

**REST API Document
For Financial Transactions
to
InnoPay API
from POS Client/Terminal**

**Last Updated On: May 15, 2020
VERSION: v1.3**

Table of Contents

1.	Introduction.....	4
2.	Environment Keys to be provided	4
3.	API Authentication	4
4.	Encryption mechanism for credit card information	4
5.	Service: chargecustomer	5
	a. Web Service Description.....	5
	b. Request JSON format	6
	i. With Credit card number.....	6
	ii. With Card token.....	7
	c. Input parameters of Web Service –	8
	d. Response JSON format.....	10
	e. Output parameters of Web Service	10
	f. Expected Response Code	11
6.	Service: refund.....	12
	a. Web Service Description.....	12
	b. Request JSON format	12
	i. With existing retref number	12
	ii. With existing order id.....	12
	c. Response JSON format	13
	i. With existing retref number	13
	ii. With existing order id.....	13
7.	Service: void	13
	a. Web Service Description.....	13
	b. Request JSON format	14
	c. Response JSON format	14
8.	Service: memberbill	15
	a. Web Service Description.....	15
	b. Request JSON format	15
	c. Response JSON format	16
9.	Service: paymentmethod.....	17
	a. Web Service Description.....	17
	b. Request JSON format	17
	c. Response JSON format	18

10. Service: keyauth.....	18
a. Web Service Description.....	18
b. Request JSON format	18
c. Response JSON format	19
11. Service: enctxt.....	19
a. Web Service Description.....	20
b. Request JSON format	20
c. Response JSON format	20
12. Service: transactionrec	20
d. Web Service Description	21
e. Request JSON format.....	21
13. Encryption Examples	22

1. Introduction

This API serves to allow external Point of Service systems to connect to InnoPay for financial transactions. Currently, this would supply Credit/Debit charge and refund transactions. It is also intended to extend to Gift Cards (stored value). In this manner, our customer uses one point of integration with us to support multiple financial conveniences to the customer. We will expand if needed to rewards/loyalty cards, etc.

Purpose of this document is to describe request/response objects of RESTful web services for InnoPay POS Client API as a gateway to financial transactions.

This version specifies the Credit/Debit transaction messages. June 7, 2019

2. Environment Keys to be provided

Following values will be provided by InnoPay system to the client:

- Username and Password (Service Authentication)
- Secret Key (For encrypting secure info)
- Web Service End Point

3. API Authentication

We use Basic Authorization for our rest APIs. The sender (client app) places a “**username:password**” into the request header. The username and password are encoded with Base64, which is an encoding technique that converts the username and password into a set of 64 characters to ensure safe transmission.

Here’s an example of a Basic Authorization in a request header:

Authorization: Basic bG9sOnNIY3VyZQ==

4. Encryption mechanism for credit card information

We use SHA-256 digested secret key to encrypt information like Credit Card No, CCID, Credit Card Exp Month and Year etc with AES encryption technique using digested secret key and then encoded as Base64.

All modern programming languages have libraries that can aid in generating the defined encryption mechanism. Java code to encrypt a string with a secret Key would be as follows:

```

public static String encrypt(String value, String secret) {
    try {
        byte[] key = secret.getBytes("UTF-8");
        MessageDigestsha = MessageDigest.getInstance("SHA-256");
        key = Arrays.copyOf(sha.digest(key), 16);
        Cipher cipher = Cipher.getInstance("AES");
        cipher.init(Cipher.ENCRYPT_MODE, new SecretKeySpec(key, "AES"));
        return Base64.getEncoder().encodeToString(cipher.doFinal(value.getBytes("UTF-
8"))));
    } catch (Exception e) {
        logger.debug("Error while encrypting: " + e.toString());
    }
    return null;
}

```

At the end of this document are C# .NET and PHP examples of accomplishing the encryption.

Note: We will provide the secret key to the client. The secret key you use to encrypt secured data must be kept secret. You cannot embed this key in a webpage or in the code of a mobile app. Any rest service calls you make encrypted data must come from your own servers. This is to prevent a malicious user from discovering your secret key.

