SecureFrame Integration Guide

Document Control

This is a control document

DESCRIPTION	SecureFrame Integration Guide		
CREATION DATE	02/10/2013	**CREATED BY**	SecurePay
VERSION	1.10	**DATE UPDATED**	21/10/2022
CHANGES	Section 1.3: - Added fiserv FDMSA as supporting institutions		

Table of Contents

1 Introduction

1.1 About this Guide

This guide provides technical information about integrating and configuring SecurePay's SecureFrame within your shopping cart or application.

SecureFrame offers a secure and flexible hosted service that helps you meet your PCI DSS obligations.

1.2 Intended Audience

This document is intended for developers, integrating SecurePay's SecureFrame with their applications or shopping cart.

It is recommended that someone with web site, HTML and application programming experience reads this guide and implements SecureFrame.

1.3 System Overview

SecurePay's SecureFrame provides merchants with the ability to process card payments in a secure environment.

SecurePay partners with the following major banks and financial institutions in the provision of the SecurePay Payment Gateway:

- ANZ
- American Express
- BankWest
- Commonwealth Bank
- Diners Club
- National Australia Bank
- St George (including Bank of SA)
- Westpac
- fiserv FDMSA

SecureFrame supports card transactions only.

It is a fully hosted service and comes with a variety of options to help you integrate such as:

- Full page templates
- Call-back of results to your application
- Redirect options for receipt page
- FraudGuard
- Surcharging
- Card storage and tokenisation

1.4 Feedback

Continuous improvement is one of SecurePay's core values. We welcome any feedback you have on our integration guides to help us improve any future changes to our products.

If you wish to leave feedback, please email us at support@securepay.com.au.

2 Additional Payment Choices

SecureFrame also provides the following payment options. Please contact the Payment Choice provider for details on how to sign-up.

- PayPal

2.1 Payment Choice - PayPal

If you wish to enable your SecurePay account to accept PayPal transaction via SecureFrame please follow the steps below:

Step 1: Inform the SecurePay Support team of your intention to use PayPal.

Step 2: If you don't have a Business PayPal account, establish an account with PayPal.

Step 3: Login to the SecurePay Merchant Login.

Step 4: Navigate to the following location:

- Click on [Manage] dropdown list and click on [PayPal Settings].
- Click on [Change Settings] button.

Step 5: Click on the [Retrieve API Credentials] link on the page.

Note: A popup window will appear. Please ensure you have popups enabled in your web browser.

Step 6: Login to PayPal using your credentials.

Step 7: Copy and paste the credentials into [Step 8] and close the popup window.

Step 8: Add the following PayPal credentials obtained from [Step 7] to the SecurePay PayPal settings page:

- API Username
- API Password
- API Signature

Step 9: Add the company logo URL. The URL must be publicly accessible and securely hosted (HTTPS).

Step 10: Save changes.

Note: Once PayPal has been enabled and configured successfully, you can view PayPal transactions processed via SecureFrame through the SecurePay Merchant Login.

2.1.1 *Technical Overview - PayPal*

PayPal uses a secure page, hosted by PayPal and presented to your customer as part of the payment authorisation. To enable PayPal transactions through SecureFrame (from your end), the website must pass through the tender type of "PAYPAL" within the "card_types" SecureFrame input field.

Step 1: Generate a Fingerprint

A Fingerprint is generated in your website code by a HMAC SHA-256 hash comprised of your SecurePay Merchant ID, transaction password, transaction type, transaction reference, the payment amount, and a timestamp. This value is then presented on your payment form as a hidden field. Use your transaction password as the secret key. SHA-1 support is being decommissioned over the coming months and once support is fully dropped, only HMAC SHA-256 will be used. In the transition period, to use HMAC SHA-256, provide "isSHA256=" in the callback URL. It can be appended at the end. Do

not provide a value, just leave blank as is. Once support for SHA-1 is stopped, you will not need to provide the parameter in the callback URL but can continue to send as it will not be used by SecureFrame. See examples for callback URL in section 4.1.5.5.

For HMAC SHA-256 fingerprint

<input type="hidden" name="callback_url"
value="http://www.myserver.com.au/result.asp?isSHA256=">

Step 2: Website submits transaction to SecureFrame

Your website submits the transaction details to SecureFrame, which will interpret the optional "card_types" input to determine if "PAYPAL" is included.

