Payo Payment Gateways API for Smart COD

Instruction Document

This document has been created by SOW PHILIPPINES FULFILLMENT Inc. (PAYO). Its contents may be changed without prior notice. External web links are provided for information only. PAYO does not claim liability for access to and correctness of the referenced content.

COPYRIGHT

Published and Printed in Philippines

Version 2.0.2

Date Released: Nov 7, 2018

CONTACT INFORMATION

For questions relating to this document please contact:

Email: info@payo.asia

Contents

1. Introduction

1.1 Document Purpose

The purpose of this document is to describe the functional requirements of Payo API for Payment Gateways Smart COD clients.

1.2 Target Audience

This document is principally for the application developer who will integrate PAYO API to their system. The application developer should have working knowledge on the following:

- SHA-256 Secure Messaging
- REST Integration
- SOAP Integration
- JSON Conventions

1.3 Software Requirements

- Internet connection supporting HTTPS (or alternatively on a dedicated TCP/IP connection)
- Working knowledge of SOAP, REST and JSON Convention
- Basic knowledge of PHP, Java, Perl, C++ or any other supported programming language
- TLSv1 supporting 128-bit encryption (or stronger)

1.4 Character Set

The character set suggested for setting up HTTPS connectivity is UTF-8, a compressed version of Unicode that uses a single byte for the ASCII 0-127 characters. A plain ASCII string is therefore also a valid UTF-8 string. UTF-8 is the default character set in XML specifications. It supports most international character sets.

1.5 Version History

Date	Version	Description	Author
Apr 3, 2017	1.2	First publication of PAYO public API	ID
Jul 9, 2018	2.0	PAYO API v2	ID
Nov 7, 2018	2.0.2	Added "Instant Cash Collected Notification" (ICCN)	ID

1.6 TEST URLs

Create Order REQUEST

POST Request endpoint: http://api.payo.asia/order/create

Query Order Status

GET request endpoint: http://api.payo.asia/order/get

Shipping Cost Request

GET Request endpoint: http://api.payo.asia/shipping

2. Order Processing

2.1 Create Order REQUEST

Data should be sent as application/x-www-form-urlencoded list of fields (both field names and values should be urlencoded).

Request Fields for COD clients:

Field	Mandatory	Example	Notes	Default Value
contact[firstname]	Yes	John	Contact firstname	
contact[lastname]	Yes	Peterson	Contact last name	
contact[email]	Yes	john@email.com	Contact email	
contact[phone]	Yes	639123456789	Contact phone number	
shipping[country]	Yes	Philippines	Shipping country	
shipping[state]	Yes	Metro Manila	Shipping state	
shipping[city]	Yes	Manila	Shipping city	
shipping[street]	Yes	SomeStreet	Shipping street	
shipping[zip]	Yes	12345	Shipping zip	
shipping[barangay]	No	SomeBarangay	Shipping barangay	
shipping[house]	No	8	Shipping house no	
shipping_method	No	COD	Shipping method (ex. COD or Paid)	Paid
date_to_deliver	No	2018-01-01	Date to deliver	
billing[country]	No	Philippines	Billing country	
billing[state]	No	Metro Manila	Billing state	
billing[city]	No	Manila	Billing city	

billing[street]	No	Astreet	Billing street	
billing[zip]	No	12345	Billing zip	
Items[0][id]	Yes	6x1309936	Item product ID	
Items[0][price]	Yes	1500	Total price of all the items delivered	
Items[0][quantity]	Yes	1	1	
Charges	No	50	Extra charges (shipping etc) or discounts (in this case put it as negative number)	
order_id	Yes	17234	Order ID	
Description	No	“2 baby dresses”	Order description, extra info. Up to 255 characters.	
client_id	Yes	test	The ID assigned to the client (merchant) by Payo.	
pgw_id	Yes	card	Payment Gateway API assigned by Payo.	
Signature	Yes	b5ed930d6917daaf5fcd433dde7e8d9810d2d559f9d2c7ec55bac34cba54fba2	Signature	

COD Clients Remarks:

- items[0][id] - This id is created by the system and sent to the client. Use this product id for all the items, in every order.
- Items[0][price] – The total price of all the items in the order.
- Items[0][quantity] – The quantity of the items will always be 1.

  The order may include several same or different items. However, they are treated as a single item, thus the quantity will always be 1.

  The place to describe the quantity and the details of the items is in the description field.

