BluePay 2.0 “Daily Report” V2 Interface

Abstract

This interface, “bpdailyreport2,” is intended to be polled by a merchant for the purpose of returning transaction details.

Documentation Last Updated: 2020-10-29

URL

https://secure.bluepay.com/interfaces/bpdailyreport2

Input Format

Input to this web service is formatted as a standard HTTP "POST" is formatted. The parameters to the service should be URI-encoded in the body of the request, as is standard for a POST.

Output Format

Output will be in the immediate response to the POST, in standard HTTP response format with a header and a body separated by a newline. The header contains a HTTP status line which will indicate a status of 200 for success and 400 for an error or other failure.

If successful, the body of the response will contain a number of lines, one for each matching transaction. Each line contains a comma-separated list of values you may use to tie the transaction back to data in your own system.

If failed, the output will contain only a single line, containing a message indicating the reason for failure.

Any commas existing in the data will be escaped by preceding them with a backslash character: ',' becomes '\,'.

Input Fields

ACCOUNT_ID
Required: Yes
Description:

Set to your 12-digit BluePay 2.0 Account ID.

REPORT_START_DATE, REPORT_END_DATE

Required: Date Yes, Time No Description:
These are the dates within which transactions will be reported.
Dates are formatted in ISO standard format: YYYY-MM-DD HH:MM:SS. Time defaults to 00:00:00 if not supplied.
E.g., if wishing to get the list of all transactions on 2018-08-08, one would set REPORT_START_DATE to 2019-08-08 and REPORT_END_DATE to 2019-08-09. This would query 2019-08-08 00:00:00 through 2019-08-09 00:00:00.

RESPONSEVERSION

Required: No
Default: 1 Latest:
11
Description:
If not set, response will contain version 1 fields only.

TAMPER_PROOF_SEAL

Required: Yes
Description:
Hash for security, using selected algorithm (either TPS_HASH_TYPE or account's “Hash Type in API’s” value). See TAMPER_PROOF_SEAL section below for more details.

DO_NOT_ESCAPE

Default: 0 Values:
0 or 1 Description:
Causes the output to have all commas and quotes removed from the data and to be simply comma-separated.

QUERY_BY_SETTLEMENT

Default: 0 Values: 0 or 1 Description: bpdailyreport2 queries by issue date of the transaction by default. This switches it to query by settlement date. NOTE: using this means you will only receive settled transactions in the output (i.e. no auths, voids, etc).

QUERY_BY_HIERARCHY

Default: 0 Values:
0 or 1 Description:
Causes the output to include transactions from child accounts. The account_id field will be returned in the output.

EXCLUDE_ERRORS

Default: 0 Values:
0 or 1 Description:
Causes the report to only include approvals and declines.

MODE

Default: LIVE
Values: LIVE or TEST
Description:
Allows you to choose to query LIVE transaction data or TEST transaction data.

TPS_HASH_TYPE

The algorithm used to compute the TAMPER_PROOF_SEAL. Accepted values are 'MD5', 'SHA256', 'SHA512', 'HMAC_SHA256', or 'HMAC_SHA512'. Merchant's 'Hash Type in APIs' value is used if this parameter is not present. See TAMPER_PROOF_SEAL section below for more details.

TPS_DEF

** NOTICE: THE USE OF THIS FIELD CAN POSSIBLY WEAKEN YOUR SECURITY. PLEASE BE SURE YOU UNDERSTAND HOW THE TAMPER_PROOF_SEAL WORKS BEFORE YOU CONSIDER USING THIS OPTION. **Space-separated list of input fieldnames in the order they are to be used in the calculation of the TAMPER_PROOF_SEAL. If set as blank or not sent, it will default to:"ACCOUNT_ID REPORT_START_DATE REPORT_END_DATE". The merchant's Secret Key is always used in the calculation of the TAMPER_PROOF_SEAL, but should NOT be included in the TPS_DEF. See TAMPER_PROOF_SEAL section below for more details.

Aggregate fields

It's possible to request an aggregate report instead of a detailed report. When you do this, the columns returned in the output will be the count of the transactions, the sum of their amounts, and whatever columns you request as below, instead of the standard output.

AGG_QUERY

Default: 0 Values:
0 or 1 Description:
Turns on the aggregate reporting feature

AGG_FIELDS

Default: n/a
Values: space-separated list of fieldnames Description:
Fieldnames are mostly the same as in output, below, with the exception that you cannot aggregate by "id" or "amount." Example (after uri-encoding):
"AGG_FIELDS=payment_type%20trans_type%20card_type"

Further Filtering

If you aim to retrieve a subset of transactions, these additional filters can be used to filter the results to fit your purpose.

