



# **YagoutPay**

## **API Payment**

## **Gateway**

**Integration Guide &**  
**Developer Documentation**

# 1. Introduction

YagoutPay provides secure and efficient payment solutions for merchants. This SDK enables seamless integration for collecting payments via cards, wallets, and more. It supports both Aggregator Hosted and API-based integration methods.

# 2. Environment

YagoutPay operates two environments: Test and Production. Credentials are shared upon registration.

## 2.1 Test Environment Configuration

Parameter	Value
<i>Post URL</i>	<a href="https://uatcheckout.yagoutpay.com/ms-transaction-core-1-0/paymentRedirection/checksumGatewayPage">https://uatcheckout.yagoutpay.com/ms-transaction-core-1-0 /paymentRedirection/checksumGatewayPage</a>
<i>Merchant ID</i>	Shared by us
<i>Encryption Key</i>	Shared by us
<i>Aggregator ID</i>	yagout

Parameter	Value
<i>Post URL</i>	<a href="https://uatcheckout.yagoutpay.com/ms-transaction-core-1-0/paymentRedirection/checksumGatewayPage">https://uatcheckout.yagoutpay.com/ms-transaction-core-1-0 /paymentRedirection/checksumGatewayPage</a>
<i>Merchant ID</i>	Shared by us
<i>Encryption Key</i>	Shared by us
<i>Aggregator ID</i>	yagout

## 2.2 Production Environment Configuration

Parameter	Value
<i>Post URL</i>	Shared by us
<i>Merchant ID</i>	Shared by us
<i>Encryption Key</i>	Shared by us
<i>Aggregator ID</i>	yagout

Parameter	Value
<i>Post URL</i>	Shared by us
<i>Merchant ID</i>	Shared by us
<i>Encryption Key</i>	Shared by us
<i>Aggregator ID</i>	yagout

### 3. Encryption & Decryption Logic

YagoutPay uses AES-256-CBC encryption with a shared merchant key and a static IV.

Sample PHP functions for encryption and decryption:

```
1 function encrypt($text, $key, $type)
2 {
3     $iv = "0123456789abcdef";
4     $size = 16;
5     $pad = $size - (strlen($text) % $size);
6     $padtext = $text . str_repeat(chr($pad), $pad);
7     $crypt = openssl_encrypt($padtext, "AES-256-CBC",
8     base64_decode($key), OPENSSL_RAW_DATA |
9     OPENSSL_ZERO_PADDING, $iv);
10    return base64_encode($crypt);
11 }
12 function decrypt($crypt, $key, $type)
13 {
14     $iv = "0123456789abcdef";
15     $crypt = base64_decode($crypt);
16     $padtext = openssl_decrypt($crypt, "AES-256-CBC",
17     base64_decode($key), OPENSSL_RAW_DATA |
18     OPENSSL_ZERO_PADDING, $iv);
19     $pad = ord($padtext[strlen($padtext) - 1]);
20     if ($pad > strlen($padtext)) {
21         return false;
22     }
23     if (strspn($padtext, $padtext[strlen($padtext) - 1],
24     strlen($padtext) - $pad) != $pad) {
25         $text = "Error";
26     }
27     $text = substr($padtext, 0, -1 * $pad);
28     return $text;
29 }
30 }
```

YagoutPay supports two main integration methods for merchants to securely process payments:

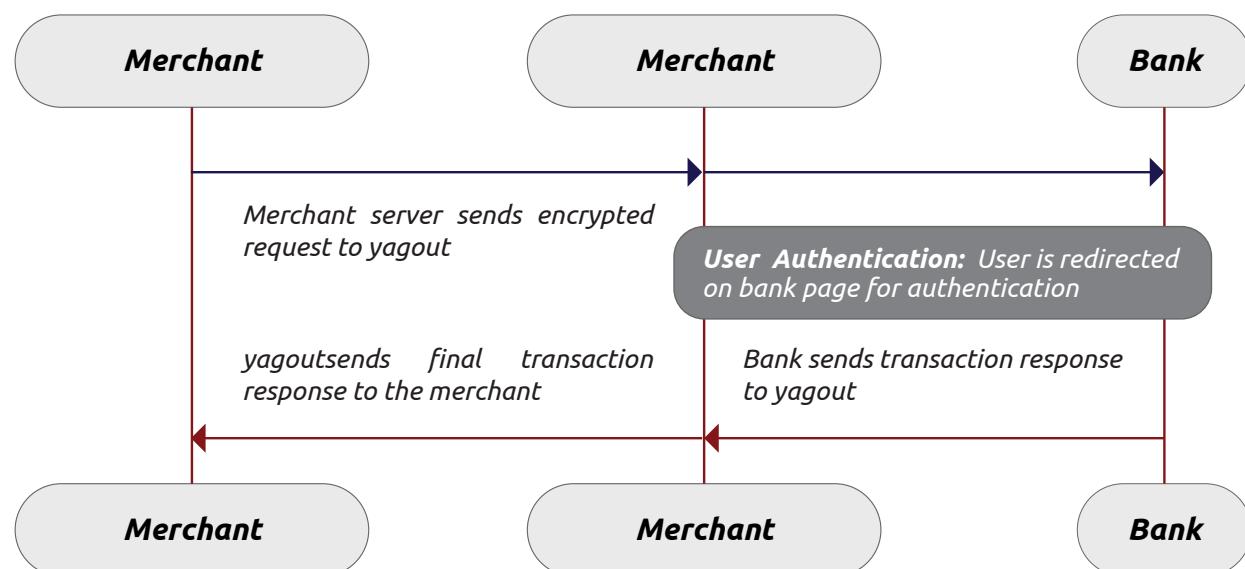
- Aggregator Hosted (Non-Seamless)
- API Integration

#### 4.1. Aggregator Hosted (Non-Seamless)

This method redirects the customer to a secure, pre-built payment page hosted by YagoutPay. It requires minimal PCI compliance effort from the merchant side and offers a streamlined setup.