Request Fields for Warehouse clients:

Field	Mandatory	Example	Notes	Default Value
contact[firstname]	Yes	John	Contact firstname	
contact[lastname]	Yes	Peterson	Contact last name	
contact[email]	Yes	john@email.com	Contact email	
contact[phone]	Yes	639123456789	Contact phone number	
shipping[country]	Yes	Philippines	Shipping country	
shipping[state]	Yes	Metro Manila	Shipping state	
shipping[city]	Yes	Manila	Shipping city	
shipping[street]	Yes	SomeStreet	Shipping street	
shipping[zip]	Yes	12345	Shipping zip	
shipping[barangay]	No	SomeBarangay	Shipping barangay	
shipping[house]	No	8	Shipping house no	
shipping_method	No	COD	Shipping method (ex. COD or Paid)	Paid
date_to_deliver	No	2018-01-01	Date to deliver	
billing[country]	No	Philippines	Billing country	
billing[state]	No	Metro Manila	Billing state	
billing[city]	No	Manila	Billing city	
billing[street]	No	Astreet	Billing street	
billing[zip]	No	12345	Billing zip	
Items[0][id]	Yes	6x1309936	Item 1 product ID	
Items[0][price]	Yes	1500	Item 1 price	
Items[0][quantity]	Yes	1	Item 1 quantity	
Items[n][id]	Yes	6x1309936	Item n product ID	

Items[n][price]	Yes	1500	Item n price	
Items[n][quantity]	Yes	1	Item n quantity	
Charges	No	50	Extra charges (shipping etc) or discounts (in this case put it as negative number)	
order_id	Yes	17234	Order ID	
Description	No	“2 baby dresses”	Order description, extra info. Up to 255 characters.	
client_id	Yes	test	The ID assigned to the client (merchant) by Payo.	
pgw_id	Yes	card	Payment Gateway API assigned by Payo.	
Signature	Yes	b5ed930d6917daaf5fcd433dde7e8d9810d2d559f9d2c7ec55bac34cba54fba2	Signature	

General Remarks:

- Shipping[state] – It is very important to send a real and accurate province name, otherwise the order cannot be processed by the system.

- contact[phone] – The format for the phone number should be: 639…

- Description – Describe here the names and the quantity of the items in the order, and additional information.

- API key:

    1. In case of a Payment Gateway API client - the Payment gateway API key that was assigned by Payo.

    2. In case of a regular client – the client API key that was assigned by Payo.

- Creating the Signature:

1. Concatenate the following fields into one string (no extra concatenating characters like dots, spaces etc.) using the same order:

   contact[email] + contact[firstname] + contact[lastname] + number of items + client_id + pgw_id + your Payment Gateway API key

   Example:

   Contact[firstname] = John
   Contact[lastname] = Peterson
   Contact[email] = john@email.com
   Contact[phone] = 639123456789
   Shipping[country] = Philippines
   Shipping[state] = Metro Manila
   Shipping[city] = Manila
   Shipping[street] = Astreet
   Shipping[zip] = 12345
   Order_id = 17234
   Items[0][id] = 6x1309936
   Items[0][quantity] = 1
   Items[0][price] = 1500
   Description = “2 baby dresses”
   Client[id] = test
   Pgw[id] = card
   Api_key = 098f6bcd4621d373cade4e832627b4f6

   The string to be signed:
   ohn@email.comJohnPeterson1testcard098f6bcd4621d373cade4e832627b4f6

2. Hash that string using SHA256 algorithm.
   b5ed930d6917daaf5fcd433dde7e8d9810d2d559f9d2c7ec55bac34cba54fba2

Sending the request

```
curl -d "contact%5Bfirstname%5D=John&contact%5Blastname
%5D=Peterson&contact%5Bemail%5D=john%40email.com&contact%5Bphone
%5D=639123456789&items%5B0%5D%5Bid
%5D=6x1309936&items%5B0%5D%5Bprice%5D=1500&items%5B0%5D%5Bquantity
%5D=1&shipping%5Bcountry%5D=Philippines&shipping%5Bstate
%5D=Metro+Manila&shipping%5Bcity%5D=Manila&shipping%5Bstreet
%5D=Astreet&shipping%5Bzip
```

```
%5D=12345&order_id=17234&description=2+baby+dresses&client_id=test&pgw_id=card
&signature=b5ed930d6917daaf5fcd433dde7e8d9810d2d559f9d2c7ec55bac34cba54fba2
" -X POST http://api.payo.asia/order/create
```

Response

Response is a JSON object:

if operation was successful:

{ "success" : true, "invoice_no": "xxxxx"} xxxxx – is the order number in the system. Keep it to track the order status later.

if operation failed:

{ "success" : false, "error": "Description of error"}

Errors

Error	Description
The *** field is required.	Validation failed. The required field was not sent.
Invalid Client.	client_id is not in our database.
Invalid signature.	The signature is not valid
Invalid Payment Gateway.	pgw_id is not in our database.
Client does not exist.	Client payment gateway does not match the pgw_id
Items required.	No items were sent

