

Info type: Confidential
Company: NTT Data Payment Services India
Info. owner: Product Management Group



Link Based Payment APIs

API Document



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 in particular, 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	Link Based Payment APIs
Owner	Product Management Group
Author	Aviral Tripathi

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	Link Based Payment APIs	25-07-2022	
1.1	Aviral Tripathi	Changes in Invoice Settings to restrict NDPS from sending link to customer	04-05-2023	
1.2	Aviral Tripathi	Changes in 'Create Payment Link API' to initiate TPV transaction	12-06-2023	

Index

I. Introduction	4
II. Get Template API.....	4
Description.....	4
Sample Request.....	4
Sample Response	4
III. Get Template Details API	5
Description.....	5
Sample Request.....	5
Sample Response	6
IV. Create Payment Link API.....	9
Description.....	9
Sample Request.....	9
Sample Response	12
V. Resend Payment Link API.....	13
Description.....	13
Sample Request.....	14
Sample Response	14
VI. Cancel Payment Link API	15
Description.....	15
Sample Request.....	15
Sample Response	15
Status Codes	16
Java Code for API Implementation	16
Class code for encryption and decryption	16
Class code for 'Signature' element.....	19
i) Create Payment Link.....	19
ii) Resend/Cancel Payment Link	19
UAT environment details	21
Request parameters to be received from NTT Data payment services	21



I. Introduction

This document elaborates upon the APIs pertaining to link-based payment system wherein the merchants use the Merchant Console to perform various actions to generate and accept payment via payment link.

II. Get Template API

Description

On using the Get Template API, the merchant receives template number and name in response as per the templates that have been created by the merchant.

Prod URL: https://payment1.atomtech.in/ots/invoice-api-service/v1/invoice/templates/merch_id/1191

Note *- The request is served using GET to capture response.

Sample Request

No parameters are to be passed to attain the template list. The URL itself is sufficient for 'Get Template'.

Note*-

'1191' in the above URL represents UAT configuration, merchant will use the provided merchant ID in URL.

Sample Response

```
{
  "merchId": 1191,
  "statusCode": "OTS0000",
  "description": "Request Executed Successfully",
  "templates": [
    {
      "templateId": 13749,
      "templateName": "PIMS"
    },
    {
      "templateId": 15000,
      "templateName": "Virgin Atlantis"
    },
    {
      "templateId": 15406,
      "templateName": "RCI_Demo"
    },
    {
      "templateId": 16426,
      "templateName": "Greenwoods Society"
    },
    {
      "templateId": 16687,
      "templateName": "Greenwoods"
    }
  ]
}
```

```

    "templateId": 56046,
    "templateName": "TESTING"
  }
]
}

```

Specifications of the Response parameters –

Parameter Name	Mandatory/Optional	Data Type	Description
merchId	Mandatory	Integer	MID obtained on registration of Merchant URL's and IP
statusCode	Mandatory	String	Status Code
description	Mandatory	String	Status Description Message
templateId	Mandatory	Integer	All the template Ids will be sent using the array data type
templateName	Mandatory	String	All the template Names will be sent using the array data type corresponding to the template Ids passed in the Template Id parameter

III. Get Template Details API

Description

The Get Template API enables the merchant to receive all details pertaining to the specific template for which the API is triggered.

Prod URL: https://payment1.atomtech.in/ots/invoice-api-service/v1/invoice/fetch_template_details

Note *- The request is served using **POST** method to capture response.

Sample Request

```

{
  "merchId":"1191",
  "templateId":"56046"
}

```

Specifications of the Request parameters –

Parameter Name	Mandatory/Optional	Data Type	Description
merchId	Mandatory	Integer	MID obtained on registration of Merchant URL's and IP
templateId	Mandatory	Integer	The template Id of the template for which the parameter details are required will be passed



Sample Response

```
{
  "merchId": 1191,
  "templateId": 56046,
  "statusCode": "OTS0000",
  "description": "Request Executed Successfully",
  "custEmailId": "aishwarya.s @atomtech.in",
  "invoiceByEmail": true,
  "invoiceByMobileNo": true,
  "custMobileNo": "8291148911",
  "mailSubject": "Test email",
  "mailContent": "this email is test email",
  "fields": [
    {
      "name": "productId",
      "displayName": "ACT ID",
      "type": "dropdown",
      "defaultValue": null,
      "editable": false,
      "chargesType": null
    },
    {
      "name": "productBasePrice",
      "displayName": "PBP",
      "type": "text",
      "defaultValue": null,
      "editable": false,
      "chargesType": null
    },
    {
      "name": "quantity",
      "displayName": "QUANT",
      "type": "text",
      "defaultValue": null,
      "editable": false,
      "chargesType": null
    },
    {
      "name": "description",
      "displayName": "DESC",
      "type": "text",
      "defaultValue": null,
      "editable": false,
      "chargesType": null
    },
    {
      "name": "merchantTxnId",
      "displayName": "TXN ID",
      "type": "text",
      "defaultValue": null,
      "editable": false,
      "chargesType": null
    },
    {
      "name": "amount",
      "displayName": "AMT",