##### **Process Flow**

1. *Customer selects product/service and proceeds to payment.*
2. *Customer is redirected to YagoutPay-hosted payment page.*
3. *Customer enters payment information.*
4. *YagoutPay connects to bank/processor to complete transaction.*
5. *YagoutPay sends transaction status back to the merchant.*



## ***YagoutPay Test Environments for Aggregator Hosted (Non-Seamless)***

YagoutPay supports two main integration methods for merchants to securely process payments:

### **Parameter Value**

<i>Post URL</i>	<a href="https://uatcheckout.yagoutpay.com/ms-transaction-core-1-0/paymentRedirection/checksumGatewayPage">https://uatcheckout.yagoutpay.com/ms-transaction-core-1-0/paymentRedirection/checksumGatewayPage</a>
<i>Merchant ID</i>	202504290002
<i>Encryption Key</i>	neTdYIKd87JEj4C6ZoYjaeBiCoeOr40ZKBEI8EU/8lo=
<i>Aggregator ID</i>	yagout

## **Request Parameters:**

### ***Section: Txn\_Details (Required Parameters)***

<b>Parameter Name</b>	<b>Mandate</b>	<b>Type (length)</b>	<b>Description</b>
<i>ag_id</i>	YES	Alphabets (11)	Aggregator id is "yagout".
<i>me_id</i>	YES	Numeric(20)	Merchant ID issued to you by yagout.
<i>order_no</i>	YES	Alphanumeric (70)	This ID is used by merchants to identify the order. Ensure that you send a <b>unique id</b> with each request. As it generates a unique payment reference number for each order which is sent by the merchant.
<i>Amount</i>	YES	Numeric(10,2)	This is your Transaction Amount. You can use upto 2 decimal places. E.g., "150.00"
<i>Country</i>	YES	Alphabets (3)	Country code needs to be passed, Country code as "ETH" for Ethiopia.

<b>Parameter Name</b>	<b>Mandate</b>	<b>Type (length)</b>	<b>Description</b>
<i>Currency</i>	YES	Alphabets (3)	Currency in which you want to process the transaction. ETB - Ethiopian birr
<i>txn_type</i>	YES	Alphabets (10)	Use Txn_Type as "SALE" which is static with each request
<i>success_url</i>	YES	Alphanumeric (200) <b>Characters allowed:</b> Alphabet (A-Z), (a-z), Numbers, / (slash), _ (underscore)	This will be the return URL where response is accepted for successful transaction.
<i>failure_url</i>	YES	Alphanumeric (200) <b>Characters allowed</b> Alphabet (A-Z), (a-z Numbers, / (slash), _ (underscore))	This will be the return URL where response is accepted for failed transaction.
<i>Channel</i>	YES	Alphabets (10)	Use "WEB" in case of website-based payment or "MOBILE" in case of mobile app. Static value with each request.

### **Section: pg\_details**

<i>pg_id</i>	No	Numeric(11)	<i>Value must be blank for Aggregator Hosted (Non-Seamless)</i>
<i>Paymode</i>	No	Alphabets	<i>Value must be blank for Aggregator Hosted (Non-Seamless) in the request.</i>
<i>Scheme</i>	No	Numeric(11)	<i>Value must be blank for Aggregator Hosted (Non-Seamless)</i>
<i>wallet_type</i>	No	Alphabets	Enter the wallet

## ***Section: card\_details***

<i>card_no</i>	No	Numeric(19)	Value must be blank for Aggregator Hosted (Non-Seamless)
<i>exp_month</i>	No	Numeric (2)	Value must be blank for Aggregator Hosted (Non-Seamless)
<i>exp_year</i>	No	Numeric (4)	Value must be blank for Aggregator Hosted (Non-Seamless)
<i>cvv</i>	No	Numeric (3)	Value must be blank for Aggregator Hosted (Non-Seamless)

## ***Section: cust\_details***

<i>card_name</i>	No	Alphabets (50)	<b>Value must be blank for Aggregator Hosted (Non-Seamless)</b>
<i>email_id (Mandatory)</i>	Optional	Alphabets (50)	Customer Name. This is required for fraud protection.
	Yes	Alphanumeric (100) <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers @ (at), dot, _ (underscore)	Customer's Email ID. This is required for fraud protection.
<i>mobile_no (Mandatory)</i>	Yes	Numeric(15)	Customer's Mobile Number. This is required for fraud protection.
	No	Alphanumeric(100)	This is required in case of <b>Stored card feature.</b>
<i>is_logged_in</i>	Yes	Alphabet (1)	Pass "Y" if user is logged in.

## ***Section: Bill\_details***

<i>bill_address</i>	Optional	Alphanumeric (400) <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot,- (hyphen) Space in between words	Customer's Billing Address. This is required for fraud protection.
<i>bill_city</i>	Optional	Alphabets (50)	Customer's Billing City. This is required for fraud protection.
<i>bill_state</i>	Optional	Alphabets (50)	Customer's Billing State. This is required for fraud protection.
<i>bill_country</i>	Optional	Alphabets (50)	Customer's Billing Country. This is required for fraud protection.
<i>bill_zip</i>	Optional	Alphanumeric (20) <b>Characters allowed:</b> Alphabet (A-Z),(a-z). Numbers	Customer's Billing Zip/Pin Code. This is required for fraud protection.

## ***Section: Ship\_details***

<b><i>ship_state</i></b>	<b>Optional</b>	<b>Alphabets (50)</b>	<b>Customer's Shipping State. This is required in case of physical delivery of goods.</b>
<i>ship_country</i>	Optional	Alphabets (50)	Customer's Shipping Country. This is required in case of physical delivery of goods.
<i>ship_zip</i>	Optional	Alphanumeric (20) <b>Characters allowed:</b> Alphabet (A-Z),(a-z). Numbers	Customer's Shipping Zip/Pin Code. This is required in case of physical delivery of goods.
<i>ship_days</i>	Optional	Numeric(11)	Approximate number of days required for delivery of goods. This is required in case of physical delivery of goods.
<i>address_count</i>	Optional	Numeric(11)	If customer has more than one addresses stored in your system.

