenated encrypted string in order as shown as above	Checksum key will be shared separately along with logic and algo

3.4.2.1 Sample

```
{
 "source":"merchant123",
 "channel":"api",
 "terminalId":"merchant123-123",
 "extTransactionId":"XYZ211001adkhff123233adnfddadf",
 "data":[
   {
     "upiTxnId":"COSB123123123KSKJFSFNSDNFJFD",
     "respCode":"0",
     "respMessage":"error message or success",
     "txnTime":"20:20:19",
     "amount":"2000.00",
     "upiId":"akash@upi",
     "custRefNo":"134312519812"
   }
 ],
 "status":"SUCCESS",
 "checksum":"idfaiofasdiofsdfusd=="
}
```

3.5 Transactions Report

This API is used to pull the transactions status in bulk based on two timestamps. Response time of this API may vary depending upon the volume of data. It is advised to narrow down the search using date/timestamp.

3.5.1 Request

Endpoint: /evok/cm/v2/report

Key	Description	Possible values/Example

source	merchantid/merchantname. This will be assigned by NPST at the time of onboarding	Merchant123
channel	For server to server communication, value of this parameter will be “api” and web portal access it will be “portal”	api/sms/portal
terminalId	Merchant terminal id in cases of sub merchant	merchant123-001
startDate	Start time stamp There will be validation over date range to fetch data upto certain days.	2021-01-01 00:00:00
endDate	End time stamp There will be validation over date range to fetch data upto certain days.	2021-01-01 19:20:00
pageSize	No of page Size	10
pageNo	Page Number; first page is mentioned as zero and incremented accordingly	0
checksum	concatenated encrypted string in order as shown as above	Checksum key will be shared separately along with logic and algo

3.5.1.1 Sample

```
{

  "source":"merchant123",

  "channel":"api",

  "terminalId":"merchant123-001",

  "startDate":"2021-01-01 00:00:00",

  "endDate":"2021-01-02 20:00:00",

      "pageSize":"10",
      "pageNo":"0",
   "checksum":"djfaisdjfiadsjfasdfjsdf=="
}
```

3.5.2 Response

Key	Description	Possible values/Example
source	Same as received in request	Merchant123
channel	Same as received in request	api/sms/portal
upiId	customer payer upi/VPA	akash@upi
terminalId	Merchant terminal id in cases of sub merchant	merchant123-001
custRefNo	Transaction RRN	134312519111
amount	Amount value upto 2 decimal	2000.00
customerName	Banking name of the customer	"Akash Kumar"
respCode	Response of transaction submission for collect	0 for success in case of successful submission
respMessage	Error code description or message	"Insufficient balance"
upiTxnId	Upi transaction id, please associate with your order id / extTransactionId	COSB12NJEEWQJRBEWRE EERBWER
status	Status of API SUCCESS/FAILURE PENDING - in case of response awaited	
checksum	concatenated encrypted string in order as shown as above	Checksum key will be shared separately along with logic and algo

3.5.2.1 Sample

```
{
 "source":"merchant123",
 "channel":"api",
 "data":[
   {
      "upiTxnId":"COSB123123123KSKJFSFNSDNFJFD",
      "extTransactionId":"XYZ211001adkhff123233adnfddadf",
      "respCode":"0",
      "respMessage":”SUCCESS",
      "txnTime":"2021-01-01 20:20:01",
      "requestTime":"2021-01-01 18:20:01",
      "amount":"2000.00",
      "upiId":"akash@upi",
      "customerName":"Akash Kumar",
      "terminalId":"merchant123-001",
           "custRefNo":"134312519111"
   }
 ],
 "status":"SUCCESS",
 "checksum":"iueqiwruqwieurwerwery=="
}
```

Note: “data” will have an array with more than one record based upon the search result.

3.6 Transaction Callback

If the merchant shares a URL, then the Evok will hit the merchant’s URL with the details of the transaction. This will be called only for “Raise Collect” request.

3.6.1 Request

Endpoint: To be shared by merchants. REST API should be supported with JSON format.