Some filters support searching for emptiness by submitting a key without a value. An example of this would be: "KEY="

COMPANY_NAME

Max length: 64 Usage:
To search for transactions with a company called "abcxyz":
"COMPANY_NAME=abcxyz"

FIRST_NAME

Max length: 32 Usage:
To search for transactions with a first name of "abc":

"FIRST_NAME=abc"

LAST_NAME

Max length: 32 Usage:
To search for transactions with a last name of "abc":
"LAST_NAME=abc"

EMAIL

Max length: 128 Usage:
To search for transactions with an email of email@email.email:
"EMAIL=email%40email.email" (uri-encoded)

ORIGIN

Valid values:
bp10emu bp20post asbyemu anet-aim VTERM REJECT FIXER PAYOUT AGG BATCH CAPQUEUE FRAUDSCRUB REBILL IVR Usage:
To search for transactions from bp10emu:
"ORIGIN=bp10emu"

TRANSACTION_ID

Usage:
To search for a transaction by ID (100000123456):
"TRANSACTION_ID=100000123456"

MASTER_ID

Usage:
To search for a transaction by Master ID (100000123456):
"MASTER_ID=100000123456"

REBILLING_ID

Usage:
To search for a transaction by Rebilling ID (100000123456):
"REBILLING_ID=100000123456"

SETTLEMENT_ID

Usage:
To search for a transaction by Settlement ID (100000123456):
“SETTLEMENT_ID=100000123456”

PROCESSOR_ID

Usage:
To search for a transaction by Processor ID (100000123456):
“PROCESSOR_ID=100000123456”

STATUS

Valid values:
1 = APPROVED 0 = DECLINED E = ERROR Usage:
To search for approved transactions:
“STATUS=1”

TRANS_TYPE

Valid values:
AUTH SALE CAPTURE REFUND CREDIT VOID UPDATE Usage:
To search for sale transactions:
“TRANS_TYPE=SALE”

PAYMENT_TYPE

Valid values:
CREDIT ACH SEPA Usage:
To search for credit transactions:
“PAYMENT_TYPE=CREDIT”

CARD_TYPE

Valid values:

VISA MC DISC DCCB AMEX JCB ENRT BNKC SWTC SOLO Usage:
To search for VISA transactions:
“CARD_TYPE=VISA”

CARD_PRESENT

Valid values:
1 0
Usage:
To search for transactions with card present:
“CARD_PRESENT=1”
To search for transactions without a card present:
“CARD_PRESENT=0”

CUSTOM_ID

Max length: 16 Usage:
To search for transactions with a custom_id of “abc”:
“CUSTOM_ID=abc”

CUSTOM_ID2

Max length: 64 Usage:
To search for transactions with a custom_id2 of “abc”:
“CUSTOM_ID2=abc”

ORDER_ID

Max length: 128 Usage:
To search for transactions with an order_id of “abc”:
“ORDER_ID=abc”

INVOICE_ID

Max length: 64 Usage:
To search for transactions with a invoice_id of “abc”:
“INVOICE_ID=abc”

BACKEND_ID

Max length: 64

Usage:

To search for transactions with a backend_id of “100000123456”:

“BACKEND_ID=100000123456”

AUTH_CODE

Max length: 8 Usage:

To search for transactions with an auth_code of “abc123”:

“AUTH_CODE=abc123”

AMOUNT

Usage:

To search for transactions with an amount of “$1.00”:

“AMOUNT=1.00”

Output Fields

Output is versioned.

If version is omitted from the request input, the interface will respond with version 1.

If the requested version is greater than the default, the output will include each of the versions below and up to it.
For example, requesting RESPONSEVERSION 3 will output the fields from versions [1,2,3]

inclusive.

RESPONSEVERSION 1 (default)

Version 1 is the default version.

id

Length: 12
Description:
The 12-digit transaction ID assigned to this transaction by BluePay.

payment_type

Maximum length: 8
Description:

- ‘ACH’ for ACH transactions,
- ‘CREDIT’ for credit card transactions

amount

Maximum length: 9
Description:
The monetary amount for which the transaction was run.

card_type

Maximum length: 4
Description:

A four-character indicator of the credit card type used, if any.

AMEX	American Express
MC	MasterCard
DISC	Discover
VISA	VISA
JCB	JCB
DCCB	Diner’s Club or Carte Blanche
ENRT	EnRoute
BNKC	BankCard
SWTC	Switch
SOLO	Solo

payment_account

Maximum length: 32

Description:

The payment account used for the transaction.

If a credit card is used, 12 x’s followed by the last four digits will be returned.

If a check is used, the following string is returned.

