Dragonpay Online Payment

Merchant Payout API

Version 0.33 – 30 Sept 2016

Table of Contents

1. About this Document

This document describes the Application Programming Interface (API) between the Dragonpay Payout (PO) System and the Merchant’s e-commerce back-end. The PO is responsible for performing mass payout to multiple recipients. Upon completing the request, the merchant’s payout postback url will be invoked. An email summary will also be generated.

If you have any questions please do not hesitate to contact **sales@dragonpay.ph**.

2. Intended Audience

The intended audience for this document is technical personnel or programmers with background knowledge of programming and e-commerce. The examples in this document are written in Microsoft C# .NET. However, the programmer is free to implement the interfaces using other programming languages as long as they conform to Web standards such as HTTP GET, Name-Value Pair, and SOAP/XML Web Services calls.

3. Change Log

Version	Date	Changes
0.25	Oct 12, 2015	Removed enrollment module documentation
0.30	Feb 27, 2016	Added return codes for RequestPayoutEx
0.32	Aug 2, 2016	Updated Appendix 3
0.33	Sept 30, 2016	Added email and mobileNo to RequestPayoutEx

4. Introduction

Merchants that need to send payments to multiple recipients are faced with the challenge of writing hundreds, if not thousands, of checks or manually depositing to various bank accounts in different banks. This process is very tedious, takes up a lot of man-hours, and is prone to error.

Mass payout solutions simplify this process by allowing Merchants to just send a list of recipients and amounts, or programmatically request the payout, and it will handle everything behind-the-scenes. The recipient can choose their preferred method of receiving the payment whether through bank transfers, mobile payments, or physical cash pickup.

4.1 How does mass payout work?

The system generally follows this pattern:

1. Merchant sends the payout details (bank id, account no, name, amount) to Dragonpay via SOAP
2. Dragonpay handles the payout and notifies Merchant through http postback or merchant can manually check the progress status through the admin portal

The Dragonpay Payout System will perform the actual payment based on its agreed cut-off times with the Merchant. It is not carried out in real-time. Merchant can also query the payout system in real-time to check on the status of a payout request.

5. Payout API

This section of the document describes the Payout (PO) API in detail, covering the various functions used, as well as, codes that can be used to integrate them.

5.1 System Requirements

In order to integrate with the PO, Merchant must fulfill the following prerequisites:

1. Merchant site must be capable of getting the required data from customer (ex. Name, address, email).
2. Merchant site can send http request data to PO system to request a payout.
3. Optionally, Merchant site may have a Postback URL to accept real-time notifications from PO of status changes or rely on email summaries.

Each Merchant is assigned the following:

- merchant id – unique code identifying the Merchant
- password – a unique password for logging in to admin website
- api key – a unique string that is used for API calls

Although this document uses Microsoft .NET conventions, it should be implementable under other operating environments (ex. Linux, PHP, Perl, Java).

5.2 Message Passing (Merchant -> PO)

This section describes how the Merchant will pass a request to the PO for payout requests and vice versa. There is currently one integration model available –the Web Services Model.

5.2.1 SOAP/XML Web Service Model

The Merchant may choose to implement the API using the XML Web Services model. Under this model, the parameters are are exchanged directly between the Organization back-end system and PO servers through SOAP calls.

You may use the following URL’s as the Web Service entry point.

Web Service Production URL:

```
https://live.dragonpay.ph/DragonPayWebService/PayoutService.asmx
```

Web Service Test URL:

```
http://test.dragonpay.ph/DragonPayWebService/PayoutService.asmx
```

5.2.1.1 Requesting a Simple Payout

These are the parameters passed by the Merchant to the PO to request for a simple payout with no enrollment involved.

Web Method: RequestPayoutEx

Parameter	Data Type	Description
apiKey	Varchar(40)	A unique code assigned to Merchant for API
merchantTxnId	Varchar(40)	Unique txnid on the side of merchant referring to this request
userName	Varchar(30)	Account name
amount	Numeric(15,2)	Amount to pay
currency	Char(3)	Currency (currently only PHP is supported)
description	Varchar(180)	Text description as to what this is about
procId	Varchar(4)	Payout channel selected (see Appendix 4)
procDetail	Varchar(40)	Account/mobile no of payout channel
runDate	Date	Date when to execute the payout
email	Varchar(80)	Email address of payout recipient
mobileNo	Varchar(20)	Mobile number of payout recipient

The *RequestPayoutEx()* method will return an integer value of zero (0) if successfully requested. Merchant has to check the Admin portal to see updates on payout status.

Possible return values:

Code	Description
0	Successfully created payout request
-1	Invalid credentials or apiKey
-2	(reserved)
-3	(reserved)
-4	Unable to create payout transaction (internal error)
-5	Invalid account no / details
-6	Invalid pre-dated run date
-7	Amount exceeds limit for payout channel
-8	A payout has been previously requested for the same merchant txn id

5.2.1.2 Querying Transaction Status

These are the parameters passed by the Merchant to the PO to request for a simple payout with no enrollment involved.

Web Method: GetTxnStatus

Parameter	Data Type	Description
apiKey	Varchar(40)	A unique code assigned to Merchant for API
merchantTxnId	Varchar(40)	Unique txnid on the side of merchant referring to this request

These are the parameters passed by the Merchant to the PS via SOAP request for a transaction status.

The *GetTxnStatus()* method will respond with a single *status* string:

Parameter	Description
status	The result of the payment. Refer to Appendix 5 for codes.

For more details on error codes due to FAILURE, or reference numbers for SUCCESS or PENDING, please access the web-based administrator page.

Appendix 1 – Currency Codes

Code	Description
PHP	Philippine Peso
USD	US Dollar

Appendix 2 – Error Codes

Code	Description
000	Success
102	Incorrect secret key
103	Invalid reference number
104	Unauthorized access
106	Currency not supported
107	Transaction cancelled
108	Insufficient funds
109	Transaction limit exceeded
110	Error in operation
111	Invalid parameters
201	Invalid Merchant Id
202	Invalid Merchant Password

Appendix 3 – Payout Processor Codes

Code	Description
BDO	Banco de Oro CA/SA
BPI	BPI CA/SA
CBC	Chinabank CA/SA
EWB	EastWest CA/SA
LBP	Landbank CA/SA
MBTC	Metrobank CA/SA
PNB	PNB individual CA/SA
RCBC	RCBC CA/SA, RCBC Savings Bank CA/SA, RCBC MyWallet
SBC	Security Bank CA/SA
UBP	Unionbank CA/SA, EON
UCPB	UCPB CA/SA
CEBL	Cebuana Lhuillier Cash Pick-up
PSB	PSBank CA/SA (reserved)
GCSH	Gcash
SMRT	Smart Money (reserved)

Appendix 4 – Payout Transaction Status Codes

Code	Description
S	Successfully completed
F	Failed
P	Pending
G	In progress
V	Voided