NTT DATA
Payment Services

Info type: Confidential
Company: NTT Data Payment Services India
Info. owner: Product (PMG)

AIPay APIs

NTT DATA
Payment Services

Confidentiality Disclaimer

The information included in this document is confidential information relating to the business of NTT Data Payment Services, India(NDPS). It is being presented to you based on the understanding that it will not be used for any reason other than consideration of a commercial relationship with NDPS and, will not be used in connection with any decision to trade in securities of NDPS. Please be advised that any disclosure of the information contained in this document/presentation to any other person, or any use of this information in connection with the trading of NDPS securities, may be a violation.

This document, or any part of it, may not be reproduced, copied, circulated and/or distributed nor quoted without prior written approval from NDPS.

Document Information

Document Attributes	Information
ID	API Document_AIPay
Owner	Product Management & Engineering
Author	P&E – Product Team

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	API Document ~ AIPay	25/02/2021	
1.1	Nital Kapadia	Addition of Surcharge & UDF Parameter details in API Request/ Response	06/12/2021	

Version	Primary Author	Description of Version	Date Completed	Reviewed By
1.2	Aviral Tripathi	Signature Generation Sequence added, APIs updated to OTS-based format and document updated as per NTT Data Guidelines.	08-08-2022	Nital K.
1.3	Aviral Tripathi	Multiproduct Transaction parameter to Auth API added	23-12-2022	
1.4	Aviral Tripathi	Added parameter 'subChannel'	20-03-2023	
1.5	Aviral Tripathi	Change in Status Code description in Requery API response with details added in 'Point to be Noted' section	02-06-2023	

Index

I. AI Pay

1. Introduction

"AI Pay Flash Checkout" provides best payment experience to end customer and results into a better and faster payment experience. AI Pay will ensure a higher possibility of success in completing the NTT Data Payment Gateway transactions.

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.

2. Key Features

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

3. Payment Process

- Once the end customer arrives at the merchant's checkout page, the merchant needs to call the NDPS' 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 NDPS 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 NDPS and acquire the token Id sent in the response
- This token Id is to be used for calling the NDPS' Java script function which will then open the AI 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, NDPS' 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

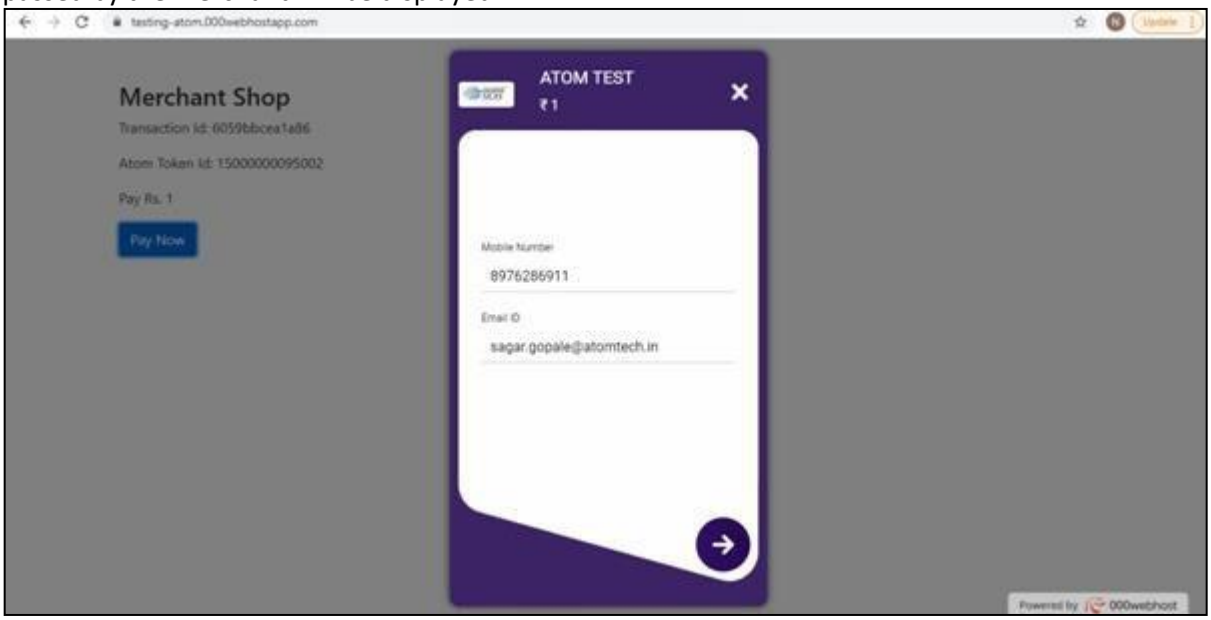
request timed out"}

- Such errors are to be handled explicitly by the merchant and Rest API request is to be reinitiated

4. Transaction on Insta Pay Page

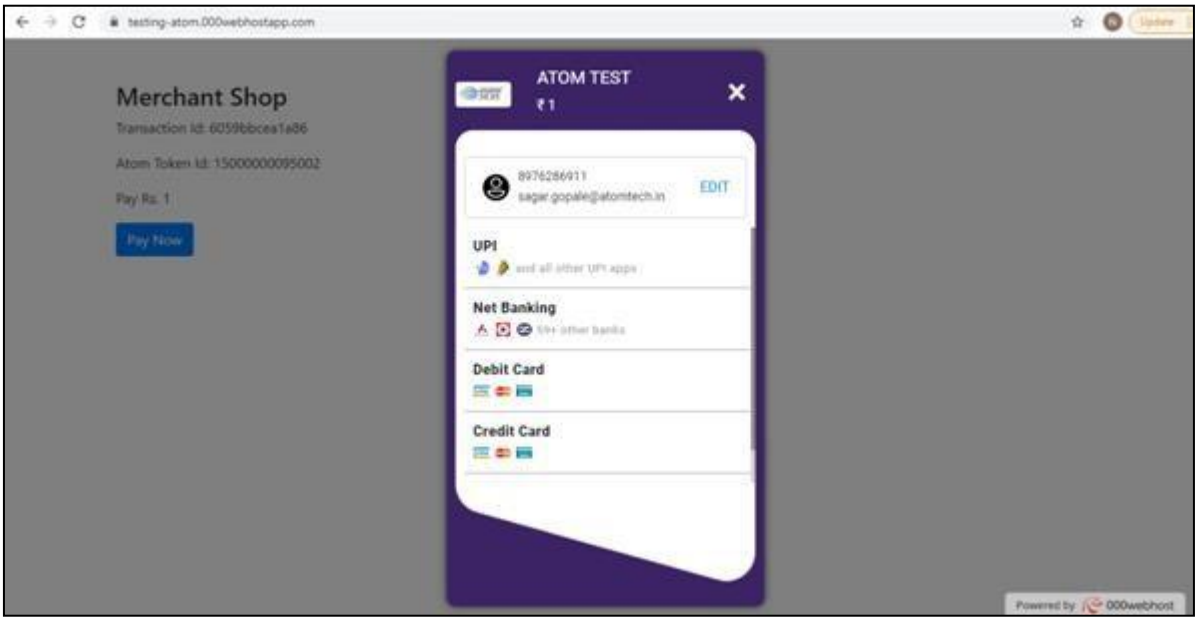
- Once the end customer arrives at the merchant's checkout page, and can click on the Pay Now button as shown below

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

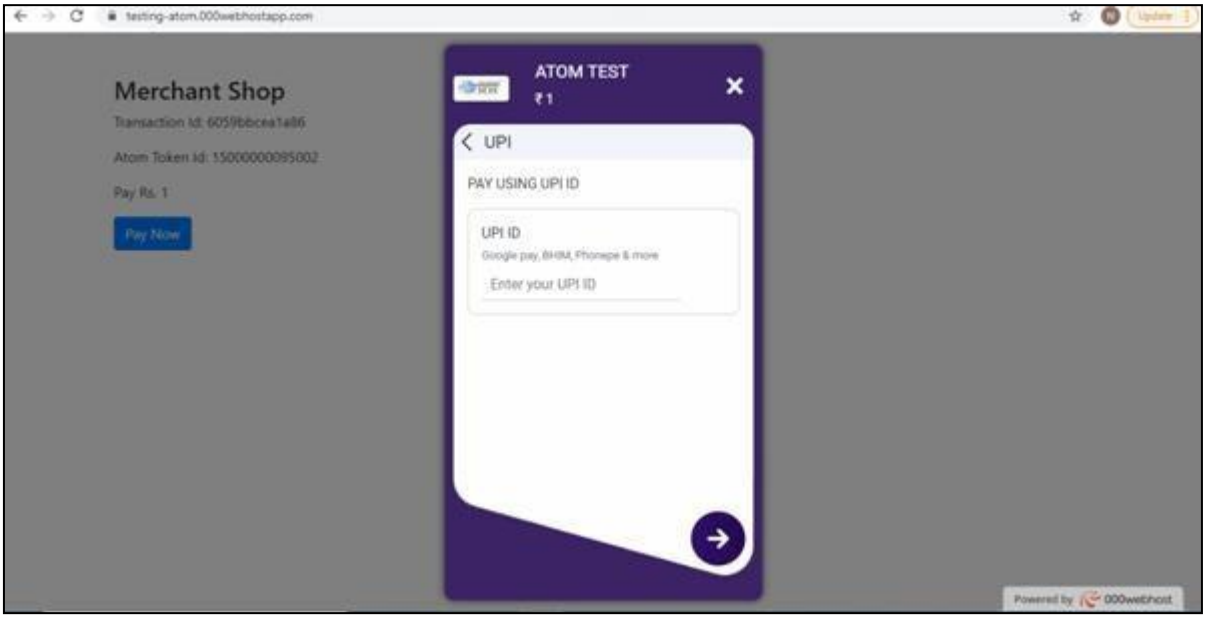


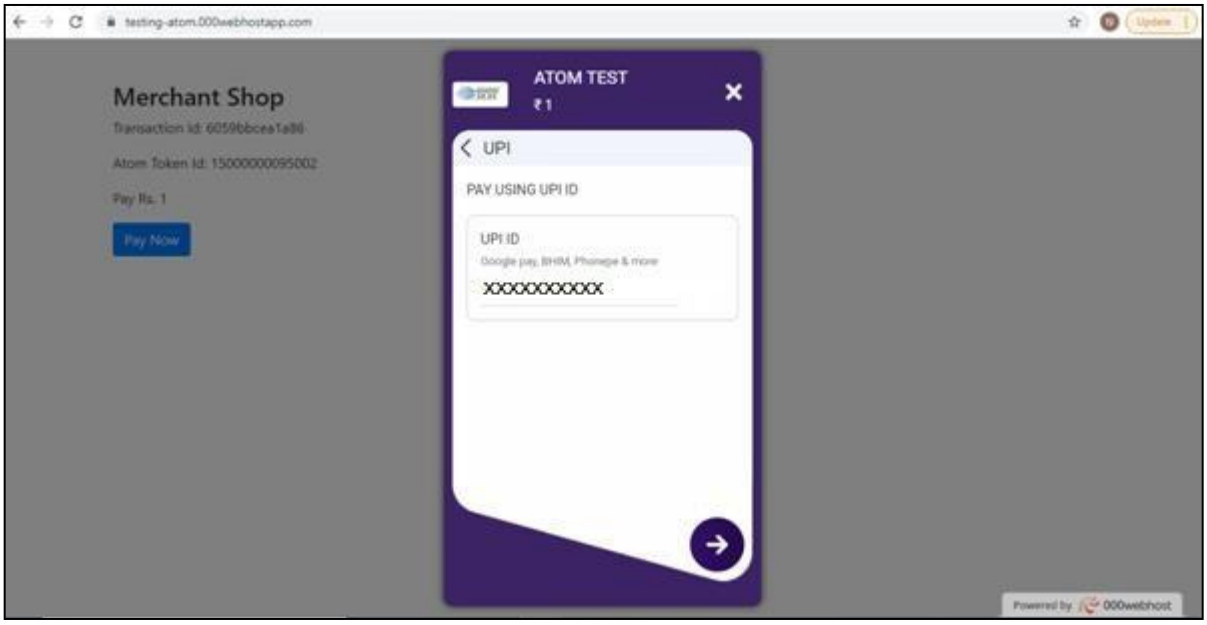
- Clicking on the arrow will allow the user to see the available payment options i.e. UPI, Net Banking, Debit Card and Credit Card

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



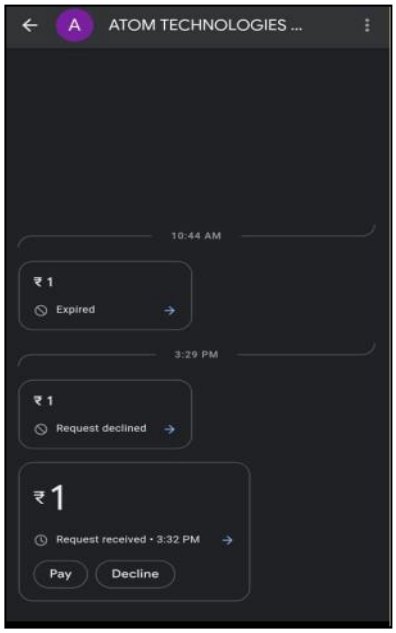
a) UPI Transaction

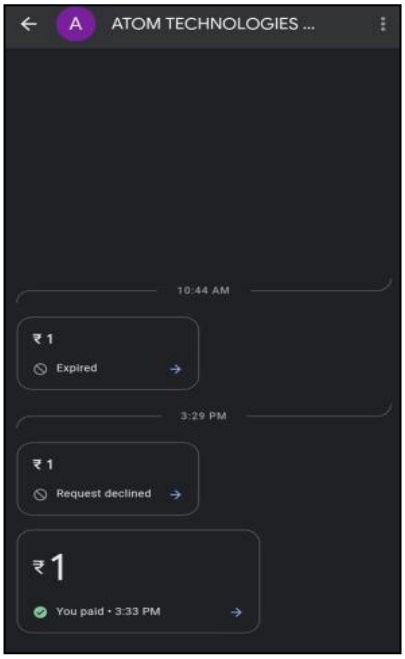




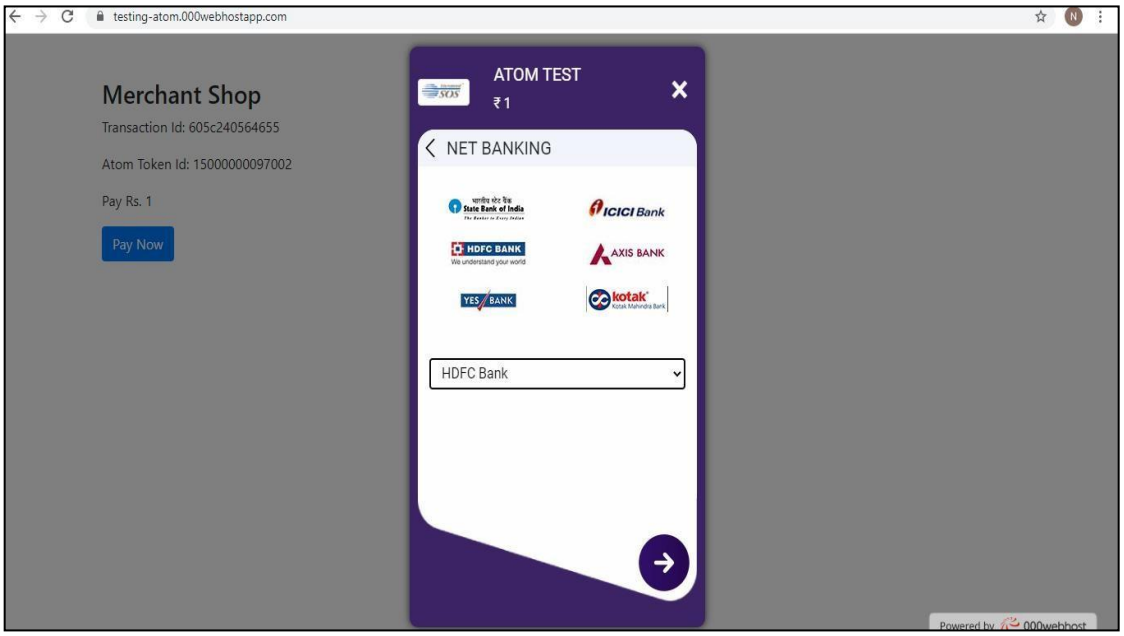
- The collect request will be sent to entered VPA

User will receive the collect request notification to make the payment



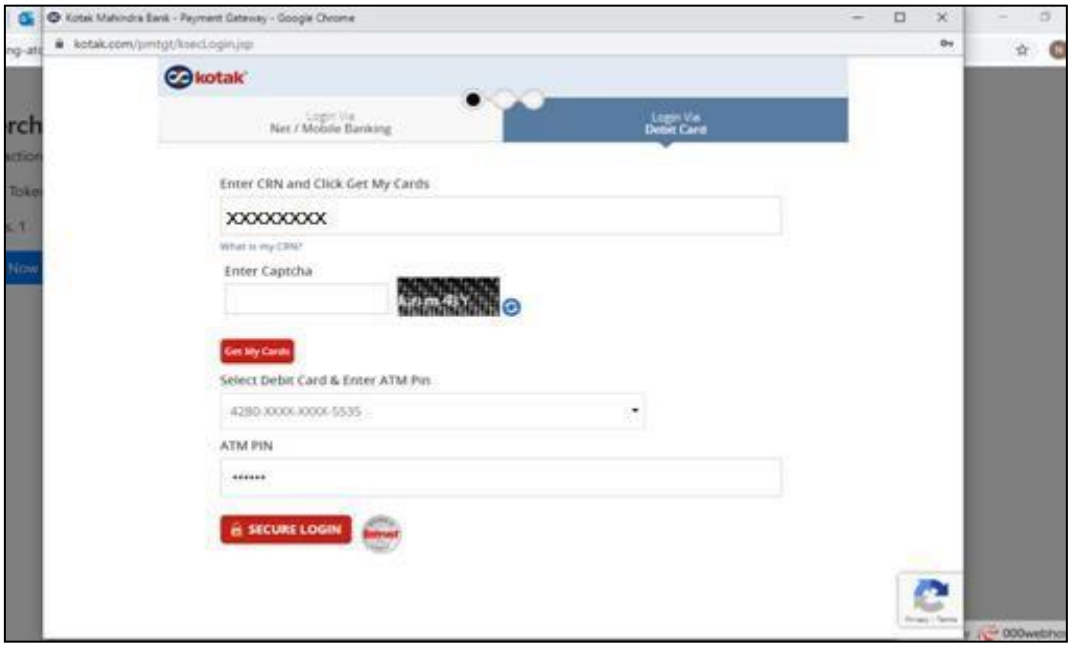


- Clicking on Net Banking Payment option from the AI Pay popup will open the below page

