Atom Insta Pay

Contents

1.Document Information

Document Attributes	Information
ID	Atom Insta Pay
Owner	Product Management & Audit
Author	Nital Kapadia
CR Number	Product Note

Revision Chart

This chart contains a history of this document's revisions.

Version	Primary Author	Description of Version	Date Completed	Reviewed By
1.0	Nital Kapadia	Product Note for Atom Insta Pay	25/03/2021	
1.1	Nital Kapadia	Addition of Surcharge & UDF Parameter details in API Request/ Response	06/12/2021	

Introduction

"Atom Flash check out" provides best payment experience to end customer and results into a better and faster payment experience. Atom Insta Pay solution will ensure a higher possibility of success in completing the ATOM PAYMENT gateway transaction.

It is optimized with device screen size and helps customer to pay easily and securely over web or on the mobile. It supports payment options such as Credit card, Debit card, Net banking and UPI. It also provides retry option to the customer which helps to reduce customer drop off in case of transaction failure. Customer can retry the transaction by selecting alternate payment option.

Key Features

- Easy to integrate with atom Insta pay
- It provides optimized UI
- PCI DSS complaint
- User friendly customer UI
- Provides retry option, reducing customer drop

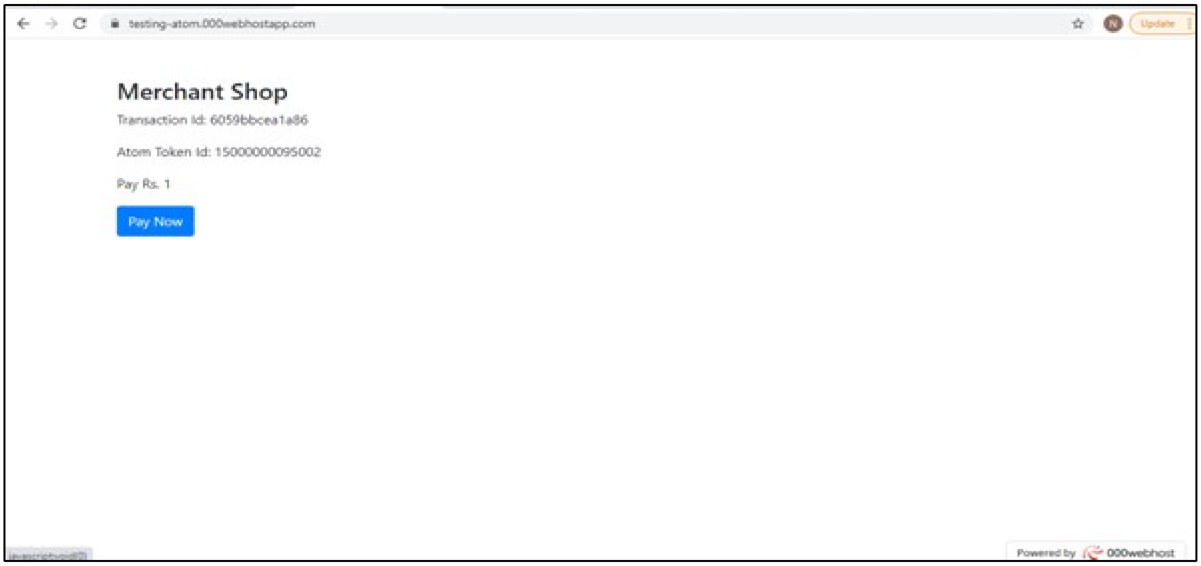
Payment Process

- Once the end customer arrives at the merchant's checkout page, the merchant needs to call the Atom's Rest API
- Merchant needs to initiate the Rest API request for authentication
- Rest API request needs to be encrypted using the AES-256-CBC algorithm
- Token Id will be generated by Atom in response to the Rest API request received from the merchant
- This generated token Id will be sent as the Rest API response to the merchant
- The response will also be encrypted using the AES-256-CBC algorithm
- The merchant needs to decrypt the response sent by Atom and acquire the token Id sent in the response
- This token Id is to be used for calling the Atom's Java script function which will then open the Atom's Insta Pay Payment Popup
- End customer can edit the mobile number and email Id which was sent by the merchant in the Rest API request and continue with the payment
- Payment can be done by the customer using UPI, Net Banking, Credit card or Debit card payment option
- Once the payment is done, Atom will send the response to the merchant showing success, failed or timeout transaction

- In case if the token Id is not received by the merchant or the response is not received in the specified time period then the error message will be shown to the merchant i.e. "Endpoint request timed out"}
- Such errors are to be handled explicitly by the merchant and Rest API request is to be reinitiated

2.Transaction on Atom Insta Pay Page

- Once the end customer arrives at the merchant's checkout page, and can click on the Pay Now button as shown below



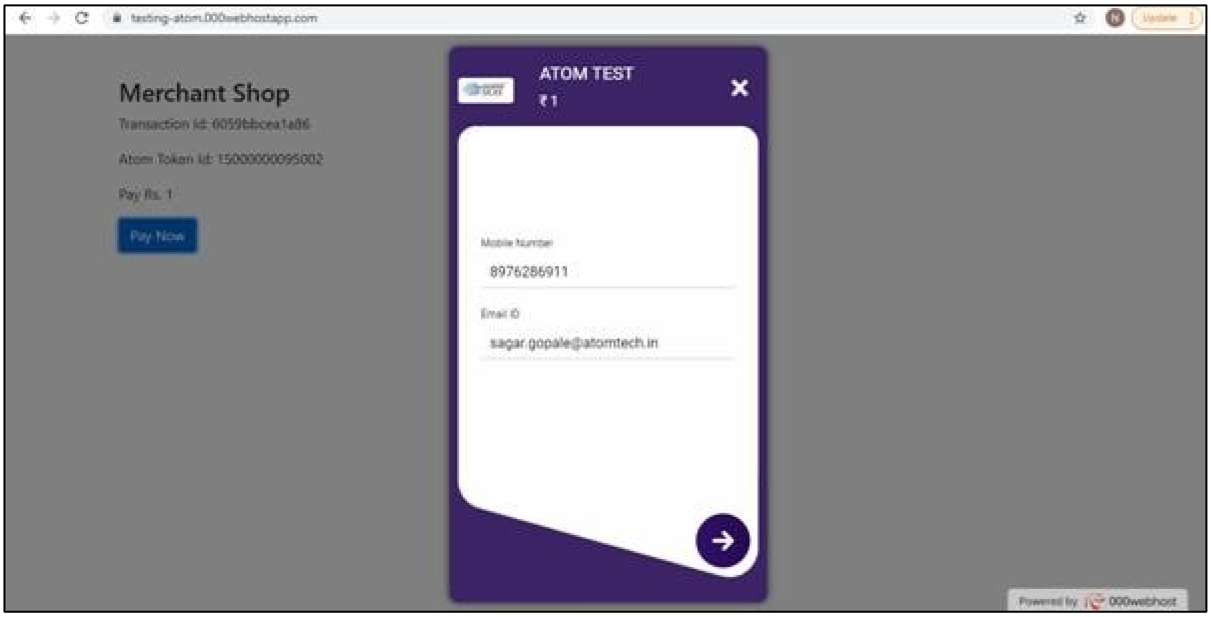
Merchant Shop

Transaction Id: 6059bbcea1a86

Atom Token Id: 15000000095002

Pay Rs. 1

- Clicking on the Pay Now button will open the Atom Insta Pay popup where the user's mobile number and email id passed by the merchant will be displayed



Merchant Shop

Transaction Id: 6059bbcea1a86

Atom Token Id: 15000000095002

Pay Rs. 1

ATOM TEST

₹ 1

Mobile Number

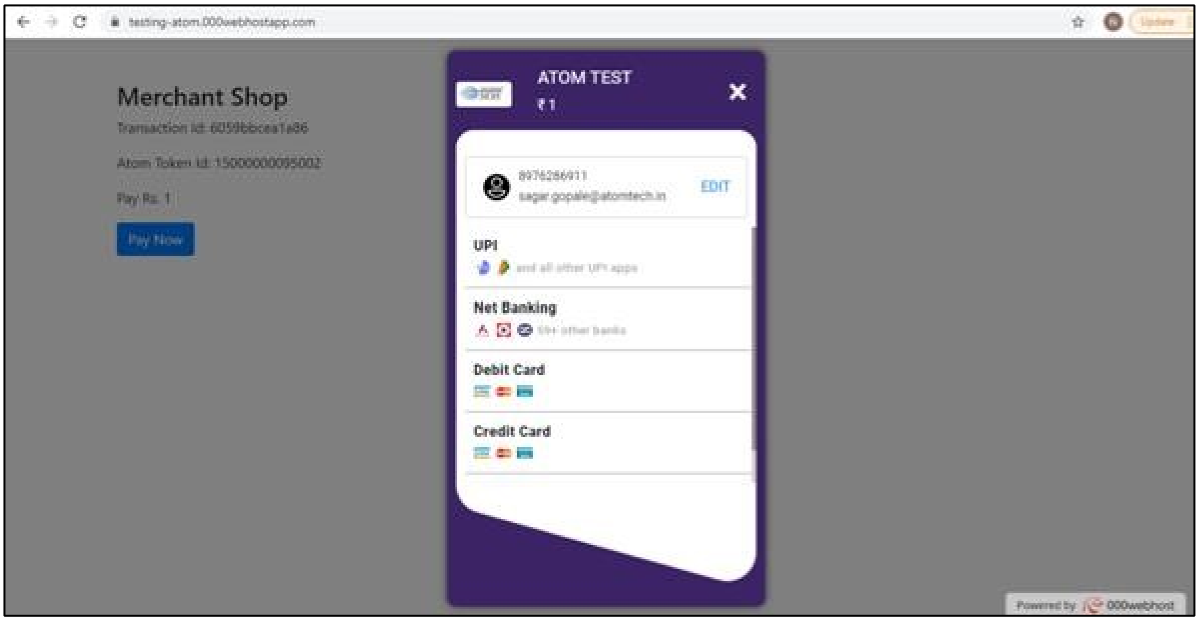
8976286911

Email ID

sagar.gopale@atomtech.in

- Clicking on the arrow will allow the user to see the available payment options i.e. UPI, Net Banking, Debit Card and Credit Card

User will also have an option to edit the email number and mobile number passed by the merchant



testing-atom.000webhostapp.com

Merchant Shop

Transaction Id: 6059bbcea1a86

Atom Token Id: 15000000095002

Pay Rs. 1

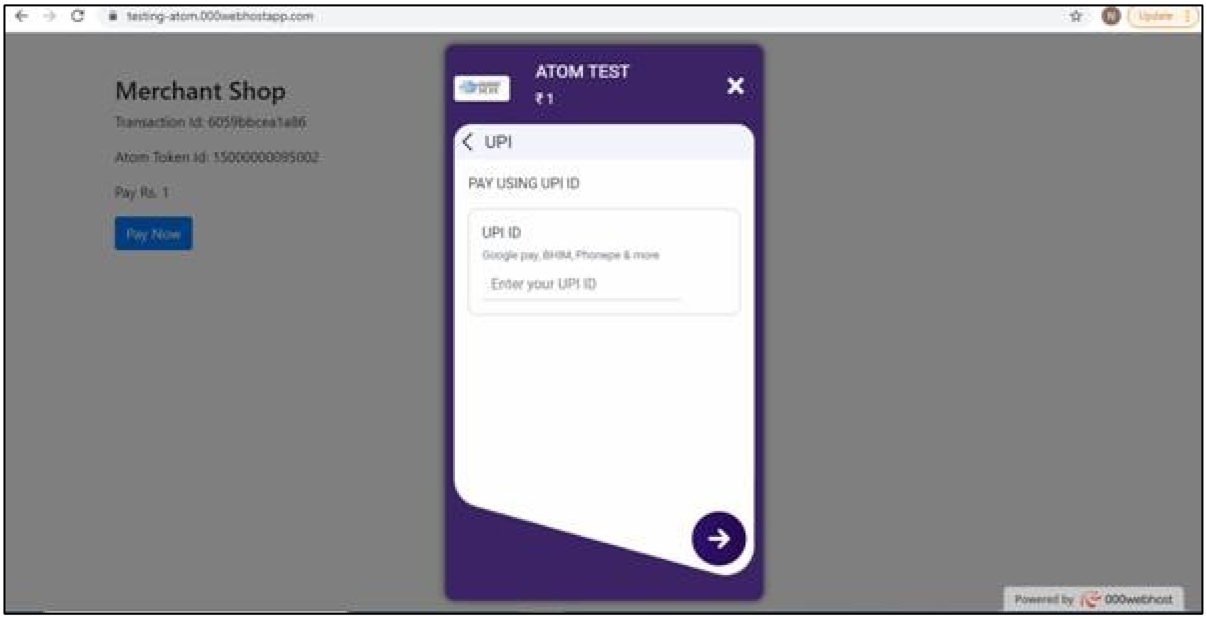
ATOM TEST

₹ 1

8976286911

sagar.gopale@atomtech.in

UPI Transaction



testing-atom.000webhostapp.com

Merchant Shop

Transaction Id: 6059bbcea1a86

Atom Token Id: 15000000095002

Pay Rs. 1

ATOM TEST

₹ 1

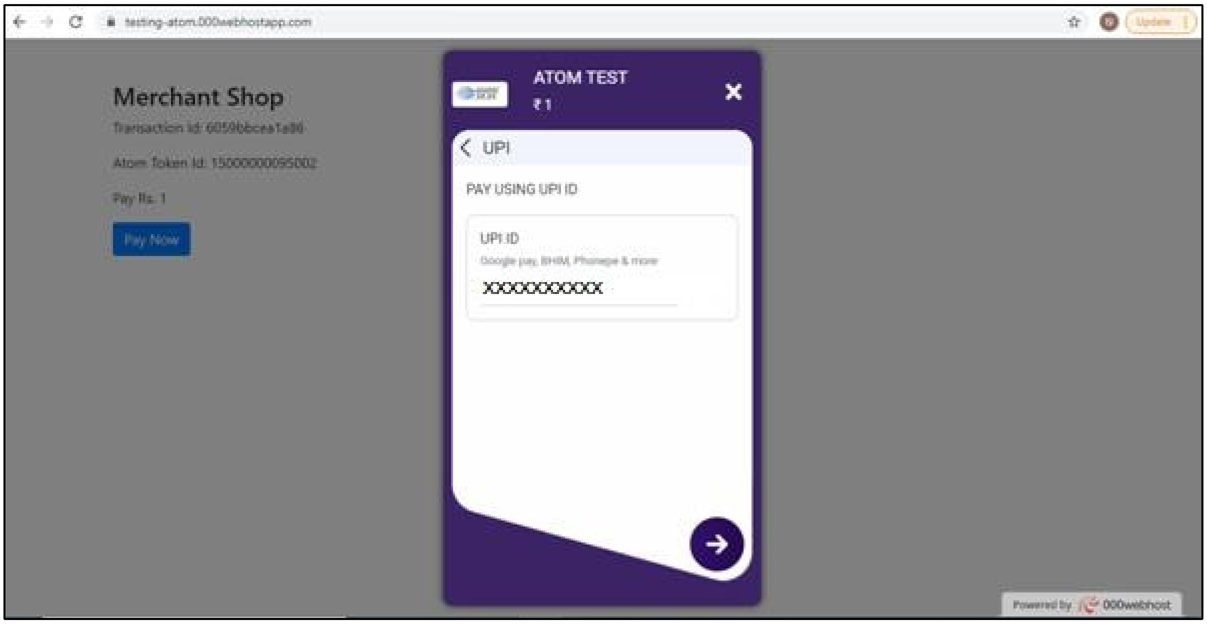
PAY USING UPI ID

UPI ID

Google pay, BHIM, Phonepe & more

Enter your UPI ID

-



testing-atom.000webhostapp.com

Merchant Shop

Transaction Id: 6059bbcea1a86

Atom Token Id: 15000000095002

Pay Rs. 1

ATOM TEST

₹ 1

UPI

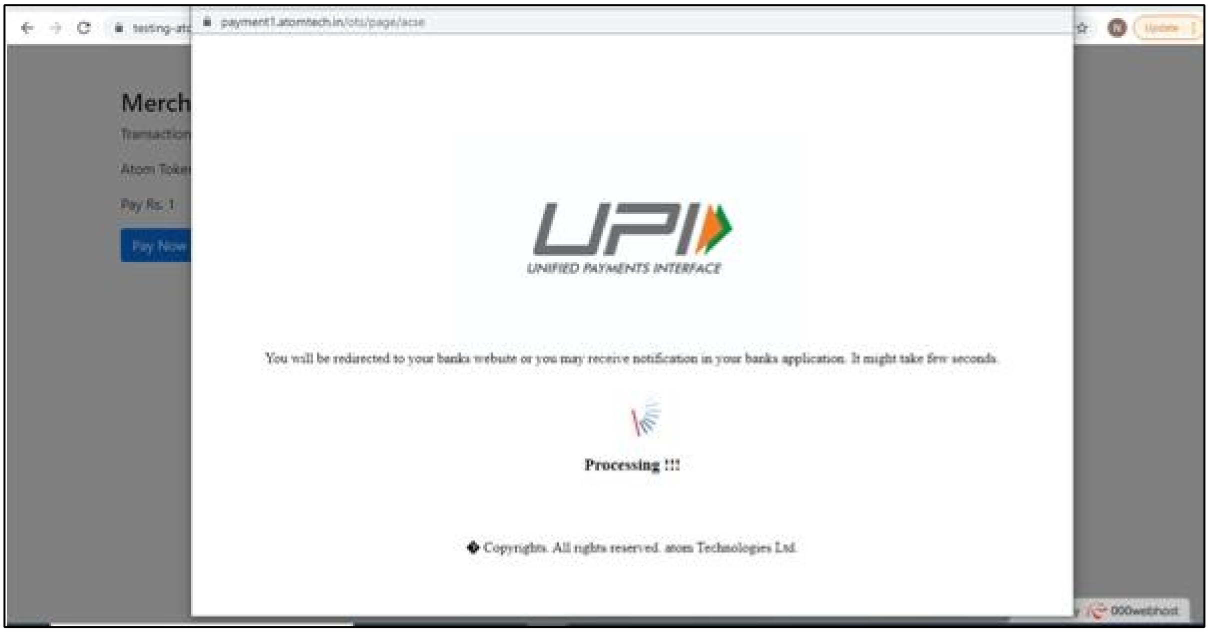
PAY USING UPI ID

UPI ID

Google pay, BHIM, Phonepe & more

XXXXXXXXXXX

- The collect request will be sent to entered VPA



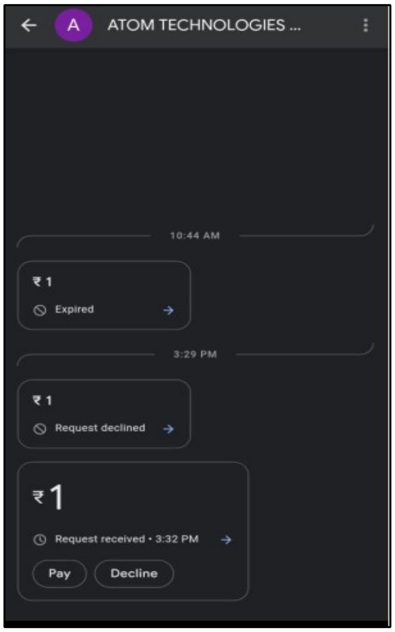
payment1.atomtech.in/ots/page/acse

You will be redirected to your banks website or you may receive notification in your banks application. It might take frw seconds.