“<account type>:<routing number>:<x’s><last four digits of account number>” E.g.

“C:123123123:xxxxxx4321” Account Type: ‘C’ is checking and ‘S’ is savings.

order_id

Maximum length: 128

Description:

The merchant-supplied or system supplied order ID.

invoice_id

Maximum length: 64

Description:

The merchant-supplied or system supplied invoice ID.

custom_id

Maximum length: 16
Description:
The merchant-supplied value for Custom ID .

custom_id2

Maximum length: 64
Description:
The merchant-supplied value for Custom ID 2.

master_id

Maximum length: 12
Description:
The trans_id if the current transaction was generated from a previous transaction.

status

Length: 1 Description:
● ‘1’ for approved, ● ‘0’ for declined, ● ‘E’ for error.

f_void

Length: 1
Description:
A flag to identify if the transaction was voided.

message

Maximum length: 64
Description:
Some human parsable text describing the reason of the transaction. For settlements this normally just reads “Approved.”

origin

Maximum length: 16
Description:
Where the transaction originated.

bp10emu	Post & Redirect API
bp20post	Post API
asbyemu	AssureBuy Emulation/XML
a.net-aim	Authorize.net Emulator
VTerm	Virtual Terminal
AGG	Aggregation System
BATCH	File Upload
FRAUDSCRUB	Fraud Management System
REBILL	Recurring Billing System
REJECT	Automatic reject or chargeback from bank

issue_date

Length: 19 Description:
The date that the transaction was entered into BluePay (i.e. "YYYY-MM-DD HH:MM:SS").

settle_date

Length: 19 Description:
The date and time of settlement for the transaction (i.e. "YYYY-MM-DD HH:MM:SS").

rebilling_id

Length: 12
Description:
ID of the recurring billing schedule if this transaction was initiated by a rebill.

settlement_id

Length: 12
Description:
The settlement ID for the transaction.

card_expire

Length: 4 Description:
Credit card expiration date in MMYY format

bank_name

Length: 64
Description:
Bank name associated with the payment method used.

addr1

Length: 64

addr2

Length: 64

city

Length: 32

state

Length: 16

zip

Length: 16

memo

Length: 4096

phone

Length: 16

email

Length: 64

auth_code

Length: 8 Description:
Contains the credit card authorization code in the case of a successful TX. This field will display the reject code on voids from ACH reject.

name1

Length: 32

name2

Length: 32

company_name

Length: 64

backend_id

Length: 2048

f_captured

Length: 1
Description:
A flag to identify if a transaction was captured. Only applies to authorizations.

account_id

Length: 12
Description:
This field will only appear when QUERY_BY_HIERARCHY=1. If the field appears, it will always appear as the last column regardless of RESPONSEVERSION.

RESPONSEVERSION 2

RESPONSEVERSION 2 includes RESPONSEVERSION 1 response fields, plus the following.

avs_result

Length: 1
Description:
Address Verification System (AVS) response code received on the transaction.

cvv_result

Length: 1 Description:
Card Verification Value 2 response code. Result of the validation of the CVV2 value entered by the payer.

merchdata

Description:
All the MERCHDATA values combined into a single field.

card_present

Length: 1
1 for a swiped transaction. 0 for not present or a non-swiped transaction.

RESPONSEVERSION 3

RESPONSEVERSION 3 includes all lower response version values plus the following.

level_3_data

Description:
All the LV3_ITEMx_* vfalues combined into a single field.

RESPONSEVERSION 4

RESPONSEVERSION 4 includes all lower response version values plus the following.

remote_ip

Description:
Either the REMOTE_IP value received in the transaction request or the customer's IP address when the post came from the customer's web browser.

connected_ip

Description:
The IP address of the computer that accessed the BluePay system.

RESPONSEVERSION 5

RESPONSEVERSION 4 includes all lower response version values plus the following.

level_2_data

Description:
All the LV2_ITEMx_* values combined into a single field.

RESPONSEVERSION 6

RESPONSEVERSION 6 includes all lower response version values plus the following.

acct_update_id

Length: 12 Description:
The ID of any updated account data for the payment account used for the transaction. Updated account data only provided for merchants using Account Updater.

RESPONSEVERSION 7

RESPONSEVERSION 7 includes all lower response version values plus the following.

vehicle_rental_data

Description:
All vehicle rental fields combined into a single field.

lodging_data

Description:
All lodging fields combined into a single field.

RESPONSEVERSION 8

RESPONSEVERSION 8 includes all lower response version values plus the following.

f_refunded

Length: 1
Description:
A flag to identify if a transaction was refunded.

processor_id

Length: 12
Description:
ID of the processor that processed the transaction.