## ***Section: Item\_details***

<i>item_count</i>	Optional	Alphanumeric(11)	Count of items purchased by customer.
<i>item_value</i>	Optional	Alphanumeric(200 ,2)	Commaseparated item wise cost
<i>item_category</i>	Optional	Alphanumeric(200 )	Commaseparated item category. E.g., "DVD"

<i>ship_address</i>	Optional	Alphanumeric (400) <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen) Space in between words	Customer's Shipping Address. This is required in case of physical delivery of goods.
<i>ship_city</i>	Optional	Alphabets (50)	Customer's Shipping City. This is required in case of physical delivery of goods.

## Section: UPI\_details

<i>udf_1</i>	Optional	<b>Alphanumeric (100)</b> <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets,/(slash), dot, -(hyphen) Space in betweenwords	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the response.
<i>udf_2</i>	Optional	<b>Alphanumeric (100)</b> <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets,/(slash), dot, -(hyphen) Space in betweenwords	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the response.
<i>udf_3</i>	Optional	<b>Alphanumeric (100)</b> <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets,/(slash), dot, -(hyphen) Space in betweenwords	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the response.
<i>udf_4</i>	Optional	<b>Alphanumeric (100)</b> <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets,/(slash), dot, -(hyphen) Space in betweenwords	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the response.
<i>udf_5</i>	Optional	<b>Alphanumeric (100)</b> <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets,/(slash), dot, -(hyphen) Space in betweenwords	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the response.

## Note:

Value in each section will be encrypted value generated in the following way. Only me\_id (Merchant ID) will be in plain text.

Each attribute in every section is combined and separated with pipe (|). Make sure that you are sending the request in the HTTP Form post method.

### Please find the format shared below:

The transaction request is not compatible with the Postman Platform.

Transactions are only permitted through registered merchant domains. If “localhost” or local IP is not permitted, you will get an “Invalid Referral URL” error in such case.

Transaction requests must be in the manner below when it triggers to our Payment Gateway URL. Merchants must send the following parameters to the yagout PG, which is mandatory, but values might be blank if they are optional or not required.

### Example of the

### Transaction request in HTTP Form post method: Sample HTML transaction posting:

- Make sure that you are sending the request in the HTTP Form post method.
- Transactions are only permitted through registered merchant domains. If “localhost” or local IP is not permitted, you will get “Invalid Referral URL” error.
- Transaction request must be in the manner below when it triggers to our Payment Gateway URL.
- Merchants must send the following parameters to the yagout PG which is mandatory, but values might be blank in case of optional or not required.

**Please note only HTML Form Submission with the ‘POST’ method is supported.**

**Sending payment requests using any REST API call is not supported.**

## HTML Form Submission

```
1      <!DOCTYPE html>
2      <html>
3      <body>
4          <form name="paymentForm" method="POST"      enctype=
5              "application/x-www-form-urlencoded"
6              action="https://uatcheckout.yagoutpay.com/ms-transaction-cor
7              e-1-0/paymentRedirection/checksumGatewayPage" >
8              <input name= "me_id"      value="202402140001"
9              type="hidden">
10
11
12          <inputname="merchant_request"      value=
13              "VueAaFgKrAMO1sA+p8YKdC0r5vIdTBmRzJr/tmLSx+5Kmw6lu
14              p47Q3wm2z3kKZJdbHmO3i+5ww0nzcqo1vRliKNqCJCPrk/nVL
15              Mnf0ehrUexFICmhGL+qzsL24Rv3Ugs2rZHLLrRDCUeSqeHiE87
16              zeqxWlvGJEaqKE34DfKT7cjeNztZEDpzsfVQn0As22obQzrxVM
17              OFUlnd3LKB07JBfJmOZKB+87Yg72iYY11KA/QPWwTJ3RqXCJx
18              Baogz8Lm" type="hidden" >
19
20
21          <inputname= "hash "      value =
22              "0FHhRpcLlgr8Hb88AaM+5uhGP3HwMueYTlzCKM5hQTW9JW
23              bDwA0/X+k1/a46dYFw3Vw29id/b2dvp9Y3iuzoKFGPEIlGw8U2
24              ObdDSzef+tM=" type="hidden">
25
26          <inputtype="submit"      name="submit"      value="Pay
27              Now">
28      </form>
29      </body>
30      </html>
```

## Sample Request Before Encryption:

```
me_id: 202504290002merchant_request:  
yagout|202504290002|49340|1|ETH|ETB|SALE|http://localhost/YagoutPay/transaction//Respo  
nse.php|http://localhost/YagoutPay/transaction//Response.php|WEB~|||~|||~|||~Y~|||~|||~|||~|||  
~|||  
hash:  
LuzwiRKvTGHRXZ/Z6FX9SDKzZpEK2P5UWD1nUVfgStT/cBoxond5C6JdAQm3rPBjWUpPMKW  
w1kIZ9d6gfgLLAibWJtxJMPbqvUTlztIP+fs=
```

## Sample Request After Encryption:

```
me_id: 202504290002  
merchant_request:  
VueAaFgKrAMO1sA+p8YKdG3nVX8q1aX2dZ6LqPzxY3lj+3W9gmlwOXvaHWyWw4mUR+JnftIn  
JUGOy2CCVpsJ9OMJ4kiHnqDLa889i9G2MyOalCksMDjsRFkLLETUGElu8muC6//u2roewGFX/X  
RBtL1u4xuQ9o6nf0UkjLtC3JxFPa0F9HaZc4mUtL1iFni3kKI+UHn5pQevqv8shX/IZG1kr8Ht+Lmk  
rPgcV2pjfDfkeZyCR3lAuXwmU3vhNdW  
hash:  
LuzwiRKvTGHRXZ/Z6FX9SDKzZpEK2P5UWD1nUVfgStT/cBoxond5C6JdAQm3rPBjWUpPMKW  
w1kIZ9d6gfgLLAibWJtxJMPbqvUTlztIP+fs=
```

## **Merchant\_request will get encrypt by aes256 (Details shared below):**

```
$all_values =  
$txn_details.'~'.$pg_details.'~'.$card_details.'~'.$cust_details.'~'.$bill_details.'~'.$ship_details.  
'~'.$item_details. '~'.$upi_details'~'.$other_details;  
$merchant_request['merchant_request'] = $aes->encrypt($all_values, $merchant_key, 256);
```