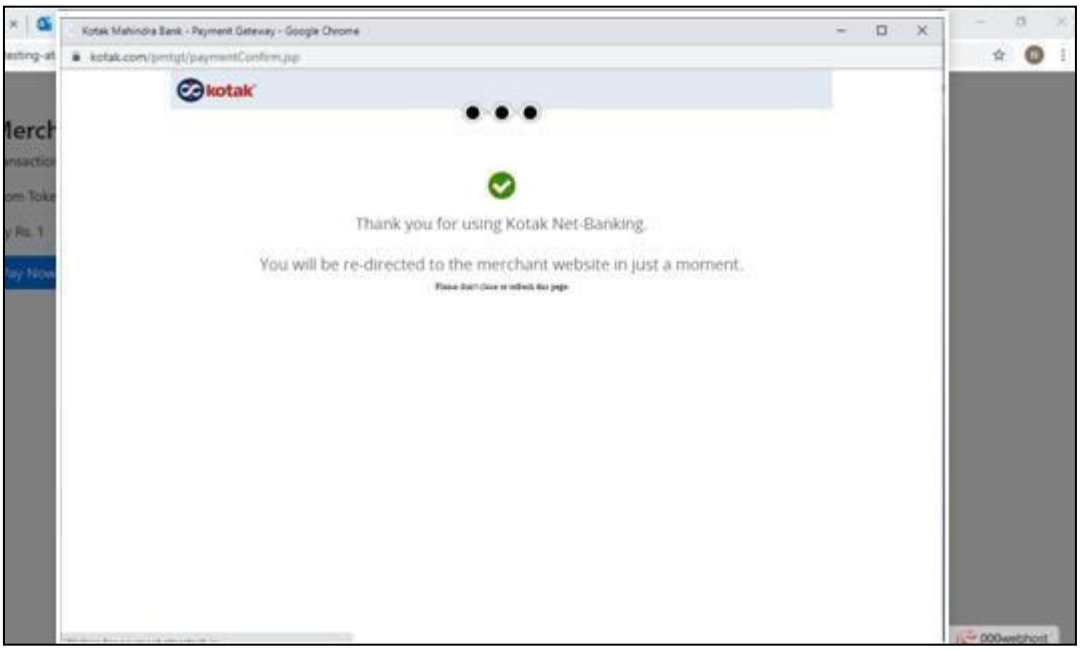


- User can select any of the net banking bank from the available 59+ banks

- Bank Internet Banking page exhibits

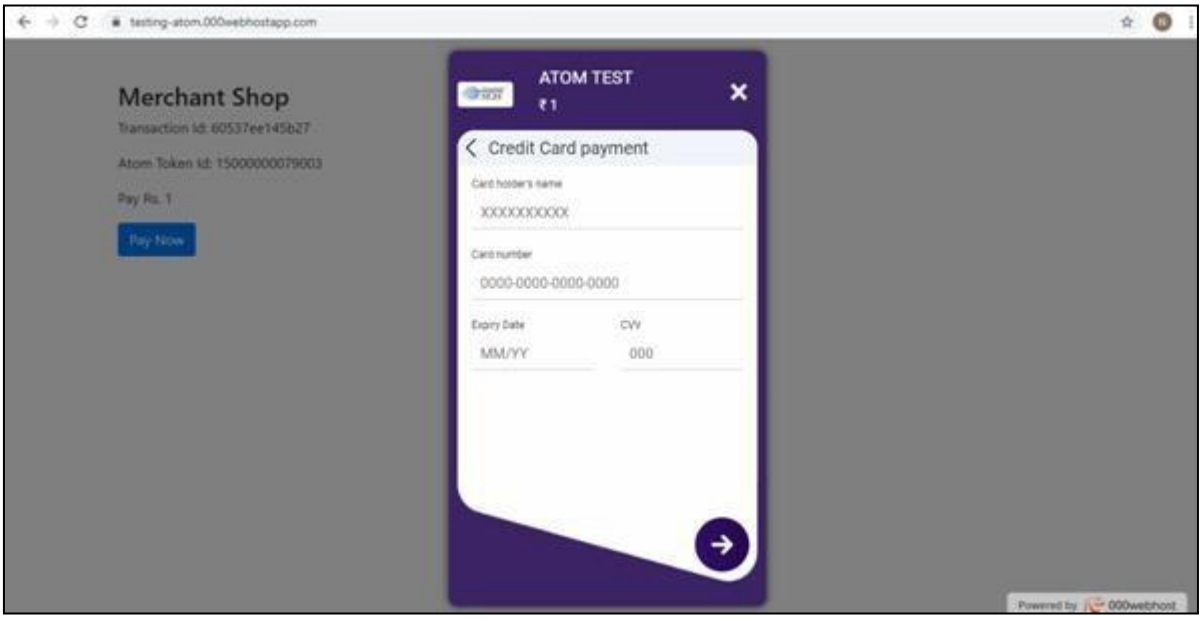


- Payment Completion

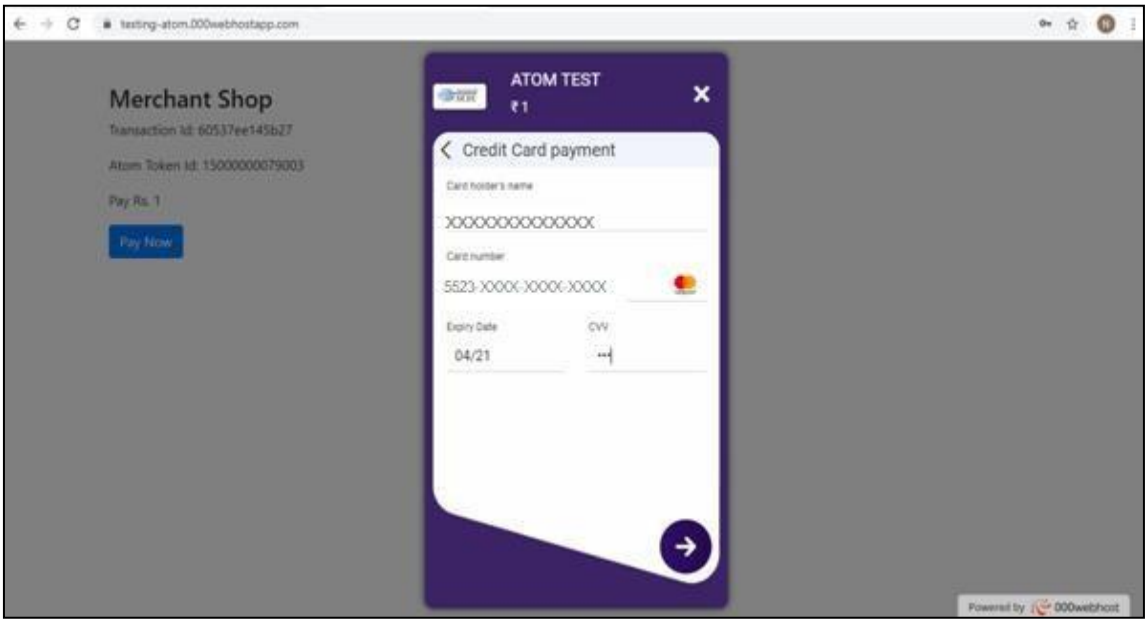


- Transaction response is acquired

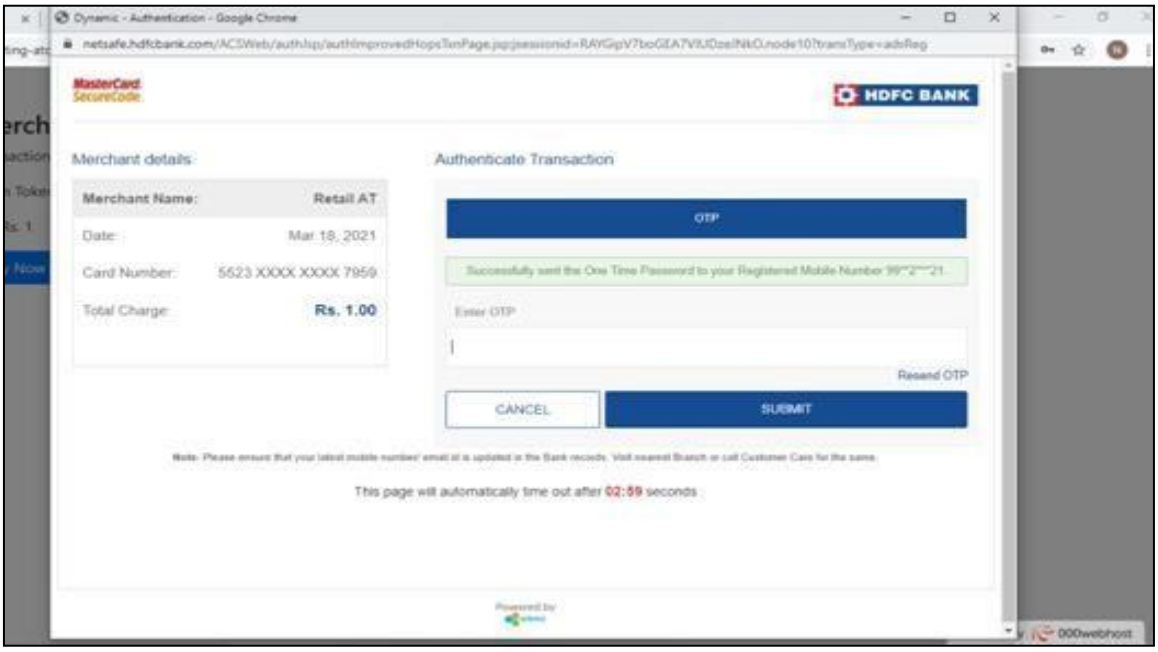
b) Credit Card Transaction



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



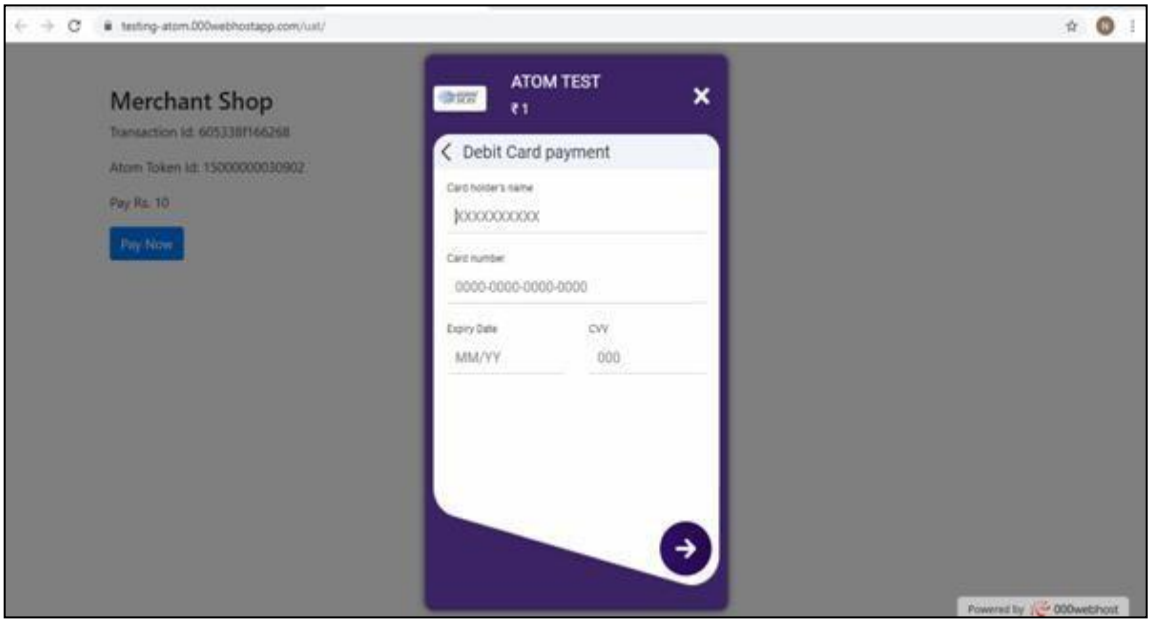
- Bank Page Exhibits



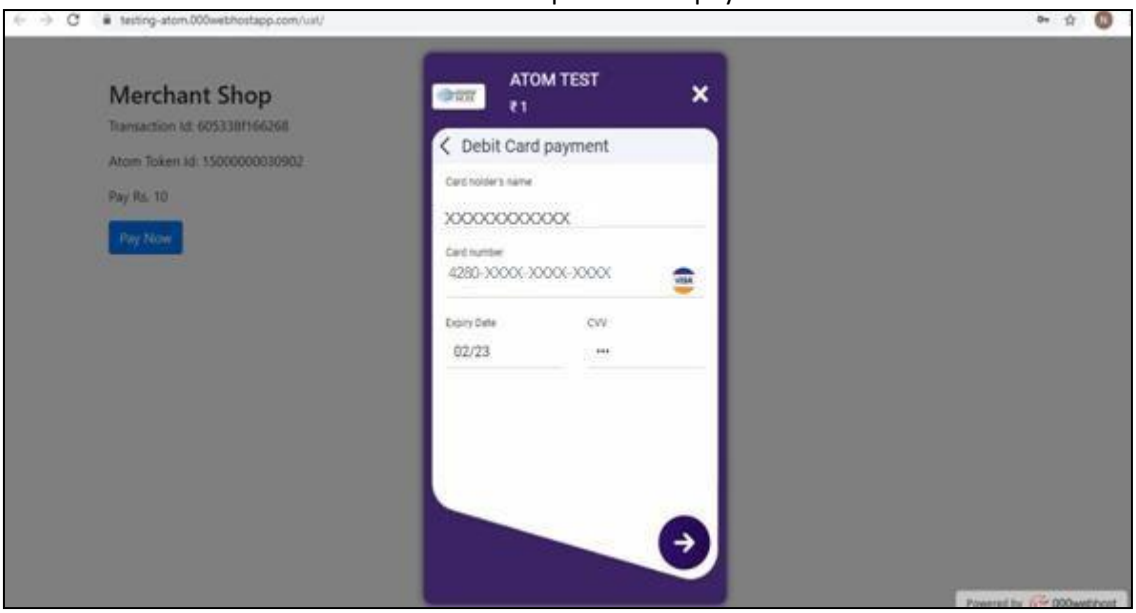
- Transaction response is acquired

c) Debit Card Transaction

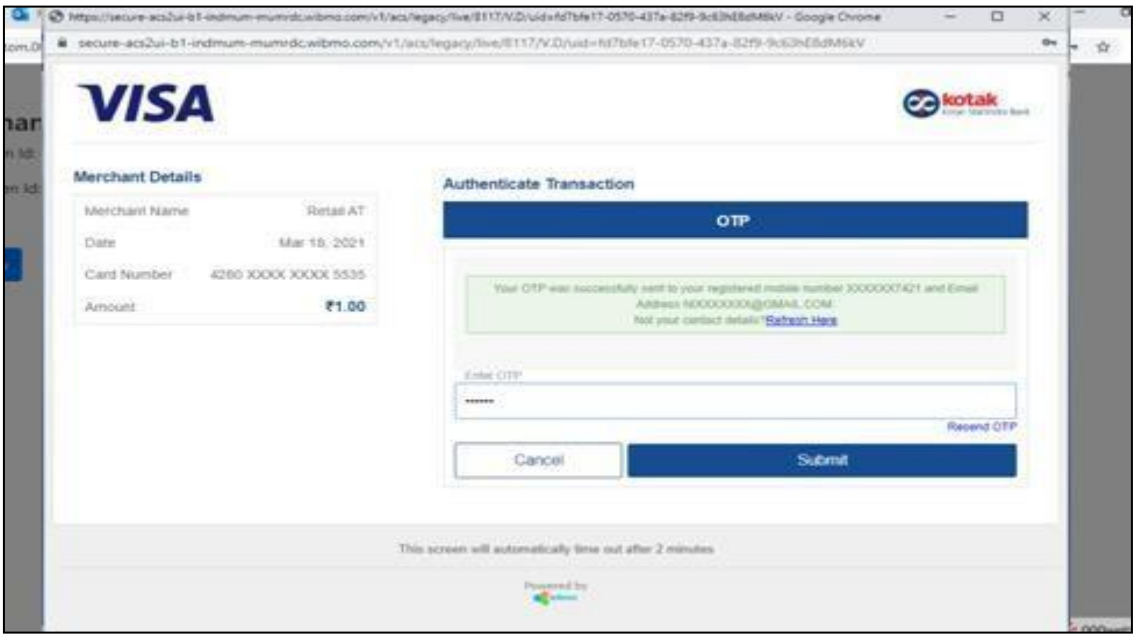
- Clicking on Credit Card Payment option from the AI Pay popup will open the below page



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



- Bank transaction page exhibits



- Transaction response is acquired

5. Merchant APIs

a) 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 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
20	subchannel	String	2	O

The below are the Possible Values for Parameter 'subChannel' –

Field Value	Description
NB	In case the merchant wishes to show only **Net Banking** as payment option to customer.
CC	In case the merchant wishes to show only **Credit Card** as payment option to customer.
DC	In case the merchant wishes to show only **Debit Card** as payment option to customer.
MW	In case the merchant wishes to show only **Wallet** as payment option to customer.
EM	In case the merchant wishes to show only **EMI** as payment option to customer.
CH	In case the merchant wishes to show only **Challa**n as payment option to customer.
MV	In case the merchant wishes to show only **BharatQR** as payment option to customer.
UP	In case the merchant wishes to show only **Unified Payment Interface (UPI)** as payment option to customer.

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": "5.00",
   "product": "NSE",
   "custAccNo": "213232323",
   "txnCurrency": "INR"
  },
  "custDetails": {
   "custEmail": "sagar.gopale@atomtech.in",
   "custMobile": "8976286911"
  },
  "extras": {
   "udf1": "",
   "udf2": "",
   "udf3": "",
   "udf4": "",
   "udf5": ""
  },
  "payModeSpecificData": {
   "subChannel": {
    "NB"
   }
  }
 }
}
```


Note*- Merchant can specify multiple products in 'prodDetails' if the same is pre-configured. The amounts across those products can be variable, but the total amount across all the products should be equal to that paid by the end-user.

If there are various products across which the payment is to be distributed, the below mentioned product parameter shall be replaced by the prodDetails header as depicted in the example below –

The "product": "NSE",' shall be replaced by :-