Key	Description	Possible values/Example
merchant	Merchant Id issued by TIMEPAY	merchant123
source	Same as received in request	TIMEPAY
channel	Same as received in request	api/sms/portal
extTransactionId	Same as received in request	XYZ211001adkhff123233adnfddadf
upiId	customer payer upi/VPA	akash@upi
terminalId	Merchant terminal id in cases of sub merchant	merchant123-001
amount	Amount value upto 2 decimal	2000.00
customerName	Banking name of the customer	"Akash Kumar"
respCode	Response of transaction submission for collect	0 for success in case of successful submission
respMessage	Error code description or message	"Insufficient balance"
upiTxnId	Upi transaction id, please associate with your order id / extTransactionId	COSB12NJEEWQJRBEWRE EERBWER
status	Status of API SUCCESS/FAILURE PENDING - in case of response awaited	
custRefNo	Transaction RRN	134312519111
checksum	concatenated encrypted string in order as shown as above	Checksum key will be shared separately along with logic and algo

3.6.1.1 Sample

```
{ merchant: [ 'merchant123' ],
 message: '{
 "source":"TIMEPAY",
 "channel":"api",
 "extTransactionId":"XYZ211001adkhff123233adnfddadf",
 "data":[
   {
     "upiTxnId":"COSB123123123KSKJFSFNSDNFJFD",
     "respCode":"0",
     "respMessage":”SUCCESS",
     "txnTime":"2021-01-01 20:20:01",
     "requestTime":"2021-01-01 18:20:01",
     "amount":"2000.00",
     "upiId":"akash@upi",
     "customerName":"Akash Kumar",
     "terminalId":"merchant123-001",
           "custRefNo":"134312519111"
   }
 ],
 "status":"SUCCESS",
 "checksum":"iueqiwruqwieurwerwery=="
}'
}
```

3.6.2 Response

The Evok will not store the response from the merchant.

3.7 Generate QR

This API is designed for generation of dynamic/static QR if a merchant is already onboarded in the system and wants to generate real time QR at Web page or POS devices etc.

3.7.1 Request

Endpoint: /evok/qr/v1/dqr

Key	Description	Possible values/Example
source	Same as received in request	TIMEPAY
channel	Same as received in request	api/sms/portal
extTransactionId	Same as received in request to identify the transaction later at merchant side	XYZ211001adkhff123233adnfddadf
sid	Sub merchant id of merchant	merchant123
terminalId	Merchant terminal id in cases of sub merchant	merchant123-001
amount	Amount value upto 2 decimal	2000.00
type	Type of QR D dynamic or S static	D/S
remark	Transaction note to be shown in upi	
requestTime	Time of request	
minAmount	This field can be left blank or min amount for QR can be provided.	
receipt	URL of invoice can be provided if available	
param1	Future use	
param2	Future use	
param2	Future use	
Checksum	concatenated encrypted string in order as shown as above	

Sample Request:

{

```
  "minAmount":"1.00",
  "amount":"5.00",
  "extTransactionId":"NPST040120220010",
  "channel":"api",
  "remark":"QR SIT testing",
  "source":"NPSTPAY001",
  "terminalId":"NPSTPAY001-001",
  "type":"D",
  "param3":"param3",
  "Param2":"param2",
  "param1":"param1",
  "sid":"NPSTPAY001-001",
  "requestTime":"2022-01-30 16:32:36",
  "reciept":"http:\/\/google.com",
   "checksum":""
}
```

3.7.2 Response

Key	Description	Possible values/Example
source	Same as received in request	TIMEPAY
channel	Same as received in request	api/sms/portal
extTransactionId	Same as received in request to identify the transaction later at merchant side	XYZ211001adkhff123233adnfddadf
sid	Sub merchant id of merchant	merchant123
terminalId	Merchant terminal id in cases of sub merchant	merchant123-001
amount	Amount value upto 2 decimal	2000.00
type	Type of QR D dynamic or S static	D/S
remark	Transaction note to be shown in upi	
requestTime	Time of request	

minAmount	This field can be left blank or min amount for QR can be provided.	
receipt	URL of invoice can be provided if available	
param1	Future use	
param2	Future use	
param2	Future use	
qrString	Merchant can render the QR based on this QR string for payments	
Status	success/failure of QR generation	
Checksum	concatenated encrypted string in order as shown as above	

Sample Response:

```
 {
   "source":"NPSTPAY001",
   "sid":"NPSTPAY001-001",
   "terminalId":"NPSTPAY001-001",
   "channel":"api",
   "amount":"5.00",
   "minAmount":"1.00",
   "remark":"QR SIT testing",
   "extTransactionId":"NPST040120220010",
   "reciept":"http://google.com",
   "type":"D",
"qrString":"upi://pay?ver=01&mode=15&am=5.00&mam=1.00&cu=INR&pa=npstpay@timeco
smos&pn=NPST PAY&mc=6012&tr=NPST040120220010&tn=QR SIT
testing&mid=NPSTPAY001&msid=NPSTPAY001-001&mtid=NPSTPAY001-
001&category=02&url=http://google.com",
   "status":"SUCCESS",
   "param1":"param1",
   "param2":"param2",
   "param3":"param3",
   "errorMsg":"",
   "checksum":"wewewqew"
}
```

3.8 QR Transaction Status by RRN

Merchants can request a status check based upon the UPI RRN generated from the UPI transaction. Although the system sends a call back to the merchant after the transaction with status.

