NTT DATA
Payment Services

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

Refund Status API

1. 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.

A. Document Information

Document Attributes	Information
Document Name	Refund Status API
Document Version	1.01
Owner	PMG
Author	Aviral Tripathi
Approved	Pavan Nikumbh

B. Revision Chart

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

Version	Primary Author	Description of Version	Date Completed	Reviewed By
1.0	Aviral Tripathi	Refund Status API	02/08/2022	Pavan N.
1.01	Aviral Tripathi	Password Description Updated	16/06/2023	

NTT DATA
Payment Services

Contents

1. Description

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

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": " VGVzdEAxMjM0"
  },
  "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)	VGVzdEAxMjM0	Password Provided by NDPS encoded to base64 format
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):

59B80AF255F3D63E3711DD833E01F98FC098E50B9E09B3C39D0605CC6E7F8AB998E84CBE365D6FD578FB3B770F4B1FF9509C56A3CF7059FF2DA3FEC198AC427B6FB4B5B8A14D7B7FBB9483E1E98BEC097E20FE90A1E770D7DA85F725F0DEE236C58CC6FD971C4EFCE70B74CDA3D36C526FC65394716963D584423B02800DB32C8E4A56082726E74099BA11D40644E669950C9A0CDCED2A8D7740553478847E1AD73FB9EDF9403C5E165B25795494B8B87164B25DD52047AB1BD18A0229147A88AA6042A2BBB6E12445D6A5E08C53C77B6AC63CF059A86E95EEABCA0EB8CCC4F9DE6B2A7E7AB9EA021152ED29D1B854CBB97FCCA34456A88D4AC8A6E1DAB7839A78DAC48AB74DFD66A575202963BC6E51A0F55EA63FF3BADF3FA55918B20FB58B3FE20CF0B6F530C30043A634652B2DB87332EB80296347A1B37443FF370BC5B29DC39E78904F2E8553276F56F28E8EBC4B1B89E378D3121CA064E98DF687F2B5

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

https://caller.atomtech.in/ots/payment/status?merchId=9135&encData=59B80AF255F3D63E3711DD833E01F98FC098E50B9E09B3C39D0605CC6E7F8AB998E84CBE365D6FD578FB3B770F4B1FF9509C56A3CF7059FF2DA3FEC198AC427B6FB4B5B8A14D7B7FBB9483E1E98BEC097E20FE90A1E770D7DA85F725F0DEE236C58CC6FD971C4EFCE70B74CDA3D36C526FC65394716963D584423B02800DB32C8E4A56082726E74099BA11D40644E669950C9A0CDCED2A8D7740553478847E1AD73FB9EDF9403C5E165B25795494B8B87164B25DD52047AB1BD18A0229147A88AA6042A2BBB6E12445D6A5E08C53C77B6AC63CF059A86E95EEABCA0EB8CCC4F9DE6B2A7E7AB9EA021152ED29D1B854CBB97FCCA34456A88D4AC8A6E1DAB7839A78DAC48AB74DFD66A5752029963BC6E51A0F55EA63FF3BADF3FA55918B20FB58B3FE20CF0B6F530C30043A634652B2DB87332EB80296347A1B37443FF370BC5B29DC39E78904F2E8553276F56F28E8EBC4B1B89E378D3121CA064E98DF687F2B5

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": " VGVzdEAxMjM0"
  },
  "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):

59B80AF255F3D63E3711DD833E01F98FC098E50B9E09B3C39D0605CC6E7F8AB998E84CBE365D6FD578FB3B770F4B1FF9509C
56A3CF7059FF2DA3FEC198AC427B6FB4B5B8A14D7B7FBB9483E1E98BEC097E20FE90A1E770D7DA85F725F0DEE236C58CC6FD
971C4EFCE70B74CDA3D36C526FC65394716963D584423B02800DB32C8E4A56082726E74099BA11D40644E669950C9A0CDCE
D2A8D7740553478847E1AD73FB9EDF9403C5E165B25795494B8B89DAA15AD882880B3B1B12D8F217B9F11D04445B3B14C16
F95339C8BE0C9C9945DB9A856DF700C7EE0D8639003F6B63E946519CCF758478AF6F82B05A65B6C9ACA09FEEA802F148DA21
C5177A0FA69DF71B7FBB668B1D9849B574E5333B708D18F153472B45B2F51C79597694067668744BD012D1E18CE230577B4B
2A0BFD3562F28A43490220000CB664785651F597B0814BC99BF252EE1579CF3F3216A1B118C231E4CB920C5630A2B5CDAC34
E08DAEB503EE73A690FCB97CEF61BFD7D6A4560CDEAC47BDA034CDC7B7D76E73315E9D69AA8E9D9D97B61BA8A9FAE986DF
F6A1

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

https://caller.atomtech.in/ots/payment/status?merchId=9135&encData=59B80AF255F3D63E3711DD833E01F98FC098E50B9E
09B3C39D0605CC6E7F8AB998E84CBE365D6FD578FB3B770F4B1FF9509C56A3CF7059FF2DA3FEC198AC427B6FB4B5B8A14D7B
7FBB9483E1E98BEC097E20FE90A1E770D7DA85F725F0DEE236C58CC6FD971C4EFCE70B74CDA3D36C526FC65394716963D584
423B02800DB32C8E4A56082726E74099BA11D40644E669950C9A0CDCED2A8D7740553478847E1AD73FB9EDF9403C5E165B2
5795494B8B89DAA15AD882880B3B1B12D8F217B9F11D04445B3B14C16F95339C8BE0C9C9945DB9A856DF700C7EE0D863900
3F6B63E946519CCF758478AF6F82B05A65B6C9ACA09FEEA802F148DA21C5177A0FA69DF71B7FBB668B1D9849B574E5333B70

8D18F153472B45B2F51C79597694067668744BD012D1E18CE230577B4B2A0BFD3562F28A43490220000CB664785651F597B0
814BC99BF252EE1579CF3F3216A1B118C231E4CB920C5630A2B5CDAC34E08DAEB503EE73A690FCB97CEF61BFD7D6A4560CD
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 EncryptionKey [Pg. 10]
and MID in the decryption method as illustrated below:

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

DataType	Name	Value	Description
String	decstr	BFC23F835C2840C82CCA60671	Encrypted responseto the encrypted request triggered, that needs to be decrypted
String	Key	Key provided by NDPS, to decrypt theresponse	Key provided by NDPS, to decrypt theresponse
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. Postencryption, 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
4E8050B811AF99555A40946D9977D91ED0D3B75A15CF7AB2694846EA394548A37804013297B3A8AA67C44068169833720D
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
DD5E3ED0B1AC4876C85101138D36727E5C66E59CA9E559127095C3E52A60738204FB4A7FE1D71B9A1CFBB34D2DB0A26675
F500DE2A0B1A7974254B4D13AD956EC53619D16796D34919738B039CC7040E7A07734BA9D4583184DD5A04595BDDA551
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

4. 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

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
        }
}
}
```

5. 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 Refund Status API.

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