RESPONSEVERSION 9

RESPONSEVERSION 9 includes all lower response version values plus the following.

fancy_name

Description:
First and last name combined into a single value.

country

Description:
Country value supplied in the transaction request.

owner_id

Length: 12 Description:
ID number of the user associated with the transaction. Transactions processed using an API will be associated with the default user on the account.

mode

Length: 4 Description: LIVE or
TEST f_rebill_master

Length: 1 Description:
1 = True, 0 = False
Whether a rebill schedule is using this transaction as the master of transactions created by the rebill schedule.

f_will_capture

Length: 1 Description:
1 = True, 0 = False
Whether a transaction is flagged for auto-capture.

f_corporate

Length: 1 Description:
1 = True, 0 = False
Whether IS_CORPORATE was set to 1 in the transaction request.

cvv2_status

Length: 1 Description:
1 if a CVV2 value was supplied on the transaction. 0 if a CVV2 value was not supplied.

account_name

Description:
Merchant gateway account name. update_id

Length: 12 Description:

If a transaction was updated by the Card Account Udpater service this will be the ID of the update used.

f_unheld

Length: 1
Description:
0 if the transaction was never on hold.
1 if the transaction was on hold at some point, but later unheld or declined.

unhold_id

Length: 12
Description:
Transaction ID of the original transaction unheld by this transaction.

RESPONSEVERSION 10

RESPONSEVERSION 10 includes all lower response version values plus the following.

amount_tip

Length: 9
Description:

The monetary amount of the tip for this transaction.

RESPONSEVERSION 11

RESPONSEVERSION 11 includes all lower response version values plus the following.

f_transarmor

Length: 1
Description:
0 or null if no TransArmor token was used.
1 if TransArmor token was used.

Field Quick Reference

Below is a table with the fieldname, minimum response version, and expected length.

fieldname	RESPONSE VERSION	Max Length
id	1	12
payment_type	1	8
trans_type	1	8
amount	1	9
card_type	1	4
payment_account	1	32
order_id	1	128
invoice_id	1	64
custom_id	1	16
custom_id2	1	64
master_id	1	12
status	1	1
f_void	1	1
message	1	64
origin	1	16
issue_date	1	19
settle_date	1	19

rebilling_id	1	12
settlement_id	1	12
card_expire	1	4
bank_name	1	64
addr1	1	64
addr2	1	64
city	1	32
state	1	16
zip	1	16
memo	1	4096
phone	1	16
email	1	64
auth_code	1	8
name1	1	32
name2	1	32
company_name	1	64
backend_id	1	2048
f_captured	1	1
avs_result	2	1
cvv_result	2	1

merchdata	2	

card_present	2	1
level_3_data	3	
remote_ip	4	
connected_ip	4	
level_2_data	5	
acct_update_id	6	12
vehicle_rental_data	7	
lodging_data	7	
f_refunded	8	1
processor_id	8	12
fancy_name	9	
country	9	
owner_id	9	12
mode	9	4
f_rebill_master	9	1
f_will_capture	9	1
f_corporate	9	1
cvv2_status	9	1
account_name	9	

update_id	9	12
f_unheld	9	1
unhold_id	9	12
amount_tip	10	
f_transarmor	11	1
account_id *	1	12

* “account_id” only appears if QUERY_BY_HIERARCHY=1 is present in the request. It will always appear as the last column of the request.

################################
TAMPER_PROOF_SEAL
################################

BluePay uses cryptographic hash (or "digest") functions as a means of both protecting transaction data from being altered and ensuring that the transaction is genuine. A cryptographic hash function is an algorithm that maps data of any size to a bit string of a fixed size that cannot be deciphered.

All merchants have a default hash type assigned to their account. This can be viewed and updated on the merchant's Account Admin page of BluePay's Gateway (https://secure.bluepay.com) under "Hash Type in APIs". Merchants may override their default by including the
TPS_HASH_TYPE
field in the transaction request.

The default hash type and the TPS_HASH_TYPE may be any of the following algorithms (in hexadecimal form):

MD5 Use md5sum or a similar program to calculate a 128-bit hash, then convert it into hexadecimal form; result is 32 hexadecimal characters.

SHA256 Use sha256sum or a similar program to calculate a 256-bit hash, then convert it into hexadecimal form; result is 64 hexadecimal characters.
SHA512 Use sha512sum or a similar program to calculate a 512-bit hash, then convert it into hexadecimal form; result is 128 hexadecimal characters.

HMAC_SHA256 A 128-bit hash, resulting in a sequence of 64 hexadecimal characters.
HMAC_SHA512 A 128-bit hash, resulting in a sequence of 128 hexadecimal characters.