If included, SecureFrame will render the alternative payment page, which allows the customer to select either to pay with any of the accepted card types, or to select PayPal.

If the customer selects PayPal and the "Continue" button, they will be presented with the PayPal login/registration page.

Step 3: Customer's PayPal account

Your customer logs into or registers their PayPal account, confirms shipping and billing details (as required), and selects the payment tender. Your customer will then submit the payment for processing.

Step 4: Redirect to SecureFrame result page

Upon completion of the transaction, the customer is redirected to the SecureFrame result page.

Step 5: Customer returned to the online store

The customer selects to "Continue" from the SecureFrame result page and using the "return_url" or "return_url_target" they will be redirected to the URL specified. This also passes the result parameters back to your system. Your system should check the Fingerprint, update your database and display the receipt to the Customer.

Note: to ensure that the result is passed back to your system successfully, it is recommended that you use the "callback_url" field. This passes the results back to your system as soon as the transaction is complete, rather than waiting for the customer to select the "Continue" button. This prevents missing transactions caused by customers closing the web browser without selecting to "Continue".

SHA-1 support is being decommissioned over the coming months and once support is fully dropped, only HMAC SHA-256 will be used. In the transition period, to use HMAC SHA-256, provide "isSHA256=" in the callback URL. It can be appended at the end. Do not provide a value, just leave blank as is. Once support for SHA-1 is stopped, you will not need to provide the parameter in the callback URL but can continue to send as it will not be used by SecureFrame. See examples for callback URL in section 4.1.5.5.

Important Information: Processing a PayPal transaction

- Only "txn_type=0" is supported to process a PayPal transaction. Any other transaction type, used in conjunction with "card_types=PAYPAL" will return the error: "Unsupported Transaction Type for PayPal"
- SecurePay FraudGuard cannot be used in conjunction with PayPal payments.

3 Integration

3.1 General Information

3.1.1 *TLS Support*

Clients integrating with SecurePay must be configured to use TLS v1.2. TLS versions 1.1 and below, as well as all SSL versions, are not supported.

In addition, insecure ciphers such as Triple-DES (3DES) and RC4 are not accepted. Merchants should configure their webservers to follow a similar TLS profile, only permitting secure cipher suites and TLS v1.2 and above.

More information and additional detail on a secure TLS configuration can be found at the following publications:

- Australian Government's Australian Signals Directorate (ASD) TLS and HTTPS configuration guidelines.

- New Zealand Government Communications Security Bureau (NZ GCSB) Information Security Manual.

- NIST Guidelines for the Selection, Configuration, and Use of Transport Layer Security.

3.1.2 *Case Sensitivity*

All field “name” and “value” attributes should be treated as case sensitive.

3.1.3 *Sending Data*

SecureFrame accepts POST or GET data from your application to initiate a transaction, however POST is the preferred action type.

When using an HTML form, the following “form” tags are used to encapsulate SecureFrame inputs:

```
<form method="post" action="https://…">
…
</form>
```

All INPUT fields must occur between the “form” tags for correct submission of information to the SecureFrame Live and Test servers.

You may also add the “name” attribute or any other form functionality that you require.

3.1.4 *Acceptable Input Fields*

Any HTML form tags may be used to submit information to SecureFrame.

This document deals predominantly with the “input” tag however, you may use any form tag to create the necessary name/value data pairs that form the information interpreted by the service.

Most data is normally passed as “hidden” type input fields. Some fields that may be entered by your customer is typically passed as “text” type input fields.

Form inputs follow the structure:

<input type=”*field_type*” name=”*field_name*” value=”*field_value*”>

3.2 Transaction URLs

Listed below are the live and test URLs for performing several functions.

3.2.1 *Test URL*

```
https://test.payment.securepay.com.au/secureframe/invoice
```

3.2.2 *Live URL*

```
https://payment.securepay.com.au/secureframe/invoice
```

3.3 Mandatory Input Fields

3.3.1 *Bill Name*

The Bill Name defines the flow of the transaction process.

"bill_name" must be set to "transact"

```
Example: Set the bill name for the default transaction process:
<input type="hidden" name="bill_name" value="transact">
```

3.3.2 *Merchant ID*

The Merchant ID field, "merchant_id", is mandatory. It is the SecurePay account used to process payments.