3.8.1 Request

Endpoint: /evok/qr/v1/qrStatusRRN

Key	Description	Possible values/Example
source	merchantid/merchantname. This will be assigned by NPST at the time of onboarding	Merchant123
channel	For server to server communication, value of this parameter will be “api” and web portal access it will be “portal”	api/sms/portal
terminalId	Merchant terminal id in cases of sub merchant	merchant123-001
extTransactionId	RRN of the UPI transaction generated in upi system	XYZ211001adkhff123233adnfddadf
checksum	concatenated encrypted string in order as shown as above	Checksum key will be shared separately along with logic and algo

Sample Request:

```
{
  "extTransactionId":"merchant123",
  "channel":"api",
  "checksum":"1231",
  "source":"merchant123",
  "terminalId":"merchant123-001"
}
```

3.8.2 Response

Key	Description	Possible values/Example
source	merchantid/merchantname.	Merchant123

	This will be assigned by NPST at the time of onboarding	
channel	For server to server communication, value of this parameter will be “api” and web portal access it will be “portal”	api/sms/portal
extTransactionId	RRN of the UPI transaction generated in upi system	XYZ211001adkhff123233adnfddadf
terminalId	Merchant terminal id in cases of sub merchant	merchant123-001
status	Status of API success/failure	
respMessge	Transaction status	
respCode	Transaction error code	
upiTxnId	UPI Transaction id	
txnTime	Transaction Time	
amount	Transaction Amount	
upiId	Customer VPA	
extTransactionId	Reference id of upi transaction/extTransaction id	
custRefNo	UPI RRN no.	
remark	Upi transaction note	
customerName	Name of customer	
checksum	concatenated encrypted string in order as shown as above	Checksum key will be shared separately along with logic and algo

Sample Response:

```
  {
  "source":"001",
  "channel":"api",
  "terminalId":"001-001",
  "extTransactionId":"200616380004",

"checksum":"4785e397efa4946cedb168313418de707fb2b3433250862024f47ca4effea228",
  "status":"SUCCESS",
  "data":[
   {
       "respMessge":"SUCCESS",
        "respCode":"0",
         "customerName":"abc",
       "upiTxnId":"COB66C739A28D3A43CBBCAEFE80544F8A17",
       "txnTime":"2022-01-06T16:38:47.847 05:30",
       "amount":"50.00",
       "upiId":"7387551812@cosb",
       "extTransactionId":"NPST06012022004",
       "custRefNo":"200616380004",
       "remark":"upiPayment"
     }

  ]
}
```

3.9 QR Transaction Status by ExtTransactionId

3.9.1 Request

Endpoint: /evok/qr/v1/qrStatus

Key	Description	Possible values/Example
source	merchantid/merchantname. This will be assigned by NPST at the time of onboarding	Merchant123
channel	For server to server communication, value of this parameter will be “api” and web portal access it will be “portal”	api/sms/portal
terminalId	Merchant terminal id in cases of sub merchant	merchant123-001

extTransactionId	Unique id prefix with merchant assigned 3-4 characters. Prefix will be assigned by NPST	XYZ211001adkhff123233adnfddadf
checksum	concatenated encrypted string in order as shown as above	Checksum key will be shared separately along with logic and algo

Sample Request:

```
{
  "extTransactionId":"merchant123",
  "channel":"api",
  "checksum":"1231",
  "source":"merchant123",
  "terminalId":"merchant123-001"
}
```