# API Integration

## YagoutPay Test Environments for API Integration

<b>Post URL</b>	<a href="https://uatcheckout.yagoutpay.com/ms-transaction-core-1-0/apiRedirection/apiIntegration">https://uatcheckout.yagoutpay.com/ms-transaction-core-1-0/apiRedirection/apiIntegration</a>
<b>Merchant ID</b>	202505090001
<b>Encryption Key</b>	6eUzH0ZdoVVBMTHrgdA0sqOFyKm54zojV4/faiSirkE=
<b>Aggregator ID</b>	yagout

## YagoutPay Production Environments

<b>Post URL</b>	<i>Shared by us</i>
<b>Merchant ID</b>	Shared by us
<b>Encryption Key</b>	Share by us
<b>Aggregator ID</b>	Shared by us

## Encryption Logic

<b>Encryption</b>	AES with 256 Bits
<b>Content-Type</b>	Application/json
<b>IV</b>	0123456789abcdef
<b>Crypt</b>	AES-256-CBC

## Request Parameters

Merchant must send the following parameters to YagoutPay PG for processing and.

### ***Section: Txn\_Details (Required Parameters)***

Parameter Name	Mandatory	Type (length)	Description
ag_id	YES	Alphabets (11)	Aggregator id is “yagout”.
me_id	YES	Numeric(20)	Merchant ID issued to you by yagout.
order_no	YES	Alphanumeric (70)	This ID is used by merchants to identify the order. Ensure that you send a <b>unique id</b> with each request. As it generates a unique payment reference number for each order which is sent by the merchant.
Amount	YES	Numeric (10,2)	This is your Transaction Amount. E.g., “150.00”
card_no	No	Numeric(19)	<i>Value must be blank for AggregatorHosted(Non-Seamless)</i>
exp_month	No	Numeric (2)	Value must be blank for AggregatorHosted(Non-Seamless)
exp_year	No	Numeric (4)	Value must be blank for Aggregator Hosted (Non-Seamless)
cvv	No	Numeric (3)	Value must be blank for AggregatorHosted(Non-Seamless)
card_name	No	Alphabets (50)	Value must be blank for AggregatorHosted(Non-Seamless)

## ***Section: cust\_details***

<i>cust_name</i>	Optional	Alphabets (50)	Customer Name. This is required in case you required fraud protection.
<i>email_id (Mandatory)</i>	Yes	Alphanumeric (100) <b>Characters allowed:</b> Alphabet (A-Z), (a-z).Numbers @ (at), dot, _ (underscore)	Customer's Email ID. This is required in case You required fraud protection.
<i>mobile_no (Mandatory)</i>	Yes	Numeric(15)	Customer's Mobile Number. This is required in case you required fraud protection.
<i>unique_id</i>	No	Alphanumeric (100)	This is required in case of <b>Stored card feature</b> .
<i>is_logged_in</i>	Yes	Alphabet (1)	Pass "Y" if user is logged in.

## ***Section: Bill\_details***

<i>bill_address</i>	Optional	Alphanumeric (400) <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen) Space in between words	Customer's Billing Address. This is required incase you required fraud protection.
<i>bill_city</i>	Optional	Alphabets (50)	Customer's Billing City. This is required in case you required fraud protection.
<i>bill_state</i>	Optional	Alphabets (50)	Customer's Billing State. This is required in case you required fraud protection.
<i>bill_country</i>	Optional	Alphabets (50)	Customer's Billing Country. This is required in case you required fraud protection.
<i>bill_zip</i>	Optional	Alphanumeric (20) <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers	Customer's Billing Zip/Pin Code. This is required in case you required fraud protection.

## ***Section: Ship\_details***

<i>ship_address</i>	Optional	Alphanumeric (400) <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen) Space in between words	Customer's Shipping Address. This is required in case of physical delivery of goods.
<i>ship_city</i>	Optional	Alphabets (50)	Customer's Shipping City. This is required in case of physical delivery of goods.
<i>ship_state</i>	Optional	Alphabets (50)	Customer's Shipping State. This is required in case of physical delivery of goods.
<i>ship_country</i>	Optional	Alphabets (50)	Customer's Shipping Country. This is required in case of physical delivery of goods.
<i>ship_zip</i>	Optional	Alphanumeric (20)	Customer's Shipping Zip/Pin Code. This is required in case of physical delivery of goods.
<i>ship_days</i>	Optional	Numeric(11)	Approximate number of days required for delivery of goods. This is required in case of physical delivery of goods.
<i>address_count</i>	Optional	Numeric(11)	If customer has more than one addresses stored in your system.

## ***Section: Item\_details***

<i>item_count</i>	Optional	Alphanumeric(11)	Count of items purchased by customer.
<i>item_value</i>	Optional	Alphanumeric (200,2)	Comma separated item wise cost
<i>item_category</i>	Optional	Alphanumeric(200)	Comma separated item category. E.g., "DVD"

## ***Section: Other\_details***

<i>udf_1</i>	Optional	Alphanumeric (100) <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen) Space in between words	This parameter can be used for sending additional information about the transaction. PG will send this parameter in the response.
--------------	----------	---	---

<i>udf_2</i>	Optional	Alphanumeric (100) <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen) Space inbetween words	This parameter can be used for sending additional information about the transaction. PG will send this parameter inthe response.
<i>udf_3</i>	Optional	Alphanumeric (100) <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen) Space inbetween words	This parameter can be used for sending additional information about the transaction. PG will send this parameter inthe response.
<i>udf_4</i>	Optional	Alphanumeric (100) <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen) Sp acein between words	This parameter can be used for sending additional information about the transaction. PG will send this parameter inthe response.
<i>udf_5</i>	Optional	Alphanumeric (100) <b>Characters allowed:</b> Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, - (hyphen) Space inbetween words	This parameter can be used for sending additional information about the transaction. PG will send this parameter inthe response.

<i>udf_6</i>	Optional	Alphanumeric <b>(100) Characters</b> <b>allowed:</b> Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, -(hyphen) Space inbetween words	This parameter can be used for sending additional information about the transaction. PG will send this parameter inthe response.
<i>udf_7</i>	Optional	Alphanumeric <b>(100) Characters</b> <b>allowed:</b> Alphabet (A-Z), (a-z). Numbers # (hash), Comma, circular brackets, /(slash), dot, -(hyphen) Space in between words	This parameter can be used for sending additional information about the transaction. PG will send this parameter inthe response.