```
"prodDetails": [
{
"prodName": "NSE",
"prodAmount": 11
},
{
"prodName": "BSE",
```

"prodAmount": 12
}
],

One aspect to be considered here is sum of all 'prodAmount' should equal amount to be paid.

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=11C0BE23D1A9B9A03A80BBBE07DBEF89902F
7DA9F553156D98895794D6BCEA4078BFA0FB4F796687DB5983ECB1AAD813DDB66DCABC82D8FD
CC3CCDB8AEC94BB0CB94B2BA3B463A994ECF2E7387875E5F624FE651150719DC8203F142761F
CECF65A3C54176D852D7C694910ED5EFEB7030C2F2195CD645CE4FD7174BF8B3625587B9989F
2384AA43ADA01DD8F872D35290606D34840C32ED0199E3399C453ADBC2848C5F681C8EDCE2B8
5792E90E9FA52A67D8669BA79A2D8C0D2A796473BE9326FB10EB3B7E1457B4B4FDD842262260
EA0399B22B5314FEDC62648D6D47D059D60D6A392AAC0D4B3DDBE44EC36A5B3F405862BA2E54
47F9A42A23E5A3F4A47C4513A5E0EB23E2CA9FC3E10ED9E894D1F1A08EBAC32D71E29D516435
A014B12F8F16A102209F798FEF7B77C2F3ABDDD659CF5AC51A73D829064C3F477F29C3D651CE
DD88094CC47D01BF57B27C8787B0140298466221252359 83CFF3097EED44B62BE87A10F4FEAB
031576777358A9316FCE03891BC2014A1A81B5FA1B6D67811479D493643D6733B287655F7868
9C20798803FFEF4E0038A02B37B13A0AFFAAB6491CD50163CB45FEEDDAB07DF2722C27482150
EA2CF53B964F9FB733658B056641A741AE13F2189E3EADF806246D6604D2A2E170D728D0614B
ED340FC93E5789772C3356F2149CDA1D90DCB0AA9198B9FF073B91A30BBAA62EF611326FEAE4
B746682C69A15B17B109DFB7ADF2BA4B8C8AD905BD27218923C23F574B2D3A32052D579FFE62
AEA74FB5A0CBC55625AE1D6C9B199955273BA020C7D4F88B81ACCB47EFE2209ABAA2068CE543
302437AE7FEB880896E1F21361E96C671D4AFC3B5019BFB2309754A87C82B26F30E6D70FBFCB
3BA4D6942883B617553740D1F75A12ED3F86FE06C379FFC61EA1F48B309BEACB0B0AD6A4277B
3E6A9EE236757547DAD8EB75B489D853EA09AABE4E028D3514BDB3C25C614EAFDEB2F5D8B4D4
CED99470AB4BA04C2A55F410CA34FEE0AC08BBAF0DA9AC4B318ECA77CF628481ABDA7619C517
87CB47A72D8FC6D32AAA7F89DB813E204FACAAD0973835EB505199DCC1342DBC9D40559563B2
246B2B46F0BAA4B1990CC983D3FFAC485DE90E6778CAD45FA7722C4036426785598E184BAC4D
7FCD66DE46F4BF569B91B37A17C3E30F8E557EF3723483471BAA0DD98593D33EC7944A4646FD
2CA7E799D994A085C7ECDFB0DDCAEE0FF2FE3A32FC9D8BE3A0A829A136560A8BAF74063BA8C3
A35D3ADCCAF072795CE59C7BA6B9247BFE2EAA5BD0436B9D8ACFB894DCAB0ED59EDEDA3A&mer
chId=8952

For Production:

https://payment1.atomtech.in/ots/aipay/auth?encData=11C0BE23D1A9B9A03A80BBBE07DBEF
89902F7DA9F553156D98895794D6BCEA4078BFA0FB4F796687DB5983ECB1AAD813DDB66DCA
BC82D8FDCC3CCDB8AEC94BB0CB94B2BA3B463A994ECF2E7387875E5F624FE651150719DC82
03F142761FCECF65A3C54176D852D7C694910ED5EFEB7030C2F2195CD645CE4FD7174BF8B3
625587B9989F2384AA43ADA01DD8F872D35290606D34840C32ED0199E3399C453ADBC2848C

5F681C8EDCE2B85792E90E9FA52A67D8669BA79A2D8C0D2A796473BE9326FB10EB3B7E1457
B4B4FDD842262260EA0399B22B5314FEDC62648D6D47D059D60D6A392AAC0D4B3DDBE44EC3
6A5B3F405862BA2E5447F9A42A23E5A3F4A47C4513A5E0EB23E2CA9FC3E10ED9E894D1F1A0
8EBAC32D71E29D516435A014B12F8F16A102209F798FEF7B77C2F3ABDDD659CF5AC51A73D8
29064C3F477F29C3D651CEDD88094CC47D01BF57B27C8787B0140298466221252359 83CFF3
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":150000
00033303,
"responseDetails":{
"txnStatusCode":"OTS0
000",
"txnMessage":"SUCCES
S",
"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 NDPS in the response will be received by the merchant
- For Production:

merchId=8952&encData=5500FEA2F09DA7EF128CFE7D2D01F2533706602C7F5CF1
7176C85222692E684F1EC38C6EC4D2A30305F58B09A5DAA5285A4CCB10A2298D74
3622C0835BD2D70CE5AF22C9B8610B169544715B6ACDBE3B85532198D73B28B06
63F5B96395AEC61CAD1B2CCFD6E63A8B1FBDBFA3E4F22EFE3C7A4F9683EA1E49CA
371F00E4732CABFEB1589E4D93C24C8C33219674BFD45EA96150EA37D7ED8C3BA0
18B7879C809DE5B283FFE03850C452AED7A43F82B23

b) 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>
```

c) Final Transaction Response

- Once the transaction is completed by the end user the final transaction response will be sent to the merchant by NDPS
- 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":{
                        "merchDe
                        tails":{
                                "merchId":8952,
                                "merchTxnId":"60583241b
                                0c1c",
                                "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":"5678a713fdfac844fe6349107996c336b43ae86413f3b2523
                                7211022f38e3be2b286243bbc2019aa1885cae1e80e35edbccd5210e92
                                d5587d54d72c0e27a8017",
                                "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.	

Signature Generation

- For above mentioned API's request and their response signature fields to be generated using the shared Hashing code.
- Signature type is HMACSHA512.

String to be encrypted:-
[merchId + atomTxnId + merchTxnId + totalAmount + txnStatusCode + subChannel + bankTxnId **]**

a) Sample Request

- 9132Test@123OTS77SL3.00INRREFUNDINIT

b) Generated Signature

fb0798ae1602bc0d395b7a12e8f08ab5d0ec64e90a2b9e7f09ed3db847ac86820edb1a0d557c51aec2e0166fae97d4aa966a6f0d8512fd454e4deb906959c1bf

c) Hash Keys

Below are the merchant id, request & response hash key –

MerchId	reqHashKey	respHashKey
9135	KEY1234567234	KERESPY1234567234

d) Encryption Key

- Below are the merchant id, request & response encryption key –

MerchId	encReqKey	encResKey
9135	58BE879B7DD635698764745511C704AB	7813E3E5E93548B096675AC27FE2C850

e) Signature Generation Java Code:

```
import java.io.PrintStream;
import java.io.UnsupportedEncodingException;
import java.security.InvalidKeyException;
import java.security.Key;
import java.security.NoSuchAlgorithmException;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;

public class AtomSignature {
public static String generateSignature(String hashKey, String[] param) {
        String resp = null;

        StringBuilder sb = new StringBuilder();
        for (String s: param) {
                sb.append(s);
        }

        try {
                System.out.println("String =" + sb.toString());
                resp = byteToHexString(encodeWithHMACSHA2(sb.toString(), hashKey));
        } catch (Exception e) {
                System.out.println("Unable to encocd value with key :" + hashKey + " and input :" +
sb.toString());
                e.printStackTrace();
        }
        return resp;
}
private static byte[] encodeWithHMACSHA2(String text, String keyString)
throws NoSuchAlgorithmException, InvalidKeyException, UnsupportedEncodingException {
        Key sk = new SecretKeySpec(keyString.getBytes("UTF-8"), "HMACSHA512");
        Mac mac = Mac.getInstance(sk.getAlgorithm());
        mac.init(sk);
        byte[] hmac = mac.doFinal(text.getBytes("UTF-8"));

        return hmac;
}
public static String byteToHexString(byte byData[]) {
```

```
                StringBuilder sb = new StringBuilder(byData.length * 2);

                for (int i = 0; i < byData.length; i++) {
                        int v = byData[i] & 0xff;
                        if (v < 16)
                                sb.append('0');
                        sb.append(Integer.toHexString(v));
                }
                return sb.toString();
}
```

Response in Encrypted format

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

Encrypted Data:

merchId=8952&encData=0CD63904F407DC4D24167F8D8F2BEE997C80D4536815B71C1EDDB8FDCBA
CE8B9F6D72174CC18263D88AD7728C821A1F7B789A3F6C8EE43C28933A6748D3239346F4AAEC04A
FFD7687FB31FFDA9AAF0F64A2911990F315B425F11F1196EAF59E9471644C321A0050BF64C72DA0E
FE6A11D14F4AA3890267552FA4E3FA84B21275150F7E98CCF84BD145DCB6F00EF7494FAEE9D2385
2507F46517B67D8F9680941BF887A761D9B26879B2A369E9B6CEF835DD51DEE86982255CCB98A8C9
D8C3662FB87AF859B9E0DF167D7D48E89BE6896226C5D8AED6B2CA1986325D030B646ADAD4C394
4A8B851D3D3448C7F0CD37DB8E466A51AFFD43E38307E2ECA29675614AC4DAA072CB6595974E4E
1EEEA70F031B7BA3C5FDF2D4F599854B658B9D9D2DAA3FEAA67B534BFADE98FB5D123BB8FDB5
63F840B7B28C71054CCA358F50348CE842C3AA05FDBF0400892F03B5290FF5CA7E01C174FCEBC0B
65012FF9F0AB1C19F483570599F9C681001F46C342CD6A491B2B4619FC70D30FF0CEAB49F2B5201A
DCE25C510027496A4A64EA3A2661D55DA89741600
9769F2F501CE864F8360EC8EB7558CB24AAC577E338EA3DB515DA5199D159688C0A624D0C548135
A6001D2543D7274BEBA6C2AB648042C08CDF4732CB63FB0B57DDD967E8F577427EF19D669E0A16
AF77CA605491C1657744A1395065C1A459232A12743E62DEF8B6FC9F6075C8DA7E27C2AE6874698F
AD367A9C23CB28FDC001AE5501693FB5435BD871DDCC5BB0E236867F60AA04E52A7582DB8F25D0
FC4CFD7C4D223D67DADF248E0C920F0BCA7C9E774604FAEB56FAE025091FA595A7797C4331D3FC
F9CEB881A1392F231A4F05F4A4FFA58099802A20F7345EC7120BCE82ACD7C536B2627E4DD22F2CE
177C27F38B544ED991EF83A1F888A6217B48511E6A1FCFEF065FD84C459E4695E39AD5BB078B4275
1A798B780258E64B2DA311A782D3B6FCD987515E870A54465F7C5302660B6F3A0CF88908EDE72743
FBD55741B8D7B9C6BBAE1A9D6F174DC4A9C52EB8129A8DE69F49A340BA65AF128E935FED44D57
906B274905E6297C00358FBEE401742C5812C9BE54517770F48143B9E8E9EA39617834968AAC4A1494
E3D5CF3B69BD5D1ED0392CF84B86C64E4B29CDCB81C7FEA1ADEA09B9C27D6F31A5DB9EE6A27F
1E0914D2180D5A6247A5BE6F2119FA13919DC696B0C4EA95E5D96F850CB6D32B05F5505F48E2354F
4953A2D2897F929D346ECC39F816DBF4079893333A44AFA473B4A86FF96773295B48BA192FAAFEF7
9126FE0E944E1B1CE0D09C2BC745F62C237C3E57EC61209ACF426E8FAB1DB3F8DDDE7AC6374866
964178

Decrypted Final response:


```
{
  "payInstrument": {
    "merchDetails": {
      "merchId": 8952,
      "merchTxnId": "Test123450",
```

```
    "merchTxnDate": "2021-12-06 T11:40:18"
   },
   "payDetails": {
    "atomTxnId": 11000000174925,
    "prodDetails": [
     {
      "prodName": "NSE",
      "prodAmount": 100
     }
    ],
    "amount": 100,
    "surchargeAmount": 1.18,
    "totalAmount": 101.18,
    "custAccNo": "213232323",
    "clientCode": "1234",
    "txnCurrency": "INR",
    "signature":
"2b12c8bfc0e3a8268eddb6f406bf4187d4d0a0064d035544698651145392 2c27e38367a97fff85863d48c147a8218
e9e2d5003ab121f6f61ce3914030c60caac",
    "txnInitDate": "2021-12-06 11:40:19",
    "txnCompleteDate": "2021-12-06 11:40:23"
   },
   "payModeSpecificData": {
    "subChannel": [
     "NB"
    ],
    "bankDetails": {
     "otsBankId": 2001,
     "bankTxnId": "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."
   }
```

```
 }
}
```

II. Transaction Status (Requery) API

1. Description

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

a) Transaction Status Tracking Process*:*

- Merchant can track the transaction initiated/completed (Success/Fail) by end user via Transaction Status (Requery) API. Merchant sends the MID credentials as provided by NDPS in login parameter along with the transaction details as encrypted data [Pg. 5] pertaining to transaction details of the transaction/refund of which the status is enquired.
- On initiating the Requery API, merchant will receive the status in encrypted response [Pg. 6]. Merchant will decrypt this response through decryption method [Pg. 7] .
- Merchant must incorporate the encryption logic [Pg. 9] provided by NDPS at their end, to send the encrypted data in request and to decrypt the encrypted response.

Note*: This API is a **Server-to-Server** Call.

2. Request Format

a) Transaction Status Sample Request (Open Request-JSON):

- Transaction Status API request UAT URL: https://caller.atomtech.in/ots/payment/status?
- Production URL : https://payment1.atomtech.in/ots/payment/status?
- Request and Response of Transaction Status (Requery) API will be encrypted using AES 512.

Request Parameters are to be shared in the format illustrated below:

```
{
"payInstrument" : {
"headDetails" : {
"api" : "TXNVERIFICATION",
```

"source" : "OTS"
},
"merchDetails" : {
"merchId" : 9135,
"password" : "Test@123",
"merchTxnId" : "250420221",
"merchTxnDate" : "2022-04-25"
},
"payDetails" : {
"amount" : 1.00,
"txnCurrency" : "INR",
"signature" :
"abaf4b4011b6813c0a16896302a6fab404035df377d3b25e60b8a6766dffb6383891a7443f603fc99b643e2bf4049d34eccc74e3253
3c742c25580f60e17ab2a"
}
}
}

b) Transaction Status Sample Request (Encrypted Request):

https://caller.atomtech.in/ots/payment/status?merchId=9135&encData=400E949F8F951060A21D462EC57CB03341BBB5627A23973D22BB39E312AA7491788FE15430FC7019851C8EC737B613F70740488E050EBA67B08083103AB5C0D139A35E47FE9C41A36C44C9EB181DA931FED63BE0F68C82B26F791DC805395AB0580AB7B7CF942061980C94AEE5AC50DD01225B6135A5F3F7AFA5646130D3CEFFC241AB316CBC2ECD34AD757B05CC8EFCBC0F91961325604CD71DFDE35AFD48F0070FD94136986DFE1573ABDF2F1DD1027B1581BA59FB6D635FF8DF1FDC70C27A011197E80E069F2D473493C614B000FEB951DA47EF3433ED1B3E00F8A5625379290CE453DBC792BDDEF825F11224BFDB5FF02BCDCD0B08901E97D7010F394BC50DD1E891BE655768B2E34CEBCDBA52B5423E079E9ADFF5F64E6560F86499DC1E6124242F46DC077C128A6A8574EEE6148177B7EBB0F8CC09CD7CDA85505C390CD49C10F79191680FCF1D048DEF669CF1DD42D0F2E66A7AA81E56D537796C7DF271BFC181D32C71425B26FE0F2574DA406250854CDC8045CE9AE5E06309BDA8F1885EC40CB5E5B9FC2210079232F08E0DF2B5A88DE5B0AA8B433AFD2DFBEE616C3EC946D5A48517E4F6AEC3 CA9172CE3F

c) Sample Encrypted Request Data:

400E949F8F951060A21D462EC57CB03341BBB5627A23973D22BB39E312AA7491788FE15430FC7019851C8EC737B613F70740488E050EBA67B08083103AB5C0D139A35E47FE9C41A36C44C9EB181DA931FED63BE0F68C82B26F791DC805395AB0580AB7B7CF942061980C94AEE5AC50DD01225B6135A5F3F7AFA5646130D3CEFFC241AB316CBC2ECD34AD757B05CC8EFCBC0F91961325604CD71DFDE35AFD48F0070FD94136986DFE1573ABDF2F1DD1027B1581BA59FB6D635FF8DF1FDC70C27A011197E80E069F2D473493C614B000FEB951DA47EF3433ED1B3E00F8A5625379290CE453DBC792BDDEF825F11224BFDB5FF02BCDCD0B08901E97D7010F394BC50DD1E891BE655768B2E34CEBCDBA52B5423E079E9ADFF5F64E6560F86499DC1E6124242F46DC077C128A6A8574EEE6148177B7EBB0F8CC09CD7CDA85505C390CD49C10F79191680FCF1D048DEF669CF1DD42D0F2E66A7AA81E56D53

7796C7DF271BFC181D32C71425B26FE0F2574DA406250854CDC8045CE9AE5E06309BDA8F1885EC40CB5E5B9
FC2210079232F08E0DF2B5A88DE5B0AA8B433AFD2DFBEE616C3EC946D5A48517E4F6AEC3CA9172CE3F

d) Specifications of the parameters of API Request:

Parameter Name	Conditional/ Optional/ Mandatory	Data Type & Max Length	Sample Value	Content/ Remarks
api	Mandatory	String (20)	TXNVERIFICATION	For Transaction verification "TXNVERIFICATION" fixed
source	Mandatory	String (3)	It has to be only "OTS"	It's static, only OTS
merchId	Mandatory	int(15)	9135	Unique ID assign by NDPS to merchant
password	Mandatory	String (50)	Password provided by NDPS	Password Provided by NDPS
merchTxnId	Mandatory	String(50)	1234567890	Unique transaction ID provided by merchant
merchTxnDate	Mandatory	Date	2022-04-25	Transaction date must be in yyyy-mm-dd
signature	Mandatory	String (256)	eced0d634d6fa6637 6d40cbc1c5a812c5f 0fbc9c2fe756c459f5 e39c3455ba7ebb90 53cd3d3b37191d9d 20e794baf4923b3e1 f45c59c8ca34f9be6 03f89157	Signature generation using logic provided by NDPS - encrypts merchID, password, merchTxnID, amount, txnCurrency, api, reqHashKey
amount	Mandatory	double (12,2)	10.00	Transaction Amount
txnCurrency	Mandatory	string (5)	INR	The currency code in which the payment is made

3. Signature Generation Logic:

- For any given transaction, the Transaction Status API's request and the following response signature fields to be generated using the shared Hashing code.
- Signature generation sequence [merchID + password + merchTxnID + amount + txnCurrency + api]
- The UAT request and response hash key are:

MerchId	reqHashKey	respHashKey
9135	ea59e6ee036c81d8b5	ea59e6ee036c81d8b6

4. Response Format:

Response to the transaction status request will comprise of the below illustrated encrypted Data.

It needs to be decrypted as per AES Decryption logic provided by NDPS.

a) Sample Encrypted Response Data:

encData=F5140AF9DC1B3DB7AFA300D9675EE72A54618F3A30059A96C668F7148BCA948D88451AEF2A7E37F625DD
C2B4E1822587E62A3287CA5788F51C2D78186D40C11ECA66EF7048D3B64306CD22251D3078D061D9B141A6E2DEA4
67F6BEF8C8E0A78674DF6650A4011E6462C853930F30D441CFF3F9AD4CA7443A5A3837E222E502BFF3F2D815A4DE
67F29A5AB49D1DE6FA3D5E11075B8399686F554AA764E710E3D5A7FBAF44A3D490E21851F92B8F2BBEB2C1572EB
FF02F15D14D34A5C94EC909FC1B7F9F5510EE8176E968C3C52DD3C41719DE50B4223C3AA87A9D8E228245F3057CA
102A37E7229B23CBF21ABF44B19D40B3E5B32C736FF5A910EA29EB2ECCE2035A3448C20DBC3FA1FC13235F93FF2
89216A758C597E7C770E3B510B607D6E5491FE16E82C068ECA6A88C29D80922332DC1C29F362F8A9EB938C66417CA
3900067EE15F394C5179C3DFC08C1EF92A1277638D8D9DDE1F3836E4219FB4B5BB00A45E58D531C4CEBA66A9BE0
C9DE2FB4F346F1C396EB252F98F260FB3B4977B01F56B809F4153307765B73CF56829857E4F4AD90542382AE7384D7
FCCA121BFD01D9F2DE6E216F903BBA123A95D51D09D8FECF5174BB78986AB8D48C21AEC6AA5B2D03C16BAB1F
903EAC9BC714AE92CFA0BCAB3B71BF1499B6407251582E3A9F5E93BE2CC7E3077AEB726F4D676FC100D618144E3
01AAD6720303FE2 094FF26FE868F137311883D6E580C15BBD9D13EBEA&merchId=9135

b) Sample Encrypted Data from obtained Response to Decrypt:

F5140AF9DC1B3DB7AFA300D9675EE72A54618F3A30059A96C668F7148BCA948D88451AEF2A7E37F625DDC2B4
E1822587E62A3287CA5788F51C2D78186D40C11ECA66EF7048D3B64306CD22251D3078D061D9B141A6E2DEA46
7F6BEF8C8E0A78674DF6650A4011E6462C853930F30D441CFF3F9AD4CA7443A5A3837E222E502BFF3F2D815A4
DE67F29A5AB49D1DE6FA3D5E11075B8399686F554AA764E710E3D5A7FBAF44A3D490E21851F92B8F2BBEB2C
1572EBFF02F15D14D34A5C94EC909FC1B7F9F5510EE8176E968C3C52DD3C41719DE50B4223C3AA87A9D8E2282
45F3057CA102A37E7229B23CBF21ABF44B19D40B3E5B32C736FF5A910EA29EB2ECCE2035A3448C20DBC3FA1
FC13235F93FF289216A758C597E7C770E3B510B607D6E5491FE16E82C068ECA6A88C29D80922332DC1C29F362F
8A9EB938C66417CA3900067EE15F394C5179C3DFC08C1EF92A1277638D8D9DDE1F3836E4219FB4B5BB00A45E5
8D531C4CEBA66A9BE0C9DE2FB4F346F1C396EB252F98F260FB3B4977B01F56B809F4153307765B73CF56829857
E4F4AD90542382AE7384D7FCCA121BFD01D9F2DE6E216F903BBA123A95D51D09D8FECF5174BB78986AB8D48
C21AEC6AA5B2D03C16BAB1F903EAC9BC714AE92CFA0BCAB3B71BF1499B6407251582E3A9F5E93BE2CC7E30
77AEB726F4D676FC100D618144E301AAD6720303FE2094FF 26FE868F137311883D6E580C15BBD9D13EBEA

c) Decryption of Response:

Merchant must pass the encrypted response along with Merchant Specific Response Encryption Key

[Pg. 10] and MID in the decryption method as illustrated below:

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

DataType	Name	Value	Description
String	decstr	BFC23F835C2840C8 2CCA60671	Encrypted response to the encrypted request triggered, that needs to be decrypted
String	Key	Key provided by NDPS, to decrypt the response	Key provided by NDPS, 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.

d) Sample Decrypted Response – Open Data:

Post decrypting the response successfully, merchant will get corresponding data in the below JSON format.