Steps to find the HMAC of either SHA256 (HMAC_SHA256) or SHA512 (HMAC_SHA512):

1. Compare the length of the key (the merchant's Secret Key) to the hash's input blocksize.
SHA256 blocksize = 64, SHA512 blocksize = 128.
If length of key is > blocksize, set the key's value to the hash of the original key.
If length of key is < blocksize, pad the key to the right with zeros until its length equals the blocksize.

2. Create the inner key (inner_key):
Create an inner padding value of 0x36 repeated the blocksize number of times.
Perform a bitwise exclusive-OR (XOR) on the key and the inner padding to create the inner key.

3. Create the outer key (outer_key):
Create an outer padding value of 0x5c repeated the blocksize number of times.
Perform a bitwise exclusive-OR (XOR) on the key and the outer padding to create the outer key.

4. Calculate the hash of the inner key concatenated with the text string, then calculate the hash of the outer key concatenated with the previous hash result:
hash(outer_key + hash(inner_key + string))

5. Convert the result into a hex string.

When using a program or function to calculate the TAMPER_PROOF_SEAL, make sure that it will
accept a text string (or "message") argument and will return the hashed string (or "message digest") in hexadecimal form.

#####################################
Calculating the TAMPER_PROOF_SEAL

####################################

STEP ONE
Concatenate the values of the fields that make up the TPS_DEF in same order that they are listed. Use ""(empty string - no space) as the value for any fields that are empty or unsent. When no TPS_DEF is sent ('+' represents string concatenation, and the field names represent the contents of the respective fields):

message = ACCOUNT_ID + REPORT_START_DATE + REPORT_END_DATE

STEP TWO
- If TPS_HASH_TYPE is "" or is not sent, the merchant's 'Hash Type in APIs' value will determine which hash function to use.

- If TPS_HASH_TYPE is 'MD5', 'SHA256', or 'SHA512', find the md5sum, sha256sum, or sha512sum
of (the merchant's Secret Key + message) in hex format.

- If TPS_HASH_TYPE is 'HMAC_SHA256' or 'HMAC_SHA512', find the HMAC_SHA256 or HMAC_SHA512 of
(the merchant's Secret Key, message) in hex format.

EXAMPLE:
Merchant A's account information is as follows:
Secret Key = "abcdabcdabcdabcd"
ACCOUNT_ID = "123412341234"
Hash Type in APIs (default hash type) = "MD5"

If Merchant A wanted a report of all transactions issued on 2018-02-28, excluding errors, the request might include:
TPS_DEF = "ACCOUNT_ID MODE REPORT_START_DATE REPORT_END_DATE EXCLUDE_ERRORS"
ACCOUNT_ID = "123412341234"
EXCLUDE_ERRORS = "1"
REPORT_START_DATE = "2018-02-28"
REPORT_END_DATE = "2018-03-01"

To calculate the TAMPER_PROOF_SEAL, Merchant A would need to:

STEP ONE
Concatenate the values in the TPS_DEF to create a message string. Remember, if the field isn't sent or if the value is undefined, use an empty string as that field's value.
message = ACCOUNT_ID + MODE + REPORT_START_DATE + REPORT_END_DATE + EXCLUDE_ERRORS
= "123412341234" + "" + "2018-02-28" + "2018-03-01" + "1"
= "1234123412342018-02-282018-03-011"

STEP TWO
This step will vary depending on which TPS_HASH_TYPE is sent (if any).

-- If TPS_HASH_TYPE = "" or was not sent, the merchant's default hash type must be used
TAMPER_PROOF_SEAL = md5sum(Secret Key + message) in hex format
= md5sum("abcdabcdabcdabcd" + "1234123412342018-02-282018-03-011") in hex format
= "0d76fcbca9501abe9cc3985e03a62d7d"

-- If TPS_HASH_TYPE = "SHA256"
TAMPER_PROOF_SEAL = sha256sum(Secret Key + message) in hex format
= sha256sum("abcdabcdabcdabcd" + "1234123412342018-02-282018-03-011") in hex
format
= "c8981328bba52de2e931fe5b67bbfb76d92c16262ff04896ae62bf9d1d30e076"

-- If TPS_HASH_TYPE = "HMAC_SHA256"
TAMPER_PROOF_SEAL = HMAC_SHA256(Secret Key, message) in hex format
= HMAC_SHA256("abcdabcdabcdabcd", "1234123412342018-02-282018-03-011") in hex
format
= "08859f8e5fe469bb9dd93cb4da6334f7473d9f76a98d7056bd2e8e62c656ad5d"