5. Service: chargecustomer

Service Name	Method Type	Service URL
chargecustomer	POST	hostname/ebizservice/posapi/rest/ebiz/chargecustomer

a. Web Service Description

This web service is used to register your customer with the InnoPay system, provide them with a VCard Account, set up the Credit Card swiped at the POS as a Virtual Card Funding Source, and make a call to Card Connect to authenticate customer card and charge customer for order. This service returns a response for Approval or Denial w/reason code. The monies are deposited into a Datamotio holding account for the stored value on the cards. This stored value will be transferred to the Client Merchant in a subsequent vCard transaction.

This will be a REST service call with below mentioned input parameters in JSON format.

12 of these fields are required. The others are either optional or not required.

If you are using an end-to-end encryption point for security, and your scanner tokenizes a card, then you will send the token to us. If there is no tokenization then send the billing fields, including the credit card number.

b. Request JSON format

i. With Credit card number

```
{
  "request_id":"111111000",
  "store_id":"a1101",
  "terminal_device_id":"111111",
  "vendor_id":"abc111",
  "customer":{
    "first_name":"John",
    "middle_initial":"",
    "last_name":"",
    "customer_id":"100121",
    "account_number":"1236540",
    "address1":"Address 1",
    "address2":"Address 2",
    "city":"Atlanta",
    "state":"GA",
    "zip":"45120"
  },
  "order":{
    "order_id": "88590345",
    "order_amount":"100",
    "order_date":"MM/DD/YYYY",
    "payment_type":"DEBIT "
  },
  "card":{
    "credit_card_no":"7Qqv6IVnGoGmfJdPyjw4dpHiL7EpLPeC+kkQZZDOvTY="
```

```
"credit_card_type": "VISA",
"ccid": "NGT9Lm6fI9uWXYQZMxioig==",
"credit_card_exp_month": "FDPX0QwC49sK2aA08+HWwQ==",
"credit_card_exp_year": "LVLk2IWQIYCUwQMpqHF36A==",
"name_on_account": "my card123",
"billing_address": "Addr 11",
"billing_city": "city 44",
"billing_state": "CA",
"billing_zip": "12346",
"billing_phone": "1111122222"
}
}
```

ii. With Card token

```
{
  "request_id": "111111021",
  "store_id": "a1110",
  "terminal_device_id": "1111110",
  "vendor_id": " abc111",
  "customer": {
    "first_name": "Raman",
    "middle_initial": " Test6",
    "last_name": "Raj6",
    "customer_id": "1001236",
    "account_number": "12365456",
    "address1": "Address 16",
    "address2": "Address 26",
    "city": "Atlanta",
    "state": "GA",
```

```

    "zip": "45126"
  },
  "order": {
    "order_amount": "700",
    "order_id": "1027",
    "order_date": "05/28/2019",
    "payment_type": "DEBIT"
  },
  "card": {
    "card_token": "9477709629051443",
    "credit_card_exp_month": "o85qGydbbHBilD0OIWlynw==",
    "credit_card_exp_year": "VpRXduRO2bvVTKvklXWwHw=="
  }
}

```

Notes on Request ID: this is a generated unique ID. It cannot be the order ID or the customer ID.

Notes on Name: You send the Customer Name in the first portion of the request. You send the scanned name from the card in the NAME_ON_CARD field. It may be different, but the tracking is tracking a real verified customer for compliance so you must have both.

Notes on Vendor, Location, Device: We have provided defaults for your testing, but each of your clients will have their own values for these items.

Notes on Customer ID: any POS will have a unique identifier for the Customer. If there is only an account ID, then put that in Customer ID and Account ID, they can be the same. If there is only a Customer ID and no Account ID from the POS you can do the reverse and put the same value in both.

c. Input parameters of Web Service -

Note: the only values that a Scanner might not provide are the Vendor, Store ID, Customer Account # of this specific customer in your POS system, and the Order ID.

Field Name	Req ?	Max Length	Type	Description
------------	----------	---------------	------	-------------