SecurePay Customer Support will supply your Merchant ID when your account is activated. The Merchant ID will be of the format "ABC0001", where ABC00 is your unique account code.

```
Example: Setting the SecurePay Merchant ID:
<input type="hidden" name="merchant_id" value="ABC0001">
```

3.3.3 *Transaction Type*

The "txn_type" defines the payment process. It allows switch of the payment type, as well as addition of optional services such as FraudGuard. It also forms part of the Fingerprint.

3.3.3.1 *Payment*

Payments are real-time, immediately authorised card transactions. Transaction information is submitted by your customer via SecureFrame to your SecurePay account for immediate processing.

```
Example: Form fields required to make a card payment


<input type="hidden" name="merchant_id" value="ABC0001">
<input type="hidden" name="primary_ref" value="Test Reference">
<input type="hidden" name="txn_type" value="0">
<input type="hidden" name="amount" value="100">
<input type="hidden" name="fp_timestamp" value="20220228022758">
<input type="hidden" name="fingerprint"
value="33de8f9454a62513838ce534309c76ff8ac2c925bfda0364663d836254497899">
```

3.3.3.2 *Pre-Authorisation*

A pre-authorisation is a transaction that reserves funds on a card account. The Merchant can then complete the transaction later and receive the funds. If the pre-authorisation is never completed, it expires, usually after five days. After this, the reserved funds are again available to the card holder.

Pre-authorisations are often used by hotels to reserve funds at booking time and then completed when the guest checks out.

To pre-authorise an amount, submit all the fields exactly as they were for the PAYMENT transaction type above and set:

```
<input type="hidden" name="txn_type" value="1">
```

3.3.4 *Transaction Reference*

The "primary_ref" mandatory field is used to tag orders with an identifier meaningful to you. This may be your invoice number or could be a unique tracking number produced as part of your own web site.

The Payment Reference is available to the Result URL and emails, and appears as the Transaction Reference in the SecurePay Merchant Log In.

It is recommended that the Transaction Reference is unique to aid in reconciliation.

```
Example: Defining a Transaction Reference

Scenario: Your Company wants to include its invoice numbers with every
          payment.

<input type="hidden" name="primary_ref" value="642193">
```

3.3.5 *Transaction Amount*

The "amount" mandatory field is the amount that will be transacted through your SecurePay account. By default, the currency is AUD (Australian Dollars).

It must be passed in the base unit of the currency. For Australian Dollars this is cents. For example, $1.00 AUD would be passed as “100”. 100 Yen would be passed as “100”.

```
Example: Setting the transaction amount

Scenario: A customer chooses items from your shopping cart totalling
          AUD$53.20.

<input type="hidden" name="amount" value="5320">
```

3.3.6 *GMT Timestamp*

You must pass a valid Greenwich Mean Time (GMT) timestamp in the field "fp_timestamp" (also known as UTC).

```
The timestamp used to generate the fingerprint must exactly match the one
sent with the associated transaction.
```

It must be of the format "YYYYMMDDHHMMSS" where:

YYYY is the current year

MM is the current two-digit month 01 – 12

DD is the current two-digit day 01 - 31

HH is the current two-digit hour in 24-hour format 01 - 24

MM is the current two-digit minute 00 – 59

SS is the current two-digit second 00 – 59

```
Example: Setting the GMT timestamp

Scenario: Your system has generated a Fingerprint. It is currently 22:35:05
          on 20/06/2011 in Sydney (+10 hours from GMT). The time in GMT is
          12:35:05 on the same day.

<input type="hidden" name="fp_timestamp" value="20110620123505">
```

3.3.7 *Fingerprint*

The Fingerprint is a protected record of the amount to be paid. It must be generated and then included as an input field to SecureFrame. It prevents a customer modifying the transaction details when submitting their card information.

SHA-1 support is being decommissioned over the coming months and once support is fully dropped, only HMAC SHA-256 will be used. In the transition period, to use HMAC SHA-256, provide "isSHA256=" in the callback URL. It can be appended at the end. Do not provide a value, just leave blank as is. Once support for SHA-1 is stopped, you will not need to provide the parameter in the callback URL but can continue to send as it will not be used by SecureFrame. See examples for callback URL in section 4.1.5.5.