Response Parameters are obtained in the format illustrated below:


```
{
 "payInstrument": [
  {
   "settlementDetails": {
    "reconStatus": "PNRNS"
   },
   "merchDetails": {
    "merchId": 9135,
    "merchTxnId": "250420221",
    "merchTxnDate": "2022-04-25 13:14:57"
   },
   "payDetails": {
    "atomTxnId": 11000000216668,
    "product": "Mangeshtest",
    "amount": 1,
    "surchargeAmount": 0,
    "totalAmount": 1
   },
   "payModeSpecificData": {
    "subChannel": "CC",
    "bankDetails": {
     "bankTxnId": "0011000000216668325",
```

```
      "otsBankName": "Hdfc Bank",
      "cardMaskNumber": "401288XXXXXX1881"
     }
    },
    "responseDetails": {
     "statusCode": "OTS0000",
     "message": "SUCCESS",
     "description": "SUCCESS"
    }
   }
  ]
}
```


e) Specifications of API Response:

Parameter Name	Conditional/ Optional/ Mandatory	Data Type & Max Length	Sample Value	Content/ Remarks
reconStatus	Mandatory	String(10)	RNS	Reconciliation Status
merchId	Mandatory	int(15)	9135	Unique ID assign by NDPS to merchant
merchTxnId	Mandatory	String(50)	1234567890	Unique transaction ID provided by merchant system
merchTxnDate	Mandatory	datetime	2022-05-24 20:46:00	Transaction date must be in yyyy-mm-dd hh:mm:ss
atomTxnId	Mandatory	Numeric (16)	11000000216668	Unique transaction ID (NDPS)
product	Mandatory	string (50)	ACC01	Product Id provided by NDPS. Passed during the transaction initiation.
amount	Mandatory	double (12,2)	10.00	Amount to be paid
surchargeAmount	Optional	double (12,2)	0.00	surcharge amount
totalAmount	Mandatory	double (12,2)	10.00	Total amount [amount + surcharge amount]
subChannel	Mandatory	String(10)	BQ	Product used during Transaction
otsBankId	Mandatory	String(10)	2	Bank ID as per NDPS system
bankTxnId	Mandatory	String(20)	1234567	Bank Transaction ID
cardMaskNumber	Optional	String(20)	485498XXXXXX0465	Mask Card Number
statusCode	Mandatory	String(10)	0000	0000-Success
message	Mandatory	String(80)	Message for Status Code	SUCCESS

description	Mandatory	String(100)	Status Description	TRANSACTION IS SUCCESSFUL

f) Response Codes:

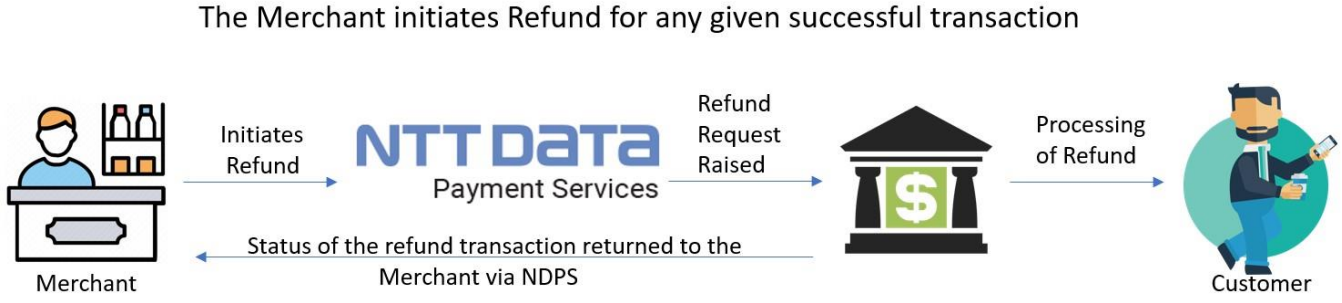
Error Code	Message	Description
OTS0000	SUCCESS	TRANSACTION IS SUCCESSFUL / FORCE SUCCESS *
OTS0101	CANCEL	TRANSACTION IS CANCELLED BY USER ON PAYMENT PAGE
OTS0201	TIMEOUT	TRANSACTION IS TIMEOUT
OTS0401	NODATA	NO DATA
OTS0451	INVALIDDATA	INVALID DATA
OTS0501	INVALIDDATA	INVALID DATA
OTS0600	FAILED	TRANSACTION IS FAILED / AUTO REVERSAL *
OTS0301	INITIALIZED	TRANSACTION IS INITIALIZED
OTS0351	INITIATED	TRANSACTION IS INITIATED
OTS0551	PENDING	TRANSACTION IS PENDING
OTS0951	SOMETHING WENT WRONG	UNEXPECTED ERROR

* Either of the mentioned description will be received.

Points to be Noted*-

1. 'OTS401' i.e., 'NO DATA' means that the transaction data is not available due to incorrect input.
2. Requery API will fetch the status only within 30 days from the day of transaction.
3. The description will be 'FORCE SUCCESS' if force success is enabled & as 'AUTO REVERSAL' if auto reversal is enabled. (For scenarios, refer part V - Callback API).

III. Refund API



1. DESCRIPTION

The **Refund API** is provided to the merchant to raise refunds against successful transactions. Request should be initiated, and the response is captured using the HTTP **POST** method.

a) URL: (Encrypted URL)

Production: https://payment1.atomtech.in/paynetz/refund?merchId=7&encData=3efsgdfdfhgdhgggfbg UAT: https://caller.atomtech.in/ots/payment/refund?merchId=9135&encData=59B80AF255F3D

Note*-
The URLs mentioned in this document are UAT. Merchants are required to be consume the Production link in a similar manner to conduct transaction.

b) Process Flow Diagram :

2. SINGLE PRODUCT REFUND

a) SINGLE PRODUCT SAMPLE REFUND REQUEST (OPEN DATA):

Request Parameters for Single Product Refund are to be shared in the format illustrated below:


```
{  "payInstrument": {
   "headDetails": {
     "api": "REFUNDINIT",
     "source": "OTS"
   },
   "merchDetails": {
     "merchId": 9135,
     "password": "Test@123",
     "merchTxnId": "11234456"
   },
   "payDetails":                {
"signature":
"7b5e32891aab691be2b7ed174caf8e34552943ba860b00e4e8320f2d2179364e4523f5c7d2459dcc1fd47383d1c
6b802703d7870301a1b4b3cc6f396675732d6",
     "atomTxnId": 11000000223788,
     "totalRefundAmount": 4000.00,
     "txnCurrency": "INR",
     "prodDetails": [
       {
         "prodName": "Mangeshtest",
         "prodRefundAmount": 4000.00,
         "prodRefundId": "189333256"
       }
     ]
   }
```

```
  }
}
```

b) Specifications of API Request:

Parameter Name	Conditional/ Optional/Mandatory	Data Type & Max Length	Sample Value	Content / Remarks
api	Mandatory	String (20)	REFUNDINIT	"REFUNDINIT" for Refund API
merchId	Mandatory	int(15)	9135	Unique ID assigned by NDPS to Merchant
password	Mandatory	String(45)	Test@123	Merchant transaction password
merchTxnId	Mandatory	String(50)	1234567890	Unique transaction ID provided by Merchant
atomTxnId	Mandatory	int(14)	100034566789	Unique transaction ID provided by NDPS
txnCurrency	Mandatory	String(5)	INR	The code of the currency in which the refund is to be made
signature	Mandatory	String(256)	7b5e32891aab691be2b7ed174caf8e34552943ba860b00e4e8320f2d2179364e4523f5c7d2459dcc1fd47383d1c6b802703d7870301a1b4b3cc6f396675732d6	Signature generation using logic provided by NDPS – merchId + pwd + merchTxnId + txn mode (RD/SL/SS) + amount + Currency + stage + reqHashKey
totalRefundAmount	Mandatory	double(12,2)	10.00	Refund Amount (Sum of all refund amount initiated against all products)
prodName	Mandatory	String(50)	NSE	Product ID passed in product name parameter.
prodRefundId	Mandatory	String(45)	234567890	Unique refund ID for Product
prodRefundAmount	Mandatory	double(12,2)	10.00	Product wise Refund amount.

SIGNATURE GENERATION LOGIC

- For above mentioned Refund API's request and their response signature fields to be generated using the following shared Hashing code.
- Signature type is HMACSHA512.
- Signature generation sequence [merchId + password + merchTxnId + paymentmode (RD/SL/SS) + total amount + txnCurrency + stage]

ii) Sample Request

- 9132Test@123OTS77SL3.00INRREFUNDINIT

iii) Generated Signature

fb0798ae1602bc0d395b7a12e8f08ab5d0ec64e90a2b9e7f09ed3db847ac86820edb1a0d557c51aec2e0166fae97
d4aa966a6f0d8512fd454e4deb906959c1bf

iv) Hash Keys

Below are the merchant id, request & response hash key –

MerchId	reqHashKey	respHashKey
9135	KEY1234567234	KERESPY1234567234

v) Encryption Key

- Below are the merchant id, request & response encryption key –

MerchId	encReqKey	encResKey
9135	58BE879B7DD635698764745511C704AB	7813E3E5E93548B096675AC27FE2C850

vi) Signature Generation Java Code:

```
import java.io.PrintStream;
import java.io.UnsupportedEncodingException;
import java.security.InvalidKeyException;
import java.security.Key;
import java.security.NoSuchAlgorithmException;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;

public class AtomSignature {
public static String generateSignature(String hashKey, String[] param) {
        String resp = null;

        StringBuilder sb = new StringBuilder();
        for (String s: param) {
                sb.append(s);
        }

        try {
```

```
                System.out.println("String =" + sb.toString());
                resp = byteToHexString(encodeWithHMACSHA2(sb.toString(), hashKey));
        } catch (Exception e) {
                System.out.println("Unable to encocd value with key :" + hashKey + " and input :" +
sb.toString());
                e.printStackTrace();
        }
        return resp;
}
private static byte[] encodeWithHMACSHA2(String text, String keyString)
throws NoSuchAlgorithmException, InvalidKeyException, UnsupportedEncodingException {
        Key sk = new SecretKeySpec(keyString.getBytes("UTF-8"), "HMACSHA512");
        Mac mac = Mac.getInstance(sk.getAlgorithm());
        mac.init(sk);
        byte[] hmac = mac.doFinal(text.getBytes("UTF-8"));

        return hmac;
}
public static String byteToHexString(byte byData[]) {
        StringBuilder sb = new StringBuilder(byData.length * 2);

        for (int i = 0; i < byData.length; i++) {
                int v = byData[i] & 0xff;
                if (v < 16)
                        sb.append('0');
                sb.append(Integer.toHexString(v));
        }
        return sb.toString();
}
```

c) SINGLE PRODUCT SAMPLE REQUEST ENCRYPTED DATA:

This is sample encrypted request data as illustrated below:

59B80AF255F3D63E3711DD833E01F98FC098E50B9E09B3C39D0605CC6E7F8AB998E84CBE365D6FD
578FB3B770F4B1FF9509C56A3CF7059FF2DA3FEC198AC427B1537DD6D7FE50E54750EE85F1D37133
86464291AF0207A9EE9709D6FA4E9D86AB393F5FF9A5EB09070D814EBCB5B0E6C37752D1182454B5
2C2FFC4477E01ABC1A7DCF565234854F774218D6DCEB109331026E2E26C102B451CDFAAE85BAAC
E6DF5966319E2D550CF8177D10AB089BBDDF9EC391C2A7DAC3C5D6471C3056FEDAD11DAC3A6C
EABBE12D13A3F9C0EAC1CB1715696DDDC740D0B2E66A777F87CDC7533A69E98D018908AAE19A6
734482B862EC31EDA59F60FEAC1BA27FCB59E8A284BC646CFFCB2414D3ACF695A509612DA251DB
84B82D2AEB3CA57950F456512D379DB0667F48D9DAE85B0078B576C02F57F8B1D1D7E718F2F7604B
8C9B3CBC9B50D6AA7EE4530C63680117ADEC402E216729DD663F1CD825DF24561B7C43163A698A2
19E7737A20C82C1AFAC550829ED58F2D625FCB1A04EBC4A7CCB334A3491B04B2592C5FD22532791
6945C190620BE3E50FA64E6834ED4A0917A61B5C4E344C74CA82AA05D7AF32ABC0220D800D38F5F
EA734157668C5AC342A0A8CDC96F1044229A96BCCEF856AA84D628CA21EA627A382E01E0F290DF
757FD3FA440A86101F5092E2093734F284D78FCA8301D5ABCB3BC215020FC653BC1BF90064E31D0B
50A7AD51962F9EA58C483B70A0D5897961C2D33A9C49DB65D3AB18BC87FC4CBE8E33B6A53ABA5
878CEE08F1ADA1893FC837D0D0B16DFFF6BD39BFBDF784771D5F356FF779288ECADE20D19F7B3B
ABCB323028B01F86413A6DD71548A25F73F37F7259AB67E07EB2BA2690F0BF6BAA2FA59FD04E5C3

34501338C05B2C4802E92E27B4DC1D5A0F226EBE4611D4628112AC97C3CEF0CA3A3C46FCAF71BB3A961F7E9

d) SINGLE PRODUCT SAMPLE REFUND REQUEST (ENCRYPTED):

https://caller.atomtech.in/ots/payment/refund?merchId=9135&encData=59B80AF255F3D63E3711DD833E01F98FC098E50B9E09B3C39D0605CC6E7F8AB998E84CBE365D6FD578FB3B770F4B1FF9509C56A3CF7059FF2DA3FEC198AC427B1537DD6D7FE50E54750EE85F1D3713386464291AF0207A9EE9709D6FA4E9D86AB393F5FF9A5EB09070D814EBCB5B0E6C37752D1182454B52C2FFC4477E01ABC1A7DCF565234854F774218D6DCEB109331026E2E26C102B451CDFAAE85BAACE6DF5966319E2D550CF8177D10AB089BBDDF9EC391C2A7DAC3C5D6471C3056FEDAD11DAC3A6CEABBE12D13A3F9C0EAC1CB1715696DDDC740D0B2E66A777F87CDC7533A69E98D018908AAE19A6734482B862EC31EDA59F60FEAC1BA27FCB59E8A284BC646CFFCB2414D3ACF695A509612DA251DB84B82D2AEB3CA57950F456512D379DB0667F48D9DAE85B0078B576C02F57F8B1D1D7E718F2F7604B8C9B3CBC9B50D6AA7EE4530C63680117ADEC402E216729DD663F1CD825DF24561B7C43163A698A219E7737A20C82C1AFAC550829ED58F2D625FCB1A04EBC4A7CCB334A3491B04B2592C5FD225327916945C190620BE3E50FA64E6834ED4A0917A61B5C4E344C74CA82AA05D7AF32ABC0220D800D38F5FEA734157668C5AC342A0A8CDC96F1044229A96BCCEF856AA84D628CA21EA627A382E01E0F290DF757FD3FA440A86101F5092E2093734F284D78FCA8301D5ABCB3BC215020FC653BC1BF90064E31D0B50A7AD51962F9EA58C483B70A0D5897961C2D33A9C49DB65D3AB18BC87FC4CBE8E33B6A53ABA5878CEE08F1ADA1893FC837D0D0B16DFFF6BD39BFBDF784771D5F356FF779288ECADE20D19F7B3BABCB323028B01F86413A6DD71548A25F73F37F7259AB67E07EB2BA2690F0BF6BAA2FA59FD04E5C334501338C05B2C4802E92E27B4DC1D5A0F226EBE4611D4628112AC97C3CEF0CA3A3C46FCAF71BB3A961F7E9

e) SINGLE PRODUCT SAMPLE REFUND RESPONSE DATA (ENCRYPTED):

This is sample encrypted response as illustrated below:

encData=F5140AF9DC1B3DB7AFA300D9675EE72A23B8826A6ED4001BD3E1B4760A58413789AF19B09B7AD993AEB53212F4D1933B49A281C3D8C99DF6BA0EDD1D629D2B4230718B7BC8E7453DC06D920D6373AA85241219116DF2C342968F45FE2CE4018DD13D4F5F7FAE2CD11D81C29CE29A5E5D9DDA03A4453D807C80A12DA8C899759A43112C727BA9644F4B56657B7916ED6F9BCF5B8A72F7E09721BE5E1A5CECF33DBAFA2CBB7B3DD3ED57207D8826D8C581350E66CBF35F1ABC8303CB0A623A903AC13462E95ED233FCE82242DEF8B3ED70EC7BADAA7B042CA761D815597A4DE4435450F62E80AEC84003AFA23CF29C7CCD506020BD01E2DF957E3A8757A5898785FEAA1473E8F67D0970AF447E11E27C3F11C8BCB58DFD02D629E0A059B27218C2E8859F706B9241779C6BE5279DACB9D48F76CB76A0E390352EF2BF5668385EBE1D979751FB0A6A761198D0ADD977DDDC017F0A16B7696FF7C77C39BA5E62B28E7B9D2852E93B8917D4070E91C0F3A6E96F0B98A0B6CC502C23C435E58925D6C61C773445277B2F5A6CCD77B1DC6550CDA8D135AE1F5E78685D4AE1C958444C146999C6A44FD328F7B0F90E7582AB20F7&merchId=9135

f) SINGLE PRODUCT SAMPLE RESPONSE DECRYPTED (OPEN DATA):

Response Parameters are obtained in the format illustrated below:

{
  "payInstrument": {
    "payDetails": {
      "atomTxnId": 11000000223788,
      "prodDetails": [

```
      {
        "prodName": "Mangeshtest",
        "prodRefundAmount": 4000,
        "prodRefundId": "189333256",
        "refundTxnId": 1519,
        "prodDescription": "Full Refund Initiated Successfuly",
        "prodStatusCode": "OTS0000"
      }
    ],
    "totalRefundAmount": 4000,
    "txnCurrency": "INR"
  },
  "responseDetails": {
    "statusCode": "OTS0000",
    "message": "SUCCESS",
    "description": "Full Refund Initiated Successfuly"
  }
 }
}
```


g) Specifications of API Response:

Parameter Name	Conditional/ Optional/Mandatory	Data Type & Max Length	Sample Value	Content / Remarks
atomTxnId	Mandatory	int (14)	100034566789	Unique transaction ID provided by NDPS
totalRefundAmount	Mandatory	double(12,2)	10.00	Refund Initiated amount (Sum of all refund amount initiated against all products)
txnCurrency	Mandatory	String(5)	INR	The code of the currency in which the refund is made
prodName	Mandatory	String(50)	NSE	Product ID passed in the product name parameter
prodRefundAmount	Mandatory	double(12,2)	10.00	Product wise Refund amount
prodRefundId	Mandatory	String(45)	234567890	Unique Refund ID for Product
refundTxnId	Mandatory	int(8)	12345678	Unique Refund Transaction id
prodDescription	Mandatory	String(50)	Refund Initiated Successfully	Product Description

prodStatusCode	Mandatory	String(7)	OTS0000	Product Status Code
statusCode	Mandatory	String(7)	OTS0000	Status code
message	Mandatory	String(10)	SUCCESS	Status Message
description	Mandatory	String(50)	Refund Initiated Successfully	Message description