```

```
"type": "number",
"defaultValue": "75",
"editable": true,
"chargesType": null
},
{
  "name": "serviceTax",
  "displayName": "SERVICE TAX",
  "type": "text",
  "defaultValue": null,
  "editable": false,
  "chargesType": null
},
{
  "name": "discount",
  "displayName": "DISCOUNT",
  "type": "text",
  "defaultValue": "50",
  "editable": true,
  "chargesType": "FLAT"
},
{
  "name": "invoiceValidFor",
  "displayName": "INVOICE VALIDITY",
  "type": "text",
  "defaultValue": null,
  "editable": false,
  "chargesType": null
},
{
  "name": "otherCharges",
  "displayName": "OTHER CHARGES",
  "type": "text",
  "defaultValue": null,
  "editable": false,
  "chargesType": null
},
{
  "name": "note",
  "displayName": "NOTE",
  "type": "text",
  "defaultValue": null,
  "editable": false,
  "chargesType": null
},
{
  "name": "currency",
  "displayName": "CURRENCY",
  "type": "dropdown",
  "defaultValue": null,
  "editable": false,
  "chargesType": null
},
{
  "name": "invoiceAttachment",
  "displayName": "Invoice Attachment",
  "type": "text",
```

```

    "defaultValue": null,
    "editable": true,
    "chargesType": null
  },
  {
    "name": "UDF1",
    "displayName": "Gender",
    "type": "radio",
    "defaultValue": "Male, Female, Transgender",
    "editable": false,
    "chargesType": null
  },
  {
    "name": "UDF2",
    "displayName": "CITY",
    "type": "dropdown",
    "defaultValue": "MUMBAI, PUNE, KOLHAPUR",
    "editable": false,
    "chargesType": null
  },
  {
    "name": "UDF3",
    "displayName": "BIRTH YEAR",
    "type": "number",
    "defaultValue": "1996,1997,1998",
    "editable": false,
    "chargesType": null
  },
  {
    "name": "UDF4",
    "displayName": "STATE",
    "type": "text",
    "defaultValue": "MAHARASHTRA, GUJRAT,GOA",
    "editable": false,
    "chargesType": null
  }
}
]
}

```

Specifications of the Response parameters –

Parameter Name	Mandatory/Optional	Data Type	Description
merchId	Mandatory	Integer	MID obtained on registration of Merchant URL's and IP
templateId	Mandatory	Integer	The template Id of the template for which the parameter details are required will be passed
statusCode	Mandatory	String	Status of the request
description	Mandatory	String	Description of the request status
custEmailId	Mandatory	String	E-mail ID of the Customer in invoice
invoiceByEmail	Mandatory	String	If invoice by E-mail is enabled
invoiceByMobileNo	Mandatory	String	If invoice by mobile is enabled
custMobileNo	Mandatory	Integer	Mobile No of the Customer in invoice
mailSubject	Optional	String	Subject line of the E-mail

mailContent	Optional	String	E-mail body content
name	Mandatory	String	Template Name of the template Id sent in request
displayName	Conditional	String	Field Display Name
type	Conditional	String	Field Type
defaultValue	Mandatory	String	Values assigned by default to the template fields
editable	Conditional	Boolean	Field editable or non-editable
chargesType	Optional	String	Type of charges – flat or percentage

IV. Create Payment Link API

Description

The Create Payment Link API is triggered in order to create invoice using a predefined template in the merchant console by the merchant which is sent along with the payment link.

Prod URL: https://payment1.atomtech.in/ots/invoice-api-service/v1/invoice/create_invoice_link

Note *- The request is served using **POST** method to capture response.