The Fingerprint is a HMAC SHA-256 hash of the below mandatory fields, plus the SecurePay Transaction Password in this order with a pipe separator "|":

- "merchant_id"
- Transaction Password (supplied by SecurePay Support)
- "txn_type"
- "primary_ref"
- "amount"
- "fp_timestamp"

```
Example: Setting the fingerprint

Fields joined with a | separator:
ABC0001|txnpassword|0|Test Reference|100|20220228022758

HMAC SHA-256 of the above string using your transaction password as the
secret key (e.g. txnpassword):
33de8f9454a62513838ce534309c76ff8ac2c925bfda0364663d836254497899

<input type="hidden" name="fingerprint"
value="33de8f9454a62513838ce534309c76ff8ac2c925bfda0364663d836254497899">
```

3.4 Transaction Options

3.4.1 *Receipt Page Redirect*

Display of the receipt page and button options on the hosted receipt page can be optionally controlled.

By default, SecureFrame will display the receipt page to the customer.

To redirect to your own receipt page, set “display_receipt=no”.

If SecureFrame displays the receipt (default), you can control the button on the receipt page via the following optional parameters:

- return_url – the fully qualified URL to embed in the button link. The system will append result parameters to this link by default.
- return_url_text – the text of the button
- return_url_target – the target of the button. This can be one of “self”, “new”, “parent” or “top”. “parent” is useful when using the iFrame template to take the browser back to full screen.

```
Example: Set the button to say “Continue...” and take the browser out of
the iFrame when clicked.

<input type="hidden" name="return_url" value="http://www.mysite.com.au">
<input type="hidden" name="return_url_text" value="Continue...">
<input type="hidden" name="return_url_target" value="parent">
```

By default, the return_url is used when a Cancel button is clicked by the card holder. This url can be explicitly set by using the “cancel_url” parameter.

3.4.2 *Card Storage*

The card number used in the transaction may be optionally stored for subsequent batch or XML transaction triggering.

By setting the field "store=yes”, the card will be stored in SecurePay’s Payor system using the “primary_ref” as the Payor ID by default.

3.4.2.1 *Payor*

This is the default card storage method.

With Payor storage, you define the Payor ID to store with the card. Cards and Payor ID’s can be edited via the Merchant Login.

You may also set “store_type=payor” to use this storage type.

You may optionally pass in an alternative value for the stored Payor ID to override the use of “primary_ref”.

Set “payor” to your required value.

```
Example: Set card storage with type Payor and my own Payor ID

<input type="hidden" name="primary_ref” value=”123456”>
<input type="hidden" name="store” value=”yes”>
<input type="hidden" name="store_type” value=”payor”>
<input type="hidden" name="payor” value=”MyCustomer”>
```

3.4.2.2 *Token*

A Token is a string that represents a stored card number. If the card number changes, so does the token, therefore card numbers and tokens cannot be edited, they may only be added or deleted.

Tokens can be used in 3rd party systems to represent card numbers.

If a card is passed to the system for storage several times, the same token is always returned.

To have SecurePay generate a token for a card or return an existing token for a pre-stored card set "store_type=token".

SecureFrame will return the token in the result parameters.

```
Example: Set card storage with type Token

<input type="hidden" name="primary_ref" value="123456">
<input type="hidden" name="store" value="yes">
<input type="hidden" name="store_type" value="TOKEN">
```

3.4.2.3 *Stored Transaction Reference*

When triggering a payment from a stored card of Payor or Token via batch or API, the Transaction Reference defaults to the Payor ID (or Token). This can be overridden by setting a specific Transaction Reference at the time of storage.

Set the "payor_ref" field to store your desired Transaction Reference against the stored card.

This is particularly useful; for tokens, as the token does not necessarily represent your customer.

```
Example: Set card storage with type Token and your own Transaction Reference

<input type="hidden" name="primary_ref" value="123456">
<input type="hidden" name="store" value="yes">
<input type="hidden" name="store_type" value="TOKEN">
<input type="hidden" name="payor_ref" value="123456">

In this example, the Payment Reference ID and the stored Transaction
Reference for future triggering are the same.
```

3.4.2.4 *Store Only*

When you choose to store a customer's card details in SecurePay's card storage system when a SecureFrame transaction is processed, you can optionally choose to store their card details without charging their card. This is known as the Store Only method.