**Note:**

- An order details should be created for every payment.
- Order number should be unique for every transaction request.
- Make sure that you are sending the request in JSON.
- Please find the format shared below:
- Encrypting and Decrypting the Request and Response with the Merchant Encryption Key

## Example of Transaction request(Plain) in JSON

```
1  {
2      "card_details": {
3          "cardNumber": "",
4          "expiryMonth": "",
5          "expiryYear": "",
6          "cvv": "",
7          "cardName": ""
8      },
9      "other_details": {
10         "udf1": "",
11         "udf2": "",
12         "udf3": "",
13         "udf4": "",
14         "udf5": "",
15         "udf6": "",
16         "udf7": ""
17     },
18     "ship_details": {
19         "shipAddress": "",
20         "shipCity": "",
21         "shipState": "",
22         "shipCountry": "",
23         "shipZip": "",
24         "shipDays": "",
25         "addressCount": ""
26     },
27     "txn_details": {
28         "agId": "yagout",
29         "meld": "202505090001",
30         "orderNo": "00012100",
31         "amount": "1",
32         "country": "ETH",
33         "currency": "ETB",
34     }
35 }
```

```
1      "transactionType": "SALE",
2      "sucessUrl": "",
3      "failureUrl": "",
4      "channel": "API"
5    },
6      "item_details": {
7        "itemCount": "",
8        "itemValue": "",
9        "itemCategory": ""
10     },
11      "cust_details": {
12        "customerName": "",
13        "emailId": "",
14        "mobileNumber": "965680964",
15        "uniqueId": "",
16        "isLoggedIn": "Y"
17     },
18      "pg_details": {
19        "pg_Id": "67ee846571e740418d688c3f",
20        "paymode": "WA",
21        "scheme_Id": "7",
22        "wallet_type": "telebirr"
23     },
24      "bill_details": {
25        "billAddress": "",
26        "billCity": "",
27        "billState": "",
28        "billCountry": "",
29        "billZip": ""
30     }
}
{
  "card_details": {
    "cardNumber": "",
    "expiryMonth": "",
    "expiryYear": ""
  }
}
```

```
1      "transactionType": "SALE",
2      "sucessUrl": "",
3      "failureUrl": "",
4      "channel": "API"
5    },
6      "item_details": {
7        "itemCount": "",
8        "itemValue": "",
9        "itemCategory": ""
10     },
11      "cust_details": {
12        "customerName": "",
13        "emailId": "",
14        "mobileNumber": "965680964",
15        "uniqueId": "",
16        "isLoggedIn": "Y"
17     },
18      "pg_details": {
19        "pg_Id": "67ee846571e740418d688c3f",
20        "paymode": "WA",
21        "scheme_Id": "7",
22        "wallet_type": "telebirr"
23     },
24      "bill_details": {
25        "billAddress": "",
26        "billCity": "",
27        "billState": "",
28        "billCountry": "",
29        "billZip": ""
30     }
  }
{
  "card_details": {
    "cardNumber": "",
    "expiryMonth": "",
    "expiryYear": ""
  }
```

```
1      "cvv": "",  
2      "cardName": ""  
3    },  
4    "other_details": {  
5      "udf1": "",  
6      "udf2": "",  
7      "udf3": "",  
8      "udf4": "",  
9      "udf5": "",  
10     "udf6": "",  
11     "udf7": ""  
12   },  
13   "ship_details": {  
14     "shipAddress": "",  
15     "shipCity": "",  
16     "shipState": "",  
17     "shipCountry": "",  
18     "shipZip": "",  
19     "shipDays": "",  
20     "addressCount": ""  
21   },  
22   "txn_details": {  
23     "agId": "yagout",  
24     "meld": "202505090001",  
25     "orderNo": "00012100",  
26     "amount": "1",  
27     "country": "ETH",  
28     "currency": "ETB",  
29     "transactionType": "SALE",  
30     "sucessUrl": "",  
       "failureUrl": "",  
       "channel": "API"  
    },  
    "item_details": {  
      "itemCount": "",  
      "itemValue": ""  
    }
```

```
1      "itemCategory": ""  
2    },  
3    "cust_details": {  
4      "customerName": "",  
5      "emailId": "",  
6      "mobileNumber": "965680964",  
7      "uniqueId": "",  
8      "isLoggedIn": "Y"  
9    },  
10   "pg_details": {  
11     "pg_Id": "67ee846571e740418d688c3f",  
12     "paymode": "WA",  
13     "scheme_Id": "7",  
14     "wallet_type": "telebirr"  
15   },  
16   "bill_details": {  
17     "billAddress": "",  
18     "billCity": "",  
19     "billState": "",  
20     "billCountry": "",  
21     "billZip": ""  
22   }  
23 }
```