request_id	Y	20	N	Generated ID from Flourish to identify this request. We suggest that you do not use the Order ID, because you may have a subsequent request on the same order_id, such as a refund or when the first card proffered is denied and the customer tries a second card.
vendor_id	Y	20	AN	IDENTIFY Client Individual from system
store_id	Y	20	AN	LOCATION, actual physical store
terminal_device_id	N	30	AN	From Scanner device
first_name	Y	20	AN	Customer First Name
middle_initial	N	10	AN	Customer Middle Initial
last_name	Y	30	AN	Customer Last Name
customer_id	Y	20	AN	If Flourish has a system based Customer ID that is not the same as the Account #.
account_number	Y	20	N	Flourish Account #
address1	N	60	AN	Customer home address
address2	N	30	AN	Customer home address
city	N	25	AN	Customer city
state	N	2	AN	Customer state
zip	N	10	AN	Customer postal code – may send 5 digits or 99999-9999
order_id	Y	20		Transaction ID in customer's system/site
order_amount	Y		N	Amount to be charged
order_date	Y		DATE	MM/DD/YYYY format
payment_type	Y	10	AN	CREDIT/DEBIT/GIFT
card_token	Y	320	AN	TOKENIZED BY CARD CONNECT
credit_card_no	C	100	N	ENCRYPTED DATA BEFORE REST SERVICE CALL
credit_card_type	C	10	AN	Credit Card Type (VISA/MasterCard/Discover)
ccid	C	50	N	ENCRYPTED DATA BEFORE REST SERVICE CALL
credit_card_exp_m onth	C	50	N	ENCRYPTED DATA BEFORE REST SERVICE CALL
credit_card_exp_ye	C	50	N	ENCRYPTED DATA BEFORE REST

ar				SERVICE CALL
name_on_account	C	60	AN	Customer Name on account
billing_address	C	60	AN	Customer Billing Address
billing_city	C	30	AN	Billing City
billing_state	C	2	AN	Billing State
billing_zip	C	10	AN	Billing Zip
billing_phone	C	14	AN	Billing Phone

d. Response JSON format

```
{
  "status": "1",
  "resp_code": "200",
  "resp_message": "Approval",
  "retref_no": "305635717722",
  "avsresp": "Z",
  "cvvresp": "M",
  "token": "9376856895050005"
}
```

e. Output parameters of Web Service

1. status

- Returns code (0 or 1) depending on the execution of requested action at server.

1 – Success

0 – Failure

2. resp_code

- Returns http response code based on the status.

3. resp_message

- Returns text description of response.

3. retref_no

- Card Connect retrieval reference number from response
- 4. avsresp
 - Card Connect address verification response code
- 5. cvvresp
 - Card Connect card verification value response code
- 6. token
 - Card Connect card token value in response for credit card number/card token

f. Expected Response Code

Response Code	Description
200	Transaction successful
4001	Data format error
4002	Data Type error
4003	Required fields
4004	Invalid Value
4005	Bad HTTP header
4006	Socket Error
500	Server Error

6. Service: refund

Service Name	Method Type	Service URL
refundrequest	POST	hostname/ebizservice/posapi/rest/ebiz/refundrequest

a. Web Service Description

This web service is used to refund partial or full amount to a customer related to a specific retref number or Order ID, credit to be applied back to the card used to charge. If refund_amount is not provided full amount refund for respective transaction will be processed.

This will be a REST service call with below mentioned input parameters in JSON format.

b. Request JSON format

i. With existing retref number

```
{
  "request_id":"111111000",
  "store_id":"a1101",
  "terminal_device_id":"111111",
  "vendor_id":"abc111",
  "customer_id":"100121",
  "refund_amount" : "125",
  "retref_no": "234645247733"
}
```

ii. With existing order id

```
{
  "request_id":"111111000",
  "store_id":"a1101",
  "terminal_device_id":"111111",
  "vendor_id":"abc111",
  "customer_id":"100121",
  "order_id":"10599401",
}
```

```
    "refund_amount": "975"  
  }
```

c. Response JSON format

i. With existing retref number

```
{  
  "status": "1",  
  "resp_code": "200",  
  "resp_message": "Approval",  
  "retref_no": "234395147812",  
  "refund_amount": "125"  
}
```

ii. With existing order id

```
{  
  "status": "1",  
  "resp_code": "200",  
  "resp_message": "Approval",  
  "retref_no": "234192149719",  
  "refund_amount": "975"  
}
```

7. Service: void

Service Name	Method Type	Service URL
voidrequest	POST	hostname/ebizservice/posapi/rest/ebiz/voidrequest

a. Web Service Description

This web service is used to cancel a transaction that is in either "Authorized" or "Queued for Capture" status. This service will return amount related to a specific Order ID or Retref No. It will be a REST service call with below mentioned input parameters in JSON format.