When you use Store Only, the amount included is ignored and is not stored against the customer's details.

Note: As there is no transaction processed with this option, the card number, expiry date and CVV are not validated with the bank at this step.

To use Store Only, additional to the standard mandatory fields you must:

- Pass through the txn_type value of 8 in your requests. This value is defined further in Section 4.2.1.3.

```
TYPICAL USE <input type="HIDDEN" name="txn_type" value="8">
```

- Set display_receipt to false – See section 4.2.5.1 for more details.
- Set confirmation to false – See section 4.2.5.7 for more details.
- Set store_type to "payor".
- Set store to "yes".
- Pass through payor to set the value for the payor id.
- Generate a fingerprint and pass this through as the fingerprint value in your requests. This is a protected record of the transaction details and prevents a customer modifying the details when submitting their card information. Your system will need to create a HMAC SHA-256 hash of the following fields in order, separated by "|". These fields are different to the standard fingerprint fields described in Section 4.2.1.7.

merchant_id|TransactionPassword|txn_type|store_type|payor|fp_timestamp

```
Example:

ABC0001|txnpassword|8|payor|PayorTest|20220228022758

TYPICAL USE <input type="HIDDEN" name="fingerprint"
value="882df414d8583ec99aea9f89177d0cb529d98b0cad400e3d58c9653a95105a2d">
```

When you use Store Only, only the following result fields are returned:

- strestext
- fingerprint
- strescode
- payor
- summarycode

SHA-1 support is being decommissioned over the coming months and once support is fully dropped, only HMAC SHA-256 will be used. In the transition period, to use HMAC SHA-256, provide "isSHA256=" in the callback URL. It can be appended at the end. Do not provide a value, just leave blank as is. Once support for SHA-1 is stopped, you will not need to provide the parameter in the callback URL but can continue to send as it will not be used by SecureFrame. See examples for callback URL in section 4.1.5.5.

3.4.3 *Currency*

If your bank supports multicurrency, you may optionally set the currency of the transaction to one other than AUD.

Set the field "currency" to any ISO three letter currency value.

```
Example: Set the currency to USD

<input type="hidden" name="currency" value="USD">
```

3.4.4 *Surcharging*

Surcharging may be applied across all card types (Visa, MasterCard, etc.), or set individually per card type

Surcharging can be either or both:

- A percentage rate of the invoice total (e.g., 1%)
- A flat fee (e.g., $2.00). The fee is added on top of any rate.

To activate surcharging set "surcharge=yes". This will not apply any surcharging unless rates and/or fees have been set. It will turn on optional result parameters.

To set a surcharge rate of 1%, set "surcharge_rate=1".

```
Example: Set a surcharge rate of 1% for all cards

<input type="hidden" name="surcharge" value="yes">
<input type="hidden" name="surcharge_rate" value="1">
```

Individual rates and fees may be also set per card.

```
Example: Set a surcharge rate of 1% for Amex only

<input type="hidden" name="surcharge" value="yes">
<input type="hidden" name="surcharge_rate_a" value="1">
```

3.4.5 *Template*

There are several options to control how the hosted pages are displayed to your customers. This is controlled by the optional "template" parameter. Responsive is the recommended template.

Set "template=responsive" to use the responsive version. This version displays only the input fields, confirmation text and form buttons. It also will perform well on multi form devices and has friendly CSS tagging.

Set "template=default" to use the full screen payment option. If the template parameter is omitted, this is the default.

Tip: use the Return URL text and target options to control the payment flow from within the iFrame.

```
Example: Set the template to responsive

<input type="hidden" name="template" value="responsive">
```

3.4.6 *Look and Feel*

Several options can be passed to the system to change the look and feel. These include the header logo, the page title and the primary reference name, among others.

```
Example: Set the logo, title and primary reference values

<input type="hidden" name="page_header_image"
value="https://www.mysite.com.au/header.jpg">
<input type="hidden" name="title" value="Payment Page">
<input type="hidden" name="primary_ref_name" value="Order Number">
```

For more look and feel options, please see the Appendices.