Processing !!!

� Copyrights. All rights reserved. atom Technologies Ltd.

User will receive the collect request notification to make the payment



ATOM TECHNOLOGIES ...

10:44 AM

₹ 1

Expired

3:29 PM

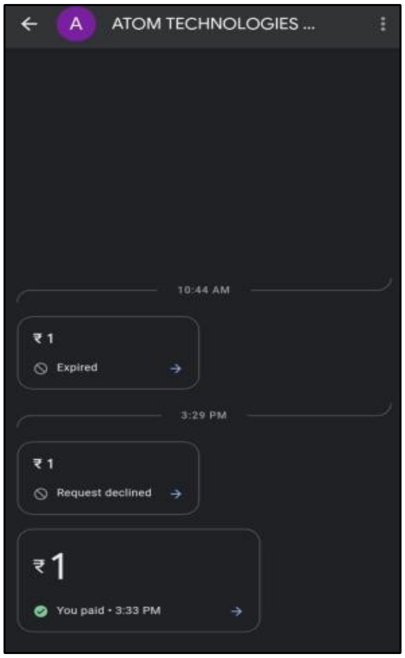
₹ 1

Request declined

₹ 1

Request received • 3:32 PM

Pay Decline



ATOM TECHNOLOGIES ...

10:44 AM

₹ 1

Expired

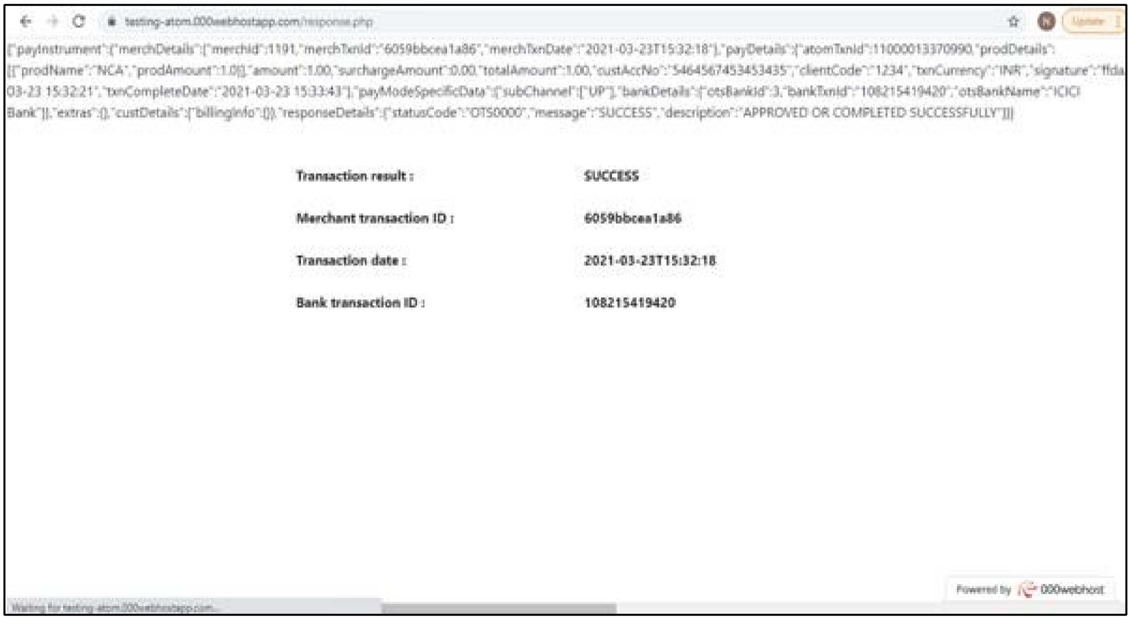
3:29 PM

₹ 1

Request declined

₹ 1

You paid • 3:33 PM

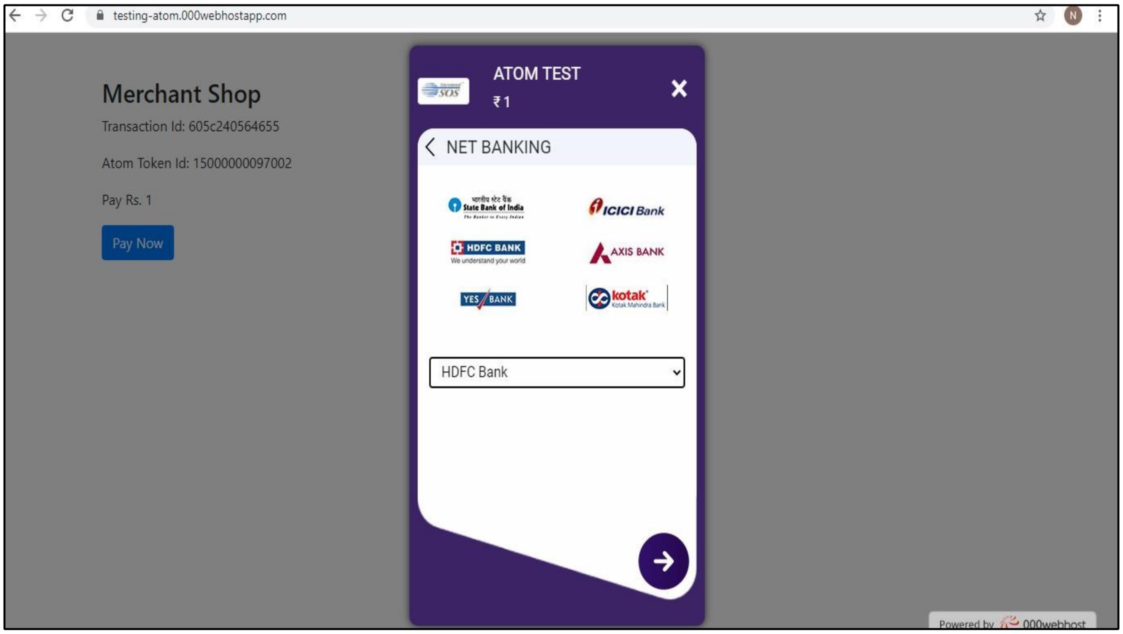


testing-atom.000webhostapp.com/response.php

```
{"payInstrument":{"merchDetails":{"merchId":1191,"merchTxnId":"6059bbcea1a86","merchTxnDate":"2021-03-23T15:32:18"},"payDetails":{"atomTxnId":11000013370990,"prodDetails":
[{"prodName":"NCA","prodAmount":1.0}],"amount":1.00,"surchargeAmount":0.00,"totalAmount":1.00,"custAccNo":"5464567453453435","clientCode":"1234","txnCurrency":"INR","signature":"ffda
03-23 15:32:21","txnCompleteDate":"2021-03-23 15:33:43"},"payModeSpecificData":{"subChannel":["UP"],"bankDetails":{"otsBankId":3,"bankTxnId":"108215419420","otsBankName":"ICICI
Bank"}},"extras":{},"custDetails":{"billingInfo":{}},"responseDetails":{"statusCode":"OTS0000","message":"SUCCESS","description":"APPROVED OR COMPLETED SUCCESSFULLY"}}}
```

Transaction result :	SUCCESS
Merchant transaction ID :	6059bbcea1a86
Transaction date :	2021-03-23T15:32:18
Bank transaction ID :	108215419420

- Clicking on Net Banking Payment option from the Atom Insta Pay popup will open the below page



testing-atom.000webhostapp.com

Merchant Shop

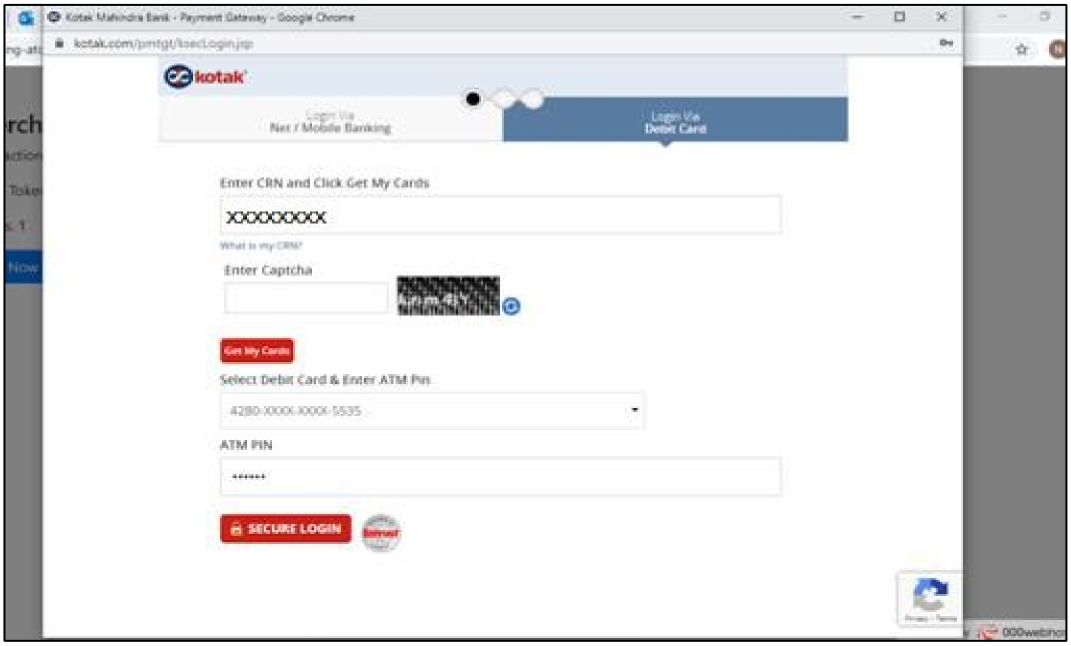
Transaction Id: 605c240564655

Atom Token Id: 15000000097002

Pay Rs. 1

Pay Now
ATOM TEST
₹ 1
NET BANKING
भारतीय स्टेट बैंक State Bank of India
ICICI Bank
HDFC BANK We understand your world
AXIS BANK
YES BANK
kotak Kotak Mahindra Bank
HDFC Bank
Powered by 000webhost

- User can select any of the net banking bank from the available 59+ banks

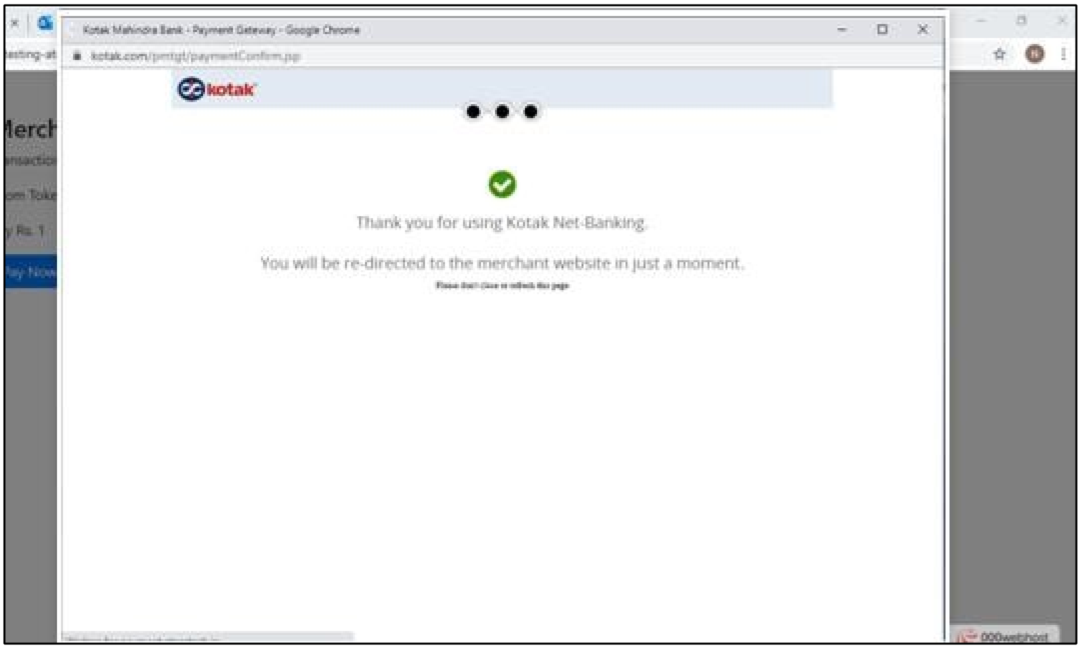


Enter CRN and Click Get My Cards: XXXXXXXXX

Enter Captcha

Select Debit Card & Enter ATM Pin: 4280-XXXX-XXXX-5535

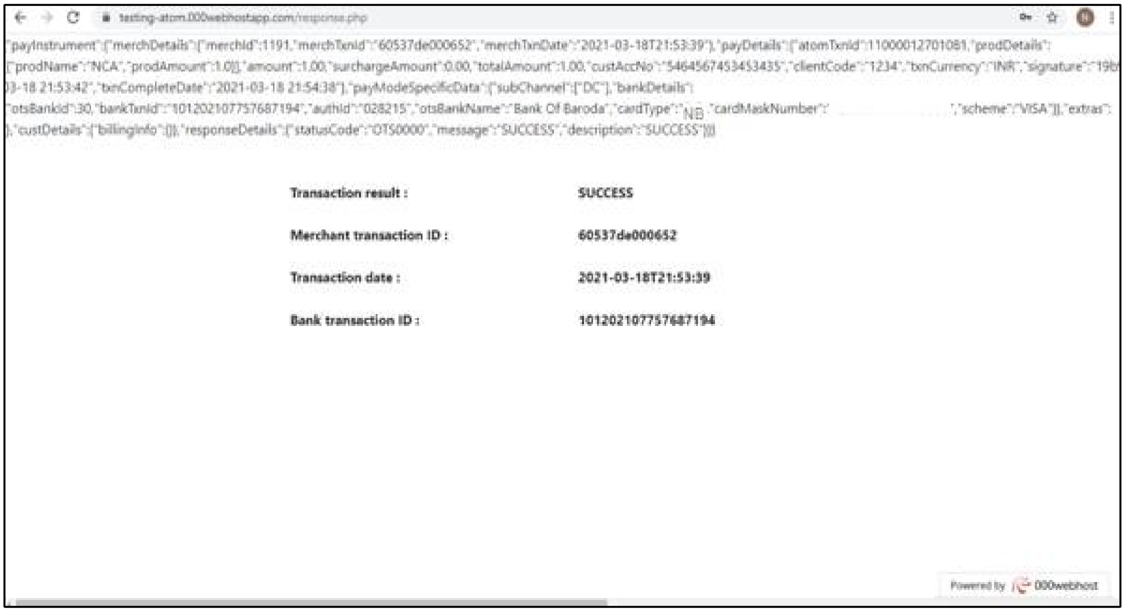
ATM PIN: ******


Kotak Mahindra Bank - Payment Gateway - Google Chrome
kotak.com/pmtgt/paymentConfirm.jsp
kotak
000webhost

Thank you for using Kotak Net-Banking.

You will be re-directed to the merchant website in just a moment.

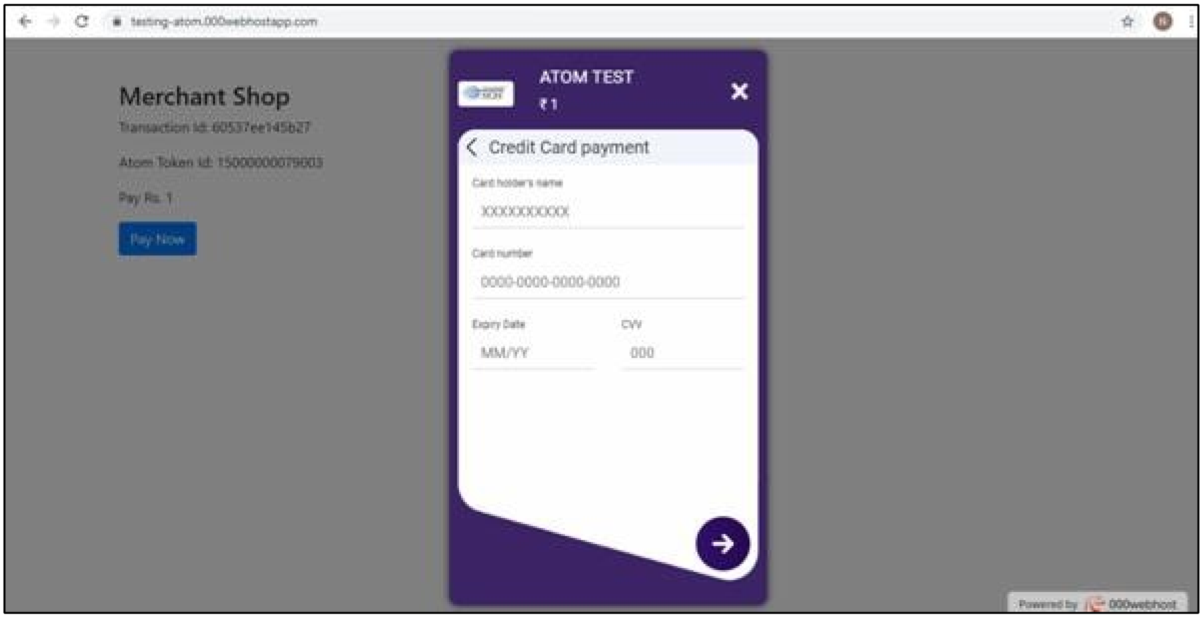
Please don't close or refresh this page



"payInstrument":{"merchDetails":{"merchId":1191,"merchTxnId":"60537de000652","merchTxnDate":"2021-03-18T21:53:39"},"payDetails":{"atomTxnId":11000012701081,"prodDetails":
[{"prodName":"NCA","prodAmount":1.0}],"amount":1.00,"surchargeAmount":0.00,"totalAmount":1.00,"custAccNo":"5464567453453435","clientCode":"1234","txnCurrency":"INR","signature":"19b
}3-18 21:53:42","txnCompleteDate":"2021-03-18 21:54:38"},"payModeSpecificData":{"subChannel":["DC"],"bankDetails":
"otsBankId":30,"bankTxnId":"101202107757687194","authId":"028215","otsBankName":"Bank Of Baroda","cardType":"NB","cardMaskNumber":"   ","scheme":"VISA"}},"extras":
},"custDetails":{"billingInfo":{}},"responseDetails":{"statusCode":"OTS0000","message":"SUCCESS","description":"SUCCESS"}}}