## Request after Encrypting the Plain Request:

lq1FilIDHAD/9Cx4innDJVF+oF2ow673w/wDf969AT48VPCrvfLwX2mdC-QLYB77SSq5n5sOqsxT/+fizkAEMyTi8b90BLqKbVIH6aR2uYv1dk9PG0XYXF4LOp6mzuTYInU1AydyCCNSEURPlxn//QC3YAZCt13YGi2a+ZPp8TVLE7YNQxjeqaZV9uloRLLl7zOXw0G5aU+xalGwSsFZLsE2GKN9hg+zQJ2VOJFZao88TBcrXhQOnMA9r88zNlXuoqBfdQoBwvcU/ERcRevRjcYmz1lum28NRDl0//Xlnx1JBXlh94XcNl6L5U+L7HwsAUeEVM6Fmb5+HvW7W8ZozspukVfeXdxNO2FdvcqccXrrEWpSJHop2RtLwEqOuA3B3e2Fpbb9OBmrcCMvgRR3KgjDTll/LMuOpb/2Ad3VLLfTfwmPoyzwTzFuKpKc3c/+A52LfNgBJ4FlmHQ37b70Nf61ORjDL4g0TdHaNb1hw7IgwOxo1IDOplr83kdHoeyF3MMGr11AqeEUOBpblU4NvLi6++dEUdetH0Ysglrilis8MRhNJAhVqrIHFnxdaag9CO2nsQRDN2d+CpYD0bEej9oz0c1EOD95RS1DnZyNT1W0pj3CaPDS9ehZlYVmOfzbDKRs7QpDJf+Lf52xU1lByCEGXy958IbsjaCo8aaBytpsVAxC/VmkWJh1TSXeg3bZC3Ck7Lu08U6Rvfinz42qEbPQaID7RyUFQPCDwKvUKjcsplIruYh9dmO2gPVYU1BSPxjpqaP7BxzjfYf9y0Eb+JMMGEzqg/MaohGeeROruAUea6h6820bAl2/NBZ7s/N8VHdZtepwjV/tUAJTg9BnZWrbzB/VbVb9SHiPNOWHr5Sz iTJV2RgmP7eXgf5EBPHnG48M2vcUPZ98Uy8Z6BuW8jv+63EunktJE5Lo5A0rPKr4F188thbEJEyBxz1sNyD0se4K1QFLx4A6vmyqukXRrTD7EmKFqDyp6rKLJINxJhPl3VraYYtFodM47DPXbFfeDKBlHMUJRVnDCZQV64QHYMps39QJ23ZF1Kle2aS5Jo3uqjwEc7vNC8JcnleD3QvAZ03p3TedcegBuXqVhfZM35Y5+T3/2L+fd3MsHmjefvGblEHh27aUOHcerULTxOnXZyFMYChZOv7fFjsHY82VwQTrMKvBED+uXjrkLFFx3BXnhzldc24Pi9Y6rV0Jn+SNTvEz2rS+4co8U8fLgBTDLGVXsZMaFjnbtCillu4eSwRHyeFYNZwCtMs0cYXmsf+hqMHiyUGMeOVatO7c9dr8n85m2bOugCGMv03KH+DiBnxFO6dK+NHgFBbWZcbFxAVAmQqzAIPIFtvSoK+G343zfHhaHgU8ljKgFaEnGJU1DGhOC4IQJppVCt5t3HBA/UvS/JNPRe0G8R0Qdvnd1rMeABnXwSIsf1ngR3a7gITTvhw17W5riqHXYW/0PXzyqO8yMm8cugMtr35LAX2+vWU7NktqKKpsGPBCLY7fuocJ7FKNesJFUVx9wWFyidJyN1A0WJ/GeG7VU0FwTdTf0BykNBfCao7JO9RKla6jWVDS8Nv75FUPtEztL1OCltUES25ZKlYAfQ64N6w6ug6kPgdQDEHe795fTxPRCVoha270w8D8jXjQksXgPsIWdZUhckZ5GqsIcD73M8jpTdJg/bBKb1IlCfW2TVYYxRhBB011oq/SpXxRtVJlL3LYhpRgLB7wwAFkUN74bJWidc+rspjiKs+CSmil+RV6PwMh09ljMfdMzEG/lZWgPyyNBlrz7Sf8oOmDyCFpr0CvoW9/YT8jwaK5VzGNs/SNdKClozKaL8jMZPrfLEM8rZsZqR3

## Final Sample Request Format:

```
{  
    "merchantId": "202505090001",  
    "merchantRequest": "lq1FilIDHAD/9Cx4innDJVF+oF2ow673w/wDf969AT48VPCrv-  
fLwX2mdC-  
    QLYB77SSq5n5sOqsxT/+fizkAEMyTi8b90BLqKbVIH6aR2uYv1dk9PG0XYXF4LOp6mzu  
    TYinU1AydyCCNSEURPlxn//QC3YAZCt13YGi2a+ZPp8TVLE7YNQxjeqaZV9uloRLLl7zO  
    Xw0G5aU+xalGwSsFZLsE2GKN9hg+zQJ2VOJFZao88TBcrXhQOnMA9r88zNlXuoqBfd  
    QoBwvcU/ERcRevRjcYmz1lum28NRDl0//Xlnx1JBXlh94XcNl6L5U+L7HwsAUeVM6F  
    mb5+HvW7W8ZozspukVfeXdxNO2FdvcqccXrrEWpSJHop2RtLwEqOuA3B3e2Fpb90  
    BmrcCMvgRR3KgjDTll/LMuOpb/2Ad3VLLFTfwmPoyzwTzFuKpKc3c/+A52LfNgBJ4Flm  
    HQ37b70NF61ORjDL4g0TdHaNb1hw7IgwOxo1IDOpIr83kdHoeyF3MMGr11AqeEUOB  
    pblU4NvLi6++dEUdetH0Ysglrlis8MRhNJAhVqlHFnxdaag9CO2nsQRDN2d+CpYD0bE  
    ej9oz0c1EOD95RS1DnZyNT1W0pj3CaPDS9ehZlYVmOfzbDKRs7QpDJf+Lf52xU1lByC  
    EXGy958IbsjaCo8aaBytpsVAxC/VmkWJh1TSXeg3bZC3Ck7Lu08U6Rvfinz42qEbPQaID  
    7RyUFQPCDwKvUKjcsplIruYh9dmO2gPVYU1BSPxjpqaP7BxzfYf9y0Eb+JMMGEzqg  
    /MaohGeeROruAUea6h6820bAl2/NBZ7s/N8VHdZtepwjV/tUAJTg9BnZWrbzB/VbVb9  
    SHiPNOWHr5SziTJV2RgmP7eXgf5EBPHnG48M2vcUPZ98Uy8Z6BuW8jv+63EunktJE5  
    Lo5A0rPKr4F188thbEJEyBxz1sNyD0se4K1QFLx4A6vmyqukXRrTD7EmKFlqDyp6rKLJI  
    /NxJhPl3VraYYtFodM47DPXbFfeDKBlHMuJRVnDCZQV64QHYMps39QJ23ZF1Kle2  
    aS5Jo3uqjwEc7vNC8JcnleD3QvAZ03p3TedcegBuXqVhfZM35Y5+T3/2L+fd3MsHmjef  
    vGblEHh27aUOHcerULTxOnXZyFMYChZov7fFjsHY82VwQTrMKvBED+uXjrkLFFx3BX  
    nhzldc24Pi9Y6rV0Jn+SNtvEz2rS+4co8U8fLgBTDLGVXsZMaFjnbtCillu4eSwRHyeFyNZ  
    wCtMs0cYXmsf+hqMHiyUGMeOVatO7c9dr8n85m2bOugCGMv03KH+DiBnxFO6dK+N  
    HgFBbWZcbFxAVAmQqzAIPlFtvSoK+G343zfHhaHgU8ljKgFaEnGJu1DGhOC4IQJppV  
    Ct5t3HBA/UvS/JNPRe0G8R0Qdvnd1rMeABnXwSIsf1ngR3a7gITTvhw17W5riqHXYW/  
    0PXzyqO8yMm8cugMtr35LAX2+vWU7NktqKKpsGPBCLY7fuocJ7FKNesJFUVx9wWFy  
    idJyN1A0WJ/GeG7VU0FwTdHdTf0BykNBFcao7JO9RKla6jWVDS8Nv75FUPtEztL1OCl  
    tUES25ZKIYAfQ64N6w6ug6kPgdQDEHe795fTxPRCVoha270w8D8jXjQksXgPsIWdZU  
    hckZ5GqsIcD73M8jpTdJg/bBKb1IlCfW2TVYYxRhBB011oq/SpXxRtVJlL3LYhpRgLB7w  
    wAFkUN74bJWidc+rspjiKs+CSmil+RV6PwMh09ljMfdMzEG/lZWgPyyNBrlrz7Sf8oOmD  
    yCFpr0CwoW9/YT8jwaK5VzGNs/SNdkClozKaL8jMZPrfLEM8rZsZqr3"  
}
```