Sample Request

```
{
  "merchId": "1191",
  "password": "NCA@1234",
  "payMode": "NB~UP",
  "templateId": "56046",
  "custEmailId": "aishwarya.sarkale@atomtech.in",
  "custMobileNo": "8291148911",
  "custName": "Aishwarya Sarkale",
  "mailSubject": "Test email",
  "mailContent": "this email is test email",
  "signature":
    "55883f5600a46197baecf5d488a3b56fe24e8ebf9730ea00841e5423012e3666e71a75d3a6e9f43934df7627263a2e381e03eda8ae34b6c10b5e3d51fb5f4cf",
  "fields": [
    {
      "name": "productId",
      "useDefaultValue": "false",
      "value": "409000590984"
    },
    {
      "name": "merchantTxnId",
      "useDefaultValue": "false",
      "value": "91000000002245"
    },
    {
      "name": "amount",
      "useDefaultValue": "true",
      "value": "5.00"
    }
  ]
}
```

```
{
  "name": "quantity",
  "useDefaultValue": "false",
  "value": "1"
},
{
  "name": "description",
  "useDefaultValue": "false",
  "value": "test Description"
},
{
  "name": "serviceTax",
  "useDefaultValue": "false",
  "value": "0.5"
},
{
  "name": "invoiceAttachment",
  "useDefaultValue": "false",
  "value": "123"
},
{
  "name": "invoiceValidFor",
  "useDefaultValue": "false",
  "value": "10.00"
},
{
  "name": "productBasePrice",
  "useDefaultValue": "false",
  "value": "1000.00"
},
{
  "name": "discount",
  "useDefaultValue": "true",
  "value": "2.00"
},
{
  "name": "otherCharges",
  "useDefaultValue": "false",
  "value": "Processing fee"
},
{
  "name": "currency",
  "useDefaultValue": "false",
  "value": "INR"
},
{
  "name": "note",
  "useDefaultValue": "false",
  "value": "Notess"
},
{
  "name": "UDF1",
  "useDefaultValue": "true",
  "value": "Male, Female, Transgender, Other"
},
{
  "name": "UDF2",
```

```

    "useDefaultValue": "true",
    "value": "MUMBAI, PUNE, KOLHAPUR, JAIPUR"
  },
  {
    "name": "UDF3",
    "useDefaultValue": "true",
    "value": "1996,1997,1998,1999"
  },
  {
    "name": "UDF4",
    "useDefaultValue": "true",
    "value": "MAHARASHTRA,GUJARAT,GOA,RAJASTHAN"
  }
]
}

```

Specifications of the Request parameters –

Parameter Name	Mandatory/Optional	Data Type	Description
merchId	Mandatory	Integer	MID obtained on registration of Merchant URL's and IP
password	Mandatory	String	Password used for Console login
payMode	Mandatory	String	Mode of payment that merchant wants to enable for that payment link; possible values
templateId	Mandatory	Integer	The template Id of the template for which the parameter details are required will be passed
accountNumber	Conditional	Integer	Account Number of the customer for TPV validation
vpa	Conditional	String	VPA of the customer for TPV validation
custEmailId	Conditional	String	E-mail ID of the Customer in invoice
custMobileNo	Conditional	Integer	Mobile No of the Customer in invoice
custName	Mandatory	String	Name of the Customer in invoice
mailSubject	Optional	String	Subject line of the E-mail
mailContent	Optional	String	E-mail body content
signature	Mandatory	String	Signature generated using NDPS logic
name	Conditional	String	Template Name of the template Id sent in request
useDefaultValue	Mandatory	Boolean	Values assigned by default to the template fields
value	Conditional	String	Value assigned to the invoice field

Note* -

- The parameters - 'custEmailId' & 'custMobileNo' will be non-mandatory in case you do not require NDPS to send the payment link to the Customer.
- 'accountNumber' is mandatory if 'NB' is passed in payMode parameter.
- 'vpa' & 'accountNumber' both are mandatory if 'UP' is passed in payMode parameter.
- Point no. 'ii' & 'iii' apply only if the same is applicable and configured at back-end.
- Only two values can be passed for 'payMode' parameter for a given request.
- The values in the 'payMode' parameter are to sent ~ seperated

To configure the settings in such manner, follow the below steps–

- i) Login to the Merchant Console
- ii) Hover to the 'Invoice Settings' sub-menu in the 'Invoice' menu
- iii) Scroll to option 'Link sent by' and select 'Merchant' from the dropdown.

This functionality enable you to Recurring Invoice

YES

Contact Detail	Name	Email	Country code	Area code	Contact No
Contact Detail1 *	name	email	country code	area code	mob no

Allow Invoice to be sent as an Attachment?

NO

Link to be sent by Merchant i

CLEAR SUBMIT

'Invoice Settings' sub-menu in Console

On configuring this setting, **NDPS will not send the payment link to the end customer** and Merchant shall be responsible for the same.

Possible Values for Parameter 'payMode'

Field Value	Effect on Payment Option Page
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.
AMEX	In case the merchant wishes to show only American Express as payment option to customer.
EMI	In case the merchant wishes to show only EMI as payment option to customer.
CH	In case the merchant wishes to show only Challan 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.

