BluePay Single Transaction Query (stq)

Single Transaction Query

Reference Guide

April 2025

http://www.fiserv.com

This document is classified as Fiserv Public.

Content

About this Document

This documentation provides technical assistance for accessing various transaction lists on the BluePay Manager website using different processing methods.

Intended Audience

This document is written for merchants, partners, and developers who will be responsible for integrating payment processing functionality with the BluePay Payment Gateway. This document provides an understanding of the integration options available using the BluePay Payment Gateway API.

Assistance & Feedback

Use the following contact information for help with the BluePay Payment Gateway integration or to provide feedback on this document.

Support Level	Contact Details
BluePay Integration Support Team	bluepay-integration@fiserv.com

Support hours are Monday through Friday 8:00am to 5:00pm (CST UTC-6).

Overview

The BluePay Gateway manager website allows users to export transaction data in a CSV (Comma-Separated Value) format. This can be useful for further analysis, integration with external systems, or various other purposes.

URL

Click the below link to access the sample Single Transaction Query response.

Sample: https://secure.bluepay.com/interfaces/stq

Input Parameters

The following input parameters are available:

ACCOUNT_ID	
Required:	Yes
Description:	Unique 12-digit BluePay ACCOUNT_ID associated with the merchant transaction.

TAMPER_PROOF_SEAL	
Required:	Yes
Description:	Hash for security, using selected algorithm (either TPS_HASH_TYPE or account's 'Hash Type in APIs' value). See TAMPER_PROOF_SEAL section below for more details.
Example:	Null, 1, 2, 3, 4, 5, 6, 7, 8, 9 or 10

REPORT_START_DATE	
Required:	Yes
Description:	The date/time associated with the transaction, it is required unless id or master_id is included as an input parameters. The valid formats are: YYYY-MM-DD HH:MM:SS
Example:	If the submitted time is not available then default value is 00:00:00

EXCLUDE_ERRORS	
Required:	Optional
Description:	An error code to avoid the error message in the search. Set the value to 1 to ignore error transactions.

MODE	
Required:	Optional
Description:	A mode to define the current state of the transaction.
Possible Value:	LIVE or TEST

f_captured / isnull_f_captured	
Required:	Optional
Description:	A value to set whether the transaction can be captured or not. Set f_captured to 1 for captured transaction. Set isnull_f_captured to 1 for uncaptured transaction.

LIMIT_ONE	
Required:	Optional
Description:	A limit value that is set to return only single transaction result. Set to 1 to return results of a single transaction if the search criteria match more than one transaction.

TPS_HASH_TYPE	
Required:	Optional
Description:	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	
Required:	Optional
Description:	Space-separated list of input field names 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.

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.

Additional Optional Input Parameters / Standard Output Parameters

The following substitution variables are in the hosted payment form:

Parameter	Description
acct_update_id	If using the Account Updater service and updated credit card information is received, this will be the transaction ID of the transaction record storing the new information.
addr1	Customer address1 details that are associated with bank account used for the current transaction.
addr2	Customer address2 details that are associated with bank account used for the current transaction.
amount	Amount value that is used in the current transaction.
auth_code	Authorization Code returned by card issuing bank.
avs_result	AVS result code
backend_id	Transaction ID of outbound ACH payment.
bank_name	Bank name that is associated with the account number used in the current transaction.
bindata	Tilde (~) separated list of transaction information returned by credit card processing network.
card_country	Country of credit card issuer
card_expire	Credit card expiration date in MMYY format
card_present	0 or 1, Whether the credit card was swiped
card_type	Card type associated with the current transaction, such as AMEX, MC, DISC, VISA, JCB, DCCB, ENRT, BNKC
city	City name associated with the current transaction.
company_name	The name of the company.
country	Country name associated with the current transaction.
custom_id	Custom id1 that is associated with bank account used for the current transaction.
custom_id2	Custom id2 that is associated with bank account used for the current transaction.
cust_token	Customer token associated with the current transaction.
cvv_result	the CVV result code that is associated with the current transaction.