3.4.7 *PayPal*

SecureFrame contains the ability to pass customers to PayPal to complete their transaction through their PayPal account. If your SecurePay account is enabled to use this feature and your website passes through "PAYPAL" under "card_types", the SecureFrame payment page will render differently to normal, enabling a tender selection between the accepted card types and PayPal.

Without passing through “PAYPAL” under “card_types” the normal SecureFrame credit card payment page will be rendered, which will accept the “card_types” passed through.

To enquire about enabling this feature on your SecurePay account please speak to SecurePay’s Payment Gateway Advisors on 1300 786 756 (option 2).

```
Example: Set the accepted tenders to Visa, MasterCard and PayPal
<input type="hidden" name="card_types" value="VISA|MASTERCARD|PAYPAL">
```

3.4.8 *FraudGuard*

FraudGuard is an optional service offered by SecurePay. It must be activated on your SecurePay account before it can be enabled by setting the “txn_type” field.

See the Appendices for more details.

Please contact SecurePay Sales for pricing and more information.

3.5 Receiving Result Parameters

The system will send back to your URL's a predefined set of result parameters.

Some parameters will only be returned if an option is activated, such as "store".

The system will POST result parameters the first time it calls your server but will send result parameters using the GET method based on RFC 2616 standards after being redirected.

Result parameters are available to you via two methods:

- Return URL – When you specify a result URL, result parameters are appended to the URL. This occurs for both receipt page redirect and for the return button on the hosted receipt page.
- Call-back URL – SecureFrame contacts your system in the background and sends result parameters

In addition, you can configure a unique cancel URL which overrides the return_url when the Cancel button is clicked on the page.

All result parameters are listed in the Appendices.

3.6 Testing

As you build your system you can test functionality when necessary by submitting parameters to the test URL. You can generate a fingerprint and then complete the transaction by using the card details listed below.

3.6.1 *Test Card Number, Type and Expiry*

Use the following information when testing transactions:

```
Card Number: 4444333322221111
Card Type:   VISA
Card CCV:    123
Card Expiry: 08 / 13 (or any date greater then today)
```

3.6.2 *Simulating Approved and Declined Transactions*

You can simulate approved and declined transactions by submitting alternative payment amounts.

If the payment amount ends in 00, 08, 11 or 16, the transaction will be approved once card details are submitted. All other options will cause a declined transaction.

```
Payment amounts to simulate approved transactions:

$1.00 (100)
$1.08 (108)
$105.00 (10500)
$105.08 (10508)
(or any total ending in 00, 08)
```

```
Payment amounts to simulate declined transactions:

$1.51 (151)
$1.05 (105)
$105.51 (10551)
$105.05 (10505)
(or any totals not ending in 00, 08)
```

Note that when using the live URL for payments, the bank determines the transaction result, independent of the payment amount.

4 Appendices

4.1 Appendix 1: Accepted Input Fields

Mandatory	Transaction	Surcharging	Flow
bill_name	currency	surcharge	display_receipt
merchant_id	display_cardholder_name	surcharge_rate	return_url
txn_type		surcharge_fee	return_url_text
amount	**Card Storage**	surcharge_rate_v	return_url_target
primary_ref	Store	surcharge_fee_v	callback_url
fp_timestamp		surcharge_rate_m	cancel_url
fingerprint	**Card Storage (Optional)**	surcharge_fee_m	cancel_url_text
	store_type	surcharge_rate_a	cancel_url_target
	payor	surcharge_fee_a	confirmation
	payor_ref	surcharge_rate_d	**Look and Feel**
		surcharge_fee_d	template
	Fraud Guard (Optional)	surcharge_rate_j	primary_ref_name
	billing_country	surcharge_fee_j	page_header_image
	delivery_country		page_footer_image
	email_address		page_title
			card_types
			page_style_url

4.1.1 *Mandatory Fields*

4.1.1.1 *bill_name*

CLASS:	Mandatory
FORMAT:	Fixed value “transact”
DESCRIPTION:	Defines the transaction process.
TYPICAL USE:	`<input type="hidden" name="bill_name" value="transact">`