Transaction result :	SUCCESS
Merchant transaction ID :	60537de000652
Transaction date :	2021-03-18T21:53:39
Bank transaction ID :	101202107757687194

Credit Card Transaction



Merchant Shop

Transaction Id: 60537ee145b27

Atom Token Id: 15000000079003

Pay Rs. 1

ATOM TEST

₹ 1

Credit Card payment

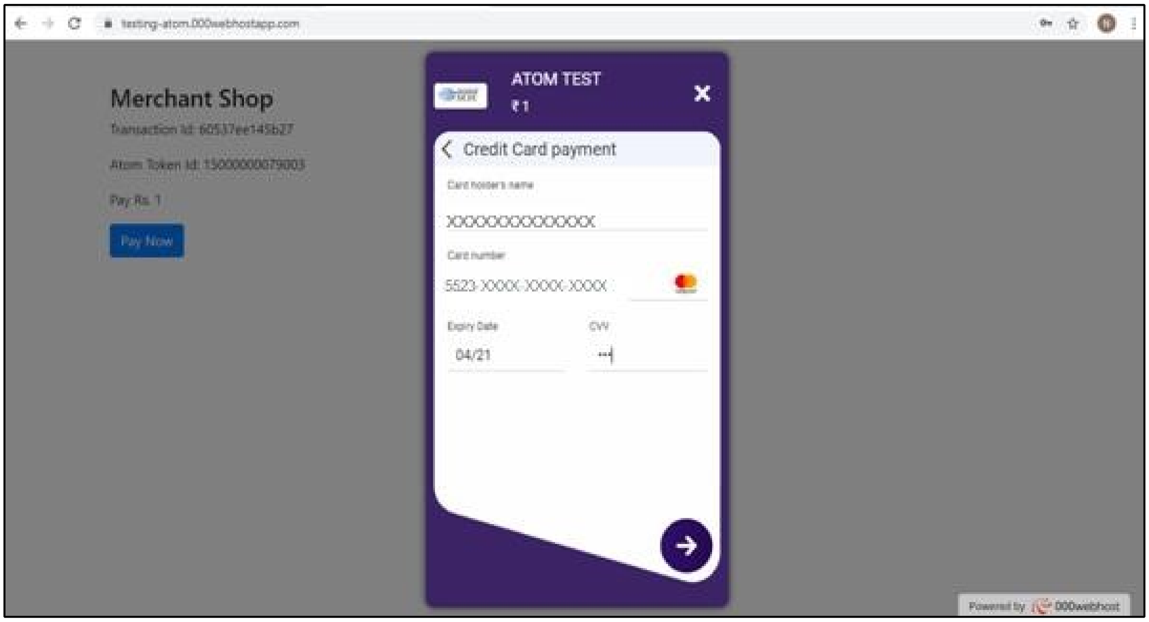
Card holder's name: XXXXXXXXXX

Card number: 0000-0000-0000-0000

Expiry Date: MM/YY

CVV: 000

- User needs to enter the credit card details and process the payment



testing-atom.000webhostapp.com

Merchant Shop

Transaction Id: 60537ee145b27

Atom Token Id: 15000000079003

Pay Rs. 1

ATOM TEST

₹ 1

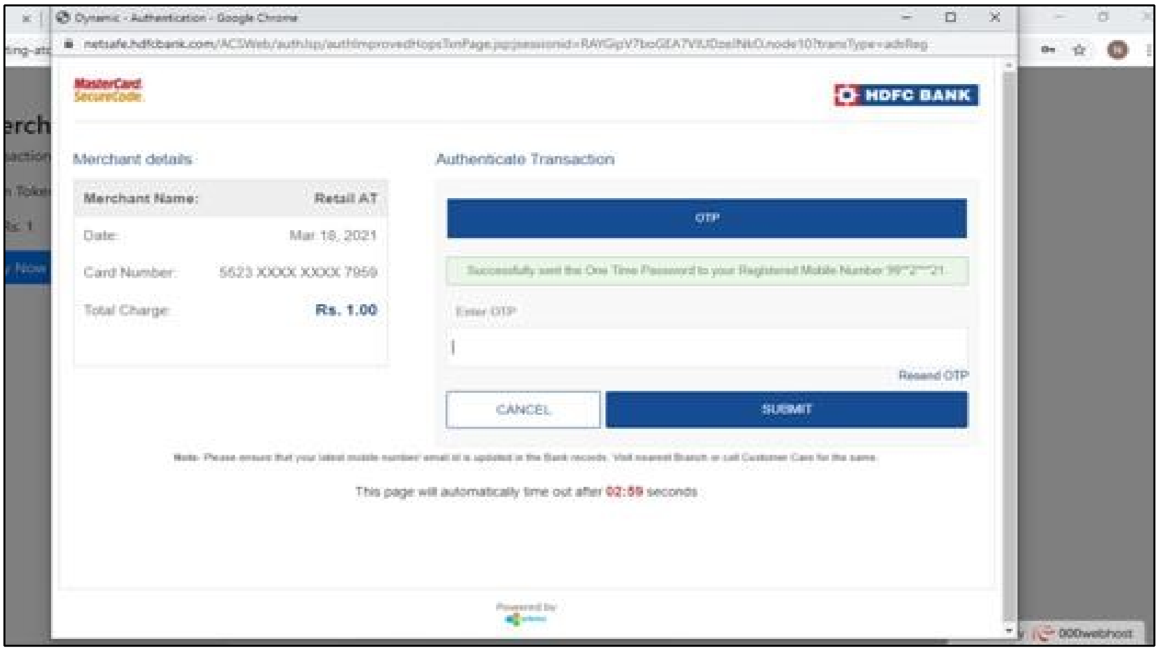
Credit Card payment

Card holder's name: XXXXXXXXXXXXXX

Card number: 5523-XXXX-XXXX-XXXX

Expiry Date: 04/21

CVV: •••



Dynamic - Authentication - Google Chrome

netsafe.hdfcbank.com/ACSWeb/authJsp/authImprovedHopsTxnPage.jsp;jsessionid=RAYGipV7boGEA7VIUDzeINkO.node10?transType=adsReg

MasterCard SecureCode

HDFC BANK

Merchant details

Merchant Name:	Retail AT
Date:	Mar 18, 2021
Card Number:	5523 XXXX XXXX 7959
Total Charge:	Rs. 1.00

Authenticate Transaction

OTP

Successfully sent the One Time Password to your Registered Mobile Number 99**2***21

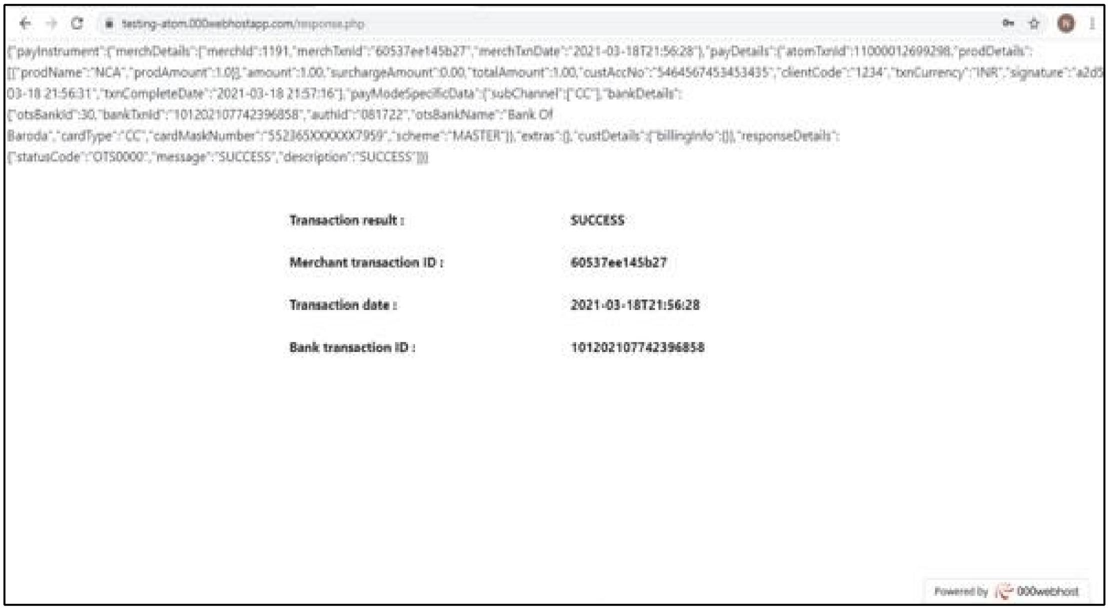
Enter OTP

Resend OTP

Note- Please ensure that your latest mobile number/ email id is updated in the Bank records. Visit nearest Branch or call Customer Care for the same.

This page will automatically time out after 02:59 seconds

Powered by

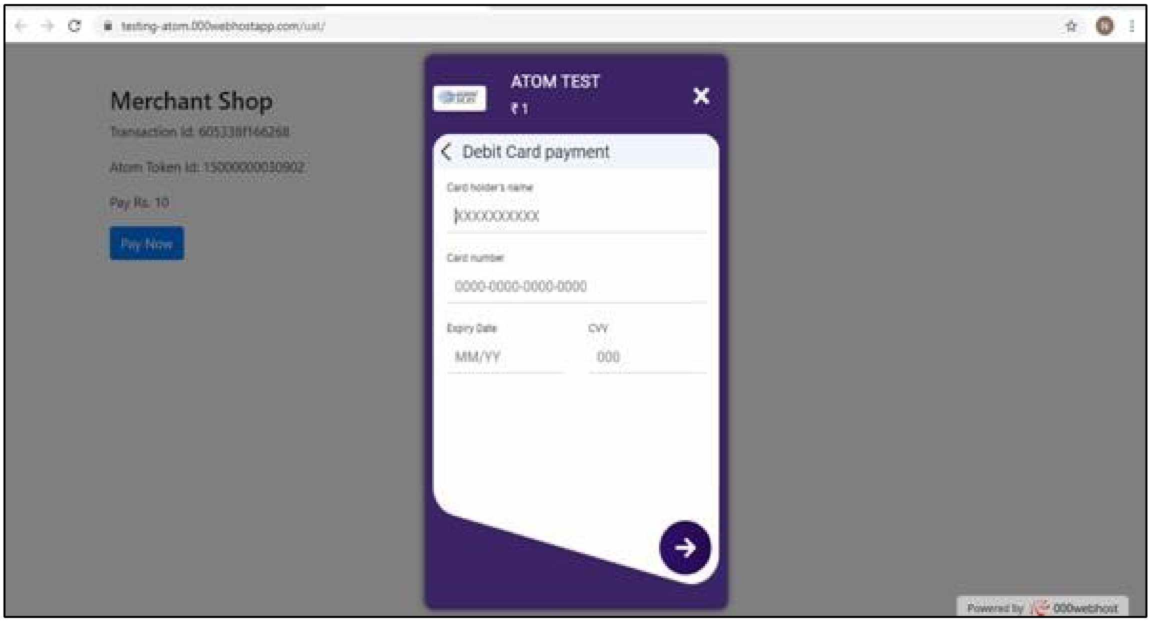


```
{"payInstrument":{"merchDetails":{"merchId":1191,"merchTxnId":"60537ee145b27","merchTxnDate":"2021-03-18T21:56:28"},"payDetails":{"atomTxnId":11000012699298,"prodDetails":
[{"prodName":"NCA","prodAmount":1.0}],"amount":1.00,"surchargeAmount":0.00,"totalAmount":1.00,"custAccNo":"546456745345 3435","clientCode":"1234","txnCurrency":"INR","signature":"a2d5
03-18 21:56:31","txnCompleteDate":"2021-03-18 21:57:16"},"payModeSpecificData":{"subChannel":["CC"],"bankDetails":
{"otsBankId":30,"bankTxnId":"101202107742396858","authId":"081722","otsBankName":"Bank Of
Baroda","cardType":"CC","cardMaskNumber":"552365XXXXXX7959","scheme":"MASTER"}},"extras":{},"custDetails":{"billingInfo":{}},"responseDetails":
{"statusCode":"OTS0000","message":"SUCCESS","description":"SUCCESS"}}}
```

Transaction result :	SUCCESS
Merchant transaction ID :	60537ee145b27
Transaction date :	2021-03-18T21:56:28
Bank transaction ID :	101202107742396858

Debit Card Transaction

- Clicking on Credit Card Payment option from the Atom Insta Pay popup will open the below page



Merchant Shop

Transaction Id: 605338f166268

Atom Token Id: 15000000030902

Pay Rs. 10

ATOM TEST

₹ 1

Debit Card payment

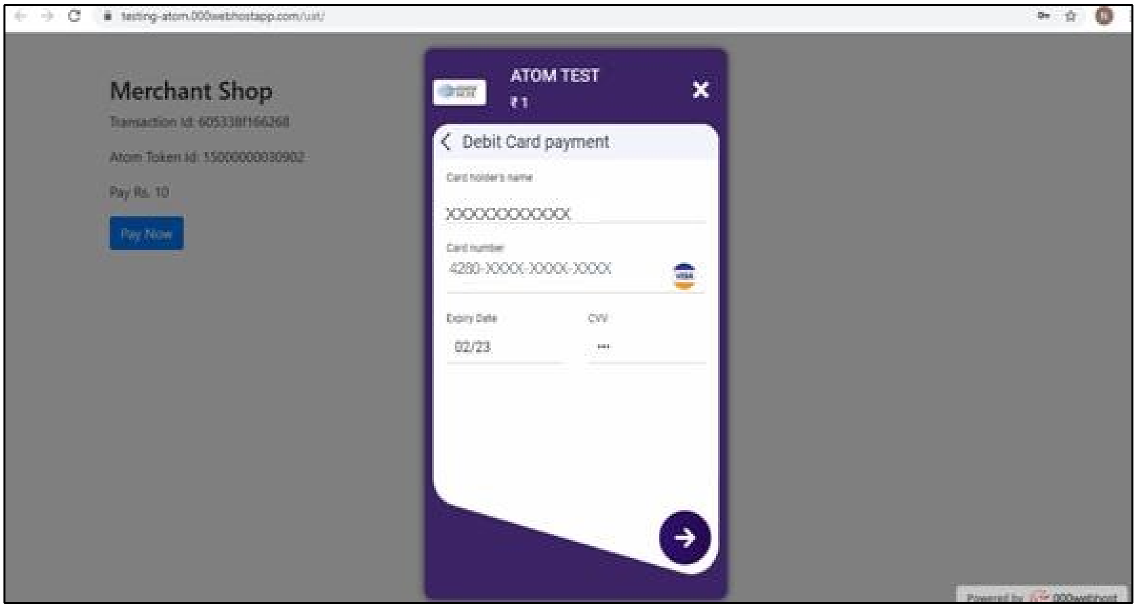
Card holder's name: XXXXXXXXXX

Card number: 0000-0000-0000-0000

Expiry Date: MM/YY

CVV: 000

- User needs to enter the debit card details and process the payment



testing-atom.000webhostapp.com/uat/

Merchant Shop

Transaction Id: 605338f166268

Atom Token Id: 15000000030902

Pay Rs. 10

ATOM TEST

₹1

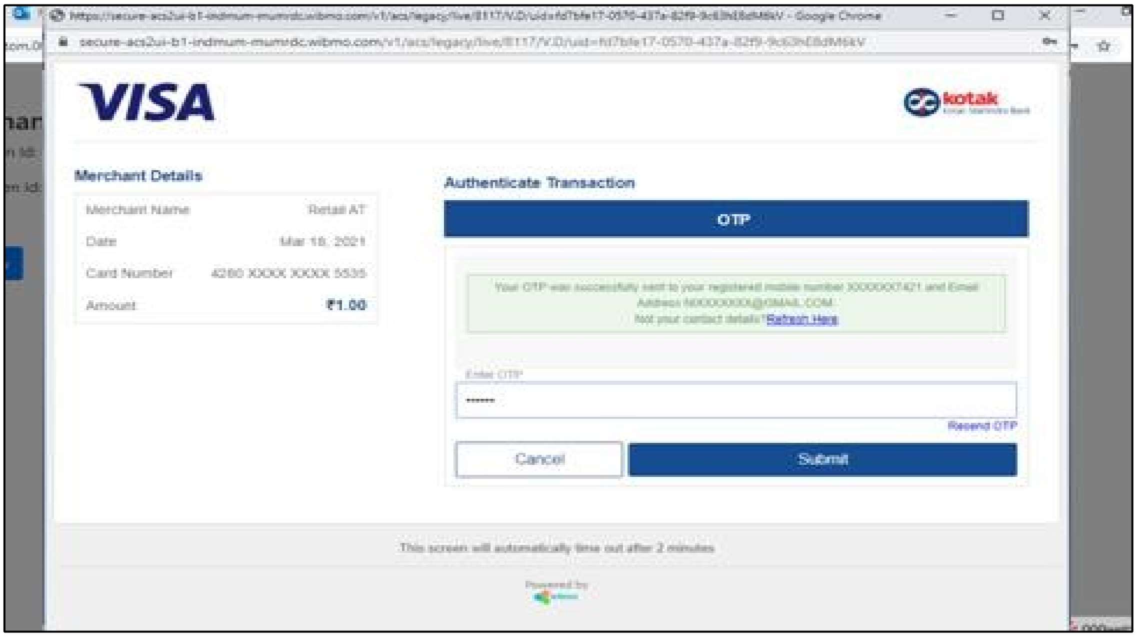
Debit Card payment

Card holder's name: XXXXXXXXXXX

Card number: 4280-XXXX-XXXX-XXXX

Expiry Date: 02/23

CVV: •••



https://secure-acs2ui-b1-indmum-mumrdc.wibmo.com/v1/acs/legacy/live/8117/V.D/uid=fd7bfe17-0570-437a-82f9-9c63hE8dM6kV - Google Chrome

secure-acs2ui-b1-indmum-mumrdc.wibmo.com/v1/acs/legacy/live/8117/V.D/uid=fd7bfe17-0570-437a-82f9-9c63hE8dM6kV