3.9.2 Response

Key	Description	Possible values/Example
source	merchantid/merchantname. This will be assigned by NPST at the time of onboarding	Merchant123
channel	For server to server communication, value of this parameter will be “api” and web portal access it will be “portal”	api/sms/portal
extTransactionId	RRN of the UPI transaction generated in upi system	XYZ211001adkhff123233adnfddadf
terminalId	Merchant terminal id in cases of sub merchant	merchant123-001
status	Status of API success/failure	

respMessge	Transaction status	
respCode	Transaction error code	
upiTxnId	UPI Transaction id	
txnTime	Transaction Time	
amount	Transaction Amount	
upiId	Customer VPA	
extTransactionId	Reference id of upi transaction/extTransaction id	
custRefNo	UPI RRN no.	
remark	Upi transaction note	
customerName	Name of customer	
checksum	concatenated encrypted string in order as shown as above	Checksum key will be shared separately along with logic and algo

Sample Response:

```
  {
  "source":"001",
  "channel":"api",
  "terminalId":"001-001",
  "extTransactionId":"200616380004",

"checksum":"4785e397efa4946cedb168313418de707fb2b3433250862024f47ca4effea228",
  "status":"SUCCESS",
  "data":[
   {
       "respMessge":"SUCCESS",
        "respCode":"0",
         "customerName":"abc",
       "upiTxnId":"COB66C739A28D3A43CBBCAEFE80544F8A17",
       "txnTime":"2022-01-06T16:38:47.847 05:30",
       "amount":"50.00",
       "upiId":"7387551812@cosb",
       "extTransactionId":"NPST06012022004",
       "custRefNo":"200616380004",
       "remark":"upiPayment"
```

```
    }

  ]
}
```

3.10 QR Report

3.10.1 Request

Endpoint: /evok/qr/v1/qrReport

Key	Description	Possible values/Example
source	merchantid/merchantname. This will be assigned by NPST at the time of onboarding	Merchant123
channel	For server to server communication, value of this parameter will be “api” and web portal access it will be “portal”	api/sms/portal
terminalId	Merchant terminal id in cases of sub merchant	merchant123-001
startDate	Start time stamp There will be validation over date range to fetch data upto certain days.	2021-01-01 00:00:00
endDate	End time stamp There will be validation over date range to fetch data upto certain days.	2021-01-01 19:20:00
pageSize	In case report record with pagination otherwise remove this from request and no needed while generating checksum as well	100 record

pageNo	In case report record with pagination otherwise remove this from request and no needed while generating checksum as well	Pagination no 0-N
checksum	concatenated encrypted string in order as shown as above	Checksum key will be shared separately along with logic and algo

Sample Request:

```
{
   "source":"merchant123",
   "channel":"api",
   "terminalId":"merchant123-001",
   "startDate":"2021-01-01 00:00:00",
   "endDate":"2021-01-02 20:00:00",
   “pageSize”:”100”,
   “pageNo”:”2”
   "checksum":"djfaisdjfiadsjfasdfjsdf=="
}
```

3.10.2 Response

Key	Description	Possible values/Example
source	merchantid/merchantname. This will be assigned by NPST at the time of onboarding	Merchant123
channel	For server to server communication, value of this parameter will be “api” and web portal access it will be “portal”	api/sms/portal
extTransactionId	RRN of the UPI transaction generated in upi system	XYZ211001adkhff123233adnfddadf
terminalId	Merchant terminal id in cases of sub merchant	merchant123-001

status	Status of API success/failure	
respMessge	Transaction status	
respCode	Transaction error code	
upiTxnId	UPI Transaction id	
txnTime	Transaction Time	
amount	Transaction Amount	
upiId	Customer VPA	
extTransactionId	Reference id of upi transaction/extTransaction id	
custRefNo	UPI RRN no.	
remark	Upi transaction note	
customerName	Name of customer	
checksum	concatenated encrypted string in order as shown as above	Checksum key will be shared separately along with logic and algo

Sample Response:

```
 {
   "source":"NPSTPAY001",
   "channel":"api",
   "terminalId":"NPSTPAY001-001",

"checksum":"4785e397efa4946cedb168313418de707fb2b3433250862024f47ca4effea228",
   "status":"SUCCESS",
    "data":[
     {
        "respMessge":"SUCCESS",
        "upiTxnId":"COB66C739A28D3A43CBBCAEFE80544F8A17",
        "txnTime":"2022-01-06T16:38:47.847 05:30",
        "amount":"50.00",
        "upiId":"7387551812@cosb",
        "extTransactionId":"NPST06012022004",
        "custRefNo":"200616380004",
        "remark":"upiPayment"
     }, {
        "respMessge":"SUCCESS",
        "upiTxnId":"COB66C739A28D3A43CBBCAEFE80544F8237",
        "txnTime":"2022-01-07T16:38:47.847 05:30",
        "amount":"500.00",
        "upiId":"7387551812@cosb",
        "extTransactionId":"NPST0601202202",
        "custRefNo":"200616380204",
        "remark":"upiPayment"
     }

   ]
}
```

3.11 QR Callback

3.11.1 Request

Endpoint: To be shared by merchant/receiver.