Sample Response

```
{
  "merchId": 1191,
  "templateId": 56046,
  "invoiceNo": 108341,
  "merchTxnId": "91000000002245",
  "invoiceValidFor": "10.00 Days",
  "custEmailId": "aishwarya.sarkale@atomtech.in",
  "custMobileNo": "8291148911",
  "custName": "Aishwarya Sarkale",
  "dateTime": "2022-06-30 14:45:53",
  "smsStatus": "Failed",
  "emailStatus": "Sent",
  "invoiceLink": "https://payment1.atomtech.in/ots/invoice/txn/30C574CA521EB92A4262FBAAC59CB41B",
}
```

```

"signature":
"55883f5600a46197baeccf5d488a3b56fe24e8ebf9730ea00841e5423012e3666e71a75d3a6e9f43934df7627263
a2e381e03eda8ae34b6c10b5e3d51fb5f4cf",
"statusCode": "OTS0000",
"description": "Request Executed Successfully"
}

```

Specifications of the Response parameters –

Parameter Name	Mandatory/Optional	Data Type	Description
merchId	Mandatory	Integer	MID obtained on registration of Merchant URL's and IP
templateId	Mandatory	Integer	The template Id of the template for which the parameter details are required will be passed
invoiceNo	Mandatory	Integer	Invoice number generated by NDPS for the invoice payment link sent
merchTxnId	Mandatory	String	Transaction ID provided by NDPS
accountNumber	Mandatory	Integer	Account Number of the customer for TPV validation
vpa	Mandatory	String	VPA of the customer for TPV validation
invoiceValidFor	Mandatory	String	Validity of the invoice
custEmailId	Mandatory	String	E-mail ID of the Customer in invoice
custMobileNo	Mandatory	String	Mobile No of the Customer in invoice
custName	Mandatory	String	Name of the Customer in invoice
dateTime	Mandatory	String	Date and time of the generated invoice
smsStatus	Mandatory	Boolean	Status of SMS service, possible values - Sent, Failed
emailStatus	Mandatory	Boolean	Status of e-mail service, possible values - Sent, Failed
invoiceLink	Mandatory	String	Payment link generated on invoice
signature	Conditional	String	Signature generated using NDPS logic
statusCode	Mandatory	String	Y will be sent as response
description	Conditional	String	Description of the request status

V. Resend Payment Link API

Description

The merchant hits this API in order to resend the invoice payment link to the customer.

Prod URL: https://payment1.atomtech.in/ots/invoice-api-service/v1/invoice/resend_invoice_link

Note *- The request is served using POST method to capture response.

Sample Request

```
{
  "merchld": "1191",
  "password": "NCA@1234",
  "invoiceNo": "108382",
  "action": "RESEND",
  "signature": "2e96092be7e324f768a46ff53b03d10a0f575ef6b39ad6900c1b7c680abf813a0ad343500915c5432a3cd2016c3f627549f57a1afe9953f00b3beb15ec368278"
}
```

Specifications of the Request parameters –

Parameter Name	Mandatory/Optional	Data Type	Description
merchld	Mandatory	Integer	MID obtained on registration of Merchant URL's and IP
password	Mandatory	String	Password used for Console login
invoiceNo	Mandatory	Integer	Invoice number generated by NDPS for the invoice payment link sent
action	Mandatory	String	Action taken using the API i.e., RESEND
signature	Mandatory	String	Signature generated using NDPS logic

Sample Response

```
{
  "merchld": 1191,
  "invoiceNo": 108382,
  "smsStatus": "Failed",
  "emailStatus": "Sent",
  "statusCode": "OTS0001",
  "description": "Invoice Link Sent Successfully"
}
```

Specifications of the Response parameters –

Parameter Name	Mandatory/Optional	Data Type	Description
merchld	Mandatory	Integer	MID obtained on registration of Merchant URL's and IP
invoiceNo	Mandatory	Integer	Invoice number generated by NDPS for the invoice payment link sent
smsStatus	Mandatory	Boolean	Status of SMS service, possible values - Sent, Failed
emailStatus	Mandatory	Boolean	Status of e-mail service, possible values - Sent, Failed
statusCode	Mandatory	String	The template Id of the template for which the parameter details are required will be passed
description	Mandatory	String	Description of the request status

VI. Cancel Payment Link API

Description

The merchant hits this API in order to cancel the sent invoice and disable the ability to make payment using the payment link.

Prod URL: https://payment1.atomtech.in/ots/invoice-api-service/v1/invoice/cancel_invoice_link

Note *- The request is served using POST method to capture response.