Error	Description
Item id is required. Item price is required. Item quantity is required.	Id, price or quantity of item is empty

2.2 Query Order Status

Request Fields:

Field	Mandatory	Example	Notes
client_id	Yes	test	The ID assigned to the client (merchant) by Payo.
invoice_no	Yes	261	The order number that was previously returned by Payo Create order API
pgw_id	Yes	card	Payment Gateway API assigned by Payo.
token	Yes	1a022c4d2ff1aaadb1f8069b74dcae54931315fcae66adc0846e866514d57380	Token (see below)

Creating the token

1. Concatenate the following fields into one string (no extra concatenating characters like dots, spaces etc) using the same order:

   - In case of a Payment Gateway client: API Key is Payment gateway API key that was assigned by Payo.

     - client_id + invoice_no + pgw_id + pgw_id + API key

   - In case of a regular client:

     - API key that was assigned by Payo.

     - client_id + invoice_no + client_id + API key

Example:

client_id=test
invoice_no=261
pgw_id=card
api key = 098f6bcd4621d373cade4e832627b4f6

The string to be signed: test261cardtest098f6bcd4621d373cade4e832627b4f6

2. Hash that string using SHA256 algorithm.
   SHA256 of it is:
   831f0f4e135314af2e181e9deaad48eb82f1ee9232da410da81eb06ae4008d88

Sending the request

Query url:

http://api.payo.asia/order/get?

client_id=test&pgw_id=card&invoice_no=261&token=831f0f4e135314af2e181e9deaad48eb82f1ee9232da410da81eb06ae4008d88

Response

Response is a JSON object.

Example of successful operation:

```
{
"success": true,
"service_status":"Pending",
"courier_status":" Pending",
"tracking_number":"12345",
“Courier”:“Ninja Van”,
“pending_reason”:“Consignee Out”,
“last_status_date”:“2019-01-02T09:15:01.000Z”,
“expected_courier_fee”:“PHP 178.58”
"order_id":"#16714"

}
```

Example of failed operation:

```
{“success”: false, "error":  "Description of error"}
```

Service and Courier Statuses list

Service Status	Courier status	Description
Created	New Order Or No status	First status when Sale is created in the system.
On Hold		A customer/merchant canceled the sale. Our team tries to save it.
Canceled	Cancel Not Shipped	The sale was cancelled and not shipped.
Waiting for Pickup		The package has been allocated to a courier.
In Transit	Shipped	The package has been picked up by the courier.
Pending	Pending	The package can be pending from different reasons connected to the customer (not home, didn’t have money, etc.). Our call center working on deliver the package successfully.
Failed	RTS ODZ	The package was failed to be delivered and is now being transferred back to the merchant.
	RTS Unclaimed/Other	
	RTS Refused/No Money	
	RTS Postpone Delivery	
	RTS Didn't Order	
Delivered	Delivered	The package was delivered successfully, and cash was collected.
Early Remitted		Only for Early Remittance clients – The ship bill moved from “delivered” to “early remitted” after the early remittance billing iteration.

Service Status	Courier status	Description
Ready To Remit		The ship bill is waiting for the billing date of remittance, after the package has been delivered and cash has been collected.
Remitted		The ship bill has gone through our billing system. Invoices and bills have been issued to the merchant, and money has been paid.
Pre Closing		Actual Courier Fee has been received. Now waiting to settle the difference between actual and expected courier fee.
Closed		Two iterations have been done on the billing, and this ship bill is now closed and paid in full.
Claim		If the package has been lost, this status is opened for the courier company to get a refund.

Errors

Error	Description
The *** field is required.	Validation failed. The required field was not sent.
Invalid Client.	client_id is not in our database.
Invalid token.	The token is not valid
Invalid Payment Gateway.	pgw_id is not in our database.
Client does not exist.	Client payment gateway does not match the pgw_id
Invalid Order.	Order not found in our database
Order not created.	There was error creating order

2.3 Shipping Cost Request

Request

Field	Mandatory	Example	Notes
amount	Yes	200	Total price of package
cheapest	Yes	0	0(All), 1, 2, 3
province	Yes	Masbate	Province Name
size	Yes	small	small, medium, large
weight	No	1	Packing weight (kg)
client_id	Yes	test	The ID assigned to the client (merchant) by Payo.
Token	Yes	1f0d35329832114813be5a56a12c9a10d388a52d61e1f8ef40cd878eb64b8162	see below

Creating the token

1. Concatenate the following fields into one string (no extra concatenating characters like dots, spaces etc) using the same order:

   amount + cheapest + province + size + weight + client_id + API key

   API key:

   - In case of a Payment Gateway API client - the Payment gateway API key that was assigned by Payo.