Key	Description	Possible values/Example
extTransactionId	Unique reference id /ext transaction of the transaction to identify the transaction by merchant and switch	XYZ211001adkhff123233adnfddadf
status	Transaction status	
errorCode	Error code in case of failure	
txnId	UPI Transaction id	
rrn	UPI RRN no.	
merchantVpa	merchant/payee VPA	
amount	Transaction Amount	
customerVpa	Customer VPA	
responseTime	Transaction end Time	
remark	Upi transaction note	

customerName	Name of customer	
checksum	concatenated encrypted string in order as shown as above	Checksum key will be shared separately along with logic and algo

Sample Request:

```
{
  "extTransactionId":"NPST06012022004",
    "status":"SUCCESS",
  "errorCode":"",
  "customerVpa":"7387551812@cosb",
  "merchantVpa":"npstpay@cosb",
  "rrn":"200712430010",
  "txnId":"COB1C9409A2298F4DB3A10598DEC6C6A52A",
  "amount":"50.0",
  "responseTime":"Fri Jan 07 12:44:43 IST 2022",
  "customerName":"",
  "remarks":"upiPayment",
  "checksum":""
}
```

Sample Response:

The Timepay Merchant switch will not store the response from the merchant.

3.12 Checksum Generation and Sample Code

Checksum generation is simple process, follow below steps to generate checksum and refer java sample code.

1. Check and refere Request Parameter table of API in above section.
2. Concatenate parameters value into a string in the order it is mentioned in the request table.
3. write the checksum code in respective language (below sample java code).
4. Use the checksum key shared with your to generate the checksum.
5. Generated checksum shall be added into the JSON request of API.

```
public static String generateChecksumMerchant(String concatenatedString, String checksumkey) throws IOException {
                        String inputString = concatenatedString + checksumkey;
                        StringBuffer sb = null;
                        MessageDigest md;
                        try {
                                    md = MessageDigest.getInstance("SHA-256");
                                    md.update(inputString.getBytes());
                                    byte byteData[] = md.digest();
                                    sb = new StringBuffer();
                                    for (int i = 0; i < byteData.length; i++) {
                                                sb.append(Integer.toString((byteData[i] & 0xff) + 0x100,16).substring(1));
```

```
                                                    }
                                        } catch (NoSuchAlgorithmException e) {

                                        }
                                        return sb.toString();
}
```

3.13 Encryption and Sample Code

After preparing JSON for an API request including checksum, Developer shall generate encypted text of json.

1. **Generate the encrypted Text from JSON prepared for API request.**
2. **Use encryption key provided.**
3. **Follow the logic mentioned below (JAVA sample) to generate encryption.**
4. **Send your encrypted text request by setting header Content-Type:text/plain .**

```
public static String encrypt(String strToEncrypt, String secret) {
                try {
                     if(secretKey==null){
                       setKey(secret);
                      }
                          Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
                          cipher.init(Cipher.ENCRYPT_MODE, secretKey);
                          return Base64.getEncoder().encodeToString(cipher.doFinal(strToEncrypt.getBytes("UTF-8")));
                          } catch (Exception e) {

                          }
                          return null;
      }


       public static void setKey(String myKey) {
                          MessageDigest sha = null;
                          try {
                                       key = myKey.getBytes("UTF-8");
                                       sha = MessageDigest.getInstance("SHA-256");
                                       key = sha.digest(key);
                                       key = Arrays.copyOf(key, 16);
                                       secretKey = new SecretKeySpec(key, "AES");
                          } catch (NoSuchAlgorithmException e) {

                          } catch (UnsupportedEncodingException e) {

                          }
             }


      public static String decryptResponse(String responseString, String encryptKey) {
             try {
                          Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5PADDING");
                          cipher.init(Cipher.DECRYPT_MODE, setMerchantKey(encryptKey));
                          return new String(cipher.doFinal(Base64.getDecoder().decode(responseString)), "UTF-8");
                          } catch (Exception e) {
                                       e.printStackTrace();
                          }
                          return null;
             }
```