Merchant Details

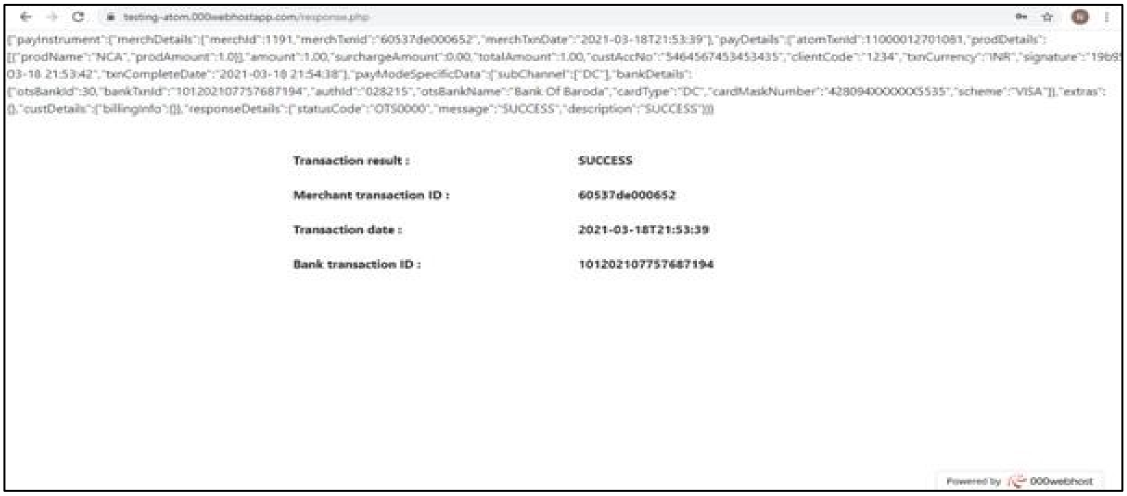
Merchant Name	Retail AT
Date	Mar 18, 2021
Card Number	4280 XXXX XXXX 5535
Amount	₹1.00

Authenticate Transaction

Your OTP was successfully sent to your registered mobile number XXXXXX7421 and Email Address NXXXXXXXX@GMAIL.COM
Not your contact details? Refresh Here

Enter OTP: ••••••

This screen will automatically time out after 2 minutes



```
{"payInstrument":{"merchDetails":{"merchId":1191,"merchTxnId":"60537de000652","merchTxnDate":"2021-03-18T21:53:39"},"payDetails":{"atomTxnId":11000012701081,"prodDetails":
[{"prodName":"NCA","prodAmount":1.0}],"amount":1.00,"surchargeAmount":0.00,"totalAmount":1.00,"custAccNo":"5464567453453435","clientCode":"1234","txnCurrency":"INR","signature":"19b9
03-18 21:53:42","txnCompleteDate":"2021-03-18 21:54:38"}],"payModeSpecificData":{"subChannel":["DC"],"bankDetails":
{"otsBankId":30,"bankTxnId":"101202107757687194","authId":"028215","otsBankName":"Bank Of Baroda","cardType":"DC","cardMaskNumber":"428094XXXXXX5535","scheme":"VISA"}],"extras":
{},"custDetails":{"billingInfo":{}},"responseDetails":{"statusCode":"OTS0000","message":"SUCCESS","description":"SUCCESS"}}}
```

Transaction result :	**SUCCESS**
Merchant transaction ID :	**60537de000652**
Transaction date :	**2021-03-18T21:53:39**
Bank transaction ID :	**101202107757687194**

3.Merchant APIs

AUTH API

- When the end user wants to make the payment, merchant needs to initiate the Rest API
- It is the authentication API used to generate the atom token Id which will required by the merchant to land on the Atom Insta Pay Payment popup

- Request to generate token Id will be server to server using POST method

AUTH API Request

Request Parameters

Sr No	Parameter	Data Type	Max Length	Mandatory/Optional
1	version	String	10	O
2	api	String	10	O
3	platform	String	10	O
4	merchId	Long	10	M
5	userId	String	45	M
6	password	String	80	M
7	merchTxnId	String	50	M

8	merchTxnDate	String		M
9	amount	Bigdecimal	12,2	M
10	product	String	50	M
11	custAccNo	String	45	For TPV its Mandatory
12	txnCurrency	String	5	M
13	custEmail	String	100	M
14	custMobile	String	20	M
15	udf1	String	45	O
16	udf2	String	45	O
17	udf3	String	45	O
18	udf4	String	45	O
19	udf5	String	45	O

Request in JSON format

```
{
    "payInstrument": {
    "headDetails": {
      "version": "OTSv1.1",
      "api": "AUTH",
      "platform": "FLASH"
    },
    "merchDetails": {
      "merchId": "8952",
      "userId": "",
      "password": "Test@123",
      "merchTxnId": "Test123450",
      "merchTxnDate": "2021-09-04 20:46:00"
    },
    "payDetails": {
      "amount":  "100.00",
      "product": "NSE",
      "custAccNo": "213232323",
      "txnCurrency": "INR"
    },
    "custDetails": {
      "custEmail": "sagar.gopale@atomtech.in",
      "custMobile": "8976286911"
    },
    "extras": {
      "udf1":"",
      "udf2":"",
      "udf3":"",
      "udf4":"",
      "udf5":""
    }
      }
}
```

Request in Encrypted format

- Merchant needs to encrypt the request using the AES -256-CBC encryption algorithm • It is mandatory to pass the encData and merchId in the REST API call
- For UAT:
  https://caller.atomtech.in/ots/aipay/auth?encData=11C0BE23D1A9B9A03A80BBBE07DBEF8990

```
   2F7DA9F553156D98895794D6BCEA4078BFA0FB4F796687DB5983ECB1AAD813DDB66DCABC82
   D8FDCC3CCDB8AEC94BB0CB94B2BA3B463A994ECF2E7387875E5F624FE651150719DC8203F1
   42761FCECF65A3C54176D852D7C694910ED5EFEB7030C2F2195CD645CE4FD7174BF8B36255
   87B9989F2384AA43ADA01DD8F872D35290606D34840C32ED0199E3399C453ADBC2848C5F68
   1C8EDCE2B85792E90E9FA52A67D8669BA79A2D8C0D2A796473BE9326FB10EB3B7E1457B4B4
FDD842262260EA0399B22B5314FEDC62648D6D47D059D60D6A392AAC0D4B3DDBE44EC36A5B
3F405862BA2E5447F9A42A23E5A3F4A47C4513A5E0EB23E2CA9FC3E10ED9E894D1F1A08EBA
C32D71E29D516435A014B12F8F16A102209F798FEF7B77C2F3ABDDD659CF5AC51A73D82906
4C3F477F29C3D651CEDD88094CC47D01BF57B27C8787B014029846622125235983CFF3097E
ED44B62BE87A10F4FEAB031576777358A9316FCE03891BC2014A1A81B5FA1B6D67811479D4
   93643D6733B287655F78689C20798803FFEF4E0038A02B37B13A0AFFAAB6491CD50163CB45
   FEEDDAB07DF2722C27482150EA2CF53B964F9FB733658B056641A741AE13F2189E3EADF806
246D6604D2A2E170D728D0614BED340FC93E5789772C3356F2149CDA1D90DCB0AA9198B9FF
073B91A30BBAA62EF611326FEAE4B746682C69A15B17B109DFB7ADF2BA4B8C8AD905BD2721
8923C23F574B2D3A32052D579FFE62AEA74FB5A0CBC55625AE1D6C9B199955273BA020C7D4
```

F88B81ACCB47EFE2209ABAA2068CE543302437AE7FEB880896E1F21361E96C671D4AFC3B50
19BFB2309754A87C82B26F30E6D70FBFCB3BA4D6942883B617553740D1F75A12ED3F86FE06
C379FFC61EA1F48B309BEACB0B0AD6A4277B3E6A9EE236757547DAD8EB75B489D853EA09AA
BE4E028D3514BDB3C25C614EAFDEB2F5D8B4D4CED99470AB4BA04C2A55F410CA34FEE0AC08
BBAF0DA9AC4B318ECA77CF628481ABDA7619C51787CB47A72D8FC6D32AAA7F89DB813E204F
ACAAD0973835EB505199DCC1342DBC9D40559563B2246B2B46F0BAA4B1990CC983D3FFAC48
5DE90E6778CAD45FA7722C4036426785598E184BAC4D7FCD66DE46F4BF569B91B37A17C3E3
0F8E557EF3723483471BAA0DD98593D33EC7944A4646FD2CA7E799D994A085C7ECDFB0DDCA
EE0FF2FE3A32FC9D8BE3A0A829A136560A8BAF74063BA8C3A35D3ADCCAF072795CE59C7BA6
B9247BFE2EAA5BD0436B9D8ACFB894DCAB0ED59EDEDA3A&merchId=8952

- For Production:
  https://payment1.atomtech.in/ots/aipay/auth?encData=11C0BE23D1A9B9A03A80BBBE07DBEF
  89902F7DA9F553156D98895794D6BCEA4078BFA0FB4F796687DB5983ECB1AAD813DDB66DCA
  BC82D8FDCC3CCDB8AEC94BB0CB94B2BA3B463A994ECF2E7387875E5F624FE651150719DC82
  03F142761FCECF65A3C54176D852D7C694910ED5EFEB7030C2F2195CD645CE4FD7174BF8B3
  625587B9989F2384AA43ADA01DD8F872D35290606D34840C32ED0199E3399C453ADBC2848C
  5F681C8EDCE2B85792E90E9FA52A67D8669BA79A2D8C0D2A796473BE9326FB10EB3B7E1457
  B4B4FDD842262260EA0399B22B5314FEDC62648D6D47D059D60D6A392AAC0D4B3DDBE44EC3
  6A5B3F405862BA2E5447F9A42A23E5A3F4A47C4513A5E0EB23E2CA9FC3E10ED9E894D1F1A0
  8EBAC32D71E29D516435A014B12F8F16A102209F798FEF7B77C2F3ABDDD659CF5AC51A73D8
  29064C3F477F29C3D651CEDD88094CC47D01BF57B27C8787B014029846622125235983CFF3
  097EED44B62BE87A10F4FEAB031576777358A9316FCE03891BC2014A1A81B5FA1B6D678114
  79D493643D6733B287655F78689C20798803FFEF4E0038A02B37B13A0AFFAAB6491CD50163
  CB45FEEDDAB07DF2722C27482150EA2CF53B964F9FB733658B056641A741AE13F2189E3EAD
  F806246D6604D2A2E170D728D0614BED340FC93E5789772C3356F2149CDA1D90DCB0AA9198
  B9FF073B91A30BBAA62EF611326FEAE4B746682C69A15B17B109DFB7ADF2BA4B8C8AD905BD
  27218923C23F574B2D3A32052D579FFE62AEA74FB5A0CBC55625AE1D6C9B199955273BA020
  C7D4F88B81ACCB47EFE2209ABAA2068CE543302437AE7FEB880896E1F21361E96C671D4AFC
  3B5019BFB2309754A87C82B26F30E6D70FBFCB3BA4D6942883B617553740D1F75A12ED3F86
  FE06C379FFC61EA1F48B309BEACB0B0AD6A4277B3E6A9EE236757547DAD8EB75B489D853EA
  09AABE4E028D3514BDB3C25C614EAFDEB2F5D8B4D4CED99470AB4BA04C2A55F410CA34FEE0
  AC08BBAF0DA9AC4B318ECA77CF628481ABDA7619C51787CB47A72D8FC6D32AAA7F89DB813E
  204FACAAD0973835EB505199DCC1342DBC9D40559563B2246B2B46F0BAA4B1990CC983D3FF
  AC485DE90E6778CAD45FA7722C4036426785598E184BAC4D7FCD66DE46F4BF569B91B37A17
  C3E30F8E557EF3723483471BAA0DD98593D33EC7944A4646FD2CA7E799D994A085C7ECDFB0
  DDCAEE0FF2FE3A32FC9D8BE3A0A829A136560A8BAF74063BA8C3A35D3ADCCAF072795CE59C
  7BA6B9247BFE2EAA5BD0436B9D8ACFB894DCAB0ED59EDEDA3A&merchId=8952

AUTH API Response

Response Parameters

Sr No	Parameter	Data Type	Max Length	Mandatory/Optional
1	atomTokenId	Long	10	M

2	txnStatusCode	String	5	M
3	txnMessage	String	50	M
4	txnDescription	String	100	M

Response in JSON format

encData [Plain] :

```
{
"atomTokenId":15000000033303,
"responseDetails":{
"txnStatusCode":"OTS0000",
"txnMessage":"SUCCESS",
"txnDescription":"ATOM TOKEN ID HAS BEEN GENERATED SUCCESSFULLY"
}
}
```

- Possible status code values and messages that will be posted in response while generating the Atom Token Id are mentioned below:

Sr. No.	txnStatus	txnMessage	txnDescription
1	OTS0000	SUCCESS	ATOM TOKEN ID HAS BEEN GENERATED
2	OTS0451	FAILED	INVALID MERCHANT INFORMATION
3	OTS0600	FAILED	AUTH SERVICE FAILED
4	OTS0600	FAILED	TOKEN ID GENERATION FAILED

Response in Encrypted format

- Merchant needs to decrypt the received response using the AES-256-CBC decryption algorithm
- Post decryption the atom token Id passed by Atom in the response will be received by the merchant
- For Production:
  merchId=8952&encData=5500FEA2F09DA7EF128CFE7D2D01F2533706602C7F5CF17176C85222692E684F1EC38C6EC4D2A30305F58B09A5DAA5285A4CCB10A2298D743622C0835BD2D70CE5AF22C9B8610B169544715B6ACDBE3B85532198D73B28B0663F5B96395AEC61CAD1B2CCFD6E63A8B1FBDBFA3E4F22EFE3C7A4F9683EA1E49CA371F00E4732CABFEB1589E4D93C24C8C33219674BFD45EA96150EA37D7ED8C3BA018B7879C809DE5B283FFE03850C452AED7A43F82B23

Java Script

- Once the token Id is received by the merchant, the Java Script function needs to be called by the merchant
- This Java Script function can be called using the button click
- Mandatory JavaScript CDN link for UAT: https://pgtest.atomtech.in/staticdata/ots/js/atomcheckout.js
- Mandatory JavaScript CDN link for Production: https://psa.atomtech.in/staticdata/ots/js/atomcheckout.js

JAVA Script in JSON format

```
 <script>    function
 openPay(){        const
 options = {
       "atomTokenId": "atomTokenId ",
       "merchId": "8952",
       "custEmail": "sagar.gopale@atomtech.in",
       "custMobile": "8888888888",
       "returnUrl":"http://your-response-url/response.php"
     }

     let atom = new AtomPaynetz(options,'uat');
   }

   </script>
```

Final Transaction Response

- Once the transaction is completed by the end user the final transaction response will be sent to the merchant by Atom
- Transaction response will provide the details such as the payment mode used, Bank Name, Card Number
- Merchant can capture the details and accordingly show the response to the end customer

Response Parameters

Sr No	Parameter	Datatype	Max Length	Mandatory/Optional

1	merchId	Long	10	M
2	merchTxnId	String	50	M
3	merchTxnDate	String		M
4	atomTxnId	Long	10	M
5	prodName	String	50	M
6	prodAmount	BigDecimal	12,2	M
7	amount	Bigdecimal	12,2	M
8	surchargeAmount	BigDecimal	12,2	M
9	totalAmount	BigDecimal	12,2	M
10	custAccNo	String	45	In case of TPV its mandatory
11	clientCode	String	45	M
12	txnCurrency	String	5	M
13	signature	String		M
14	txnInitDate	String		M
15	txnCompleteDate	String		M
16	subchannel	String	2	M
17	otsBankId	Integer	5	M
18	authId	String	45	CC-M DC-M UP-M
19	bankTxnId	String	45	M
20	otsBankName	String	25	M
21	cardType	String	15	CC-M DC-M
22	cardMaskNumber	String	20	CC-M DC-M
23	scheme	String	15	CC-M DC-M
24	statusCode	String	5	M
25	message	String	50	M
26	description	String	100	M

Response in JSON format

```
{                   "payInstrument":{

                              "merchDetails":{

                                        "merchId":8952,

                                        "merchTxnId":"60583241b0c1c",

                                        "merchTxnDate":"2021-03-22T11:29:58"

                              },

                              "payDetails":{

                                        "atomTxnId":11000000075746,

                                        "prodDetails":[{"prodName":"NSE","prodAmount":10.0}],

                                        "amount":10.00,

                                        "surchargeAmount":0.00,

                                        "totalAmount":10.00,

                                        "custAccNo":"5464567453453435",

                                        "clientCode":"1234",

          "txnCurrency":"INR",


           "signature":"5678a713fdfac844fe6349107996c336b43ae86413f3b25237211022f38e3be2b2862
43bbc2019aa1885cae1e80e35edbccd5210e92d5587d54d72c0e27a8017",

                                        "txnInitDate":"2021-03-22 11:29:59",
                                        "txnCompleteDate":"2021-03-22 11:31:22"

                              },

                              "payModeSpecificData":{

                                        "subChannel":["NB"],
 "bankDetails":{"otsBankId":30,"bankTxnId":"284603118","otsBankName":"Bank of Baroda Net Banking
Retail"}

                              },

                              "custDetails":{

                                    "billingInfo":{}

                              },

                              "extras":{

                                    "udf1":"",

                                    "udf2":"",

                                    "udf3":"",

                                    "udf4":"",

                                    "udf5":""

                              },

                              "responseDetails":{

                                    "statusCode":"OTS0000",

                                    "message":"SUCCESS",

                                    "description":"TRANSACTION IS SUCCESSFUL."

                              }

                        }

}
```

- Possible status code values, messages and description that will be posted in the final transaction response are mentioned below:

CODE	**MESSAGE**	**DESCRIPTION**	
OTS0000	SUCCESS	TRANSACTION SUCCESSFUL	
OTS0601	FAILED	IN STAGE TWO TRANSACTION, ATOM TXN ID SHOULD NOT BE NULL.	

OTS0602	FAILED	INCORRECT SURCHARGE AMOUNT.	
OTS0603	FAILED	SUBCHANNEL CAN NOT BE EMPTY.	
OTS0604	FAILED	IF SUBCHANNEL IS NB THEN BANK ID SHOULD NOT BE EMPTY.	
OTS0605	FAILED	IF SUBCHANNEL IS CC or DC THEN CARD DATA SHOULD NOT BE EMPTY.	
OTS0606	FAILED	IF SUBCHANNEL IS UP THEN VPA SHOULD NOT BE EMPTY.	
OTS0607	FAILED	IF SUBCHANNEL IS EM THEN EMI DETAILS SHOULD NOT BE EMPTY.	
OTS0608	FAILED	IN STAGE TWO TRANSACTION, SUBCHANNEL CAN NOT BE EMPTY.	
OTS0609	FAILED	CARD DETAILS ARE MISSING.	
OTS0610	FAILED	INVALID EMI TENURE.	
OTS0611	FAILED	INVALID EMI BANK NAME.	
OTS0612	FAILED	SUMMATION OF AMOUNT & SURCHANGE AMOUNT SHOULD BE EQUAL TO TOTAL AMOUNT.	
OTS0613	FAILED	RESPONSE IS ALREADY AVAILABLE IN DB FOR ATOM TXN ID.	
OTS0614	FAILED	ATOM's AMOUNT MISMATCHED WITH BANK's AMOUNT.	
OTS0615	FAILED		IT WILL SHOW IF YOU MISSED ANY MANDATORY PARAMETER
OTS0616	FAILED	INSUFFICIENT MERCHANT INFORMATION.	
OTS0617	FAILED	IN STAGE TWO TRANSACTION, DB STAGE SHOULD BE 1.	
OTS0618	FAILED	IN STAGE TWO TRANSACTION, PROVIDED AMOUNT SHOULD BE EQAUL TO DB AMOUNT.	
OTS0619	FAILED	SUBCHANNEL & CARD TYPE MISMATCHED.	
OTS0620	FAILED	CARD SCHEME MISMATCHED.	
OTS0621	FAILED	AMOUNT SHOULD BE WITHIN A RANGE.	
OTS0622	FAILED	ACCOUNT DETAILS MANDATORY FOR PENNYDROP.	