Sample Request

```
{
  "merchId": "1191",
  "password": "NCA@1234",
  "invoiceNo": "108382",
  "action": "CANCEL",
  "signature": "2e96092be7e324f768a46ff53b03d10a0f575ef6b39ad6900c1b7c680abf813a0ad343500915c5432a3cd2016c3f627549f57a1afe9953f00b3beeb15ec368278"
}
```

Specifications of the Request parameters –

Parameter Name	Mandatory/Optional	Data Type	Description
merchId	Mandatory	Integer	MID obtained on registration of Merchant URL's and IP
Password	Mandatory	String	Password used for Console login
invoiceNo	Mandatory	Integer	Invoice number generated by NDPS for the invoice payment link sent
action	Mandatory	String	Action taken using the API i.e., CANCEL
signature	Mandatory	String	Signature generated using NDPS logic

Sample Response

```
{
  "merchId": 1191,
  "invoiceNo": 108382,
  "smsStatus": "Failed",
  "emailStatus": "Sent",
  "statusCode": "OTS0001",
  "description": "Invoice Link Sent Successfully"
}
```

Specifications of the Response parameters –

Parameter Name	Mandatory	Data Type	Description
merchId	Mandatory	Integer	MID obtained on registration of Merchant URL's and IP

invoiceNo	Mandatory	Integer	Invoice number generated by NDPS for the invoice payment link sent
smsStatus	Mandatory	Integer	Status of SMS service, possible values - Sent, Failed
emailStatus	Mandatory	Integer	Status of e-mail service, possible values - Sent, Failed
statusCode	Mandatory	String	The template Id of the template for which the parameter details are required will be passed
description	Mandatory	String	Description of the request status

Status Codes :

Status Code	Description
OTS0000	Request Executed Successfully
OTS0001	Invoice Link Sent Successfully
OTS0002	Invoice Link Cancelled Successfully
OTS1001	Parameters Are Missing
OTS1002	Invalid Merch ID/Password
OTS1003	Inactive Merchant
OTS1004	Template Not Found
OTS1005	Template Details Not Found
OTS1006	Template Field/s Is/Are Missing
OTS1007	Default Value Missing
OTS1008	Value Missing (If Request Format(JSON) Is Invalid)
OTS1009	Bad Request (For Unexpected Errors)
OTS1010	Something Went Wrong
OTS1011	Signature Mismatched
OTS1012	Invoice Not Found
OTS1013	Action Not Found
OTS1014	Request Failed
OTS1015	Cancelled Invoice

Java Code for API Implementation

The API involves encryption and decryption of data. Encryption is done while the request is being sent by merchant to NTT Data payment service. This encrypted data is decrypted by NTT data payment service.

In case of response being sent to merchant, encrypted response is sent to the merchant. Merchant decrypts the response and acts accordingly on the transaction.

Class code for encryption and decryption :

```
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 = Logger.getLogger(AtomEncryption.class.getName());

    private static int pswdIterations = 65536;
    private static int keySize = 256;
    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
    }
}
```

}

Class code for 'Signature' element:

Signature is an element of the parameters being passed. It is part of the Category account validation. Signature needs to be created every time for new transaction. The code for the same is as follows:
The generateSignature method shown below needs to have two parameters -one hash Key and other Param.

The param field consists of following elements –

i) **Create Payment Link**

Signature String ->

MID + Password+ amount + merchantTxnId

ii) **Resend/Cancel Payment Link**

Signature String ->

MID + Password + InvoiceNo

The plus sign denotes concatenate operator without space.

Eg: (Create Payment Link)

Merchant Id: 1191

TxnPassword: ABC@123

MerchantTxnID: OTS77

Therefore, the param field will be of format: 1191ABC@123OTS77

There are two types of hash keys – request and response hash keys.

For encryption, the end-user will use request key and for decryption they will be using the response key.

The request and response key are merchant specific and has following characteristics:

Specifications of the Hash Keys –

Parameter Name	Mandatory	Data Type & Max Length	Sample Value	Content/ Remarks
reqHashKey	Mandatory	String(20)	ea59e6ee036c81d8b5	Need to be save per MID on boarded in MW in Merchant Table
respHashKey	Mandatory	String(20)	ea59e6ee036c81d8b6	Need to be save per MID on boarded in MW in Merchant Table

Java code for the same is as follows:

```
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();  
}
```

UAT environment details:

The UAT environment details are as follows: [13.127.25.237](#)

The above is the IP address of the UAT server. The UAT server needs to be whitelisted at the merchant's end so that we can post on the merchant side.

Request parameters to be received from NTT Data payment services.

MCC code: This will be provided to merchant based on his business operations.