   - In case of a regular client – the client API key that was assigned by Payo.

Example:

amount=200
cheapest=0
province=Masbate
size=small
weight=1
client_id=test
api key=098f6bcd4621d373cade4e832627b4f6

The string to be signed: 2000testMasbatesmall1test098f6bcd4621d373cade4e832627b4f6

2. Hash that string using SHA256 algorithm

   SHA256 of the string:
   1f0d35329832114813be5a56a12c9a10d388a52d61e1f8ef40cd878eb64b8162

Sending the request

Query url:

http://api.payo.asia/shipping?amount=200&cheapest=0&province=Masbate&size=small&weight=1&client_id=test&token=1f0d35329832114813be5a56a12c9a10d388a52d61e1f8ef40cd878eb64b8162

Response:

Response is a JSON object.

Example of successful operation:

```
{
  "success":true,
  "rates":{
    "0":{
      "name":"LBC",
      "costSuccess":182,
      "costFailed":363
    },
    "1":{
      "name":"2GO",
      "costSuccess":188,
      "costFailed":325
    },

   "2":{
     "name":"BlackArrow",
     "costSuccess":111,
     "costFailed":166
          }
  },
  "data":{
    "amount":"200",
    "province":"Manitoba",
    "size":"small",
    "weight":null,
    "cheapest":"0",
    "client_id":"2wtrade",
    "token":"5210fd35634d3f73b70f49d0dc3e06d5342d7cdaa17d251690650d69465b5c6e"
  }
}
```

Example of failed operation:

{“success”: false, "error": "Description of error"}

Errors

Error	Description
The *** field is required.	Validation failed. The required field was not sent.
Invalid Client.	client_id is not in our database.
Invalid token.	The token is not valid

2.4 Courier Allocation Notification

TBD

2.5 Instant "Cash Collected" Notification

Instant Cash Collected Notification (ICCN) is a message service that automatically notifies merchants (and payment gateways) for "Cash collected events" related to PAYO managed shipping's. Merchants can use it to automate back-office and administrative functions.

How it Works

Merchants create an ICCN listener page on their website and then specify the URL of the listener page to PAYO's support team to enable the notifications to the supplied URL.
PAYO then sends notifications of "Cash Collected" events to that URL.
The ICCN listener detects and processes ICCN messages using the merchant backend processes. The ICCN listener page contains a custom script or program that waits for the messages, validates them with PAYO, and then passes them to various backend applications for processing.

In order to enable the feature on PAYO's end, please contact support: support@payo.asia.

Once the feature was enabled on PAYO platform for you, you will receive a unique API Secret key which will enable you to confirm the message was signed by PAYO for you.

Integrations Details

To receive ICCN messages data from PAYO, your listener must follow this flow:

- Your listener listens for the HTTPS POST ICCN messages that PAYO sends with each event.

- After receiving the ICCN message from PAYO, your listener returns an empty HTTP 200 response to PAYO. Otherwise, PAYO resends the ICCN message 3 more times (After 1 minute, 10 minutes and 1 hour).

- ICCN messages are JSON object encoded in base64, the HTTP request will include a signature string in the request header (hmac-signature).

- First task is to calculate the signature by signing the encoded string using the secret api key and to compare it with the signature supplied in the request header.

- The signature is calculated with SHA256 HEX.

- Next task is to decode the body of the request with base64 decoder.

ICCN Sample message

Encoded JSON

```
eyJzaGlwQmlsbHMiOlt7InNoaXBCaWxsTmFtZSI6IlNCLTAwMDAwNjk0NDIiLCJwYXlvQXB
pT3JkZXJJZCI6IjQ0NDQ0Iiwib3JkZXJJZCI6Ijk4NzY1NCIsImNhc2hDb2xsZWN0ZWQiOj
EyMzMuMDB9XX0=
```

Decoded JSON

```
       {"shipBills": [
             {
                   "shipBillName":"SB-0000069442",
                   "payoApiOrderId":"44444",
                   "orderId":"987654",
                   "cashCollected":1233.00
             }
            ]
       }
```

HMAC

```
941dc6fb01ef88568c8c5c9e3e5684cc8631c10d4a22b96d28ee997075b35f87
(API Secret Key: " TestApiKey")
```

ICCN JSON Object

The JSON object is an array of ship bills.

Ship Bills fields:

shipBillName – Internal ID in PAYO platform

payoAPIOrderId – External ID in PAYO platform

orderId – Order ID given by the merchant at created event*

* On most cases this is the refence in the merchant's platform/ERP.

cashCollected – The amount collected by PAYO.