OTS0623	FAILED	IN SEAMLESS TRANSACTION, WHEN MERCHANT IS ON SURCHARGE THEN SURCHARGE AMOUNT SHOULD	
OTS0624	FAILED	TRANSACTION RESPONSE IS ALREADY AVAILABLE FOR ATOM TXN ID.	
OTS0625	FAILED	BANK TXN AMOUNT IS MISMATCHED WITH ATOM TXN AMOUNT.	
OTS0626	FAILED	AMOUNT & TOTAL AMOUNT OF ALL PASSED PRODUCTS SHOULD BE EQUAL.	
OTS0627	FAILED	INAPPROPRIATE WAY TO CONSUME OTS PAYMENT SERVICE.	
OTS0628	FAILED	ATOM TXN ID SHOULD NOT BE EMPTY/NULL IN BANK RESPONSE.	
OTS0629	FAILED	IN SEMI-SEAMLESS TRANSACTION, IF IT IS CARD TRANSACTION THEN SURCHARGE AMOUNT SHOULD NOT BE NULL/EMPTY/ZERO.	
OTS0630	FAILED	CARD TYPE MISMATCHED	
OTS0631	FAILED	FAILED-INVALID CARD NUMBER.	
OTS0632	FAILED	FAILED-CARD EXP MONTH/YEAR SHOULD BE IN PROPER FORMAT.	
OTS0633	FAILED	FAILED-CARD IS EXPIRED.	
OTS0634	FAILED	FAILED-INVALID CVV.	
OTS0635	FAILED	FAILED-TRANSACTION MODE NOT FOUND IN URL PATTERN.	
OTS0636	FAILED	FAILED-UNABLE TO DECRYPT CARD DETAILS	
OTS0637	FAILED	FAILED-UNABLE TO DECRYPT ENC DATA	
OTS0638	FAILED		IT WILL SHOW IF YOU MISSED ANY MANDATORY PARAMETER FROM ENCDATA
OTS0639	FAILED	FAILED-FUTURE DATE NOT ALLOWED.	
OTS0640	FAILED	FAILED-DATE FORMAT MUST BE "+OTSUtil.ATOM_DATE_PATTERN;	
OTS0641	FAILED	MERCHID OR ENCDATA MISSING.	

Response in Encrypted format

- Final Transaction response will be sent by Atom in the encrypted format
- Merchant needs to decrypt the response using AES-256-CBC decryption algorithm
- Encrypted Data:

merchId=8952&encData=0CD63904F407DC4D24167F8D8F2BEE997C80D4536815B71C1EDDB8F
DCBACE8B9F6D72174CC18263D88AD7728C821A1F7B789A3F6C8EE43C28933A6748D3239346F
4AAEC04AFFD7687FB31FFDA9AAF0F64A2911990F315B425F11F1196EAF59E9471644C321A005
0BF64C72DA0EFE6A11D14F4AA3890267552FA4E3FA84B21275150F7E98CCF84BD145DCB6F00E
F7494FAEE9D23852507F46517B67D8F9680941BF887A761D9B26879B2A369E9B6CEF835DD51D
EE86982255CCB98A8C9D8C3662FB87AF859B9E0DF167D7D48E89BE6896226C5D8AED6B2CA19
86325D030B646ADAD4C3944A8B851D3D3448C7F0CD37DB8E466A51AFFD43E38307E2ECA296
75614AC4DAA072CB6595974E4E1EEEA70F031B7BA3C5FDF2D4F599854B658B9D9D2DAA3FEAA
67B534BFADE98FB5D123BB8FDB563F840B7B28C71054CCA358F50348CE842C3AA05FDBF0400
892F03B5290FF5CA7E01C174FCEBC0B65012FF9F0AB1C19F483570599F9C681001F46C342CD6A
491B2B4619FC70D30FF0CEAB49F2B5201ADCE25C510027496A4A64EA3A2661D55DA89741600
9769F2F501CE864F8360EC8EB7558CB24AAC577E338EA3DB515DA5199D159688C0A624D0C54
8135A6001D2543D7274BEBA6C2AB648042C08CDF4732CB63FB0B57DDD967E8F577427EF19D6
69E0A16AF77CA605491C1657744A1395065C1A459232A12743E62DEF8B6FC9F6075C8DA7E27C
2AE6874698FAD367A9C23CB28FDC001AE5501693FB5435BD871DDCC5BB0E236867F60AA04E5
2A7582DB8F25D0FC4CFD7C4D223D67DADF248E0C920F0BCA7C9E774604FAEB56FAE025091FA
595A7797C4331D3FCF9CEB881A1392F231A4F05F4A4FFA58099802A20F7345EC7120BCE82ACD
7C536B2627E4DD22F2CE177C27F38B544ED991EF83A1F888A6217B48511E6A1FCFEF065FD84C
459E4695E39AD5BB078B42751A798B780258E64B2DA311A782D3B6FCD987515E870A54465F7
C5302660B6F3A0CF88908EDE72743FBD55741B8D7B9C6BBAE1A9D6F174DC4A9C52EB8129A8D
E69F49A340BA65AF128E935FED44D57906B274905E6297C00358FBEE401742C5812C9BE54517
770F48143B9E8E9EA39617834968AAC4A1494E3D5CF3B69BD5D1ED0392CF84B86C64E4B29CD
CB81C7FEA1ADEA09B9C27D6F31A5DB9EE6A27F1E0914D2180D5A6247A5BE6F2119FA13919DC
696B0C4EA95E5D96F850CB6D32B05F5505F48E2354F4953A2D2897F929D346ECC39F816DBF40
79893333A44AFA473B4A86FF96773295B48BA192FAAFEF79126FE0E944E1B1CE0D09C2BC745F
62C237C3E57EC61209ACF426E8FAB1DB3F8DDDE7AC6374866964178

- Decrypted Final response:

{"payInstrument":{"merchDetails":{"merchId":8952,"merchTxnId":"Test123450","merchTxnDate
":"2021-12-
06T11:40:18"},"payDetails":{"atomTxnId":11000000174925,"prodDetails":[{"prodName":"NSE",
"prodAmount":100.0}],"amount":100.00,"surchargeAmount":1.18,"totalAmount":101.18,"custA

ccNo":"213232323","clientCode":"1234","txnCurrency":"INR","signature":"2b12c8bfc0e3a8268
eddb6f406bf4187d4d0a0064d0355446986511453922c27e38367a97fff85863d48c147a8218e9e
2d5003ab121f6f61ce3914030c60caac","txnInitDate":"2021-12-06
11:40:19","txnCompleteDate":"2021-12-06
11:40:23"},"payModeSpecificData":{"subChannel":["NB"],"bankDetails":{"otsBankId":2001,"ban
kTxnId":"vYmy8VmFzxLF6A7uDx0H","otsBankName":"Atom
Bank"}},"extras":{"udf1":"","udf2":"","udf3":"","udf4":"","udf5":""},"custDetails":{"custEmail":"s
agar.gopale@atomtech.in","custMobile":"8976286911","billingInfo":{}},"responseDetails":{"stat
usCode":"OTS0000","message":"SUCCESS","description":"TRANSACTION IS SUCCESSFUL."}}}

4. Transaction Tracking URL (Requery API):

This API is provided to the merchant to track the status of transaction

Requery API Process:

- Merchant can track the transaction initiated/completed (Success/Fail) by end user via REQUERY API. Merchant will have to send MID provided by ATOM in login parameter and encrypted data pertaining to transaction details of refund whose status they wish to track.
- On initiating the Requery API, merchant will receive the status in encoded response. Merchant should decrypt this response through decryption method.
- Merchant will have to incorporate ATOM-Encryption logic at their end, provided by ATOM. To send encrypted data in request and to decrypt the encrypted response.

Request Format:

Encrypting Request

Parameters in the Request string

Parameters	Description
Login	Merchant ID provided by atom
Encdata	Encrypted data of '&' separated merchantid, merchanttxnid, amt, tdate

Parameters in ENCdata string –

Parameters	Description
merchantid	Merchant ID provided by atom

merchanttxnid	Unique transaction ID sent by the merchant after the first transaction request was initiated
amt	Transaction amount
tdate	Transaction date in YYYY-MM-DD format

To generate ‘encdata’, merchant will have to incorporate AES Logic in there system as below:
To encrypt the request of requery of a transaction, merchanttxnid, amt, tdate must be passed as a single string with ‘&’ separation

example: merchantid=192&merchanttxnid=abc123&amt=100.00&tdate=2018-10-31) along with **Merchant Specific Request Encryption Key** provided by ATOM 5 and same encryption key is to be provided in **Initialization Vector(32bit)** to be, as illustrated below: **DataType**

	Name	**Value**	**Description**
String	Enc	Merchantid=192&merchanttxnid=abc123&amt=100.00&tdate=2018-10-31	Values that need to be encrypted while requesting for transaction status
String	Key	Key provided by Atom, to decrypt the response	Key provided by Atom, to encrypt the request
String	IV	Same as Key	Same as Key string
String	encdata	new.ATOMAES().encrypt(enc,key,IV);	Value of this string is an object. That is used to invoke the encrypt function of ATOMAES class. Post encryption, this variable will be appended in the request along with url, and login.

Requery Sample Request:
https://paynetzuat.atomtech.in/paynetz/vftsv2?login=192&encdata=E005E4B7A8D29E3EBF3DD698CDE161873470C5C27051E840A29E13F8CD50FEEEA809B75A6AF64A53AFB1005E3D89C4F82B7A9F355EE0BB6024A66FA53DE6B217

Response Format

Parameters in the Response string:

Parameters Name	Sample Values	Description
MerchantID	192	Merchant ID provided by Atom.
MerchantTxnID	M123	Unique transaction ID sent by the merchant after the first transaction request was initiated.
AMT	100.00	Transaction amount
Verified	A. Initiated B. SUCCESS C. FAILED D. **Auto Reversal** E. **Force Success** F. Invalid date format G. Invalid Data H. Pending from Bank I. Challan Generated J. Initialized	A. Transaction is initiated, and the end user is on the payment page B. Transaction is successful C. Transaction is failed D. Amount is debited, but transaction is terminated, then the status is implicitly converted to Auto Reversal E. Amount is debited, but transaction is terminated, then the status is implicitly converted to Force Success F. If date format is other than the "yyyy-mm-dd" G. If any of the parameter is not parsed correctly or is missing H. Status is yet to be received from Bank I. When the payment method selected to pay is challan and post successful payment, challan is generated. J. Transaction is posted by merchant to atom end but due to some circumstance payment page is not rendered **Note: Merchant can either opt for "Auto Reversal" or "Force Success"**

StatusCode	A) **010** B) **001** C) **002** D) **003** E) **004** F) **005** G) **006** H) **007** I) **008**	Below Status Code will reflecting respective to the response parsed in verified parameter. **010-** "Initiated" **001- "**Success" **002- "**Failed" **003- "**Invalid Date Format" **004- "**Auto Reversal" **005- "**Force Success" **006- "**Invalid Data" **007-** "Pending from Bank" **008-** "Challan Generated"
Desc	Transaction Success	It's a description pertaining to transaction status
BID	833405033095	Reference No. received from bank.
Bankname	Net Banking - Atom bank Cards (CC and DC) – Atom PG Wallets - Wallet-Airtel Payment Bank	It varies according to mode of payment.
atomtxnId	Transaction Id generated by Atom	Transaction ID generated by atom
Discriminator	DC	Mode of Payment
Surcharge	0.00	Processing Fees charged to End Customer
CardNumber	479932XXXXXX4478	Debit Card/Credit Card number, if mode of payment is via card.
Txndate	2018-11-30 11:00:40	Date of transaction.
UDF9	-	User Defined Parameter,
ReconStatus	RS	Reconciliation status of the transaction. **NRNS** – Not Reconciled Not Settled **RNS** – Reconciled not Settled **RS** – Reconciled & Settled
sdt	2018-12-03	Settlement date of the transaction (in case of ReconStatus "RS").

Sample Encrypted Response Data:

Response to the encrypted request of requery, will be comprising of below illustrated encrypted Data. It needs to be decrypted as per AES Decryption logic provided by the ATOM:

```
AD14BE57CA9A8F520424EBDE00E3493C439E544A8C87FFF7B1DBCC6DF669C32A3A35D2BE9ED4F30DFAD0DB25A5327954A34
5B996490A5366F397C3EDD9311D11003BF1CBEB1AF49CE0C4BE74316E184DB3A0BD9E365955205A1961543DF27D1956D0E8B
FD3475328AE0DEEC626C74B1088D08069A7D7D837AB2D2D62BC0FA8BE37A398277FF1592207AC65C210DDD47524F4CD19D9
4748C1D51BA018C7B704078E7ECC1536CFD7F3E845898F0CC6E3EEE20CB50F5B2FC5514DB7E4018F8742DA18D18D2B11776E
5
14122D4EFFE6BC069B1D694B893AA4C5D4A333BAEA6BAC9AB66F8FBA1459C014A208B0820D1A4A39B99EB9028350F5819EB
0195D19F67AAC179572218376A3BE0F2E32DAD7359745DACC2399D3B2F99830A76BD9F5CAE9C0904D1E891EB8C17E5A042D
C3CBEB0F9070CE250E4311045CE7194699D25BBCE449F08FADF9B4D41E78882691B329A4F8967FFEB164834808D8D17E53207
E7969866AFD667565D748948A51B328483047B47F8EFEC3B8C0B6BC98A048D5DDE1D54BB45B7BAB6C95BC35FD5C39D8B11A
4C8D7B3B9727FD75A7A43231F9C7CF66A030E5D483D543F8E0829DBE546DEB005AC992C5FC1A10F8581CEF94B6F4B7DBA0DF
D7BE4467B10324A8C848400AABF6F2BC155C68B69612D57CDB0FBF60157D9E7A0CD38B12EEB668DEE041DC49A98C6D90424
917A53D29479A13E12B4A2561E7473677C69E038BA0B701682E241549514EAE99AD1B34586D4D058D8B99A9A03ED0DB50E3
A3CC457C0C4D043A88C2673FB05C9205CCBD7AEAC1883C956F785122E7D9D3481932EB82117B37A72950AC5B0F96466FE36C
5C14C0305F8D7C9E96A037BACCDAD7AA12ADB744C35F914C0D14A24562F54ADFE23494961F79A5556581A
```

Decryption of Response:

Merchant will have to pass the encrypted response along with Merchant Specific Response Encryption Key and MID in the decryption method as illustrated below:

decryptor = new AtomAES().decrypt(encryptedResponse, Key, iv);

DataType	**Name**	**Value**	**Description**
String	decstr	BFC23F835C2840C82CCA60671	Encrypted response to the encrypted request triggered, that needs to be decrypted
String	Key	Key provided by Atom, to decrypt the response	Key provided by Atom, to decrypt the response
String	IV	Same as Key	Same as Key string
String	dec	new.ATOMAES().decrypt(decstr,key,IV);	Value of this string, is an object. That is used to invoke the encrypt function of ATOMAES class. Post encryption, this variable will be appended in the request along with url, and login.

• Post decrypting the response successfully merchant will get the below data in JSON format:

```
[
{
"merchantID":"192",
"merchantTxnID":"M123",
"amt":"100.00",
"verified":"Success",
"statusCode":"001",
"desc": “Transaction Success”,
"bid":null,
"bankName":"ATOM PG",
"atomTxnId":"300270",
"discriminator":"DC",
"surcharge":"0.00",
"cardNumber":"189932XXXXXX4578",
"txnDate":"2018-11-30 11:00:40",
"udf9":null,
"reconStatus":"RS",
"sdt":”2018-12-03”
}
]
```

Sample Java Code For Requery API

Encryption

```
package
com.atom.reQueryTest;  import
com.atom.enc.aes.AtomAES;
import
com.atom.ipg.data.security.SecureCardData;
public class RequeryEncryptKey_3
{
public static void main(String[] args) throws Exception
{
String key = "Will be provided";
String str = "merchantid=192&merchanttxnid=M123&amt=100.00&tdate=2018-10-31";
String enc = null;
try
{
enc = new AtomAES().encrypt(str, encryptionrequestKey, IV);
System.out.println("Encrypted String is:: \n" + enc);
}
catch (Exception e)
{
e.printStackTrace();
}
SecureCardData s = new SecureCardData();
}
}
```

Output:

```
A781E72B7D101790A2FD3B0BCCA2360DEE9A87B7B882C5D6C7C4034D4CEE405D4210C99AA88
511629 2A3482C72539A2A6323185CC689B48BD9B98C89F03E342B
```

Above encrypted data is a request, that needs to be append to the Request URL as below:

https://paynetzuat.atomtech.in/paynetz/vftsv2?login=7&encdata=A781E72B7D101790A2FD3B0
BCCA23
60DEE9A87B7B882C5D6C7C4034D4CEE405D4210C99AA885116292A3482C72539A2A6323185C
C689B4
8BD9B98C89F03E342B

Decryption

```
package
com.atom.reQueryTest;  import
com.atom.enc.aes.AtomAES;
import
com.atom.ipg.data.security.SecureCardData;
public class RequeryDecryptKey_4
{
public static void main(String[] args) throws Exception
{

String Key = "Will be
provided";  String decStr=
"encrypted response";  try
{
dec= new AtomAES().decrypt(decstr, encryptionresponseKey, IV );
System.out.println("Decrypted String is:: " + dec);
}
catch (Exception e)
{
e.printStackTrace();
}
SecureCardData s = new SecureCardData();
}
}
```

5. Refund Initiation API

The API is provided to the merchant to raise online refunds on success transactions.
Request should be initiated using HTTP POST method.

Refund Format (UAT):

https://paynetzuat.atomtech.in/paynetz/rfts?merchantid=197&pwd=VGVzdEAxMjM%3D&atomtxnid=100001416131&refundamt=1.00&txndate=2018-07-17&merefundref=test123

Parameters in the Request String:

Parameters	**Description**
merchantid	Merchant ID provided by atom
pwd	**Merchant Password on Base64 with URLEncoded** Encoded Process: Password ⍰⍰Base64 ⍰⍰URLEncoded
atomtxnid	Unique transaction ID sent by the atom transaction completed successfully
refundamt	**Refund Amount** Example: 1200.00 Note: Refund Amount should not be greater than transaction amount.
txndate	**Transaction Done on Date** Format: yyyy-mm-dd Example: 2012-05-04
merefundref	Unique Merchant Refund ID to be shared by merchant.