3. SINGLE PRODUCT PARTIAL REFUND

a) SINGLE PRODUCT SAMPLE PARTIAL REFUND REQUEST (OPEN DATA):


```
{
  "payInstrument": {
    "headDetails": {
      "api": "REFUNDINIT",
      "source": "OTS"
    },
    "merchDetails": {
      "merchId": 9135,
      "password": "Test@123",
      "merchTxnId": "7777777"
    },
    "payDetails":                {
"signature":
"419e55e58cf5da634a369f66c48c8b1e6eb5635883c1596f3aeb9ea7b925313c9938b57f84b62721c3975f978c2
2be8b0c1e1921b368fa57a6c0e45073ae76d8",
      "atomTxnId": 11000000222715,
      "totalRefundAmount": 10.00,
      "txnCurrency": "INR",
      "prodDetails": [
        {
          "prodName": "Mangeshtest",
          "prodRefundAmount": 10.00,
          "prodRefundId": "567471258"
        }
      ]
    }
  }
}
```


b) SINGLE PRODUCT SAMPLE PARTIAL REFUND REQUEST DATA (ENCRYPTED):-


```
merchId=9135&encData=59B80AF255F3D63E3711DD833E01F98FC098E50B9E09B3C39D0605CC6E7F8
AB998E84CBE365D6FD578FB3B770F4B1FF9509C56A3CF7059FF2DA3FEC198AC427B1537DD6D7FE
50E54750EE85F1D3713386464291AF0207A9EE9709D6FA4E9D86AB393F5FF9A5EB09070D814EBCB5
B0E6C37752D1182454B52C2FFC4477E01ABC1A7DCF565234854F774218D6DCEB109331026E2E26C1
02B451CDFAAE85BAACE6DF5966319E2D550CF8177D10AB089BBDDF9EC391C2A7DAC3C5D6471C
3056FEDAD14283CF702F2D290F1E33C1A30C427DC28B2B048C072512FAA46563A8DC6A8CE2B1576
```

6B49563AE0812A2A8F071C43846FD4078CDC9419BBCA2218151AE3B6ED63B8347E4A3CCD73081B0
5A8FA9844FFB73029589AF18FF041F356CF0014F660CD200AC30785743BED208F37A096F8A92F3B8B
AFB80249DDED4081AA15D63C13E01A1D3FFD59829AE596293A0A2902F1B56579435E32141F79B0D
77051D5765F6656D7CF4EE410F5B476FB1A1C8AE3101B2E77A30288C42E33ACDD077E8541E78A1B
CFCA5D8181402BBE847BDDCD9186980CD954510F3FAFDD863672F4B730979C0B19333FFA0E0D6E5
40B08E7BE54DA7090E60A32372414FC0B545B4002240E272B27C986F64138CF906165ED8979B3BB6F
CFD175157CEE401079879ED8DF8544F1518C6BEE5E698B6A310FD77EF49D333D340C478603DC3365
5E3A370560D5A3DBFABB12850B7170522548819AC2E6843F61D84D91951C9FDE643747C961D41E77
2B944A5A2DB7F90F0E620C33738828228F286849D6C58EAD68AC63A345AF21C8E4F3A91AE669BD0
F320EF53E96B6C53563BE163E482D8802FB0E20E0AD34E37415777ECFA073FAC6001BFDB1D4D65C
84503271A739DE337B4E60BAFB70614646F8FFE35709F62D5929BF380EED82

c) SINGLE PRODUCT SAMPLE PARTIAL REFUND RESPONSE (ENCRYPTED):-

encData=F5140AF9DC1B3DB7AFA300D9675EE72A23B8826A6ED4001BD3E1B4760A58413789AF19B0
9B7AD993AEB53212F4D1933BFA19D7A948106E118F682B2646D9D56D43B5F5BE3A7078BE7D349D
C94BD8BA6A3E0EA82774C2B0E5EA3C7E509D0CC8FD808E5472806EEFD39235C187773C28C6F338
DE93CD3586DBF0CF7835720C3DE3A41B25E4E6A73AEEB68254D92367CA3EE37942190CF73FB674C
0262427CB3F2DC99391B5FF7D932A17346AC5F2F6E6B4C69E2EC2536A1B096979464A8BA7D64B1D
3F889A2CC720CBC479FFA749ECCF3DABF7D5DD20CA12FFCA9BCDB5C1DAE22A320F20381E2A2
C118ACA29E54F1B6C10740AB8B88F713B3FFD977E18AFB87797BBB56944704DCA98D5BB90E7506
A4754E722905B6E40A9AA39DBD1E0E7A14D5DA0653C5C76F2E7FB5AA82C8A14AADEB5EFAF0D
DF40C19D392C6A72D82727DEBC4A358727A2019519270A2764895CC993BBA41755DAE112C802AC1
8A2394D5459E51A0EA83F7566A5DCD6F6BCABF73F20195BDF525532A796E40D1BC250DCC4056BD
1018526106C1FC29CDE1B19242C65720AB1067ECB3C920670B2192518E18DE94DA8BE7FDCCE177C
D694628E96BAB5&merchId=9135

d) SINGLE PRODUCT SAMPLE PARTIAL REFUND RESPONSE (OPEN DATA):-

Response Parameters are obtained in the format illustrated below:


```
{
  "payInstrument": {
    "payDetails": {
      "atomTxnId": 11000000222715,
      "prodDetails": [
        {
          "prodName": "Mangeshtest",
          "prodRefundAmount": 10,
          "prodRefundId": "567471258",
          "refundTxnId": 1515,
          "prodDescription": "Partial Refund Initiated Successfuly",
          "prodStatusCode": "OTS0001"
        }
      ],
      "totalRefundAmount": 10,
      "txnCurrency": "INR"
    },
    "responseDetails": {
      "statusCode": "OTS0001",
```

```
    "message": "SUCCESS",
    "description": "Partial Refund Initiated Successfuly"
   }
  }
}
```


4. MULTI PRODUCT REFUND

c) MULTI PRODUCT SAMPLE REFUND REQUEST (OPEN DATA):-

Request parameters for Multi Product Refund are to be shared in the format illustrated below:


```
{
  "payInstrument": {
    "headDetails": {
      "api": "REFUNDINIT",
      "source": "OTS"
        },
        "merchDetails": {
           "merchId": 9135,
           "password": "Test@123",
           "merchTxnId": "TSTTXN543"
        },
        "payDetails": {
           "atomTxnId": 11000000229212,
"txnCurrency": "INR",          "signature":
"9cfae773f90fa8aad3e196293ffd0366ee90f987851e3b1762383b69842fb5be4c689ecbd406532f0aafc83b6047fae2a
530de2d111d9b82b079c8c51d7a61b6",
           "totalRefundAmount": 23.00,
           "prodDetails": [
              {
                 "prodName": "DHARAM_TEST",
                 "prodRefundId": "666645682",
                 "prodRefundAmount": 12.00
              },
    {
                 "prodName": "Mangeshtest",
                 "prodRefundId": "666645683",
                 "prodRefundAmount": 11.00
              }
           ]
        }
    }
}
```

d) MULTI PRODUCT SAMPLE REFUND REQUEST DATA (ENCRYPTED):-

merchId=9135&encData=59B80AF255F3D63E3711DD833E01F98FC098E50B9E09B3C39D0605CC6E7F8
AB998E84CBE365D6FD578FB3B770F4B1FF9509C56A3CF7059FF2DA3FEC198AC427B1537DD6D7FE
50E54750EE85F1D3713386464291AF0207A9EE9709D6FA4E9D86A893FF1215CC21E386967AE0C0E69
1D778A078BCC4616B702E3C7A86D907AA1C27F8666CCC143CBF0D6964C73C47E966697C18D86B6E
0332AC21724DAA395846F01D87853684385336EDE0C5B26C8F1B0C931AE006D97F42732DC16663217
5D892DEF7F8C2B91F043C96562E508992FC2D1C0DB3910EA9F774D92C7B5960B87C16098B54392F44
ED613096F24D060A60C4A611EB557DAC20D39DF9CEA26B6FAA7A95D1CF24E9E0E212D3A95AA84
3E6D388D3E537AB9EE0D2AEC83F14E6A0583E5E74B3150F0A0802885AF3CB208A9C8A55D15B950C
4BD43AFB03E85D35A93D1003958B8B1A75E6E36AE326100768D07E8DF458199911A89C3891CC965C
F4B9E78E0F262F1E30650DF0942F766172BECB3D928BA2C44C4C4CC82FF4A4E60C3D66B81DA2A51
FEB2D3C74E486CA6E3DF9496DEE04BA4E45CDB95B5DFF0B9A3A11FE59C869F8E86DF4F2FF15FB
C9A1CC0B058F51F1DA5C41258A390F7FECC052CBD22E13025924521864A42A6DC80B0B00C5F94C0
6BA40501C078D16F5951E06C58D124D10D6A4F5FE9CBB062B8EC9B2861AB6EC63AEF16A3F2EC3E
9A15820C69B1FA19B7B789936FBE3D1F83B6D970B9D02C87AC0B3EAECB06D62B90DA0F2CA36F5
ACBAA5FC98ADA3D6FFD8C9EEAEE55434DBA503C6D06A9AD648349BB32A4A62C366662C2AF58
DBDF6A6D26DAD5760C86A28B089A10AD7BC6149B7C34B5F426342BC1775085ABB9D5CE64C6B01
7DC976EF66AC1A03FDD89DB1C8689D7B84AD0E1EE745760FF626517E654822008B958A37098179D5
37BC5FB2F158D2CCB51D29FD6373C717891232198AFE5D90BECBF2F3C6836519BB9F596FFF76DA0
9123EDB5D4E002FDD6ADF616B8FAB10D69E660A0C07DA280004557DC0E5E96528E51DF075D45545
E93358193312959EEAF0B117C92E18B83F6EE4DEAE2BB7291BC6EAAC61D4F16E2EC3194EAB9B483
5FE547E36F2A9D3D5778AAF232C3687F2463CD9F042E5C49F64CC7D4281FFAEB31290FC641C1ECD
E2427300B557E8A2EAC1DB7E20BB581A6C5770B1E9DFFDC1CA909C0F94A5BFAE97C2FB73754C4E
3D91FC6FE763A032D925F6AD415D986793D000B3FB5942B5F0C6197B4BC72C3750B527681CAE7453
374CC68DB92155FCA92DFBCDF26E8F63B8CA390FC63A80A76FE20FC1524D412187115789689C496
A1D30A2980ABC6A84F4D045B63138B04B87AC1924F878A09

e) Specifications of API Request:

Parameter Name	Conditional/ Optional/Mandatory	Data Type & Max Length	Sample Value	Content / Remarks
api	Mandatory	String (20)	REFUNDINIT	"REFUNDINIT" for Refund API
merchId	Mandatory	int(15)	9135	Unique ID assign by NDPS to Merchant
password	Mandatory	String(45)	Test@123	Merchant wise password
merchTxnId	Mandatory	String(50)	1234567890	Unique Transaction ID provided by Merchant
atomTxnId	Mandatory	int (14)	100034566789	Unique transaction id provided by NDPS

txnCurrency	Mandatory	String(5)	INR	The code of the currency in which the refund is to be made
signature	Mandatory	String(256)	eced0d634d 6fa66376d 40cbc1c5a812c5f60fbc 9c2fe756c459f5 e39c3455 ba7ebb9053cd3d3b	Signature generated using the logic provided by NDPS – merchId + pwd + merchTxnId + txn mode (RD/SL/SS) + amount + Currency + stage + reqHashKey
totalRefundAmount	Mandatory	double(12,2)	10.00	Refund Amount (Sum of all refund amount initiated against all products)
prodName	Mandatory	String(50)	NSE	Product ID passed in the product name parameter.
prodRefundId	Mandatory	String(45)	234567890	Unique Refund ID for product provided by the Merchant
prodRefundAmount	Mandatory	double(12,2)	10.00	Product wise Refund Amount.

f) MULTI PRODUCT SAMPLE REFUND RESPONSE DATA (ENCRYPTED):

This is sample encrypted response as illustrated below:-

encData=F5140AF9DC1B3DB7AFA300D9675EE72A23B8826A6ED4001BD3E1B4760A58413789AF19B0
9B7AD993AEB53212F4D1933B50EB242EDE540C25CDD6D22E38AA99A36408BFC63741EA935431C3
ED8664FB7797C62B960690030B2F89A74D4FB06C9BA1C4C1AA850ED0B62A38F4CD8A5D98941CE6
EB59A06641798D808657EE46851C2A9E5AF20FBDA7288BC114B5FA275938D18C83A72C71AFE7BD
C622BA51DC6DEFE3630BDAE09131DEA1643EB42CD9C308D41B3A80D3E91CDD35396B0CC5B515
DFA6FCD3054F08F5F4BA8335AB140A6F886161CD7C6BED07011E9EE159CBEC8B3658377E2D1E492
57AEFB45D779C4C7D2963AEA53223BF5E7CCB981CCA824D71E6811336FC135AB3815CB700218A1
EB402B274B17E24B4E602E04BE39BB7D74A669164CD8C9CC6FB6AC85A864A5D518AF4C223244031
386BC95946834268FB691C75568D9F418D00182FB39291ECDB7C3640FEACED4A0D660CBCBDCCC3
B43E7B7D4AC09AEEBFD3FA47BBA933D5426949E12B02576D4FCBA547C1F080CDC8595C33C545E
E011CA9D859F8EC07F29D7F7438B5C113C5103F65B369DE360D5F4BC9D79DF55D002396677178190F
5971D7F324A2188DFFD388AF8306577C9168F38551A94BA1DCE50EFA107A794960D47683320C73AF
8A25B4BA544472EC146F9812E1409A52E60D435301130AD2C9A961D23961AE36DB2EB61D6D4DD20
787D6AF4622029B0D5684C20EE0C10FF632A9BBF53D51AA50ACF5D797BE7D52BC27107D93406469
BCA28A1005E26CAB3F6F262BEAE9E6D2594948A14D865B53EE3908B7B0DB40B90260F2EA68B3E0
26B952208F5847E6BDB693CD1593C87B78B9C6149F4022&merchId=9135

g) MULTI PRODUCT SAMPLE DECRYPTED REFUND RESPONSE (OPEN DATA):

Response parameters for Multi Product Refund are obtained in the format illustrated below:


```
{
  "payInstrument": {
    "payDetails": {
      "atomTxnId": 11000000229212,
      "prodDetails": [
        {
          "prodName": "DHARAM_TEST",
          "prodRefundAmount": 12.00,
          "prodRefundId": "666645682",
          "refundTxnId": 1569,
          "prodDescription": "Full Refund Initiated Successfuly",
          "prodStatusCode": "OTS0000"
        },
        {
          "prodName": "Mangeshtest",
          "prodRefundAmount": 11.00,
          "prodRefundId": "666645683",
          "refundTxnId": 1570,
          "prodDescription": "Full Refund Initiated Successfuly",
          "prodStatusCode": "OTS0000"
        }
      ],
      "totalRefundAmount": 23.00,
      "txnCurrency": "INR"
    },
    "responseDetails": {
      "statusCode": "OTS0000",
      "message": "SUCCESS",
      "description": "Full Refund Initiated Successfully"
    }
  }
}
```


h) Specifications of API Response:

Parameter Name	Conditional/ Optional/Mandatory	Data Type & Max Length	Sample Value	Content / Remarks
atomTxnId	Mandatory	int (14)	100034566789	Unique Transaction ID provided by NDPS

totalRefundAmount	Mandatory	double(12,2)	10.00	Refund Initiated Amount (Sum of all refund amount initiated against all products)
txnCurrency	Mandatory	String(5)	INR	The code of the currency in which the partial refund is made
prodName	Mandatory	String(50)	NSE	Product ID passed in the product name parameter.
prodRefundAmount	Mandatory	double(12,2)	10.00	Product wise Refund Amount.
prodRefundId	Mandatory	String(45)	234567890	Unique Refund ID for Product provided by the Merchant
refundTxnId	Mandatory	int(8)	12345678	Unique Refund Transaction ID
prodDescription	Mandatory	String(50)	Refund Initiated Successfully	Product Description
prodStatusCode	Mandatory	String(7)	OTS0000	Product Status Code
statusCode	Mandatory	String(7)	OTS0000	Status Code
message	Mandatory	String(10)	SUCCESS	Status Message
description	Mandatory	String(50)	Refund Initiated Successfully	Message description

5. MULTI PRODUCT PARTIAL REFUND

i) MULTI PRODUCT SAMPLE PARTIAL REFUND REQUEST DATA (Open Data):-

Request parameters for Multi Product Partial Refund are to be shared in the format illustrated below:


```
{
  "payInstrument": {
    "headDetails": {
      "api": "REFUNDINIT",
      "source": "OTS"
    },
    "merchDetails": {
      "merchId": 9135,
      "password": "Test@123",
```

```
    "merchTxnId": "TSTXN6000"
   },
   "payDetails": {
    "atomTxnId": 11000000229597,
    "txnCurrency": "INR",
    "signature":
"fb0798ae1602bc0d395b7a12e8f08ab5d0ec64e90a2b9e7f09ed3db847ac86820edb1a0d557c51aec2e0166fae97d4aa
966a6f0d8512fd454e4deb906959c1bf",
    "totalRefundAmount": 40.00,
    "prodDetails": [
     {
      "prodName": "DHARAM_TEST",
      "prodRefundId": "666645679",
      "prodRefundAmount": 30.00
     },
     {
      "prodName": "Mangeshtest",
      "prodRefundId": "666645678",
      "prodRefundAmount": 10.00
     }
    ]
   }
  }
}
```


ii) MULTI PRODUCT SAMPLE PARTIAL REFUND REQUEST DATA (ENCRYPTED):-


```
merchId=9135&encData=59B80AF255F3D63E3711DD833E01F98FC098E50B9E09B3C39D0605CC6E7F8
AB998E84CBE365D6FD578FB3B770F4B1FF9509C56A3CF7059FF2DA3FEC198AC427B1537DD6D7FE
50E54750EE85F1D3713386464291AF0207A9EE9709D6FA4E9D86AB393F5FF9A5EB09070D814EBCB5
B0E6C37752D1182454B52C2FFC4477E01ABC1A7DCF565234854F774218D6DCEB109331026E2E26C1
02B451CDFAAE85BAACE6DF5966319E2D550CF8177D10AB089BBDDF9EC391C2A7DAC3C5D6471C
3056FEDAD73D722B28F4FAE3F712BDA8355F929C0A0398A554DEB1605D0E1C33137CA40209F74CB
44A48238F436D20A97783FF04EB78C7A78D8530A12CAD276014BAA7C4C267FC46F9F126E624E383E
B8EE034592FA77CEE45CCDFCA5DFAE80C60CF2736EB948935D54C94ABC95CCB41017555EB4C317
C5B4F2A4A9C4BEC2A2EC2324AAF8662AAD58B5ED84D4E7EC13E01A08F5E89515718E26B30D26C2
9FB5061D6018502C6505A6FEA03F1176F12F80B1E873111E5643E4E07BC95597DE90E12D90DF365873
C570821AC5604133CE5FE7CF1C91FD61823B7EC410E6CCBDC98265C782B61A4510CDBF3190A29C3
9877F53DF05907BCA9689F2D62A06CBDBF4940EE57A2C1F344685C74B1C5EEF50301B202E1808033
B3F3E5FE2FD5335779B7AA63A6278716E099A6C3BE8787217F73CFCBEA66D922370F9B2981CDA4B
6C392C2A0D0B3813C4CD44C14BE7222766847478D2B927D8498B972102258A2BBF75965DB8A50BB8
070F00A2754DEFEE31FA70BED9460F4532AD4CCA0EB70554FF052BF36EAD0178140426299BE247B0
7B329E77D5DB424C30D923CE2DDB26377D4D9FD83A4C9ACDE026A1332C1F740F18606A0F651F189
A6B976D92D93519877AE2E0F90B9ACE9843679471290E429BC891EA4E3A24FFF8D77CC64D5AE9B8
2B5D8B06E7228A3C8FFAC34B43E0DDAC28F4067BD0D5241DBA5B4F3F2E4B0871160262352831D92
83784D59887CA3FA2B73A752F40759B205BC7621997E18AF60FAB7AAB20AC51845DA6096463120E4
A23711B9E0E217DD436439DC1DA66600C17CFB821FDDD64BE17F6C68A574A25EF02699EB766C9E3
```