4.1.1.2 *merchant_id*

CLASS:	Mandatory
FORMAT:	Alpha-numeric, length 7
DESCRIPTION:	A unique identifier for the Merchant within the Payment Gateway. This Merchant identifier value is an alphanumeric string allocated to you by SecurePay. This merchant identifier value is not the same as the merchant number provided by your bank.
TYPICAL USE:	`<input type="hidden" name="merchant_id" value="ABC0001">`

4.1.1.3 *txn_type*

CLASS:	Mandatory
FORMAT:	Numeric
DESCRIPTION:	Used to determine the processing type for an individual transaction. May be one of the following: • “0” - PAYMENT: A card payment/purchase transaction. Note: This is the only accepted transaction type for PayPal payments. • “1” - PREAUTH: Used to pre-authorise an amount on a card. The result parameters include the “preauthid” which must be stored and used when completing the pre-authorisation • “2” – PAYMENT with FRAUDGUARD: A card payment/purchase transaction with the optional FraudGuard service • “3” – PREAUTH with FRAUDGUARD: A card preauthorisation transaction with the optional FraudGuard service • “8” – STORE ONLY: This will store the card details without taking a payment or preauthorisation. See section 3.4.4.4. for more details
TYPICAL USE:	`<input type="hidden" name="txn_type" value="0">`

4.1.1.4 *amount*

CLASS:	Mandatory
FORMAT:	Numeric, integer, from 1 to 99999999
DESCRIPTION:	The total amount of the purchase transaction. This value must be a positive integer in the base unit of the currency, such as cents for AUD. Please be careful to correctly specify the amount as the system has no method of determining whether an amount has been correctly specified.
TYPICAL USE:	`<input type="hidden" name="amount" value="10795">`

4.1.1.5 *primary_ref*

CLASS:	Mandatory
FORMAT:	String, min length 1, max length 60
DESCRIPTION:	A string that identifies the transaction. This string is stored by SecurePay as the Transaction Reference. This field is typically a shopping cart id or invoice number and is used to match the SecurePay transaction to your application.
TYPICAL USE:	`<input type="hidden" name="primary_ref" value="My Reference">`

4.1.1.6 *fp_timestamp*

CLASS:	Mandatory
FORMAT:	String, format "YYYYMMDDHHMMSS" in GMT (UTC).
DESCRIPTION:	The GMT time used for Fingerprint generation. This value must be the same submitted to generate a fingerprint as submitted with the transaction. SecurePay validates the time to within one hour of current time. The time component must be in 24-hour time format.
TYPICAL USE:	`<input type="hidden" name="fp_timestamp" value="20220228005104">`

4.1.1.7 *fingerprint*

CLASS:	Mandatory
FORMAT:	String, length up to 60
DESCRIPTION:	SHA-1 support is being decommissioned over the coming months and once support is fully dropped, only HMAC SHA-256 will be used. In the transition period, to use HMAC SHA-256, provide “isSHA256=” in the callback URL. It can be appended at the end. Do not provide a value, just leave blank as is. Once support for SHA-1 is stopped, you will not need to provide the parameter in the callback URL but can continue to send as it will not be used by SecureFrame. A HMAC SHA-256 hash using your transaction password as the secret key of: merchant_id|TransactionPassword|txn_type|primary_ref|amount|fp_timestamp Where the “TransactionPassword” is obtained from SecurePay Support and maybe changed via the SecurePay Merchant Log In. All other fields must be exactly as sent. For a store only request (txn_type 8) merchant_id|TransactionPassword|txn_type|store_type|payor|fp_timestamp

TYPICAL USE:	<input type="hidden" name="fingerprint" value="33de8f9454a62513838ce534309c76ff8ac2c925bfda0364663d836254497899">

4.1.2 *Transaction Fields*

4.1.2.1 *currency*

CLASS:	Optional
FORMAT:	String, length 3, ISO 4217 three letter currency code
DEFAULT:	AUD
DESCRIPTION:	Used to set the transaction currency sent to the bank for processing. You must have a bank merchant facility that accepts currencies other than AUD before using this feature. Set the currency to any ISO 4217 three letter currency code. e.g., USD, NZD, GBP, etc. The format of the amount must match the currency. e.g., Yen has no decimal place.
TYPICAL USE:	`<input type="hidden" name="currency" value="NZD">`