Response Format

```
<?xml version="1.0" encoding="UTF-8"?>
<REFUND>
<MERCHANTID>197</MERCHANTID>
<TXNID>100001416131</TXNID>
<AMOUNT>10.00</AMOUNT>
<STATUSCODE>01</STATUSCODE>
<STATUSMESSAGE>Partial Refund initiated successfully</STATUSMESSAGE>
<ATOMREFUNDID>4990</ATOMREFUNDID>
</REFUND>
```

Parameters in the Response String: Parameters Description

Parameters	Description
MERCHANTID	Merchant ID provided by atom
TXNID	Unique transaction ID sent by the merchant after the first transaction request was initiated
STATUSCODE	Message Code
STATUSMESSAGE	Success/Failure reason
ATOMREFUNDID	Unique Atom Refund ID generated at the time of successful refund initiation / NA for any status code apart from ‘00’ and ‘01’.

***NOTE**: For below cases ATOMREFUNDID parameter will be populated with ‘NA’:

- ‘merefundref’ parameter is not populated by the merchant at the time of Refund API initiation.
- If the Refund API is initiated with pre-existing/utilized ‘merefundref’.
- For Multi-product transaction refund initiation.

Status Code / Status Message Description Status Code Status Message

Status Code	Status Message
00	Full Refund initiated successfully
01	Partial Refund initiated successfully
M0	Invalid merchant
M1	Refund is not allowed
M2	Invalid Transaction ID
M3	Invalid amount
M4	Insufficient amount in the account to be refund
M5	Invalid transaction date
M6	Failed Transaction
M7	Critical Parameter Missing

Production Domain:

https://payment.atomtech.in/paynetz/rfts?

6. Refund Initiation API (AES Encryption)

Single Product:

URL: (Encrypted URL)

Prod: https://psa.atomtech.in/paynetz/rfts?merchantid=7&encdata=3efsgdfdfhgdhgggfbg

UAT: https://paynetzuat.atomtech.in/paynetz/rfts?merchantid=7&encdata=3efsgdfdfhgdhgggfbg

Request

```
<Script>
Var data =
{
"merchantid":"192",
“pwd”:"1@1234",
“atomtxnid”:”1015”,
“refundamt”:”100.00”,
“txndate”:”2010-10-11”,
“merefundref”:”12345”,
“prodid”:”NSE”

}
</Script>
```

Encrypted Response:

```
981994DBECB7D8205FD686F86CF069D97280C722E1992FF9B4992456401415635CB3569D5E9C
CBD040
E1D006A05DA6844DB8DFDAC5A1BC5DEF0D6C94B6E6E04B9C566BC9A3FDC372805616A7E8A7
BBF6D7
99E97C3E9F34152F51FDAA59635BF8DE2AAFF7BC96CE6A8C9BA3D8C533D45F7132FC32B8791B8C46EB
DB669DCA51EE7906A363C981827FA2A35D8797A8C5C5E00968F912BDCB3F58458C040DD7487DD187C
85FF8579630246AADA013C3DC59B96A6D808E9EE01DA992D265418AF56BA646D4D20BFB4E04
F3AC70
D0511DF0F7C8ED3159DCA1490C3BD0FFFD04513895A85EBCFFD21557D2295F362F44CC9DD93
D17ADA
6281D2F2720660A864AD99140B36E93E3E77D488EA8A90603A232151ECFCCB6578AB89E186C5
01A98F
```

803F52490C7EF3BBD7BE6AFEB271D7E0230F17192F953A3ADB8574962656776BF3987B1839B0
E546FA E4632019671E933202F01

Decrypted Response:

```
[

{

"merchantId":"192",

"TxnId":"1015",

"amount":"100.00",

“prodid”:”NSE”,

"statusCode":"00",

"statusmessage":" Full Refund initiated successfully”,

“atomrefundid”:”456”,

“merefundrefid”:”1023”

}

]
```

Multi Product:

URL:

Prod https://psa.atomtech.in/paynetz/rfts?merchantid=7&encdata=3efsgdfdfhgdhgggfbg

UAT: https://paynetzuat.atomtech.in/paynetz/rfts?merchantid=7&encdata=3efsgdfdfhgdhgggfbg

Request:

```
<Script>
Var data =
{
```

```
"merchantid":"192",
“pwd”:"1@1234",
“atomtxnid”:”1015”,
“refundamt”:”100.00”,
“txndate”:”2010-
10-11”,
“prodid”:”Multi”,
“mprod”:
{
“products”:
{
“product”:
{
“id”:”prod1”,
 “merefundref”:”12345”,
“amount”:”20”
}
“product”:
{
“id”:”prod2”,
“merefundref”:”1234”,
“amount”:”20”
}
“product”:
{
“id”:”prod3”,
“merefundref”:”12345”,
“amount”:”20”
}

“product”:

{
“id”:”prod4”,
“merefundref”:”12346”,
“amount”:”20”
}
“product”:
{
“id”:”prod5”,
“merefundref”:”12345”,
“amount”:”20”
}
}
}
}
</script>
```

Encrypted Response:

```
981994DBECB7D8205FD686F86CF069D97280C722E1992FF9B4992456401415635CB3569D5E9C
CBD040
E1D006A05DA6844DB8DFDAC5A1BC5DEF0D6C94B6E6E04B9C566BC9A3FDC372805616A7E8A7
BBF6D7
99E97C3E9F34152F51FDAA59635BF8DE2AAFF7BC96CE6A8C9BA3D8C533D45F7132FC32B8791B8C46EB
DB669DCA51EE7906A363C981827FA2A35D8797A8C5C5E00968F912BDCB3F58458C040DD7487DD187C
85FF8579630246AADA013C3DC59B96A6D808E9EE01DA992D265418AF56BA646D4D20BFB4E04F3AC70
D0511DF0F7C8ED3159DCA1490C3BD0FFFD04513895A85EBCFFD21557D2295F362F44CC9DD93
D17ADA
6281D2F2720660A864AD99140B36E93E3E77D488EA8A90603A232151ECFCCB6578AB89E186C5
01A98F
803F52490C7EF3BBD7BE6AFEB271D7E0230F17192F953A3ADB8574962656776BF3987B1839B0
E546FA E4632019671E933202F01
```

Decrypted Response:

```
[

{

"merchantId":"192",

"TxnId":"1015",
"amount":"100.00",

“prodid”:”Multi”,
“mprod”:
{
“products”:
{
“product”:
{
“id”:”prod1”,
“merefundid”:”1273”,
“atomrefundid”:”1058”,
“amount”:”20”
}
“product”:
{
“id”:”prod2”,
“merefundid”:”1203”,
“atomrefundid”:”1056”,
“amount”:”20”
}
“product”:
{
“id”:”prod3”,
“merefundid”:”1283”,
“atomrefundid”:”1055”,
“amount”:”20”
}
“product”:
{
“id”:”prod4”,
“merefundid”:”1243”,
“atomrefundid”:”10567”,
“amount”:”20”
}
“product”:
{
“id”:”prod5”,
“merefundid”:”456”, “atomrefundid”:”1059”,
“amount”:”20”
}
}
}

"statusCode":"00",

"statusmessage":" Full Refund initiated successfully”

}

]
```

Status Code

Status Code in bold are the new status that needs to be included in response

Status Code	**Status Message**
M0	Invalid merchant
M1	Refund is not allowed
M2	Invalid Transaction ID or Failed Transaction
M3	Invalid amount
M4	Insufficient amount in the account to be refund
M5	Invalid transaction date
M6	Failed Transaction
M7	Critical Parameter Missing
P0	**Full Refund for Product (Product ID) initiated successfully**
P0	**Partial Refund for Product (Product ID) initiated successfully**
P1	**Full Refund for Product (Product ID) and Partial Refund for Product (Product Id) initiated successfully**
P2	**Prod (Prod Id) is not valid**
P3	**Prod (Prod Id) is missing**
P4	**Refund Amount for Prod (Prod Id) is greater than transaction amount**
P5	**Refund Amount for Prod (Prod id) is missing**
P6	**Refund Amount is Invalid for Prod:**
P7	**Duplicate refund reference number for Product:**
P9	**Please send this products as this a multi-product transaction.**
P10	**Insufficient amount in the account to be refund for Product :**

Description of Status Code

Status Code	**Description**
M0	MID is not valid
M1	Merchant is Broker or transaction pertains to Challan [CH], Wallet [MW], NEFT/RTGS [NR]
M2	Transaction ID is not Valid
M3	Refund amount is greater than transaction amount

M4	Insufficient amount in the account to be refund.
M5	Invalid transaction date
M6	Failed Transaction
M7	Merchant Id, PWD, Atomtxnid, txndate, ProdId (Parent) is missing. Also, ProdId (Parent) is invalid
P0	**Refund details for all Product (Product ID) is displayed**
P0	**Refund details for all Product (Product ID) is displayed**
P1	**Refund details for all Product (Product ID) is displayed, Full and Partial, respectively.**
P2	**Child Prod (Prod Id) is not valid**
P3	**Child Prod (Prod Id) is missing**
P4	**Child refund amount for Prod (Prod Id) is greater than transaction amount**
P5	**Child refund amount for Prod (Prod id) is missing**
P6	**Child refund amount for Prod is invalid**
P7	**Name for Prod (Prod Id) is missing**
P9	**Child Product information is missing**
P10	**Child Insufficient amount in the account to be refund.**

Refund blocked for Transaction/Merchant type:

1. Challan Transaction Discriminator: CH

2. NEFT/RTGS Transaction
   Discriminator: NR

3. Wallet Transaction: Discriminator: MW

4. Member/Broker Merchant Type

7. Refund Status API:

This API is provided to the merchant to track the status of online refunds raised against their successful transactions.

Refund Status API Process

- Merchant will have to setup a system at their end to incorporate ATOM-Encryption logic, to send encrypted data in request and to decrypt the response.
-

- Merchant can track their refunds initiated on successful transactions via REFUND STATUS API provided below, merchant will have to send MID provided by ATOM in login parameter and encrypted data pertaining to transaction details of refund whose status they wish to track. This is further explained under Request Format section in page number '4'.
- On initiating the REFUND STATUS API, merchant will receive the status in encoded response as further explained under Response Format section in page number '6'.

Merchant should decrypt this response via the decryption method in the shared ATOM-Encryption logic.

- Request Format

https://paynetzuat.atomtech.in/refundstatus/rsfts?login=197&encdata=EBB925BEA9EF082570
55AA9
13D0149A2D29EE705892C0D7D88DB963E8EBCA297DF7E4AE96DC78B57E08CE2DE0759876D

Parameters in the Request string

Parameters	Description
encdata	Encrypted data of Atom transaction ID, Merchant ID, Product ID., Merchant specific key and Merchant ID
login	Merchant ID provided by atom

- **'encdata'**:
  - ✦ User must encrypt the Merchant ID and Product ID using the ATOM-Encryption Java file, which will be shared separately at the time of integration.
  - ✦ To generate this 'encdata' the merchant will have to configure their system in the following manner:
    - Atom transaction ID of the transaction whose refund status is to be found out, merchant ID provided by ATOM and their Product Id must passed as a single string with pipe-separation (example: txnid=100001470088&merchantid=197&product=NSE) along with Merchant Specific Request Encryption key provided by ATOM and merchant id provided by ATOM to the encryption method, as illustrated below: **encdata= new.AtomAES().encrypt(enc, merchant specific key,mid);**

Example:

```
 String enc = “100001470088&197&NSE”;
 String reqEncriptionKey = “Will be provided at the time of integration”;
                               String  iv=  “8E41C78439831010F81F61C344B7BFC7”;    //Key
                               encdata= new.AtomAES().encrypt(enc, merchant specific key,
                               iv);
```

Response Format:

Encrypted Data:

```
    AD14BE57CA9A8F520424EBDE00E3493C439E544A8C87FFF7B1DBCC6DF669C32A3A35D2BE
    9ED4F3
    0DFAD0DB25A5327954A345B996490A5366F397C3EDD9311D11003BF1CBEB1AF49CE0C4B
    E74316E
    184DB3A0BD9E365955205A1961543DF27D1956D0E8BFD3475328AE0DEEC626C74B1088D
    08069A
    7D7D837AB2D2D62BC0FA8BE37A398277FF1592207AC65C210DDD47524F4CD19D94748C1
    D51BA0
    18C7B704078E7ECC1536CFD7F3E845898F0CC6E3EEE20CB50F5B2FC5514DB7E4018F8742D
    A18D18
    D2B11776E514122D4EFFE6BC069B1D694B893AA4C5D4A333BAEA6BAC9AB66F8FBA1459C
    014A20
    8B0820D1A4A39B99EB9028350F5819EB0195D19F67AAC179572218376A3BE0F2E32DAD73
    59745D
    ACC2399D3B2F99830A76BD9F5CAE9C0904D1E891EB8C17E5A042DC3CBEB0F9070CE250E4
    311045
    CE7194699D25BBCE449F08FADF9B4D41E78882691B329A4F8967FFEB164834808D8D17E5
    3207E79
    69866AFD667565D748948A51B328483047B47F8EFEC3B8C0B6BC98A048D5DDE1D54BB45
    B7BAB6
    C95BC35FD5C39D8B11A4C8D7B3B9727FD75A7A43231F9C7CF66A030E5D483D543F8E0829
    DBE54
    6DEB005AC992C5FC1A10F8581CEF94B6F4B7DBA0DFD7BE4467B10324A8C848400AABF6F2
    BC155C
    68B69612D57CDB0FBF60157D9E7A0CD38B12EEB668DEE041DC49A98C6D90424917A53D2
    9479A1
  3E12B4A2561E7473677C69E038BA0B701682E241549514EAE99AD1B34586D4D058D8B99A9A03ED
  0DB50E3A3CC457C0C4D043A88C2673FB05C9205CCBD7AEAC1883C956F785122E7D9D3481932EB8
  2117B37A72950AC5B09C8A5B4D15F89C16E5883DE1421EBA61A245ED003318DF7961D5B1E04A56
  1E45C9BEF6326BC55C2BA125534E974E84FB6C2A6DE2B74A480C602F30E04C820A6AA468FE78F2C
    E044E57DAA49FEF369BF481E71EC94181D07787E6CC1824C297612CA4C136205541A7B62
    C4A45C
```

```
4AF4EC0274E39FD40DD57A2DEB17EDBBF3F7EAA062FE8EA55FF57414FD28612BA90698C03061DD
6D95A2A10CA0895306D7459C465ADC2B1911A7C72A1E29D1800331FCB63462E82C352E298737F9
  C5DD8A142990A04F0DE5126A26193188BC90F0230C26B722BA01FDB727017F12B3F079C9
  2EA5F6
  0448276D546DE651F7BFE2434990981AE2002E4F652CF0B3DCA73932B4418CFE1098809DF
  7C88B5
  BBBF3B008C40A0E8CDDF5D747C25618941998D2ED4A54E
```

Decryption of Response:

- The response will be received by merchant in an encrypted format as illustrated above. This data must be decrypted using the same ATOM-Encryption Java file.
- Merchant will have to pass the encrypted response along with Merchant Specific Response Encryption Key and MID in the decryption method as illustrated below:
  decryptor = new AtomAES().decrypt(encryptedResponse, merchantSpecificKey, mid);

  Example:

  String encryptedResponse=” AD14BE57CA…”;
  String resEncriptionKey =” Will be provided at the time of
  integration”; String iv= “8E41C78439831010F81F61C344B7BFC7”;
  //Key

  decryptor = new AtomAES().decrypt(encryptedResponse, merchantSpecificKey, iv);

- Post decrypting the response successfully merchant will get the below data in XML format:

```
<?xml version="1.0"?>
<REFUNDSTATUS>
<ERRORCODE>00</ERRORCODE>
<MESSAGE>Refund Found</MESSAGE>
<DETAILS><REFUND>
<TXNID>100001470088</TXNID>
<PRODUCT>NSE</PRODUCT>
<REFUNDAMOUNT>1.0000</REFUNDAMOUNT>
<REFUNDINITIATEDATE>2018-07-26</REFUNDINITIATEDATE>
<REFUNDPROCESSDATE/>
<REMARKS/>
<MEREFUNDREF>test1</MEREFUNDREF>
</REFUND>
<REFUND><
TXNID>100001470088</TXNID>
<PRODUCT>NSE</PRODUCT>
<REFUNDAMOUNT>10.0000</REFUNDAMOUNT>
<REFUNDINITIATEDATE>2018-07-26</REFUNDINITIATEDATE><REFUNDPROCESSDATE/>
<REMARKS/>
```

```
<MEREFUNDREF>test2</MEREFUNDREF>
</REFUND>
<REFUND><TXNID>100001470088</TXNID>
<PRODUCT>NSE</PRODUCT>
<REFUNDAMOUNT>10.0000</REFUNDAMOUNT>
<REFUNDINITIATEDATE>2018-08-03</REFUNDINITIATEDATE>
<REFUNDPROCESSDATE/>
<REMARKS/>
<MEREFUNDREF>test3</MEREFUNDREF>
</REFUND></DETAILS>
</REFUNDSTATUS>
```

Parameters in the Response string:

Parameters	Description
ERRORCODE	Reflects the status of the refund initiation. Refer below 'error code' table.
MESSAGE	Refund Found / No Records Found for MerchantId, TransactionId, ProductId.
TxnID	Unique transaction ID sent by the merchant after the first transaction request was initiated.
PRODUCT	Product ID as decided and approved by the Merchant.
REFUNDAMOUNT	Refund amount initiated.
REFUNDINITIATEDATE	Refund initiation Date in YYYY-MM-DD format.
REFUNDPROCESSDATE	Refund process Date in YYYY-MM-DD format. Date when the refund process is completed.
MEREFUNDREF	Unique Merchant Refund ID to be shared by merchant, while initiating the refund via Refund initialization API.
REMARK	**Refund Closing Remarks**

NOTE:

1. **< DETAILS>** tag will only be visible when **error code** returns '**00**'.
2. **<MEREFUNDREF>** tag will contain Unique Merchant Refund ID to be shared by merchant at the time of refund initialization. Incase merchant has not shared/populated this field during refund initialization, this tag will contain no value.

Status Code / Status Message Description

Status Code	Status Message
00	Full Refund initiated successfully
01	Partial Refund initiated successfully
EE	Transaction not found / Information missing. Error details will be presented in message tag.

Production URL: https://payment.atomtech.in/refundstatus/rsfts?

8. Callback API

Introduction:

- This document will provide the overview of Callback API process flow.
- Callback API is a notification API used to post transaction response from server to server in name value pair format on configured merchant's URL in atom system.
- Success and Failed Transaction response will be posted on real-time to merchant callback URL.
- Post reconciliation Success response will be posted for Force Success transactions.
- Callback API will also post, change in the status in case of time-out transaction to Success/Failed.

General Process Flow:

1. The end customer goes on Merchant website and selects item to buy.
2. When end customer clicks on "buy" button he is redirected to atoms payment gateway page.
3. The end customer selects the "Payment option".
4. The end customer enters appropriate credentials for selected "Payment option" and completes the payment.
5. The IP and domain URL need to be white listed and configured.
6. If end customer completes payment successfully and if call back URL is configured for that merchant, then transaction response will be posted on ru (Return URL) as well as call back URL in both scenarios i.e. Success / Failure.
7. On Call back URL response is sent in name value pair format. Reposting will be done maximum 3 times in 5 seconds interval.