b. Request JSON format

i. With existing retref number

```
{  
    "request_id": "313b16d3-4ca4-4599",  
    "store_id": "58b9b09d280e2026b9",  
    "terminal_device_id": "18125PP84074309",  
    "vendor_id": "VENDORID",  
    "customer_id": "9",  
    "retref_no": "235705751502"  
}
```

ii. With existing order id

```
{  
    "request_id": "313b16d3-4ca4-4602",  
    "store_id": "58b9b09d280e2026b9",  
    "terminal_device_id": "18125PP84074309",  
    "vendor_id": "VENDORID",  
    "customer_id": "9",  
    "order_id": "01129"  
}
```

c. Response JSON format

```
{  
    "status": "1",  
    "resp_code": "200",  
    "resp_message": "Approval",  
    "retref_no": "235705751502",  
    "remaining_amount": "0.00"  
}
```

8. Service: memberbill

Service Name	Method Type	Service URL
memberbill	POST	hostname/ebizservice/posapi/rest/ebiz/memberbill

a. Web Service Description

This web service is used for subsequent monthly charges. This service requires encrypted value of card token returned from chargecustomer service. This token is completely safe for the customer to store, and can be used do subsequent charges monthly at card connect. This service will return response similar to chargecustomer service. It will be a REST service with below mentioned input parameters in JSON format.

b. Request JSON format

```
{  
  "request_id": "981111012",  
  "store_id": "SWC-37",  
  "terminal_device_id": "00000",  
  "vendor_id": "VENDORID",  
  "custommerchant" : "MERCHANTID",  
  "customproduct" : "Product1",  
  "customphone" : "8314256328",  
  "customer": {  
    "customer_id": "9811250",  
    "account_number": "9811250"  
  },  
  "order": {  
    "order_id": "9811246",
```

```
"order_amount": "1160",
"order_date": "12/30/2019",
"payment_type": "CREDIT"
},
"card": {
  "card_token": "JXrjMQVWcYyOMoGX3Jnd2IdlK9cSFRFPFeDY/Kl4eNU=",
  "credit_card_exp_month": "9swS9qIySaiJIfZ2DyWTlQ==",
  "credit_card_exp_year": "25CVpkIOpgQBs30s6SQkQQ=="
}
}
```

c. Response JSON format

```
{
  "status": "1",
  "avsresp": "Z",
  "cvvresp": "X",
  "resp_code": "200",
  "resp_message": "Transaction completed successfully",
  "retref_no": "364508233864",
  "token": "9422925921134242"
}
```


9. Service: paymentmethod

Service Name	Method Type	Service URL
paymentmethod	POST	hostname/ebizservice/posapi/rest/ebiz/paymentmethod

a. Web Service Description

This web service is used to change the subscription to another card. The User would be able to put a new card, for which we would also save the new token. It will be a REST service with below mentioned input parameters in JSON format.

b. Request JSON format

```
{
  "request_id": "981111015",
  "store_id": "SWC-37",
  "terminal_device_id": "00000",
  "vendor_id": "VENDORID",
  "name" : "Durgesh Verma2",
  "customer_id": "9811250",
  "profile_id": "17025229333995633721",
  "acct_id" : "2",
  "address" : "Address244",
  "city" : "city 244",
  "state" : "GA",
  "zip" : "14244",
  "country" : "US",
  "phone" : "2415063894",
  "card": {
    "credit_card_number":
      "yo2hTmAh4Sm/OZcoB2WseYdlK9cSFRFPFeDY/K14eNU=",
    "credit_card_exp_month": "9swS9qIySaiJIfZ2DyWTlQ==",
    "credit_card_exp_year": "25CVpkIOpgQB30s6SQkQQ=="
  }
}
```

```
}  
}
```

c. Response JSON format

```
{  
  "status": "1",  
  "profileid": "17025229333995633721",  
  "acctid": "2",  
  "defaultacct": "N",  
  "resp_code": "200",  
  "resp_message": "Profile Saved",  
  "token": "9418594164541111"  
}
```

10. Service: keyauth

Service Name	Method Type	Service URL
keyauth	POST	hostname/ebizservice/posapi/rest/ebiz/keyauth

a. Web Service Description

This web service is used to authenticate whether api key send in request is valid or not. If apikey is valid then it will return store details and their card connect merchant Ids. It is a REST service with below mentioned input parameters in JSON format.