iii) MULTI PRODUCT SAMPLE PARTIAL REFUND RESPONSE DATA (ENCRYPTED):-

encData=F5140AF9DC1B3DB7AFA300D9675EE72A23B8826A6ED4001BD3E1B4760A58413789AF19B0
9B7AD993AEB53212F4D1933BFDF939242EE65C1D846269F942D17F7CB783A5B5157D5FB5CCD05B3
A88FFDAF9FD5476DD2A195D965EE91C0E2B248A1F0E3BA63CE921E489A963B1C83052AB80DC2B3
35FBBB6B31A6D4C02670DFD8BD732829737AB583D73E4FFA13803E31AE5D97B0DA243378A70208F
3D23B7B64505F4C39F6E6AC1EC0FD3D5417F82FBC83AD50406308E5F137F526A6C171AF773649AFF
AD609A4D2B31588AD11E51B8EC5889320A6BCE2D9F0EF6D2076782C96737CA9CBCA4848BC1826E
70444952050666DDCA44F629BA5C8E402F394C6A88BB41E2500994C5421E85DE48DCF5D00882D26C
2D084C9E474144E8873080C15285AAFC561003D5FC75B1A5DC62632A9595CE0E0178217BF46F8D48
B327449B52F925D67E747D89B6FA41E21CADAAEDC32A82A4AF46FD19F3C21D9BB8F50CB6021343
41BB157720FB256FFF1CEFC9E0CE1D027687955249EF6D14665C7FCA6F6A4C2663985776C202274FF
AAD145F738B1083E38DC2F06FFE0F47AAFEC0553FBDCEDCEE2DB3A6044334A824DEDA95B7B56
CA8ED3A940965955F90757AA9A234E07F03E97617FA10E2D8186E838FAA159C1C803C5C4ECDD7A0
53D0DD40C82B0AD152CB897E53DED5F1256F6DA03FBB8C8E78CF1DEED1340DD1BA20D40264748
1650B348DCE45E4ABEDFBEBD604EBE81160C92F3344D75CCC94E72942CAE762FF13F2207E942768
9CE51590609A4B1CB16195CB7B0D7CF990338BA2A8AE3F97D311D5AC61DA70828AB8244D132F089
D160BD96278F730A3196A1DC6A82BD404F87814B0&merchId=9135

iv) MULTI PRODUCT SAMPLE DECRYPTED PARTIAL REFUND RESPONSE (Open data):-

Response parameters for Multi Product Partial Refund are obtained in the format illustrated below:


```
{
  "payInstrument": {
    "payDetails": {
      "atomTxnId": 11000000229597,
      "prodDetails": [
        {
          "prodName": "DHARAM_TEST",
          "prodRefundAmount": 30,
          "prodRefundId": "666645679",
          "refundTxnId": 1565,
          "prodDescription": "Full Refund Initiated Successfuly",
          "prodStatusCode": "OTS0000"
        },
        {
          "prodName": "Mangeshtest",
          "prodRefundAmount": 10,
          "prodRefundId": "666645678",
          "refundTxnId": 1566,
          "prodDescription": "Full Refund Initiated Successfuly",
          "prodStatusCode": "OTS0000"
        }
      ],
      "totalRefundAmount": 40,
      "txnCurrency": "INR"
    },
    "responseDetails": {
```

"statusCode": "OTS0001",
"message": "SUCCESS",
"description": "Partial Refund Initiated Successfully"
}
}
}

IV. Refund Status API

1. Description

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

a) Refund status tracking Process:

• Merchant will have to setup a system at their end to incorporate NDPS-Encryption logic [Pg. 11], to send encrypted data in request [Pg. 5] and to decrypt the response [Pg. 7,8].

• Merchant can track their refunds initiated on successful transactions via Refund Status API, wherein the merchant will have to send MID provided by NDPS 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 [Pg. 7-10].

• On initiating the Refund Status API, merchant will receive the status in encoded response [Pg. 7] as further explained under Response Format section. Merchant should decrypt this response via the decryption method in the shared NDPS-Encryption logic to receive the status of the refund.

Note*: This API is a **Server-to-Server** call; the response is captured using the HTTP POST method.

2. Request Format

i) SINGLE PRODUCT

a) Single Product Refund Status Sample Request (Open Request-JSON):

- Refund Status API request **UAT URL**: https://caller.atomtech.in/ots/payment/status?merchId=9135&encData=
- **Production URL** : https://payment1.atomtech.in/ots/payment/status?merchId=11223&encData=

Request and Response of Refund Status API will be encrypted using AES 512. ***Request Parameters are to be shared in the format illustrated below:***

```
{
 "payInstrument": {
  "headDetails": {
   "api": "REFUNDSTATUS",
   "source": "OTS_ARS"
  },
  "merchDetails": {
   "merchId": 9135,
   "password": "Test@1234"
  },
  "payDetails": {
   "atomTxnId": 11000000223788,
   "prodDetails": [
    {
     "prodName": "Mangeshtest"
    }
   ]
  }
 }
}
```


b) Specifications of the parameters of API Request:

Parameter Name	Conditional /Optional/ Mandatory	Data Type & Max Length	Sample Value	Content/ Remarks
api	Mandatory	String (20)	REFUNDSTATUS	For refund status API "REFUNDSTATUS" fixed
source	Mandatory	String	It has to be only "OTS_ARS"	It's static, only OTS_ARS
merchId	Mandatory	int(15)	9135	Unique ID assign by NDPS to merchant
password	Mandatory	String (50)	Password provided by NDPS	Password Provided by NDPS
atomTxnId	Mandatory	String(50)	1234567890	Unique transaction ID provided by merchant
prodName	Mandatory	String(50)	Mangeshtest	Product ID passed in product name parameter.

c) Single Product Sample Request Data (Encrypted):

59B80AF255F3D63E3711DD833E01F98FC098E50B9E09B3C39D0605CC6E7F8AB998E84CBE365D6FD578FB3B770F4B1FF95
09C56A3CF7059FF2DA3FEC198AC427B6FB4B5B8A14D7B7FBB9483E1E98BEC097E20FE90A1E770D7DA85F725F0DEE236C5
8CC6FD971C4EFCE70B74CDA3D36C526FC65394716963D584423B02800DB32C8E4A56082726E74099BA11D40644E669950
C9A0CDCED2A8D7740553478847E1AD73FB9EDF9403C5E165B25795494B8B87164B25DD52047AB1BD18A0229147A88AA
6042A2BBB6E12445D6A5E08C53C77B6AC63CF059A86E95EEABCA0EB8CCC4F9DE6B2A7E7AB9EA021152ED29D1B854CBB9

7FCCA34456A88D4AC8A6E1DAB7839A78DAC48AB74DFD66A575202963BC6E51A0F55EA63FF3BADF3FA55918B20FB58B3F
E20CF0B6F530C30043A634652B2DB87332EB80296347A1B37443FF370BC5B29DC39E78904F2E8553276F56F28E8EBC4B1B
89E378D3121CA064E98D F687F2B5

d) Single Product Refund Status Sample Request (Encrypted):

https://caller.atomtech.in/ots/payment/status?merchId=9135&encData=59B80AF255F3D63E3711DD833E01F98FC098E50B9E
09B3C39D0605CC6E7F8AB998E84CBE365D6FD578FB3B770F4B1FF9509C56A3CF7059FF2DA3FEC198AC427B6FB4B5B8A14
D7B7FBB9483E1E98BEC097E20FE90A1E770D7DA85F725F0DEE236C58CC6FD971C4EFCE70B74CDA3D36C526FC653947169
63D584423B02800DB32C8E4A56082726E74099BA11D40644E669950C9A0CDCED2A8D7740553478847E1AD73FB9EDF940
3C5E165B25795494B8B87164B25DD52047AB1BD18A0229147A88AA6042A2BBB6E12445D6A5E08C53C77B6AC63CF059A8
6E95EEABCA0EB8CCC4F9DE6B2A7E7AB9EA021152ED29D1B854CBB97FCCA34456A88D4AC8A6E1DAB7839A78DAC48AB74
DFD66A575202963BC6E51A0F55EA63FF3BADF3FA55918B20FB58B3FE20CF0B6F530C30043A634652B2DB87332EB802963
47A1B37443FF370BC5B29DC39E78904F2E8553276F56F28E8EBC4B1B89E378D3121CA064E98DF687F2B5

ii) MULTI PRODUCT

a) Multi Product Refund Status Sample Request (Open Request-JSON):

Request Parameters are to be shared in the format illustrated below:

```
{
 "payInstrument": {
  "headDetails": {
   "api": "REFUNDSTATUS",
   "source": "OTS_ARS"
  },
  "merchDetails": {
   "merchId": 9135,
   "password": "Test@1234"
  },
  "payDetails": {
   "atomTxnId": 11000000229597,
   "prodDetails": [
    {
     "prodName": "DHARAM_TEST"
    },
  {
      "prodName": "Mangeshtest"
    }
   ]
  }

}
}
```

b) Specifications of the parameters of API Request:

Same as mentioned in the Single Product API Request specifications. [Pg. 5]

c) Multi Product Sample Request Data (Encrypted):

59B80AF255F3D63E3711DD833E01F98FC098E50B9E09B3C39D0605CC6E7F8AB998E84CBE365D6FD578FB3B770F4B
1FF9509C56A3CF7059FF2DA3FEC198AC427B6FB4B5B8A14D7B7FBB9483E1E98BEC097E20FE90A1E770D7DA85F725
F0DEE236C58CC6FD971C4EFCE70B74CDA3D36C526FC65394716963D584423B02800DB32C8E4A56082726E74099BA
11D40644E669950C9A0CDCED2A8D7740553478847E1AD73FB9EDF9403C5E165B25795494B8B89DAA15AD882880B
3B1B12D8F217B9F11D04445B3B14C16F95339C8BE0C9C9945DB9A856DF700C7EE0D8639003F6B63E946519CCF758
478AF6F82B05A65B6C9ACA09FEEA802F148DA21C5177A0FA69DF71B7FBB668B1D9849B574E5333B708D18F153472
B45B2F51C79597694067668744BD012D1E18CE230577B4B2A0BFD3562F28A43490220000CB664785651F597B0814
BC99BF252EE1579CF3F3216A1B118C231E4CB920C5630A2B5CDAC34E08DAEB503EE73A690FCB97CEF61BFD7D6A45
60CDEAC47BDA034CDC7B7D76E73315E9D69AA8E9D9D97B61BA8A9FAE986DFF6A1

d) Multi Product Refund Status Sample Request (Encrypted):

https://caller.atomtech.in/ots/payment/status?merchId=9135&encData=59B80AF255F3D63E3711DD833E01F98FC0
98E50B9E09B3C39D0605CC6E7F8AB998E84CBE365D6FD578FB3B770F4B1FF9509C56A3CF7059FF2DA3FEC198AC42
7B6FB4B5B8A14D7B7FBB9483E1E98BEC097E20FE90A1E770D7DA85F725F0DEE236C58CC6FD971C4EFCE70B74CDA3
D36C526FC65394716963D584423B02800DB32C8E4A56082726E74099BA11D40644E669950C9A0CDCED2A8D77405
53478847E1AD73FB9EDF9403C5E165B25795494B8B89DAA15AD882880B3B1B12D8F217B9F11D04445B3B14C16F95
339C8BE0C9C9945DB9A856DF700C7EE0D8639003F6B63E946519CCF758478AF6F82B05A65B6C9ACA09FEEA802F14
8DA21C5177A0FA69DF71B7FBB668B1D9849B574E5333B708D18F153472B45B2F51C79597694067668744BD012D1E
18CE230577B4B2A0BFD3562F28A43490220000CB664785651F597B0814BC99BF252EE1579CF3F3216A1B118C231E4
CB920C5630A2B5CDAC34E08DAEB503EE73A690FCB97CEF61BFD7D6A4560CD
EAC47BDA034CDC7B7D76E73315E9D69AA8E9D9D97B61BA8A9FAE986DFF6A1

3. Response Format:

Response to the transaction status request will comprise of the below illustrated encrypted Data. It needs to be decrypted as per AES Decryption logic provided by NDPS.

i) SINGLE PRODUCT

a) Single Product Sample Response Data (Encrypted):

encData=F5140AF9DC1B3DB7AFA300D9675EE72A38E8F6216705CD1421752DDC52A603300966E76FED4CD64760761
444F0E62E3131387CE8C9AC9738B7416E7C6848CBD2B4382197ED30088314FB918093F628AA2E269F2E8F5F7E5CCF
2AE76C2F20AB482D4211F465F36D56172DF56BFBBD62A0C4CA34C40D2576BC729CB6206A093C1A3068A8687BF25
3764D51CE801F206598CA57A60B217E621F0ABBCF215C8698F60FA6E9A016CCD362858C1EB59D5AC138B6C737357
550203691064DEC864C84CCDEF5E974BB153E63E374388145F75793DE8764570915E2F3E30BA4F9F426F986713A080
B1939C1ED999A0638486706E94CDC06BADD937696BF12DB81CB6705B22BF68E3C1BE257A6646EBF27833720BD86
27F5A7F40D7B9249844C6D0E802E2D372057C706F78CDE028A365ED3DBC05061ECCD92297C08768ACD76707EAD7

EC1926B867CE28A037E3E3818C5783A68187344C7EBB0DF9482825B0ED00D81B8606A07C796A6E261CE5A098A490
4BE17D4D91DE4F3F00CDE1061F339C0E22085D84388CF24BCBA23295F4BC7854297E853&merchId=9135

b) Single Product Sample Encrypted Data from obtained Response to Decrypt:

F5140AF9DC1B3DB7AFA300D9675EE72A38E8F6216705CD1421752DDC52A603300966E76FED4CD64760761444F0E62E3131
387CE8C9AC9738B7416E7C6848CBD2B4382197ED30088314FB918093F628AA2E269F2E8F5F7E5CCF2AE76C2F20AB482D421
1F465F36D56172DF56BFBBD62A0C4CA34C40D2576BC729CB6206A093C1A3068A8687BF253764D51CE801F206598CA57A60
B217E621F0ABBCF215C8698F60FA6E9A016CCD362858C1EB59D5AC138B6C737357550203691064DEC864C84CCDEF5E974B
B153E63E374388145F75793DE8764570915E2F3E30BA4F9F426F986713A080B1939C1ED999A0638486706E94CDC06BADD93
7696BF12DB81CB6705B22BF68E3C1BE257A6646EBF27833720BD8627F5A7F40D7B9249844C6D0E802E2D372057C706F78CD
E028A365ED3DBC05061ECCD92297C08768ACD76707EAD7EC1926B867CE28A037E3E3818C5783A68187344C7EBB0DF94828
25B0ED00D81B8606A07C796A6E261CE5A098A4904BE17D4D91DE4F3F00CDE1061F339C0E22085D84388CF24BCBA23295F4
BC7854297E853

c) Decryption of Response:

Merchant must pass the encrypted response along with Merchant Specific Response Encryption Key [Pg. 10] and MID in the decryption method as illustrated below:

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

DataType	Name	Value	Description
String	decstr	BFC23F835C2840C8 2CCA60671	Encrypted response to the encrypted request triggered, that needs to be decrypted
String	Key	Key provided by NDPS, to decrypt the response	Key provided by NDPS, to decrypt the response
String	IV	Same as Key	Same as Key string
String	dec	new.ATOMAES().dec rypt(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.

d) Single Product Sample Response (Open Data-JSON):

Post decrypting the response successfully, merchant will get corresponding data in the below JSON format.

Response Parameters are obtained in the format illustrated below:


```
{
 "payInstrument": {
  "refundStatusDetails": {
   "refundDetails": [
    {
     "prodName": "Mangeshtest",
     "refundStatus": [
      {
       "refundTxnId": 1519,
       "refundAmt": 4000,
       "refundInitiatedDate": "2022-05-16",
```

```
            "remarks": "REFUND INITIATED",
            "prodRefundId": "189333256"
          }
        ]
      }
    ]
  },
  "payDetails": {
    "atomTxnId": 11000000223788
  },
  "responseDetails": {
    "statusCode": "OTS0000",
    "message": "SUCCESS",
    "description": "REFUND STATUS FETCHED SUCCESSFULLY"
  }
 }
}
```


e) Specifications of API Response:

Parameter Name	Data Type & Max Length	Sample Value	Content/ Remarks
prodName	string (50)	Mangeshtest	Product Id provided by NDPS. Passed during the transaction initiation.
refundTxnId	String(50)	1519	Unique transaction ID provided by merchant system
refundAmt	double (12,2)	4000.00	Total amount [amount + surcharge amount]
refundInitiatedDate	date (yyyy-mm-dd)	2022-05-16	Date of initiation of the refund
remarks	String(20)	REFUND INITIATED	Remark on the status of the refund
prodRefundId	String(45)	189333256	Unique Refund ID for Product
atomTxnId	Numeric (16)	11000000216668	Unique transaction ID (NDPS)
statusCode	String(10)	0000	Refund Status Code
message	String(80)	SUCCESS	Message for Status Code
description	String(100)	TRANSACTION IS SUCCESSFUL	Description of transaction Status

ii) MULTI PRODUCT

a) Multi Product Sample Response Data (Encrypted):

encData=F5140AF9DC1B3DB7AFA300D9675EE72A38E8F6216705CD1421752DDC52A603300966E76FED4CD64760761444F0E
62E3131387CE8C9AC9738B7416E7C6848CBD222CB2F4D38CB6D2E6E75E068F7EED7DD9AF6FB66B1E4FB6E9D300DBCAF451
307835A4CC76D48F81C90686EA84E1CD0DA2EC07F7535D141C3F5AA6509D03AEC0D2236C2DC59FF89943475890E580C18F
4799CDEC6BADCB99ADB0CCBCBF5757BCDDDBC151153812D3E59961608DFB4B9B30FED591512F752E77C712E8006351CB7
32841DD9E3DEC2260E252BDC8C0BB3A7BFE33D668ADD8A4F0E34EE4AC9589DC9A40AB6640C96D59BCB5C427D5D6BCFE1
D5D3CC1AE9117FB63D98918973EB2787770DE9A5184984228F33772FC55F597D11C0771E849F86657492E58AD117D7CD7E
13B37B32E3B27E885A75699CF522CC3D06FACCA18A904A149145E745308E3DC4F4DC7E64C340B0D5E92C9F70F47DAD1153

36ADD61A5269F4E5EFB9DF5EA2E32599B9FE2749B04A51D973CB0CC1CFEC2696AD31E9738711C3B21B05F32921E8F4E0896
2EC5D3F7265ADA89929CA0D44EB89912E8C3AE4C7704B7FB1A06453267F7CD758D2F52432CD20561D5D62B9A5842594DD
EEF4EBB17E72ECB3D1242A72DE9B701FF1F532D2D47DB07AAE857B99D9AD097B6579B47FB6EBCCDA8C8B2F5DDB5BA563C
10F812A3215AACA5E1A95AF8C78031161116AD897FDF71A56005D4C1FAB980318E9215AD0CFB95D18F8CDE04ED0ECD0C3
856DDAF2F27C5E5DE2DE74AD669D2C5BACE1F6D420D3A61F161B3ACF201FCAD2C962E8A7774496B973AD265D385B108E7
0630CEC48506284F4F2AA71D26CF696CE8A6A691F006A026955400D9AF59B727B3F2D5BD9CB8561DBE67F8FBC9F28C9488
4E8050B811AF99555A40946D9977D91ED0D3B75A15CF7AB2694846EA394548A37804013297B3A8AA67C440681698337 20D
9330A45982608A8F15B6303252A7C0EDFB3FDBF2A5B92446FE3F55DEE6E6A42D2505B874720B3B972D291EA83497E8C66B
9EB6684DD5E3ED0B1AC4876C85101138D36727E5C66E59CA9E559127095C3E52A60738204FB4A7FE1D71B9A1CFBB34D2DB
0A26675F500DE2A0B1A7974254B4D13AD956EC53619D16796D34919738B039CC7040E7A07734BA9D4583184DD5A04595B
DDA551D798AC&merchId=9135

b) Multi Product Sample Encrypted Data from obtained Response to Decrypt:

F5140AF9DC1B3DB7AFA300D9675EE72A38E8F6216705CD1421752DDC52A603300966E76FED4CD64760761444F0E62E3131
387CE8C9AC9738B7416E7C6848CBD222CB2F4D38CB6D2E6E75E068F7EED7DD9AF6FB66B1E4FB6E9D300DBCAF451307835A
4CC76D48F81C90686EA84E1CD0DA2EC07F7535D141C3F5AA6509D03AEC0D2236C2DC59FF89943475890E580C18F4799CDE
C6BADCB99ADB0CCBCBF5757BCDDDBC151153812D3E59961608DFB4B9B30FED591512F752E77C712E8006351CB732841DD
9E3DEC2260E252BDC8C0BB3A7BFE33D668ADD8A4F0E34EE4AC9589DC9A40AB6640C96D59BCB5C427D5D6BCFE1D5D3CC1
AE9117FB63D98918973EB2787770DE9A5184984228F33772FC55F597D11C0771E849F86657492E58AD117D7CD7E13B37B32
E3B27E885A75699CF522CC3D06FACCA18A904A149145E745308E3DC4F4DC7E64C340B0D5E92C9F70F47DAD115336ADD61
A5269F4E5EFB9DF5EA2E32599B9FE2749B04A51D973CB0CC1CFEC2696AD31E9738711C3B21B05F32921E8F4E08962EC5D3F
7265ADA89929CA0D44EB89912E8C3AE4C7704B7FB1A06453267F7CD758D2F52432CD20561D5D62B9A5842594DDEEF4EBB
17E72ECB3D1242A72DE9B701FF1F532D2D47DB07AAE857B99D9AD097B6579B47FB6EBCCDA8C8B2F5DDB5BA563C10F812A
3215AACA5E1A95AF8C78031161116AD897FDF71A56005D4C1FAB980318E9215AD0CFB95D18F8CDE04ED0ECD0C3856DDAF
2F27C5E5DE2DE74AD669D2C5BACE1F6D420D3A61F161B3ACF201FCAD2C962E8A7774496B973AD265D385B108E70630CEC
48506284F4F2AA71D26CF696CE8A6A691F006A026955400D9AF59B727B3F2D5BD9CB8561DBE67F8FBC9F28C94884E8050B
811AF99555A40946D9977D91ED0D3B75A15CF7AB2694846EA394548A37804013297B3A8AA67C44068169833720D9330A45
982608A8F15B6303252A7C0EDFB3FDBF2A5B92446FE3F55DEE6E6A42D2505B874720B3B972D291EA83497E8C66B9EB6684
DD5E3ED0B1AC4876C85101138D36727E5C66E59CA9E559127095C3E52A60738204FB4A7FE1D71B9A1CFBB34D2
DB0A26675F500DE2A0B1A7974254B4D13AD956EC53619D16796D34919738B039CC7040E7A07734BA9D4583184
DD5A04595BDDA551 D798AC

c) Decryption of Response:

Same as mentioned for Single Product API Decryption of Response. [Pg. 5 – 3-i)c]

d) Multi Product Sample Decrypted Response (Open Data):

Post decrypting the response successfully, merchant will get corresponding data in the below JSON format.


```
{
 "payInstrument": {
  "refundStatusDetails": {
   "refundDetails": [
    {
     "prodName": "DHARAM_TEST",
     "refundStatus": [
```

```
        {
          "refundTxnId": 1565,
          "refundAmt": 30,
          "refundInitiatedDate": "2022-05-31",
          "remarks": "REFUND INITIATED",
          "prodRefundId": "666645679"
        },
        {
          "refundTxnId": 1567,
          "refundAmt": 210,
          "refundInitiatedDate": "2022-05-31",
          "remarks": "REFUND INITIATED",
          "prodRefundId": "666645679"
        }
       ]
      },
      {
       "prodName": "Mangeshtest",
       "refundStatus": [
        {
          "refundTxnId": 1566,
          "refundAmt": 10,
          "refundInitiatedDate": "2022-05-31",
          "remarks": "REFUND INITIATED",
          "prodRefundId": "666645678"
        },
        {
          "refundTxnId": 1568,
          "refundAmt": 140,
          "refundInitiatedDate": "2022-05-31",
          "remarks": "REFUND INITIATED",
          "prodRefundId": "666645678"
        }
       ]
      }
    ]
  },
  "payDetails": {
    "atomTxnId": 11000000229597
  },
  "responseDetails": {
    "statusCode": "OTS0001",
    "message": "SUCCESS",
    "description": "REFUND STATUS FETCHED SUCCESSFULLY"
  }

}
}
```


e) Specifications of API Response:

Same as mentioned for Single Product API Response specifications. [Pg. 8 3-i)e]

iii) Status Codes:

Error Code	Message
OTS0522	Invalid Password
OTS0510	Total refund amount and sum of prod refund amount mismatched
OTS0509	Invalid Transaction ID
OTS0508	Invalid Merchant Transaction ID
OTS0511	Invalid Product Refund Amount
OTS0510	Invalid Product
OTS0000	REFUND STATUS FETCHED SUCCESSFULLY
OTS0401	NO RECORDS FOUND FOR MERCHANTID/ TRANSACTIONID/ PRODUCTID
OTS0951	Please Check The Request

V. Callback API

1. Description

Callback API is an API in which NTT Data payment services system posts transaction response through server-to-server mode in name value pair format on configured merchant's URL in NTT Data payment services system.

2. General process flow:

The general process flow of the call-back API is as follows:

- The end customer after purchasing product/service on merchant websites opts for payment, he either gets redirected to NTT Data payment service page or requested through API in case of seamless request.
- The user selects his choice payment option and provides credentials for authorization
- Once the authorization is completed the transaction is executed. For transaction to get successfully executed, the IP and domain URL needs to be whitelisted and configured.
- Once the end user completes transaction successfully and if callback URL is configured for that merchant, then the response will be posted on return URL as well as callback URL in both scenarios' success/failure.

3. Pre-requisite:

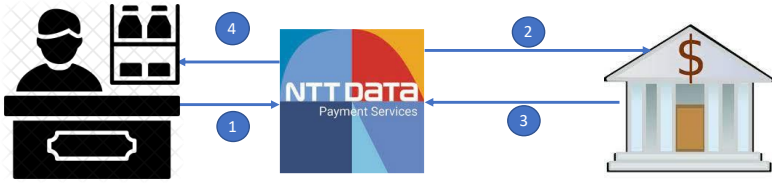
Merchant's IP and domain URL need to be whitelisted and configured at NTT data payment system and also NTT data payment server also needs to be whitelisted at their end.

4. Callback posting parameters:

Parameter Name	Conditional/ Optional/ Mandatory	Data Type & Max Length	Sample Value	Content/ Remarks
merchId	Mandatory	int(15)	8952	Unique ID assign by ATOM to merchant
merchTxnId	Mandatory	String(50)	1234567890	Unique transaction ID provided by merchant system
merchTxnDate	Mandatory	datetime	24-12-2019 20:46:00	Transaction date must be in yyyy-mm-dd hh:mm:ss
amount	Mandatory	double (12,2)	10.00	Amount to be paid
surchargeAmount	Optional	double (12,2)	0.00	surcharge amount
prodName	Mandatory	string (50)	NSE	Product Name
prodAmount	Optional	double (12,2)	10.00	Product wise amount division as provided by merchant
totalAmount	Mandatory	double (12,2)	10.00	Total amount [amount + surcharge amount]
custAccNo	Optional	String(45)	100000036600	Customer account number
clientCode	Mandatory	String(45)	12345	As provided by merchant
txnInitDate	Mandatory	datetime	03-08-2020 22:09:51	Transaction Initiation Date
txnCompleteDate	Conditional	datetime	03-08-2020 22:09:54	Transaction Completion Date for all successful txn.
subChannel	Mandatory	String (10)	BQ	Sub Channel(Payment Product will be displayed here)
otsBankId	Conditional	String (10)	2	Will share Atom bank ID
otsBankName	Mandatory	String(60)	HDFC Bank	Bank Name
bankTxnId	Mandatory	String(20)	UbiBgQp3Dwri0S347rge	Bank Txn ID
cardType	Conditional	String(20)	VISA	Card Type will be presented during a card based txn
cardMaskNumber	Conditional	String(20)	485498XXXXXX0465	Mask Card Number will be presented during a card based txn
statusCode	Mandatory	String(10)	OTS0000	OTS0000
message	Optional	String (80)	SUCCESS	Message for Status Code
description	Optional	String(100)	TRANSACTION IS SUCCESSFUL	Status Description

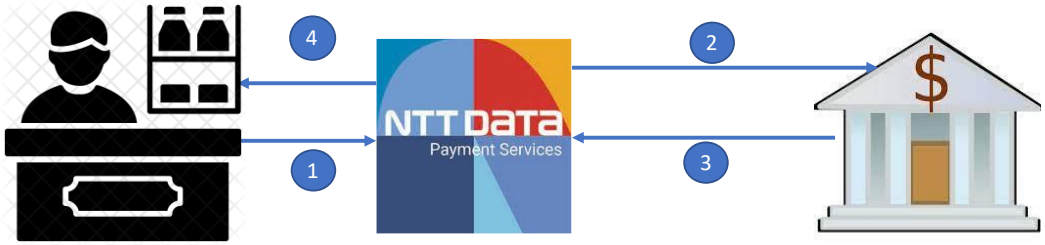
5. Flow of call-back API:

Call-back API- Non-Challan transactions



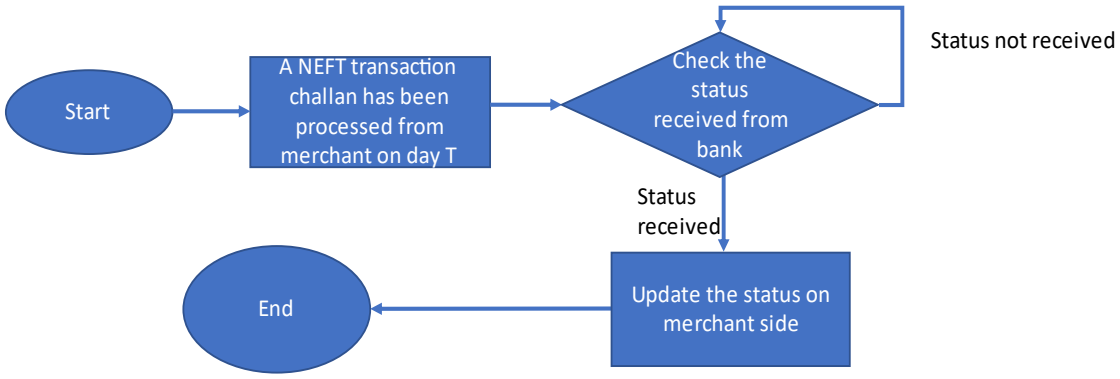
1. NTT receives payment request from merchant of the transaction carried out by end customer
2. NTT DPS forwards this request to bank
3. NTT receives response from the bank of the transaction being successful/failed till T, T+1, T+2
4. NTT posts the transaction at merchant's server through callback API

Call-back API- Non-Challan transactions after T+2 days



1. NTT receives payment request from merchant of the transaction carried out by end customer
2. NTT DPS forwards this request to bank
3. NTT receives response from the bank of the transaction being successful/failed after T+2 days
4. NTT posts the transaction at merchant's server through callback API as forced failed autoreverse the transaction to pay the end user.

Call-back API- Challan transactions



6. Sample response format:

Currently, the response is sent in JSON format to the merchant server. The response is posted in JSON format. The format is JSON.

a) Sample Encrypted Request Data:

The encrypted form of the response which will be received by merchant will be

encData=F5140AF9DC1B3DB7AFA300D9675EE72AFE09D27235E356AFAC21032BB1703A7EF4E87847263EA2FA
702E373401BED367E2E6A3EDD65389EB47D1FC8630C7E708857FF81B6A671FFF0102399E14016A9FE53552A20
6D95F5EC628A61822F0CC9506DB24AE82F3E1F478911D7495F4A6D105ADBAF90FBE56E2E04CB55EA17E03FF5
A09EBBC84980F72D37780A6D928BAB40A49593CD819DC0AFB6EA820B065E5E1E3367E9C1D77C2153B190E63
91B53FF0F26734B758EEAF32CFED1AD2D9FAF5FD8C4291AB992A7C84BBB7CC2BAA556041EE0E077F0E8979F6
1E5BDD9B5A3D2B0D494AF4AF1E6AD32346B108983D64F916E2C027FFF4E4715808826EC56CD3992B014D4B9
30496E9CBEFC9765BD4AC3267747287096B0C264131EFB964D279341D34C46B6463BF2A983F5F4E32338C576
9108DE2F011E4CFEDB3213D55A74E85615DB9A569E7C1EE53BE70CA6931D3F3C51F15E843426B60C3D320BF2
5242C6945440485210D6F76E6310D480D3ABDB2D8D7D12ED4B892DA6B412D61A36BD125339E783F0912AF64
9758F617F683C4D04421D48E570D45726FD3C82E2B0C5C2CE209247ED65830E72B27F2532F3A876ED98A783B
4036971B97C55EC3CEF5BD2E4D9B3123ACF973DC1B59AB255D6FD87312BA67433E4C15C1AD188FDC2D784EE
6B2B25C9A23E34F819A42034296997705F231C61B0C3C61AA4FD1237FE8BCAB6D238A573022E3E89F751D080
E8A18142BEA9663608A4E88172C1147680EB3B0DED793127A5E5F8935CF116A2C72F4205533B94F24836E6BB
D83D8922DF6B1DDF33668659D50C9F9410625310086609E53B2B1D1F42A137D2EB63B8547B994059A7E24277
66F0E27AE9351D9EB20D8CAC16CA2E2EB08A7B0B47CE0B8347DE157EDA051CC274AED624629649FC8970902
A8294F02C52B17ECDCB22EF839243CBC299A9227A85AF34C0363BB4E48D02854F8C6&merchId=9135

b) Sample Decrypted Response (Open Data):

{
"payInstrument":{
"merchDetails":{

```
    "merchId":9135,
    "merchTxnId":"OTS77",
    "merchTxnDate":"2022-05-02 18:43:44"
  },
  "payDetails":{
    "amount":3.00,
    "surchargeAmount":0.43,
    "atomTxnId":11000000219503,
    "prodDetails":[
     {
       "prodName":"Mangeshtest",
       "prodAmount":3.0
     }
    ],
    "totalAmount":3.43,
    "custAccNo":"123456789",
    "clientCode":"32454",
    "txnInitDate":"2022-05-02 18:44:00",
    "txnCompleteDate":null
  },
  "payModeSpecificData":{
    "subChannel":[
      "NB"
    ],
    "bankDetails":{
      "otsBankId":1,
      "otsBankName":"State Bank of India",
      "bankTxnId":"FueLffDEmCnahpAyBw2s"
    },
    "cardDetails":{
      "cardScheme":null,
      "cardMaskNumber":null
    }
  },
  "responseDetails":{
    "statusCode":"OTS0000",
    "message":"SUCCESS",
    "description":"TRANSACTION IS SUCCESSFUL."
  }
 }
}
```


Scenario 1-

If a transaction get success at atom end, on real time NDPS will post "**message** = **SUCCESS**" response with help of callback API.

Scenario 2-

If on T-day, the transaction is pending at NDPS end, after bank's re-query response on same day if transaction is success at bank end then status will be changed from pending to success at NDPS end, then at NDPS will post "**message =SUCCESS**" response on Callback API.

Scenario 3-

If on T-Day, the transaction is pending at NDPS end, on T+1-day post reconciliation if transaction is success then status gets changed from **pending** to **success** at NDPS end, then NDPS will post "**message =SUCCESS**" (Force success) response with help of Callback API. If it's Failed response will be "**message =FAILED**".

Scenario-4-

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

Scenario-5-

Until and unless we are not getting response (Success/Failed) from bank end, NDPS will not trigger Callback API. Till T+2 if we're not getting response from Bank, NDPS 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 NDPS system, same will be marked as Auto Reversal.

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 NDPS end, then NDPS will post "**message=SUCCESS**" response with help of Callback API. In This scenario, no real time response will be posted.

7. Response codes

Error Code	Message	Description
OTS0000	SUCCESS	TRANSACTION IS SUCCESSFUL
Any Other response	-	TO BE TREATED AS FAILURE

VI. Settlement API

Description: Settlement API is provided for merchants to fetch the settlement report for transactions processed for any given time interval.

a) 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 whitelisted IPs to get the required report.

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

4) If there are records in for the merchant ID for the settlement date the Settlement File isgenerated.

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

UAT Testing Credentials:

UAT:
URL= https://titanuat.atomtech.in/SettlementReport/generateReport

merchantId = 7760

settlementDate = 2019-03-25

responseType = json;

Production Credentials:

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

merchantId = Merchant Production MID

settlementDate = Settlement Date

responseType = json;

Sample Response –

.CSV

b) 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 whitelisted IPs to get the required report.
3) The parameters are “merchantId”, “settlementDate” (date should be in format YYYY-MM-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= https://titanuat.atomtech.in/SettlementReport/generateReport

merchantId = 7760

settlementDate = 2019-03-25

responseType = json;

Production Credentials:

Production:

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

merchantId = Merchant Production MID

settlementDate = Settlement Date

responseType = json;

c) 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"
}
]
```


d) 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"
}
]
```


e) 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 whitelisted IPs to get the required report.
3. The parameters are “merchantId”, “settlementDate” (date should be in format YYYY-MM-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= https://titanuat.atomtech.in/SettlementReport/generateReport
merchantId = 7760
settlementDate = 2019-03-25

responseType = json;

Production Credentials:

Production:

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

f) 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"
}
]
```


g) 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"
}
]
```


h) Status Code:

Status Code	Status Description
"00"	Mandatory parameter value can't 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
S7	No data found

VII. 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 whitelisted IPs to get the required report.

3) The parameters are “merchantId” and “settlementDate” (date should be in format YYYY-MM-DD 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:

Encrypted URL=
https://titanuat.atomtech.in/SettlementReport/generateReport?merchantId=7760&encdata=F1F7E46B85600F3CBB86E848B8F80149FF27D16E2D3B5B002081007CF73F90D15CEC7C147057045DCBCDDCF80A3661AC9460947D5940EB843C1CB31BC686F9C9C4670FD823D9F6DB062097FDB8F5337CF3B348D85BECB776CD77B4262D6D2D7C

Note*: For Open request refer Non-AES Section

Production Credentials:

Production:

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

Open Request

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

Sample Response –

.CSV

a) 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 allowsonly whitelisted IPs to get the required report.
3) The parameters are “merchantId”, “settlementDate” (date should be in format YYYY-MM-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= https://titanuat.atomtech.in/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

DatetransactionId =

Transaction Id responseType

= json;

b) Sample Encrypted Data (Settlement Date wise)–

18642F36C9661C57F6518466E8E9539BB5F48B31D83BF0F632AC9A80D3532B1C8B12
D1730A1A53329157828C84256D366654791D30553C71718A17E004C3CB85D3C605FC
E0B6B594CBA67D53DCEBC05896299468ABFEC540C52FF3D4A31361EBE6CE467F43
8D504414B8F66C11731141E6BA419C9175F5C5F0C6BA8CBBA589B78BFF61E70CB06
B756CECCFB13F39E034A60342B82FFB7CA657BEBBFF5F8DA7BAA02AD8DE79085B
DBE1A4515E1EA389304680CA62131A526B3B9E8234ED10836DF067348F407DA03778
F76E5D751BC26563B514B6D0206B2959208419F84C1ED88DC81A6E4CEA71507722D