Pre – requisite:

Merchant's IP and domain URL need to be white listed and configured at atom system.

Callback Posting Parameters –

Parameter Name	Description	Sample Value
MerchantID	Merchant Identification Code provided by Atom	160
MerchantTxnID	Merchant transaction ID generated by Merchant	123456
AMT	Transaction Amount	50.00
VERIFIED	Transaction Status	SUCCESS / FAILED
BID	Bank reference number	2125458
BankName	Bank on which customer performed transaction. In case of credit/debit card transactions, bank name will be received as Atom PG.	ATOMTEST
AtomTxnId	Atom transaction ID. The ID is generated by atom	281788
Discriminator	Payment channel	NB
Surcharge	Amount of Surcharge	0.00
CardNumber	Masked card number for Credit card / Debit card transaction or blank in case of NB / NEFT	401288XXXXXX1881
TxnDate	Transaction Date	2017-10-27 10:02:07
CustomerAccNo	Customer account number (Same as per transaction request)	00000034503902289
Clientcode	Customer identification code. In Callback response Atom will post same Client code passed by the merchant in transaction request.	ESB07874

Sample Response Format:

Scenario 1 –

If transaction get success at atom end, on real time atom will post “**VERIFIED=SUCCESS**” response with help of Callback API.

Callback Response –

MerchantID=160&MerchantTxnID=123456&AMT=50.00&**VERIFIED=SUCCESS**&BID=2125458

&BankName=ATOMTEST&AtomTxnId=281788&Discriminator=NB&Surcharge=0.00&Ca
rdNum ber=401288XXXXXX1881&TxnDate=2017-10-27
10:02:07&CustomerAccNo=00000034503902289&Clientcode=ESB07874

Scenario 2 –

If on T day transaction is pending at atom end, after bank re-query response on same day if transaction is **success** at bank end then status will be changed from **pending** to **success** at atom end, then atom will post “**VERIFIED=SUCCESS**” response on Callback API.

Callback Response –

MerchantID=160&MerchantTxnID=123456&AMT=50.00&**VERIFIED=SUCCESS**&BID=21
25458
&BankName=ATOMTEST&AtomTxnId=281788&Discriminator=NB&Surcharge=0.00&Ca
rdNum ber=401288XXXXXX1881&TxnDate=2017-10-27
10:10:25&CustomerAccNo=00000034503902289&Clientcode=ESB07874

Scenario 3 –

If on T day transaction is pending at atom end, on T+1-day post reconciliation if transaction is success then status gets changed from **pending** to **success** at atom end, then atom will post
“**VERIFIED=SUCCESS**” (Force success) response with help of Callback API. If it’s Failed response will be “**VERIFIED=FAILED**”.

Callback Response –

MerchantID=160&MerchantTxnID=123456&AMT=50.00&**VERIFIED=SUCCESS**&BID=21
25458
&BankName=ATOMTEST&AtomTxnId=281788&Discriminator=NB&Surcharge=0.00&Ca
rdNum ber=401288XXXXXX1881&TxnDate=2017-10-28
10:10:25&CustomerAccNo=00000034503902289&Clientcode=ESB07874

Callback Response –

MerchantID=160&MerchantTxnID=123456&AMT=50.00&**VERIFIED=FAILED**&BID=2125
458&B
ankName=ATOMTEST&AtomTxnId=281788&Discriminator=NB&Surcharge=0.00&Card
Number
=401288XXXXXX1881&TxnDate=2017-10-28
10:10:25&CustomerAccNo=00000034503902289&Clientcode=ESB07874

Scenario 4 –

If on T day transaction getting failed at atom end, on real time atom will post "**VERIFIED=FAILED**" response with help of Callback API

Callback Response –

MerchantID=160&MerchantTxnID=123456&AMT=50.00&**VERIFIED=FAILED**&BID=2125
458&B
ankName=ATOMTEST&AtomTxnId=281788&Discriminator=NB&Surcharge=0.00&Card
Number
=401288XXXXXX1881&TxnDate=2017-10-27
10:02:07&CustomerAccNo=00000034503902289&Clientcode=ESB07874

Scenario 5 –

Until and Unless we are not getting response (Success/Failed) from bank end, Atom will not trigger Callback API. Till T+2 if we're not getting response from Bank, Atom system will mark transaction as Fail, but Callback will not be triggered. If transaction status is received

as success from Bank's end after T+2, irrespective to the configuration [Force Success/Auto Reversal] in Atom system, same will be marked as AutoReversal.

Scenario 6 –

For **NEFT** Transactions Challan Generated on T day, on (T+1, T+2, T+3….) day post reconciliation transaction status gets changed from **challan generated** to **success** at atom end, then atom will post "**VERIFIED=SUCCESS**" response with help of Callback API. In This scenario, no real time response will be posted.

Callback Response –

MerchantID=160&MerchantTxnID=123456&AMT=50.00&**VERIFIED=SUCCESS**&BID=Z
ATOMZ
300053108941&BankName=IndusInd(NEFT)&AtomTxnId=300046083167&Discriminator
=NR&
Surcharge=0.00&CardNumber=&TxnDate=2019-06-18
12:31:19&CustomerAccNo=1236547&Clientcode=REGISTRATION/LOC 2018-19
atomTxnId 300046083167

Note*:

1. Atom will try to reach to merchant's server for 3 time.
2. If Atom receives the Status Code as "200", it depicts that the server is connected.

9. Settlement API (Without AES)

Settlement report: (In .csv format)-

1) The API will only support POST method

2) For the API merchant need to share their ip for whitelisting in our system. API allows only white listed ips to get the required report.

3) The parameters are “merchantId” and “settlementDate” (date should be in format YYYY-MMDD e.g. 2015-05-26).

4) If there are records in for the merchant id for the settlement date the Settlement File is generated.

5) In other cases, or error cases merchant will get response as csv file always.

UAT Testing Credentials:

UAT:

URL=

http://pgtest.atomtech.in:8443/SettlementReport/generateRepo

rt merchantId = 7760 settlementDate = 2019-03-25

transactionId = 400036128272 responseType = json;

Production Credentials:

Production:

URL=https://pgreports.atomtech.in/SettlementReport/generateReport

merchantId = Merchant Production MID

settlementDate = Settlement Date

transactionId = Transaction Id

responseType = json;

Sample Response –

.CSV

JSON format [with CC/DC])-

1) The API will only support POST method
2) For the API merchant need to share their ip for whitelisting in our system. API allows only white listed ips to get the required report.
3) The parameters are “merchantId”, “settlementDate” (date should be in format YYYYMM-DD e.g. 2015-05-26) and responseType : json.
4) If there are records in for the merchant id for the settlement date the Settlement File is generated.

5) In other cases or error cases merchant will get response as .JSON format always.

UAT Testing Credentials:

UAT:

URL=

http://pgtest.atomtech.in:8443/SettlementReport/generateRepo

rt merchantId = 7760 settlementDate = 2019-03-25

transactionId = 400036128272 responseType = json;

Production Credentials:

Production:

URL=https://pgreports.atomtech.in/SettlementReport/generateReport

merchantId = Merchant Production MID

settlementDate = Settlement Date

transactionId = Transaction Id

responseType = json;

Sample Response (Settlement Date wise)–

[

```
[
  {
    "srNo": "1",
    "merchantName": "BOB_DIP",
    "merchantID": "7760",
    "atomTxnID": "400036128272",
    "txnState": "Sale",
    "txnDate": "20-Mar-2019 13:23:31",
    "clientCode": "4539",
    "merchantTxnID": "47642867",
    "product": "AXISEP_SAND",
    "discriminator": "DC",
    "bankCardName": "",
    "cardType": "INTERNATIONAL",
    "cardNo": "652154XXXXXX6986",
    "cardIssuingBank": "INTERNATIONAL",
    "bankRefNo": "907918271417",
    "refundRefNo": "",
    "grossTxnAmount": "2796.60",
    "txnCharges": "0.0",
    "serviceTax": "0.0",
    "sbCess": "0.0",
    "krishiKalyanCess": "0.0",
    "totalChargeable": "0.0",
    "netAmountToBePaid": "2796.6",
    "paymentStatus": "SALE",
    "settlementDate": "25-Mar-2019 18:30:00",
    "refundStatus": "Not refunded"
  },
  {
    "srNo": "2",
    "merchantName": "BOB_DIP",
    "merchantID": "7760",
    "atomTxnID": "400036133790",
    "txnState": "Sale",
    "txnDate": "20-Mar-2019 14:21:52",
    "clientCode": "4539",
    "merchantTxnID": "47644324",
    "product": "AXISEP_SAND",
    "discriminator": "DC",
    "bankCardName": "",
    "cardType": "DOMESTIC",
    "cardNo": "652166XXXXXX7565",
    "cardIssuingBank": "HDFC",
    "bankRefNo": "907919271778",
    "refundRefNo": "",
    "grossTxnAmount": "15625.30",
    "txnCharges": "0.0",
    "serviceTax": "0.0",
    "sbCess": "0.0",
    "krishiKalyanCess": "0.0",
    "totalChargeable": "0.0",
    "netAmountToBePaid": "15625.3",
    "paymentStatus": "SALE",
    "settlementDate": "25-Mar-2019 18:30:00",
    "refundStatus": "Not refunded"
  }
]
```

Sample Response (Transaction Id wise)–

```
[
[
  {
    "srNo": "1",
    "merchantName": "BOB_DIP",
    "merchantID": "7760",
    "atomTxnID": "400036128272",
    "txnState": "Sale",
    "txnDate": "20-Mar-2019 13:23:31",
    "clientCode": "4539",
    "merchantTxnID": "47642867",
    "product": "AXISEP_SAND",
    "discriminator": "DC",
    "bankCardName": "",
    "cardType": "INTERNATIONAL",
    "cardNo": "652154XXXXXX6986",
    "cardIssuingBank": "INTERNATIONAL",
    "bankRefNo": "907918271417",
    "refundRefNo": "",
    "grossTxnAmount": "2796.60",
    "txnCharges": "0.0",
    "serviceTax": "0.0",
    "sbCess": "0.0",

"krishiKalyanCess": "0.0",
    "totalChargeable": "0.0",
    "netAmountToBePaid": "2796.6",
    "paymentStatus": "SALE",
    "settlementDate": "25-                       Mar-2019 18:30:00",
    "refundStatus": "Not refunded"
  }
]
```

JSON format [without CC/DC])-

1. The API will only support POST method
2. For the API merchant need to share their ip for whitelisting in our system. API allows only white listed ips to get the required report.
3. The parameters are “merchantId”, “settlementDate” (date should be in format YYYYMM-DD e.g. 2015-05-26) and responseType : json.
4. If there are records in for the merchant id for the settlement date the Settlement File is generated.
5. In other cases or error cases merchant will get response as .JSON format always.

UAT Testing Credentials:

UAT:

URL=
http://pgtest.atomtech.in:8443/SettlementReport/generateReport merchantId = 7760 settlementDate = 2019-03-25
transactionId = 400036128272 responseType = json;

Production Credentials:

Production:

URL=https://pgreports.atomtech.in/SettlementReport/generateReport merchantId = Merchant Production MID
settlementDate = Settlement Date transactionId =
Transaction Id responseType = json;

Sample Response (Settlement Date wise)–

```
[
{
    "srNo": "9",
    "merchantName":                              "BOB_DIP",
    "merchantID": "7760",
    "atomTxnID": "300036240200",
    "txnState": "Sale",
    "txnDate": "22-Mar-2019 04:52:17",
    "clientCode": "4539",
    "merchantTxnID": "47667147",
    "product": "AXISEP_SAND",
    "discriminator": "NB",
    "bankCardName": "Corporation Bank",
    "bankRefNo": "5593092",
    "refundRefNo": "",
    "grossTxnAmount": "3179.02",
    "txnCharges": "0.0",
    "serviceTax": "0.0",
    "sbCess": "0.0",
    "krishiKalyanCess": "0.0",
    "totalChargeable": "0.0",
    "netAmountToBePaid": "3179.02",
    "paymentStatus": "SALE",
    "settlementDate": "25-Mar-2019 00:00:00",
    "refundStatus": "Not refunded"
  }
]
```

Sample Response (Transaction Id wise)–

```
[
{
    "srNo": "9",
    "merchantName": "BOB_DIP",
    "merchantID": "7760",
    "atomTxnID": "300036240200",
    "txnState": "Sale",
    "txnDate": "22-Mar-2019 04:52:17",
    "clientCode": "4539",
    "merchantTxnID": "47667147",
    "product": "AXISEP_SAND",
    "discriminator": "NB",
    "bankCardName": "Corporation Bank",
    "bankRefNo": "5593092",
    "refundRefNo": "",
    "grossTxnAmount": "3179.02",
    "txnCharges": "0.0",
    "serviceTax": "0.0",
    "sbCess": "0.0",
    "krishiKalyanCess": "0.0",
    "totalChargeable": "0.0",
    "netAmountToBePaid": "3179.02",
    "paymentStatus": "SALE",
    "settlementDate": "25-Mar-2019 00:00:00",
    "refundStatus": "Not refunded"
  }
]
```

Status Code:

Status Code	Status Description
"00"	Mandatory parameter value cant be null.
S0	Invalid merchantId or merchant is not present.
S6	Invalid ip address or ip is not configured
S3	Invalid settlement date. Date should be in Format(yyyy-MM-dd)
S1	Invalid Response Type
S2	Invalid transactionId
S7	No data found

10. Settlement API (With AES)

Settlement report: (In .csv format)-

1) The API will only support POST method

2) For the API merchant need to share their ip for whitelisting in our system. API allows only white listed ips to get the required report.

3) The parameters are “merchantId” and “settlementDate” (date should be in format YYYY-MMDD e.g. 2015-05-26).

4) If there are records in for the merchant id for the settlement date the Settlement File is generated.

5) In other cases, or error cases merchant will get response as csv file always.

UAT Testing Credentials:

UAT:

Enc URL= http://pgtest.atomtech.in:8443/SettlementReport/generateReport?merchantId=7760&encdata=F1F7E46B85600F3CBB86E848B8F80149FF27D16E2D3B5B002081007CF73F90D15CEC7C147057045DCBCDDCF80A3661AC9460947D5940EB843C1CB31BC686F9C9C4670FD823D9F6DB062097FDB8F5337CF3B348D85BECB776CD77B4262D6D2D7C

Note*: For Open request refer NON AES Section

Production Credentials:

Production:

Enc URL= https://pgreports.atomtech.in/SettlementReport/generateReport?&encdata=F1F7E46B85600F3CBB86E848B8F80149FF27D16E2D3B5B002081007CF73F90D15CEC7C147057045DCBCDDCF80A3661ACF377EDB0545BE671F9F8192E1813D697FF9FA11D5D5DFE2F85A3B192E1C11262

Open Request

merchantId=1191|settlementDate=2020-08-04|transactionId=|responseType=json

Sample Response –

.CSV

JSON Format [with CC/DC])-

1) The API will only support POST method

2) For the API merchant need to share their ip for whitelisting in our system. API allows only white listed ips to get the required report.
3) The parameters are “merchantId”, “settlementDate” (date should be in format YYYYMM-DD e.g. 2015-05-26) and responseType : json.

4) If there are records in for the merchant id for the settlement date the Settlement File is generated.
5) In other cases or error cases merchant will get response as .JSON format always.

UAT Testing Credentials:

UAT:

URL=

http://pgtest.atomtech.in:8443/SettlementReport/generateRepo

rt merchantId = 7760 settlementDate = 2019-03-25

transactionId = 400036128272 responseType = json;

Production Credentials:

Production:

URL=https://pgreports.atomtech.in/SettlementReport/generat

eReport merchantId = Merchant Production MID

settlementDate = Settlement Date transactionId =

Transaction Id responseType = json;

Sample Response (Settlement Date wise)–

```
[
[
  {
    "srNo": "1",
    "merchantName": "BOB_DIP",
    "merchantID": "7760",
    "atomTxnID": "400036128272",
    "txnState": "Sale",
    "txnDate": "20-Mar-2019 13:23:31",
    "clientCode": "4539",
    "merchantTxnID": "47642867",
    "product": "AXISEP_SAND",
    "discriminator": "DC",
    "bankCardName": "",
    "cardType": "INTERNATIONAL",

 "cardNo": "652154XXXXXX6986",
    "cardIssuingBank":                              "INTERNATIONAL",
    "bankRefNo": "907918271417",
    "refundRefNo": "",
    "grossTxnAmount": "2796.60",
    "txnCharges": "0.0",
    "serviceTax": "0.0",
    "sbCess": "0.0",
    "krishiKalyanCess": "0.0",
    "totalChargeable": "0.0",
    "netAmountToBePaid": "2796.6",
    "paymentStatus": "SALE",
    "settlementDate": "25-Mar-2019 18:30:00",
    "refundStatus": "Not refunded"
  },
  {
    "srNo": "2",
    "merchantName": "BOB_DIP",
    "merchantID": "7760",
    "atomTxnID": "400036133790",
    "txnState": "Sale",
    "txnDate": "20-Mar-2019 14:21:52",
    "clientCode": "4539",
    "merchantTxnID": "47644324",
    "product": "AXISEP_SAND",
    "discriminator": "DC",
    "bankCardName": "",
    "cardType": "DOMESTIC",
    "cardNo": "652166XXXXXX7565",
    "cardIssuingBank": "HDFC",
    "bankRefNo": "907919271778",
    "refundRefNo": "",
    "grossTxnAmount": "15625.30",
    "txnCharges": "0.0",
    "serviceTax": "0.0",
    "sbCess": "0.0",
    "krishiKalyanCess": "0.0",
    "totalChargeable": "0.0",
    "netAmountToBePaid": "15625.3",
    "paymentStatus": "SALE",
    "settlementDate": "25-Mar-2019 18:30:00",
    "refundStatus": "Not refunded"
  }
]
```

Enc Response:

18642F36C9661C57F6518466E8E9539BB5F48B31D83BF0F632AC9A80D3532B1C8B
12D17
30A1A53329157828C84256D366654791D30553C71718A17E004C3CB85D3C605FCE0
B6B59

```
4CBA67D53DCEBC05896299468ABFEC540C52FF3D4A31361EBE6CE467F438D5044
14B8F
66C11731141E6BA419C9175F5C5F0C6BA8CBBA589B78BFF61E70CB06B756CECCF
B13F3
9E034A60342B82FFB7CA657BEBBFF5F8DA7BAA02AD8DE79085BDBE1A4515E1EA
389304
680CA62131A526B3B9E8234ED10836DF067348F407DA03778F76E5D751BC26563B
514B6
D0206B2959208419F84C1ED88DC81A6E4CEA71507722D59597D962B83E6129BD71
97D98
AE778E298326B854183014A10A2ECDE46E35E7E676E5240D7D5B0663254D6661B9
33D6F
EFDC43DD8A2204996B67785836CF6E98502545A1CFE6B4007E1E5B22832D50D234
C9200
7E08D2417785AD2AE66594F68F8B9E469F4020037AA60AB50C7704F3D545039ECC
1AD95
CC6133D9BB9EC271E85723E4D96598B42551CA09897764B3EAC43CB633B09D6ED
29218
3CB2EBDCA18A3A313594B18FEC3A949613EE3145C40037EF58C666D3455567BFC
FAA47
CAEA5182212F59D103EC6B4B30CABBEDC6EE4728ADE14A34D52AD266BD2C25A
A95193
23D2004EA76DDA11E7DBB023AC18AD82CC47F78F64CA3B82CC24B9A07C1F6283
0A9DA
CFCE410AD5D3F44F995035B947C7CBA66B88D77847C5F92487E3BD5699157CD1B
A0406
81F84818BB57A782F89F7FAE855C9CDC5A21712C397A4D69DA17AF61E4A35BCFC
D2489
E3065ACBB42125FFE36DFA65087FEB67C0507B0D52509FCC1AF56AFD58C789AC4
AC6B4
32AA7EA135FA9FB29CAA6485DF1AADF1929373A79542BED43618ECDC92543F9A7
F8529
6BCF15613215A85160FDD7F9D05687618C92D6D232CC148F3C27CBB78BB17F1E5
277CB
B77DCAC1A7A1988950C34112B293099E604BB7340B4D54690821CF33D5BF51C9A7
84589
1C11A2B73137296B07D0D4FA5737A78819BAD6B41407049F0734DC3A0AA931D04B
642AF
E7DE66247B8679DC185A6320EC674528A9CB85637E1BED60768AE5E7063EF1522F
50FA5
FF077BDA5B0DCFB866050D8324152C6C9FF5C671218127F1D97A998B2E4F7C20D
ED73F
9650865CE596C190B37348E73CFFA5407A1CF99058A5C6C28157D37B3E2FD9F2DB
8564A
```

```
0F0AD044A80B5A629E5108CBD89133EB7CE8E57F0367800DF61CCFB4B6E863691
67FF1A
DFB605358A12FD6F5E92F91FFAF22FDCD79B0B2D799CAB220B0B18617E54B9878
E2D6A
522B1E04DF2D863175D89F19252DBD46BF3AA45F4BB9D0CA6676774A2520C25B32
13DE5
462DF689D91B0F32C33DFB416B3C3CBA448EF896716276120267B4245ED5A26B50
F1607
FE138F5568BC9B24EB2073F0628E7A65D34FFFFA56C5F00BE35645367F068F6A82
9073E3
35CFC02F9EB691C6550F4D222F0ABC25A220C793C46CD442CF787DEA8C50B8203
7AA8C
1ECBB9CB10D69A85468E9B7C817C4248395D8579556C3360B4FBEF1D4513A10D4
0FF142
274A2CCE857D5792B8598B0D7D941FC195C2BAEF0962E99CE9FC1E1AF461C58E2
450CC
2FEE283D400176288D25E927A876BC41E958EB438B89DD7329BAF96ED0DEDFE43
E429F
85D7D84F49E5F22E27F13CED7682E9C1E0F17069A45DD71F4BE200774AB21C699
DC8024
C6410A398E404E2FEF1140100039C014705A3A441E4503A86284B84B9B3D62F717F
40A6C
69EDCD893F4A6A5260D9B2CA86A075640431976DBC46C3C652FB549BC9F38EA81
76CFC
FC5F242060BBD9F2E8039E1786DE2AE1206246A8EA1B1F5D86F6BB2602D88E2772
67CB7
65B3478C23A594E257F6C65E7B3E5BE85A01882150B34C233CB3DCFE3239906AE8
51E6C
D4A0D56481873230FF820DF7D087D82F901C8416E66A05C681D40319D8A00238BF
B409C
DC5026F2FF11486B368FE856558ACC8661C96E7833F39E9BC1D9901F6261E97256
5C4E5
1D0C8B8C0167715FA387AE004398F14907612425A6156CFD9D514C0A92943AA3EF
2A2A2
45EE22CA43D3BB69C002BB1D39C10ACF8A72080298403328976C776EFC8A78CCF
3719C
272577D3EE3F05D1701F9F18EC971CE7A6D024B17E8AE19D65B36ADE7341D2264
1D62F7
72EECC17EF8F98A37D68903E717AB32CB0429E2D4E09F03BB071D1397C4F93EA3
BE5E9
977E8733ADF2DFA2C9EE2EA3D946D6A3398B4BD793E43B04CE510A7A7A3FAB043
65B89
7FAA6677824FDE0EB89A05820E61D685B1DCCE5003F13D128BEAC16808F599BFE
E02AE
4BE429CB7F1843C2CDAD7F9D62CB2DD3DAD8B2B7E6B63D0818D
```

Sample Response (Transaction Id wise)–

```
[
[
  {
    "srNo": "1",
    "merchantName": "BOB_DIP",
    "merchantID": "7760",
    "atomTxnID": "400036128272",
    "txnState": "Sale",
    "txnDate": "20-Mar-2019 13:23:31",
    "clientCode": "4539",
    "merchantTxnID": "47642867",
    "product": "AXISEP_SAND",
    "discriminator": "DC",
    "bankCardName": "",
    "cardType": "INTERNATIONAL",
    "cardNo": "652154XXXXXX6986",
    "cardIssuingBank": "INTERNATIONAL",
    "bankRefNo": "907918271417",
    "refundRefNo": "",
    "grossTxnAmount": "2796.60",
    "txnCharges": "0.0",
    "serviceTax": "0.0",
    "sbCess": "0.0",
    "krishiKalyanCess": "0.0",
    "totalChargeable": "0.0",
    "netAmountToBePaid": "2796.6",
    "paymentStatus": "SALE",
    "settlementDate": "25-Mar-2019 18:30:00",
    "refundStatus": "Not refunded"
  }
]
```

Enc Data:

18642F36C9661C57F6518466E8E9539BB5F48B31D83BF0F632AC9A80D3532B1C8B
12D17
30A1A53329157828C84256D366654791D30553C71718A17E004C3CB85D3C605FCE0
B6B59
4CBA67D53DCEBC05896299468ABFEC540C52FF3D4A31361EBE6CE467F438D5044
14B8F
66C11731141E6BA419C9175F5C5F0C6BA8CBBA589B78BFF61E70CB06B756CECCF
B13F3
9E034A60342B82FFB7CA657BEBBFF5F8DA7BAA02AD8DE79085BDBE1A4515E1EA
389304

680CA62131A526B3B9E8234ED10836DF067348F407DA03778F76E5D751BC26563B
514B6
D0206B2959208419F84C1ED88DC81A6E4CEA71507722D59597D962B83E6129BD71
97D98
AE778E298326B854183014A10A2ECDE46E35E7E676E5240D7D5B0663254D6661B9
33D6F
EFDC43DD8A2204996B67785836CF6E98502545A1CFE6B4007E1E5B22832D50D234
C9200
7E08D2417785AD2AE66594F68F8B9E469F4020037AA60AB50C7704F3D545039ECC
1AD95
CC6133D9BB9EC271E85723E4D96598B42551CA09897764B3EAC43CB633B09D6ED
29218
3CB2EBDCA18A3A313594B18FEC3A949613EE3145C40037EF58C666D3455567BFC
FAA47
CAEA5182212F59D103EC6B4B30CABBEDC6EE4728ADE14A34D52AD266BD2C25A
A95193
23D2004EA76DDA11E7DBB023AC18AD82CC47F78F64CA3B82CC24B9A07C1F6283
0A9DA
CFCE410AD5D3F44F995035B947C7CBA66B88D77847C5F92487E3BD5699157CD1B
A0406
81F84818BB57A782F89F7FAE855C9CDC5A21712C397A4D69DA17AF61E4A35BCFC
D2489
E3065ACBB42125FFE36DFA65087FEB67C0507B0D52509FCC1AF56AFD58C789AC4
AC6B4
32AA7EA135FA9FB29CAA6485DF1AADF1929373A79542BED43618ECDC92543F9A7
F8529
6BCF15613215A85160FDD7F9D05687618C92D6D232CC148F3C27CBB78BB17F1E5
277CB
B77DCAC1A7A1988950C34112B293099E604BB7340B4D54690821CF33D5BF51C9A7
84589
1C11A2B73137296B07D0D4FA5737A78819BAD6B41407049F0734DC3A0AA931D04B
642AF
E7DE66247B8679DC185A6320EC674528A9CB85637E1BED60768AE5E7063EF1522F
50FA5
FF077BDA5B0DCFB866050D8324152C6C9FF5C671218127F1D97A998B2E4F7C20D
ED73F 9650865CE596C5ED09FC2132AA69D2CF319C54FE93B9E

JSON Format [without CC/DC])-

2. The API will only support POST method

3. For the API merchant need to share their ip for whitelisting in our system. API allows only white listed ips to get the required report.

4. The parameters are "merchantId", "settlementDate" (date should be in format YYYYMM-DD e.g. 2015-05-26) and responseType : json.
5. If there are records in for the merchant id for the settlement date the Settlement File is generated.
6. In other cases or error cases merchant will get response as .JSON format always.

UAT Testing Credentials:

UAT:

Enc URL= http://pgtest.atomtech.in:8443/SettlementReport/generateReport?merchantId=7760&encdata=F1F7E46B85600F3CBB86E848B8F80149FF27D16E2D3B5B002081
007CF7
3F90D15CEC7C147057045DCBCDDCF80A3661AC9460947D5940EB843C1CB31BC686F9C9C4670F
D823D
9F6DB062097FDB8F5337CF3B348D85BECB776CD77B4262D6D2D7C

Note*: For Open request refer NON AES Section

Production Credentials:

Production:

Enc
URL= https://pgreports.atomtech.in/SettlementReport/generateReport?&encdata=F1F7E46B856
00F3CBB86E848B8F80149FF27D16E2D3B5B002081007CF73F90D15CEC7C147057045DCBCDDC
F80A3661ACF377EDB0545BE671F9F8192E1813D697FF9FA11D5D5DFE2F85A3B192E1C11262

Open Request

merchantId=1191|settlementDate=2020-08-04|transactionId=|responseType=json

Sample Response (Settlement Date wise)–

```
[
{
    "srNo": "9",
    "merchantName": "BOB_DIP",
    "merchantID": "7760",

"atomTxnID": "300036240200",
    "txnState": "Sale",
    "txnDate": "22-Mar-2019 04:52:17",
    "clientCode": "4539",
    "merchantTxnID": "47667147",
    "product": "AXISEP_SAND",
    "discriminator": "NB",
    "bankCardName": "Corporation Bank",
    "bankRefNo": "5593092",
    "refundRefNo": "",
    "grossTxnAmount": "3179.02",
    "txnCharges": "0.0",
    "serviceTax": "0.0",
    "sbCess": "0.0",
    "krishiKalyanCess": "0.0",
    "totalChargeable": "0.0",
    "netAmountToBePaid": "3179.02",
    "paymentStatus": "SALE",
    "settlementDate": "25-Mar-2019 00:00:00",
    "refundStatus": "Not refunded"
  }
]
```

EncData:

```
69E0C5BD05E52F084A1956AAA52585C9AF83B895120423D1EC36753063437D82D759313DAAC
68BA4D
B5F83C4CA01503F42A64E9E0C0A7676E6521EE4325254429E86D486BDC6754A4639B06FA145C
210169
C655C8BC18A7D823D7F103A8C84CAF5470295D5D6EB4864932192349AAEDCCF94070EF404130
E91F1C
EDEE692067A1F838E0861E86956C7CFEFEFB742A7E91556ED0B370F63FD96DC51107031D0D53
EA653C
1113FADAD791209E3091B5E1EC10FE0C8FA179ED3E5841AF475182BF9B3F3F2B366E711CDE34
2896064
C705F7BE3D8F8F60D4F5F4258405BC760871EA7B0ABD72495A43DDD24207CEDD2BB69884E36
CAFEAD
B1C802CB78F9F73F63C5CDF8BF0987A4214E19AFFEDFB75D3DF2B2AC14FAE67EC911B39C84BF
F21841
726A18E5AAB117D49259AF9ABFCC20DFC8738BC1055D1C67FB93C37446ABF7EA7BD2B7361A5
2F0EB5
7C7E6CBEF308864A738815BBCCF55BDD04B7C7EBECA4AF0D6CEF5B4F4D687A3980A5A577D3E
3A9643
18A15F0EEACEE134A1C1A4D5AB7D759DACC5880E8EE981C5808850BE75C1961CEC04E2D31768
C8D5F
```

8EE15F030FCAEC9E1A1CC335BEF745CCB975BDEB73207656B4932D45735E8DF202E2A982CAD6
B728D6
786F50FAA3ECD5C884AD28D3D79DC22432DAE22875670C2EC0E894668B788AB021FDDDFF78D
389BB
D65814138020D18B59967D7962B58F9A6A24A0EC5049602DF8E28632A0B41A09F163EB464410
58AA5B
3C25A7CE47B391EFC6704DFBAAD14C835E77F586944A46887D24D7D918DA50D3498481BEF60
9E88F5
7653C120F398B0645CB7CE4E22F5459B7CED7DEFF42CC5CE0E1AC945191FA2A29578A6A4F05C
4A3206
FF2C7F575D81673DBC6F56F889A6176EEC15983135CD63938D79CA3ECABB6DB81F35BCF1089D
A82E47
FC45C5B44AF9FA9B95AE3DA1AAE896C5F5C46B33230F70447BC1C2376C867175F8857174611D
01F20E
79ED9A8198F5D573E15EC868860A574AC9871BF59D827A7E7AFD0B5E73B0204B7F03B7FCD5A0
D63F5 ADF5

Sample Response (Transaction Id wise)–

```
[
{
    "srNo": "9",
    "merchantName": "BOB_DIP",
    "merchantID": "7760",
    "atomTxnID": "300036240200",
    "txnState": "Sale",
    "txnDate": "22-Mar-2019 04:52:17",
    "clientCode": "4539",
    "merchantTxnID": "47667147",
    "product": "AXISEP_SAND",
    "discriminator": "NB",
    "bankCardName": "Corporation Bank",
    "bankRefNo": "5593092",
    "refundRefNo": "",
    "grossTxnAmount": "3179.02",
    "txnCharges": "0.0",
    "serviceTax": "0.0",
    "sbCess": "0.0",
    "krishiKalyanCess": "0.0",
    "totalChargeable": "0.0",
    "netAmountToBePaid": "3179.02",
    "paymentStatus": "SALE",

"settlementDate": "25-Mar-2019 00:00:00",
"refundStatus": "Not                                      refunded"
  }
]
```

EncData:

```
EE0B35DCD326B903B5EEE756DDD30EBA57642E9A5E0A494577CE51ABDBC04E8F950E0ECF7198
FC7411
50A6CB72665DA522E5485EE2B89D204436D7623D95B9B74829346A71B06F4F154A19DF548E31
24D6E5
85F19F0DD7DE89E4DFE660234E7BEE5E17C254CB00CF1EC48CA05D6CC2FF262C5C735A45245F
383606F
9463EFB0775DE1182CF0267F7901820D8D3E186D991A47159C81E8498039380F983FF34F75462
2708E9
8DA6B4484370BA4D66BB28CE4CBD58209C514DDA9524E9B37B17DB47BEE5BFA2C16A41B3920
CBCCD
CFA1ADED6E18FCD6862276410BA7B9597693F4CC895879C2E7C30CE8D679BF2AAA74D0002CE
077D4A
C54DBEFB3D8B66759D5691E901CE547B216CEBC9A9BFF9CE4A5C2528DBA7692F4FFB054F8F70
3E68FA
F413529BFB90FA5227AEA8EFE404AF1706EA18A6571A14C5BF169BDC2A64C673C7E7EAA012DE
327AB1
DFECCD3EA39FE3FCBC19303DAB211759CDD3353512E8B71103C47F9576A790FA0AA7F65CBEA
35FB78
92A6BB1C6E1530D963BBF0638F59013AA882C022B0BDFCD3D5EDC1AC55B2A5836ACD1635353
A96223
C5D06036EE6BB74F517EA5ECA6DC8DAE8B193ED168F2B94DA6118EF464E98E9879F6D2C1FDD
86037A
BC5AB4653A8AF3FAAD2B2B3BCF447494AB18EFCA67A722D36344AD7575C31071EB9D706FBF2
8B9A05
90E328DC1EDA0766895DF672DF86C59AAD970A8278C8C0B3CE71245DD0FD387058C26FD7A54
6EB831
A33D21EA2511327D3AFB75834AD4F441E440A40C8803C4392284B50EB283A61693C6E1BF4D45
EA1DE
791CB2749BA519C354D80F5C88982B3918E3F8CD0808AE684DB95D8C19F7A66A22D18B1F64E2
316231
445C33764724023A96A6274E2D2FF6EF847CCDA94B9B4BB6B466CA7805B27B88E8DAE5D33657
0271B4
B2128F9F4011C1DAC90C78709628D22637835636404CE8E4CA92358BF6E2A3D221A481526A79
73D9CE
054EBFF7A110C08C0121E73A4FDA2819706098882688A92CA8F79168686E9A830037A4483E012
7A02B 39396B26F63C7A3B64A335FAB34B7FEFB609182E59E5078AC7145A8406734609A66
```

Status Code:

Status Code	Status Description
"00"	Mandatory parameter value cant be null.
S0	Invalid merchantId or merchant is not present.
S6	Invalid ip address or ip is not configured
S3	Invalid settlement date. Date should be in Format(yyyy-MM-dd)
S1	Invalid Response Type
S2	Invalid transactionId
S7	No data found

Note*:

1. Settlement Date and Transaction Id cannot be passed together
2. If both are passed together, error "Provide Either TransactionID or SettlementDate. Both cannot be provided same together." With Status code "00" will be exhibited
3. If responseType= JSON is not provided in request, .csv file will be downloaded
4. For testing make use of transaction details provided in attached file

To get settlement status for particular transaction –

https://psa.atomtech.in/paynetz/vsfts?merchantid=1646&merchanttxnid=123&amt=50&tdate=20 16-06- 21&atomtxnid=17075765

Sample Response –

<?xml version="1.0" encoding="UTF-8"?>

```
<VerifyOutput TxnDate="2016-06- 21 18:21:24"
CardNumber="652189XXXXXX0233" surcharge="0.00" discriminator="DC"
atomtxnId="17075765" bankname="ATOM PG" BID="617318000165"
VERIFIED="SUCCESS" AMT="50" MerchantTxnID="123"
MerchantID="1646" SettlementAmount="49.52" ReconStatus="RNS"/>
```

Please note that response would be given in case of RNS and RS transactions, for NRNS transaction response would be NODATA.

Abbreviation:

Transaction State	Description
NRNS	Not Reconciled Not Settled
RNS	Reconciled Not Settled
RS	Reconciled Settled