## Screenshot of POSTMAN FYR:

The screenshot shows the POSTMAN interface with a POST request to <https://uatcheckout.yagoutpay.com/ms-transaction-core-1-0/apiRedirection/apiIntegration>. The Body tab is selected, displaying the following JSON payload:

```

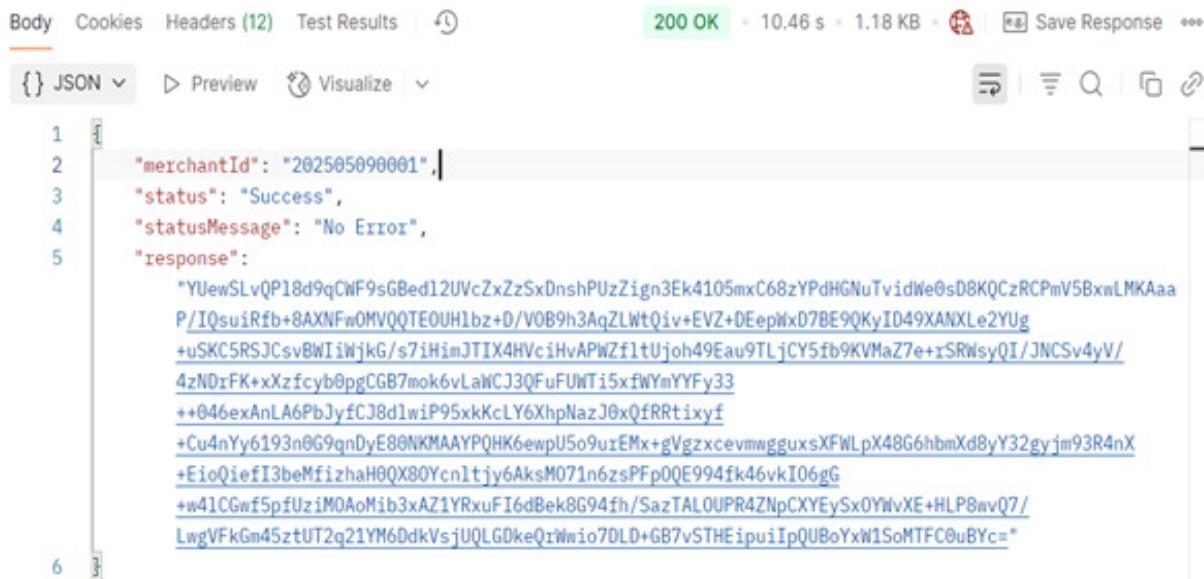
1  {
2   "merchantId": "202505090001",
3   "merchantRequest": "1q1FiiIDHAD/9Cx4innDJVF+oF2ow673w/wDf969AT48VPCrvfLwX2mdCQLYB7SSq5n5s0qsxT/
+fiZkAEMyTi8b90BLqKbVIH6aR2uYv1dk9PG0XYXF4L0p6mzuTYinU1AydyCCNSEURP1xn//QC3YAZCt13YGi2a
+ZPp8TVLE7YNQxjeqaZV9uIoRLl17z0Xw0G5aU+xa1GwSsFZLsE2GKN9hg
+zQJ2VOJFZao88TBcrXhQ0nMA9r88zN1XuoqBfdQoBwvcU/ERcRevRjcYmz1lum28NRD10//X1nx1JBX1h94xCn16L5U
+l7HwsAUEeVM6Fmb5+HvW7W8ZozspukVfeExdxN02FdvcqccXriEWpSJHop2RtlwEqouA3B3e2Fpb90BmzccCMvgRR3KgjDT11/
LMu0pb/2Ad3VLLfTfwmPoyzwTzFuKpKc3c/
+A52LfNgBJ4FlmhQ37b70Nf610RjDL4g0TdHaNb1hw7IgwOxo1ID0pIr83kdHoeyF3MMGr11AqeEU0Bpb1U4NvLi6
++dEudefH0Ysglrlilis8MRhNJAhVqr1HFnxdaAg9C02nsQRDN2d
+CpYD0bEej9oz0c1E0D95RS1DnZyNT1W0pj3CaPDS9ehZ1YVm0fzbDKRs7QpDJf
+Lf52xU11ByCEXGy958IbsjaCo8aaBytpsVAXC/"