59597D962B83E6129BD7197D98AE778E298326B854183014A10A2ECDE46E35E7E676
E5240D7D5B0663254D6661B933D6FEFDC43DD8A2204996B67785836CF6E98502545
A1CFE6B4007E1E5B22832D50D234C92007E08D2417785AD2AE66594F68F8B9E469F
4020037AA60AB50C7704F3D545039ECC1AD95CC6133D9BB9EC271E85723E4D96598
B42551CA09897764B3EAC43CB633B09D6ED292183CB2EBDCA18A3A313594B18FEC
3A949613EE3145C40037EF58C666D3455567BFCFAA47CAEA5182212F59D103EC6B4
B30CABBEDC6EE4728ADE14A34D52AD266BD2C25AA9519323D2004EA76DDA11E7D
BB023AC18AD82CC47F78F64CA3B82CC24B9A07C1F62830A9DACFCE410AD5D3F44
F995035B947C7CBA66B88D77847C5F92487E3BD5699157CD1BA040681F84818BB57
A782F89F7FAE855C9CDC5A21712C397A4D69DA17AF61E4A35BCFCD2489E3065ACB
B42125FFE36DFA65087FEB67C0507B0D52509FCC1AF56AFD58C789AC4AC6B432AA7
EA135FA9FB29CAA6485DF1AADF1929373A79542BED43618ECDC92543F9A7F85296B
CF15613215A85160FDD7F9D05687618C92D6D232CC148F3C27CBB78BB17F1E5277C
BB77DCAC1A7A1988950C34112B293099E604BB7340B4D54690821CF33D5BF51C9A7
845891C11A2B73137296B07D0D4FA5737A78819BAD6B41407049F0734DC3A0AA931D
04B642AFE7DE66247B8679DC185A6320EC674528A9CB85637E1BED60768AE5E7063
EF1522F50FA5FF077BDA5B0DCFB866050D8324152C6C9FF5C671218127F1D97A998
B2E4F7C20DED73F9650865CE596C190B37348E73CFFA5407A1CF99058A5C6C28157
D37B3E2FD9F2DB8564A0F0AD044A80B5A629E5108CBD89133EB7CE8E57F0367800D
F61CCFB4B6E86369167FF1ADFB605358A12FD6F5E92F91FFAF22FDCD79B0B2D799
CAB220B0B18617E54B9878E2D6A522B1E04DF2D863175D89F19252DBD46BF3AA45F
4BB9D0CA6676774A2520C25B3213DE5462DF689D91B0F32C33DFB416B3C3CBA448E
F896716276120267B4245ED5A26B50F1607FE138F5568BC9B24EB2073F0628E7A65D
34FFFFA56C5F00BE35645367F068F6A829073E335CFC02F9EB691C6550F4D222F0AB
C25A220C793C46CD442CF787DEA8C50B82037AA8C1ECBB9CB10D69A85468E9B7C8
17C4248395D8579556C3360B4FBEF1D4513A10D40FF142274A2CCE857D5792B8598B
0D7D941FC195C2BAEF0962E99CE9FC1E1AF461C58E2450CC2FEE283D400176288D
25E927A876BC41E958EB438B89DD7329BAF96ED0DEDFE43E429F85D7D84F49E5F2
2E27F13CED7682E9C1E0F17069A45DD71F4BE200774AB21C699DC8024C6410A398E
404E2FEF1140100039C014705A3A441E4503A86284B84B9B3D62F717F40A6C69EDCD
893F4A6A5260D9B2CA86A075640431976DBC46C3C652FB549BC9F38EA8176CFCFC5
F242060BBD9F2E8039E1786DE2AE1206246A8EA1B1F5D86F6BB2602D88E277267CB
765B3478C23A594E257F6C65E7B3E5BE85A01882150B34C233CB3DCFE3239906AE8
51E6CD4A0D56481873230FF820DF7D087D82F901C8416E66A05C681D40319D8A0023
8BFB409CDC5026F2FF11486B368FE856558ACC8661C96E7833F39E9BC1D9901F626
1E972565C4E51D0C8B8C0167715FA387AE004398F14907612425A6156CFD9D514C0A
92943AA3EF2A2A245EE22CA43D3BB69C002BB1D39C10ACF8A72080298403328976C
776EFC8A78CCF3719C272577D3EE3F05D1701F9F18EC971CE7A6D024B17E8AE19D
65B36ADE7341D22641D62F772EECC17EF8F98A37D68903E717AB32CB0429E2D4E09
F03BB071D1397C4F93EA3BE5E9977E8733ADF2DFA2C9EE2EA3D946D6A3398B4BD7
93E43B04CE510A7A7A3FAB04365B897FAA6677824FDE0EB89A05820E61D685B1DCC
E5003F13D128BEAC16808F599BFEE02AE4BE429CB7F1843C2CDAD7F9D62CB2DD3
DAD8B2B7E6B63D0818D

c) 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"
}
]
```


d) Sample Encrypted Data (Transaction Id wise)–

18642F36C9661C57F6518466E8E9539BB5F48B31D83BF0F632AC9A80D3532B1C8B12
D1730A1A53329157828C84256D366654791D30553C71718A17E004C3CB85D3C605FC
E0B6B594CBA67D53DCEBC05896299468ABFEC540C52FF3D4A31361EBE6CE467F43
8D504414B8F66C11731141E6BA419C9175F5C5F0C6BA8CBBA589B78BFF61E70CB06
B756CECCFB13F39E034A60342B82FFB7CA657BEBBFF5F8DA7BAA02AD8DE79085B
DBE1A4515E1EA389304680CA62131A526B3B9E8234ED10836DF067348F407DA03778
F76E5D751BC26563B514B6D0206B2959208419F84C1ED88DC81A6E4CEA71507722D
59597D962B83E6129BD7197D98AE778E298326B854183014A10A2ECDE46E35E7E676
E5240D7D5B0663254D6661B933D6FEFDC43DD8A2204996B67785836CF6E98502545
A1CFE6B4007E1E5B22832D50D234C92007E08D2417785AD2AE66594F68F8B9E469F
4020037AA60AB50C7704F3D545039ECC1AD95CC6133D9BB9EC271E85723E4D96598
B42551CA09897764B3EAC43CB633B09D6ED292183CB2EBDCA18A3A313594B18FEC
3A949613EE3145C40037EF58C666D3455567BFCFAA47CAEA5182212F59D103EC6B4
B30CABBEDC6EE4728ADE14A34D52AD266BD2C25AA9519323D2004EA76DDA11E7D
BB023AC18AD82CC47F78F64CA3B82CC24B9A07C1F62830A9DACFCE410AD5D3F44
F995035B947C7CBA66B88D77847C5F92487E3BD5699157CD1BA040681F84818BB57
A782F89F7FAE855C9CDC5A21712C397A4D69DA17AF61E4A35BCFCD2489E3065ACB
B42125FFE36DFA65087FEB67C0507B0D52509FCC1AF56AFD58C789AC4AC6B432AA7
EA135FA9FB29CAA6485DF1AADF1929373A79542BED43618ECDC92543F9A7F85296B
CF15613215A85160FDD7F9D05687618C92D6D232CC148F3C27CBB78BB17F1E5277C
BB77DCAC1A7A1988950C34112B293099E604BB7340B4D54690821CF33D5BF51C9A7
845891C11A2B73137296B07D0D4FA5737A78819BAD6B41407049F0734DC3A0AA931D
04B642AFE7DE66247B8679DC185A6320EC674528A9CB85637E1BED60768AE5E7063
EF1522F50FA5FF077BDA5B0DCFB866050D8324152C6C9FF5C671218127F1D97A998
B2E4F7C20DED73F9650865CE596C5ED09FC2132AA69D2CF319C54FE93B9E

e) 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"
}
]
```


f) 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 whitelisted IPs to get the required report.
3. The parameters are “merchantId”, “settlementDate” (date should be in format YYYY-MM-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 isgenerated.
5. In other cases, or error cases merchant will get response as .JSON format always.

UAT Testing Credentials:

UAT:

Encrypted URL= https://titanuat.atomtech.in/SettlementReport/generateReport?merchantId=7760&encdata=F1F7E46B85600F3CBB86E848B8F80149FF27D16E2D3B5B002081007CF73F90D15CEC7C147057045DCBCDDCF80A3661AC9460947D5940EB843C1CB31BC686F9C9C4670FD823D9F6DB062097FDB8F5337CF3B348D85BECB776CD77B4262D6D2D7C

Note*: For Open request refer Non-AES Section

Production Credentials:

Production:

Encrypted URL=
https://pgreports.atomtech.in/SettlementReport/generateReport&encdata=F1F7E46B856
00F3CBB86E848B8F80149FF27D16E2D3B5B002081007CF73F90D15CEC7C147057045DCBC
DDC
F80A3661ACF377EDB0545BE671F9F8192E1813D697FF9FA11D5D5DFE2F85A3B192E1C112
62

Open Request

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

g) Sample Encrypted Data (Settlement Date wise)–

69E0C5BD05E52F084A1956AAA52585C9AF83B895120423D1EC36753063437D82D759313DAAC68BA4DB5F83C
4CA01503F42A64E9E0C0A7676E6521EE4325254429E86D486BDC6754A4639B06FA145C210169C655C8BC18A7
D823D7F103A8C84CAF5470295D5D6EB4864932192349AAEDCCF94070EF404130E91F1CEDEE692067A1F838E08
61E86956C7CFEFEFB742A7E91556ED0B370F63FD96DC51107031D0D53EA653C1113FADAD791209E3091B5E1E
C10FE0C8FA179ED3E5841AF475182BF9B3F3F2B366E711CDE342896064C705F7BE3D8F8F60D4F5F4258405BC76
0871EA7B0ABD72495A43DDD24207CEDD2BB69884E36CAFEADB1C802CB78F9F73F63C5CDF8BF0987A4214E19
AFFEDFB75D3DF2B2AC14FAE67EC911B39C84BFF21841726A18E5AAB117D49259AF9ABFCC20DFC8738BC1055
D1C67FB93C37446ABF7EA7BD2B7361A52F0EB57C7E6CBEF308864A738815BBCCF55BDD04B7C7EBECA4AF0D6
CEF5B4F4D687A3980A5A577D3E3A964318A15F0EEACEE134A1C1A4D5AB7D759DACC5880E8EE981C5808850B
E75C1961CEC04E2D31768C8D5F8EE15F030FCAEC9E1A1CC335BEF745CCB975BDEB73207656B4932D45735E8D
F202E2A982CAD6B728D6786F50FAA3ECD5C884AD28D3D79DC22432DAE22875670C2EC0E894668B788AB021F
DDDFF78D389BBD65814138020D18B59967D7962B58F9A6A24A0EC5049602DF8E28632A0B41A09F163EB46441
058AA5B3C25A7CE47B391EFC6704DFBAAD14C835E77F586944A46887D24D7D918DA50D3498481BEF609E88F
57653C120F398B0645CB7CE4E22F5459B7CED7DEFF42CC5CE0E1AC945191FA2A29578A6A4F05C4A3206FF2C7F
575D81673DBC6F56F889A6176EEC15983135CD63938D79CA3ECABB6DB81F35BCF1089DA82E47FC45C5B44AF9
FA9B95AE3DA1AAE896C5F5C46B33230F70447BC1C2376C867175F8857174611D01F20E79ED9A8198F5D573E1
5EC868860A574AC9871BF59D827A7E7AFD0B5E73B0204B7F03B7FCD5A0D63F5ADF5

h) 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"
}
]
```


i) Sample Encrypted Data (Transaction Id wise)–

EE0B35DCD326B903B5EEE756DDD30EBA57642E9A5E0A494577CE51ABDBC04E8F950E0ECF7198FC741150A6CB
72665DA522E5485EE2B89D204436D7623D95B9B74829346A71B06F4F154A19DF548E3124D6E585F19F0DD7DE
89E4DFE660234E7BEE5E17C254CB00CF1EC48CA05D6CC2FF262C5C735A45245F383606F9463EFB0775DE1182C
F0267F7901820D8D3E186D991A47159C81E8498039380F983FF34F754622708E98DA6B4484370BA4D66BB28C
E4CBD58209C514DDA9524E9B37B17DB47BEE5BFA2C16A41B3920CBCCDCFA1ADED6E18FCD6862276410BA7B
9597693F4CC895879C2E7C30CE8D679BF2AAA74D0002CE077D4AC54DBEFB3D8B66759D5691E901CE547B216
CEBC9A9BFF9CE4A5C2528DBA7692F4FFB054F8F703E68FAF413529BFB90FA5227AEA8EFE404AF1706EA18A657
1A14C5BF169BDC2A64C673C7E7EAA012DE327AB1DFECCD3EA39FE3FCBC19303DAB211759CDD3353512E8B71
103C47F9576A790FA0AA7F65CBEA35FB78
92A6BB1C6E1530D963BBF0638F59013AA882C022B0BDFCD3D5EDC1AC55B2A5836ACD1635353A96223C5D060
36EE6BB74F517EA5ECA6DC8DAE8B193ED168F2B94DA6118EF464E98E9879F6D2C1FDD86037ABC5AB4653A8A
F3FAAD2B2B3BCF447494AB18EFCA67A722D36344AD7575C31071EB9D706FBF28B9A0590E328DC1EDA076689
5DF672DF86C59AAD970A8278C8C0B3CE71245DD0FD387058C26FD7A546EB831A33D21EA2511327D3AFB7583
4AD4F441E440A40C8803C4392284B50EB283A61693C6E1BF4D45EA1DE791CB2749BA519C354D80F5C88982B3
918E3F8CD0808AE684DB95D8C19F7A66A22D18B1F64E2316231445C33764724023A96A6274E2D2FF6EF847CCD
A94B9B4BB6B466CA7805B27B88E8DAE5D336570271B4B2128F9F4011C1DAC90C78709628D226378356364O4C
E8E4CA92358BF6E2A3D221A481526A7973D9CE054EBFF7A110C08C0121E73A4FDA2819706098882688A92CA8
F79168686E9A830037A4483E0127A02B39396B26F63C7A3B64A335FAB34B7FEFB609182E59E5078AC7145A840
6734609A66

j) 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"
}
]
```


k) 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

Note*- 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

Additional Parameter in Multi-Product Settlements

The following is the additional parameter received in the Response in case there are multi-product transactions in the Settlement Report –

Parameter Name - **"prodDetails"**

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",
"prodDetails":
[{"prodName":"Pennydrop1","prodAmount": 10.00},
{"prodName":"Pennydrop2","prodAmount": 10.00}],
"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"
}
]
```

AES Encryption/Decryption Logic for Settlement API

a) Request Encryption Code –

```
private String encrypt(String plainText) throws Exception
 {
  byte[] saltBytes = this.salt.getBytes("UTF-8");
  SecretKeyFactory factory = SecretKeyFactory.getInstance("PBKDF2WithHmacSHA1");
  PBEKeySpec spec = new PBEKeySpec(this.password
   .toCharArray(), saltBytes, pswdIterations, keySize);
  SecretKey secretKey = factory.generateSecret(spec);
  SecretKeySpec secret = new SecretKeySpec(secretKey.getEncoded(), "AES");
  IvParameterSpec localIvParameterSpec = new IvParameterSpec(this.ivBytes);
  Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
  cipher.init(1, secret, localIvParameterSpec);
  byte[] encryptedTextBytes = cipher.doFinal(plainText.getBytes("UTF-8"));
  return byteToHex(encryptedTextBytes);
 }
```

b) Response Decryption Code –

```
 private String decrypt(String encryptedText)
  throws Exception
 {
  byte[] saltBytes = this.salt.getBytes("UTF-8");
  byte[] encryptedTextBytes = hex2ByteArray(encryptedText);
  SecretKeyFactory factory = SecretKeyFactory.getInstance("PBKDF2WithHmacSHA1");
  PBEKeySpec spec = new PBEKeySpec(this.password
   .toCharArray(), saltBytes, pswdIterations, keySize);
  SecretKey secretKey = factory.generateSecret(spec);
  SecretKeySpec secret = new SecretKeySpec(secretKey.getEncoded(), "AES");
  IvParameterSpec localIvParameterSpec = new IvParameterSpec(this.ivBytes);
  Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
  cipher.init(2, secret, localIvParameterSpec);
  byte[] decryptedTextBytes = (byte[])null;
  decryptedTextBytes = cipher.doFinal(encryptedTextBytes);
  return new String(decryptedTextBytes);
 }
```

VIII. AES Encryption Logic

- Transaction Status (Requery) API's request and returned response are shared via AES-512 encryption.
- The following KEY are to be used for UAT:

MerchId	encResKey	encReqKey
9135	58BE879B7DD635698764745511C704AB	7813E3E5E93548B096675AC27FE2C850

1. AES Encryption Java Code:

```
import java.util.logging.Logger;
import javax.crypto.Cipher;
import javax.crypto.SecretKey;
import javax.crypto.SecretKeyFactory;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.PBEKeySpec;
import javax.crypto.spec.SecretKeySpec;

public class AtomEncryption {
static Logger log = Logg\
er.getLogger(AtomEncryption.class.getName());
private static int pswdIterations = 65536;
private static int keySize = 512;
private static final byte[] ivBytes = {
        0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15
};

public static String encrypt(String plainText, String key) {
        try {
                byte[] saltBytes = key.getBytes("UTF-8");

                SecretKeyFactory factory = SecretKeyFactory.getInstance("PBKDF2WithHmacSHA512");
                PBEKeySpec spec = new PBEKeySpec(key.toCharArray(), saltBytes, pswdIterations, keySize);

                SecretKey secretKey = factory.generateSecret(spec);
                SecretKeySpec secret = new SecretKeySpec(secretKey.getEncoded(), "AES");

                IvParameterSpec localIvParameterSpec = new IvParameterSpec(ivBytes);
                Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
                cipher.init(1, secret, localIvParameterSpec);

                byte[] encryptedTextBytes = cipher.doFinal(plainText.getBytes("UTF-8"));

                return byteToHex(encryptedTextBytes);
        } catch (Exception e) {
                log.info("Exception while encrypting data:" + e.toString());
        }
        return null;
}

public static String decrypt(String encryptedText, String key) {
        try {
                byte[] saltBytes = key.getBytes("UTF-8");
                byte[] encryptedTextBytes = hex2ByteArray(encryptedText);

                SecretKeyFactory factory = SecretKeyFactory.getInstance("PBKDF2WithHmacSHA512");
                PBEKeySpec spec = new PBEKeySpec(key.toCharArray(), saltBytes, pswdIterations, keySize);
                SecretKey secretKey = factory.generateSecret(spec);
                SecretKeySpec secret = new SecretKeySpec(secretKey.getEncoded(), "AES");
                IvParameterSpec localIvParameterSpec = new IvParameterSpec(ivBytes);
                Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
```

```
                cipher.init(2, secret, localIvParameterSpec);
                byte[] decryptedTextBytes = (byte[]) null;
                decryptedTextBytes = cipher.doFinal(encryptedTextBytes);
                return new String(decryptedTextBytes);
        } catch (Exception e) {
                log.info("Exception while decrypting data:" + e.toString());
        }
        return null;
}

private static String byteToHex(byte[] byData) {
        StringBuffer sb = new StringBuffer(byData.length * 2);
        for (int i = 0; i < byData.length; ++i) {
                int v = byData[i] & 0xFF;
                if (v < 16)
                        sb.append('0');
                sb.append(Integer.toHexString(v));
        }
        return sb.toString().toUpperCase();
}

private static byte[] hex2ByteArray(String sHexData) {
        byte[] rawData = new byte[sHexData.length() / 2];
        for (int i = 0; i < rawData.length; ++i) {
                int index = i * 2;
                int v = Integer.parseInt(sHexData.substring(index, index + 2), 16);
                rawData[i] = (byte) v;
        }
        return rawData;
}

public static void main(String[] args) {
        try {
                String encryptedData = AtomEncryption.encrypt("1235", "ASWKLSLLFS4sd4g4gsdg");
                System.out.println("encryptedData : " + encryptedData);
        } catch (Exception e) {
                // TODO: handle exception
        }
}
}
```

UAT environment details:

The UAT environment details are as follows:
13.127.25.237

The above is the IP address of the UAT server for scenarios pertaining to AIPay API.

UAT server:
The UAT server needs to be whitelisted at the merchant's end so that we can post on the merchant side.