b. Request JSON format

```
{  
  "api_key": "HkEtvQid80D21iUNK5VnpSKCTuOE5N",  
  "vendor_id": "VENDORID"  
}
```

c. Response JSON format

```
{
  "status": "1",
  "resp_code": "100",
  "resp_message": "Api Key Authenticated.",
  "vendor_id": "VENDORID",
  "stores": [
    {
      "store_name": "SWC BONITA SPRINGS, LLC",
      "store_id": "SWC-11",
      "merchant_id": "496160873888"
    },
    {
      "store_name": "SWC BRANDON, LLC",
      "store_id": "SWC-37",
      "merchant_id": "496160873888"
    },
    {
      "store_name": "CBD-HEMP WEB SITE SHOPIFY",
      "store_id": "CBD-HEMP",
      "merchant_id": "496160873888"
    }
  ]
}
```

11. Service: enctxt

Service Name	Method Type	Service URL
enctxt	POST	hostname/ebizservice/posapi/rest/ebiz/enctxt

a. Web Service Description

This web service is used to return encrypted value in response for **TEST environment** only. Any new client can use this service to verify that his encryption logic matches with us or not by comparing encrypted value from response. It is a REST service with below mentioned input parameters in JSON format.

b. Request JSON format

```
{
  "vendor_id": "VENDORID",
  "store_id" : "SWC-37",
  "text_to_encrypt" : "9422925921134242"
}
```

c. Response JSON format

```
{
  "status": "1",
  "resp_code": "100",
  "resp_message": "Text Encrypted Successfully.",
  "vendor_id": "VENDORID",
  "encrypted_text": "JXrjMQVWcYyOMoGX3Jnd2IdlK9cSFRFPFeDY/K14eNU="
}
```

12. Service: transactionrec

Service Name	Method Type	Service URL
transactionrec	POST	hostname/ebizservice/posapi/rest/ebiz/ transactionrec

d. Web Service Description

This web service is used to report transactions done by outside card processors, to transfer real-time data to the Merchant Portal. If api key is valid then it will store details of the customer, order, charge info. It is a REST service with below mentioned input parameters in JSON format.

e. Request JSON format

```
{
  "request_id": "981111019",
  "store_id": "ONLINE-STORE1",
  "terminal_device_id": "00000",
  "vendor_id": "MASTER",
  "customer": {
    "first_name": "Raman",
    "middle_initial": " H",
    "last_name": "NOODLES",
    "customer_id": "1001236",
    "account_number": "12365456",
    "address1": "Address 16",
    "address2": "Address 26",
    "city": "Atlanta",
    "state": "GA",
    "zip": "45126",
    "email": "boo@gmail.com"
  },
  "order": {
    "order_id": "9811222",
    "order_amount": "1165",
    "order_date": "12/12/2019",
    "payment_type": "CREDIT"
  },
  "card": {
    "card_token": "9477709629051443",
    "credit_card_exp_month": "4+TBVVaj1BBwsf4q1NGp5Q==",
    "credit_card_exp_year": "LpHRrCxjRYncowtTL00QJA==",
    "transaction_id": "3847203498",
    "confirmation_id": "876553839"
  }
}
```

-- response --

```
{
  "status": "1",
```

```
"resp_code": "200",
"resp_message": "Transaction record saved successfully.",
"transaction_id": "3847203498",
"confirmation_id": "876553839"
}
```

Note: it is not necessary to save the data or acknowledgement of the reported record sent. It is optional.

13. Encryption Examples

Note: JavaScript and Node.js are not suitable for encryption at this PCI level. Please use other language.

.NET C#

```
public static String Encrypt(String strToEncrypt, String secretKey) {
    AesManaged aes = new AesManaged();
    byte[] saltBytes = SHA256.Create().ComputeHash(Encoding.UTF8.GetBytes(secretKey));
    Array.Resize(ref saltBytes, 16 );
    aes.Key = saltBytes;
    aes.Mode = CipherMode.ECB;
    aes.Padding = PaddingMode.PKCS7;
    ICryptoTransform crypto = aes.CreateEncryptor();
    byte[] plain = Encoding.UTF8.GetBytes(strToEncrypt);
    return Convert.ToBase64String(crypto.TransformFinalBlock(plain, 0, plain.Length));
}
```