```

## Encrypted Response Parameters:

```
{
  "merchantId": "202505090001",
  "status": "Success",
  "statusMessage": "No Error",
  "response": "YUewSLvQPl8d9qCWF9sGBedl2UVcZxZzSxDnsh-
PUzZign3Ek41O5mxC68zYP-
dHGNuTvidWe0sD8KQCzRCPmV5BxwLMKAaaP/IQsuiRfb+8AXNFwOMVQQTEOUHlb
z+D/VOB9h3AqZLWtQiv+EVZ+DEepWxD7BE9QKyID49XANXLe2YUg+uSKC5RSJCsvB
WliWjkG/s7iHimJTIx4HVciHvAPWZfltUjoh49Eau9TLjCY5fb9KVMaZ7e+rSRWsyQI/JN
CSv4yV/4zNDrFK+xXzfcyb0pgCGB7mok6vLaWCJ3QFuFUWTi5xfWYmYYFy33++046e
xAnLA6PbjyfCJ8dlwiP95xkKcLY6XhpNazJ0xQfRRtixyf+Cu4nYy6193n0G9qnDyE80N
KMAAYPQHK6ewpU5o9urEMx+gVgzxcevmwgguxsXFWLpX48G6hbmXd8yY32gyjm9
3R4nX+EioQiefI3beMfizhaH0QX8OYcnltjy6AksMO71n6zsPFpOQE994fk46vkIO6gG+
w4lCGwf5pfUziMOAoMib3xAZ1YRxuFI6dBek8G94fh/SazTALOUPR4ZNpCXYEySxOY
WvXE+HLP8wvQ7/LwgVFkGm45ztUT2q21YM6DdkVsjuQLGDkeQrWwio7DLD+GB7vS
THEipuipQUBoYxW1SoMTFC0uBYc="
}
```

## Screenshot of POSTMAN FYR:



The screenshot shows a POSTMAN interface with the following details:

- Body:** JSON
- Response Status:** 200 OK
- Time:** 10.46 s
- Size:** 1.18 KB
- Save Response:** Enabled

```

1
2   "merchantId": "202505090001",
3   "status": "Success",
4   "statusMessage": "No Error",
5   "response":
6     "YUewSLvQP18d9qCWF9sG8ed12UVcZxZzSxDnshPUzZign3Ek4105mxC68zYPdHGNuTvidWe0sD8KQCzRCPmV5BxwLMKAaa
P/IQsuiRfb+8AXNFwOMVQQTEOUH1bz+D/VOB9h3AqZLWtQiv+EVZ+DEepWxD7BE9QKyID49XANXLe2YUg
+uSKC5RSJCsvBWIiWjkG/s7iHimJTIx4HVciHvAPWZfltUjoh49Eau9TLjCY5fb9KVMaZ7e+rSRWsyQI/JNCSv4yV/
4zNDrFK+xXzfcyb0pgCGB7mok6vLaWCJ3QFuFUWTi5xfWYmYYFy33
++046exAnLA6PbJyfCJ8d1wiP95xkKcLY6XhpNazJ0xQfRRtixyf
+Cu4nYy6193n0G9qnDyE80NKMAAYPQHK6ewpU5o9urEMx+gVgzxcevmwgguxsXFWLpX48G6hbmXd8yY32gyjm93R4nX
+EioQiefI3beMfizhaH0QX80Ycnltjy6AksMO71n6zsPFpOQE994fk46vkIO6gG
+w4lCGwf5pfUziMOAoMib3xAZ1YRxuFI6dBek8G94fh/SazTALOUPR4ZNpCXYEySx0YwvXE+HLP8wvQ7/
LwgVFkGm45ztUT2q21YM6DdkVsjUQLGDkeQrWwio7DLD+GB7vSTHEipuiIpQUBoYxW1SoMTFC0uBYc="
    
```

## Encrypted Response Parameters:

```
{
  "merchantId": "202505090001",
  "status": "Success",
  "statusMessage": "No Error",
  "response": "YUewSLvQP18d9qCWF9sG8ed12UVcZxZzSxDnsh-
PUzZign3Ek4105mxC68zYP-
dHGNuTvidWe0sD8KQCzRCPmV5BxwLMKAaaP/IQsuiRfb+8AXNFwOMVQQTEOUH1b
z+D/VOB9h3AqZLWtQiv+EVZ+DEepWxD7BE9QKyID49XANXLe2YUg+uSKC5RSJCsvB
WliWjkG/s7iHimJTIx4HVciHvAPWZfltUjoh49Eau9TLjCY5fb9KVMaZ7e+rSRWsyQI/JN
CSv4yV/4zNDrFK+xXzfcyb0pgCGB7mok6vLaWCJ3QFuFUWTi5xfWYmYYFy33++046e
xAnLA6PbJyfCJ8d1wiP95xkKcLY6XhpNazJ0xQfRRtixyf+Cu4nYy6193n0G9qnDyE80N
KMAAYPQHK6ewpU5o9urEMx+gVgzxcevmwgguxsXFWLpX48G6hbmXd8yY32gyjm9
3R4nX+EioQiefI3beMfizhaH0QX80Ycnltjy6AksMO71n6zsPFpOQE994fk46vkIO6gG+
w4lCGwf5pfUziMOAoMib3xAZ1YRxuFI6dBek8G94fh/SazTALOUPR4ZNpCXYEySx0Y
WvXE+HLP8wvQ7/LwgVFkGm45ztUT2q21YM6DdkVsjUQLGDkeQrWwio7DLD+GB7vS
THEipuiIpQUBoYxW1SoMTFC0uBYc="
}
```

## Encrypted Response Parameters:

### Encryption API

The Encryption API is used to encrypt the data.

**Post URL** <https://uatcheckout.yagoutpay.com/ms-transaction-core-1-0/othersRedirection/encryption>

### Encryption API

Method: POST

Content-Type: Application-Text

Parameters to be passed in the Header Section:

me\_id : 202505090001 (This is just a Test MID)

### Plain Request:

SG1a1jGJ84L16p9JARDmmSHZdffP++0FzQ1Ppa7Kt4qZkl23vPX3DYEXAgQ-  
mP2u7ZrYvYzGZA1muQnnVCMWYCw==  
//// (Enter the data that you want to decrypt)

### Response:

```
{Status=Success, Response= {  
    "me_id": "202505090001",  
    "amount": "100"  
} }
```