Parameter	Description
email	Email address of the customer that is associated with bank account used for the current transaction.
f_captured	0 or 1, Whether the transaction has been captured.
f_transarmor	0 or 1, Whether the transaction used/created a TransArmor Token.
f_void	0 or 1, Whether the transaction has been voided.
id	The identification number associated with the current transaction.
invoice_id	Invoice number associated with the current transaction.
issue_date	Date and time of the previous transaction in YYYY-MM-DD HH:MM:SS format.
lodging_data	All the LODGING* values combined into a single field.
memo	Memo associated with the current transaction.
master_id	Transaction ID of a transaction that was used as the master of the current transaction.
message	comments.
name1	Name1 associated with the current transaction.
name2	Name2 associated with the current transaction.
order_id	Alphanumeric merchant-assigned order ID that is associated with the merchant account processing the current transaction orders.
origin	Gateway interface that the transaction was received on.
payment_account	The format used is xxxxxxxxxxxx1234 to input the last 4 digits for searching.
payout_date	Date and time when the transaction was funded, provided in the YYYY-MM-DD HH:MM:SS format.
payment_type	CREDIT, ACH
phone	Phone number associated with the current transaction.
processor_id	the id of the processor on the account that is used to process the transaction.
rebilling_id	The id attached to the rebill schedule.
remote_ip	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.
settle_date	The date and time of settlement for the transaction.
settlement_id	ID for grouping of transactions when sent for settlement.

Parameter	Description
state	State of the customer that is associated with bank account used for the current transaction.
status	1=Approved, 0=Declined, E=Error, #=In Process
trans_type	AUTH, SALE, CAPTURE, REFUND, REBCANCEL
vehicle_rental_data	All the VEHICLE_* values combined into a single field.
zip	Zip code of the customer that is associated with bank account used for the current transaction.
ach_check_num	Provides the check number associated with the transaction.
ach_noc_id	• ACH Notice of Change (NOC) ID. A 12-digit number identifying which NOC was applied to the transaction. Null if no NOC was applied. • Banks provide a NOC if information used on a transaction should be changed on subsequent transactions. • BluePay saves NOC information and automatically applies it new transactions when the previously corrected information is used.
ach_trace_number	• The unique 15-digit trace control number assigned to a transaction whenever it is added to the NACHA settlement file. It is required by the originators to identify the individual entries. • It identifies the transaction during the transmission from BluePay to the orginating bank. The originating bank assigns a new trace number when they transmit the transaction to the Federal Reserve. • The receiving bank will have the Federal Reserve trace number, not this number. BluePay Support can request the Federal Reserve trace number from the originating bank if needed. • The first eight digits represent the Processor Routing Number, and the remaining seven digits are sequentially numbered from "1" to "9999999" across multiple files on multiple days. The sequential number starts again at "1" once it reaches its maximum limit of "9999999."
validation_result	Status value that includes bank account validation results. Possible status values are: • '15': Known bad bank account. The transaction is immediately declined (validation fee charged for LIVE transactions) • '20': Unknown bank account but with a valid format (validation fee charged for LIVE transactions) • '25': Unknown bank account (validation fee charged for LIVE transactions) • '35': Bank account found but pending transaction settlement (validation fee charged for LIVE transactions) • '45': Known good bank account (validation fee charged for LIVE transactions) • 'B': Account validation bypassed (no validation fee charged) • 'R': Bank Account received known bad previously (no validation fee charged) • 'E': Error, Bank account validation failed (no validation fee charged) • null: Account validation not performed (no validation fee charged)

Appendix

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):

Hexadecimal Value	Description
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 SHPF_TPS, 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 are as follows

Secret Key	= "abcdabcdabcdabcd"
ACCOUNT_ID	= "123412341234"
Hash Type in APIs (default hash type)	= "MD5"

If Merchant A wanted to find a TEST transaction issued on 2018-02-28, the request might include:

TPS_DEF	= "ACCOUNT_ID EXCLUDE_ERRORS REPORT_START_DATE REPORT_END_DATE LIMIT_ONE"
ACCOUNT_ID	= "123412341234"
LIMIT_ONE	= "1"
REPORT_START_DATE	= "2018-02-28"
REPORT_END_DATE	= "2018-03-01"
MODE	"TEST"

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 + EXCLUDE_ERRORS + REPORT_START_DATE + REPORT_END_DATE + LIMIT_ONE = "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"

Revision History

Version	Revision Date	Reason for Change
1.1	April 2025	• Updated the layout format of the document