4.1.2.2 *display_cardholder_name*

CLASS:	Optional
FORMAT:	String, values “yes” or “no”
DEFAULT:	yes
DESCRIPTION:	By default, a Cardholder name input field is not displayed to the cardholder. Setting the “display_cardholder_name” field to “yes” will display the cardholder’s name input field on the payment form. This will allow the cardholder to enter their name as part of the payment.
TYPICAL USE:	`<input type="hidden" name="display_cardholder_name" value="yes">`

4.1.3 *Card Storage Fields*

4.1.3.1 *store*

CLASS:	Mandatory for Card Storage
FORMAT:	String, values “yes” or “no”
DEFAULT:	no
DESCRIPTION	“yes” to enable card storage
TYPICAL USE	`<input type="hidden" name="store" value="yes">`

4.1.3.2 *store_type*

CLASS:	Optional
FORMAT:	String, values “PAYOR” or “TOKEN”
DEFAULT:	PAYOR
DESCRIPTION	Type PAYOR will store the card in the SecurePay Payor database. The “primary_ref” field will be used as the Payor ID unless overridden with “payor”. Type TOKEN will either create and store a new token that represents the card number or return a pre-existing token if the card has been stored previously. Tokens are stored as non-editable Payors.
TYPICAL USE:	`<input type="hidden" name="store_type" value="payor">`

4.1.3.3 *payor*

CLASS:	Optional if store_type=PAYOR
FORMAT:	String, length up to 20
DEFAULT:	If not specified, “primary_ref” is used
DESCRIPTION	The Payor ID to store with the Payor. This will become the Transaction Reference for future triggered payments against that Payor unless overridden with “payor_ref”
TYPICAL USE:	`<input type="hidden" name="payor" value="MyPayorID">`

4.1.3.4 *payor_ref*

CLASS:	Optional
FORMAT:	String, length up to 30
DESCRIPTION	Sets the Transaction Reference for future triggered payments. If not set, the system will log the Payor as the Transaction Reference when a payment is triggered
TYPICAL USE:	`<input type="hidden" name="payor_ref" value="MyTransactionReference">`

4.1.4 *Surcharge Fields*

4.1.4.1 *surcharge*

CLASS:	Mandatory for Surcharging
FORMAT:	String, values “yes” or “no”
DEFAULT:	no
DESCRIPTION	“yes” to enable surcharge calculation and result parameters
TYPICAL USE	`<input type="hidden" name="surcharge" value="yes">`

4.1.4.2 *surcharge_rate*

CLASS:	Optional
FORMAT:	Numerical, 0.0001 to 99.9999
DEFAULT:	0
DESCRIPTION	Percentage surcharge to apply across all card types. May be overridden by applying individual card type rates.
TYPICAL USE	`<input type="hidden" name="surcharge_rate" value="1.00">`

4.1.4.3 *surcharge_fee*

CLASS:	Optional
FORMAT:	Numerical, integer, 0 to 999999 cents (or base currency unit)
DEFAULT:	0
DESCRIPTION	Fee amount to be added to the payment amount for all card types.
TYPICAL USE	`<input type="hidden" name="surcharge_fee" value="100">`

4.1.4.4 *surcharge_rate_v/m/a/d/j*

CLASS:	Optional
FORMAT:	Numerical, 0.01 to 99.9
DEFAULT:	0
DESCRIPTION	Percentage surcharge to apply to a specific card type. Overrides “surcharge_rate”.
TYPICAL USE	`<input type="hidden" name="surcharge_rate_v" value="1">`

4.1.4.5 *surcharge_fee_v/m/a/d/j*

CLASS:	Optional
FORMAT:	Numerical, 0 to 999999 cents (or base currency unit)
DEFAULT:	0
DESCRIPTION	Fee amount to be added to the payment amount for a specific card type. Overrides “surcharge_fee”.
TYPICAL USE	`<input type="hidden" name="surcharge_fee_v" value="100">`

4.1.5 *Transaction Flow Fields*

4.1.5.1 *display_receipt*

CLASS:	Optional
FORMAT:	String, values “yes” or “no”
DEFAULT:	yes
DESCRIPTION	Define which system displays the receipt page to the card holder. “Yes” means SecurePay displays the receipt page. To redirect to your own receipt page set this value to “no” and use the return_url parameter.
TYPICAL USE	`<input type="hidden" name="display_receipt" value="no">`

4.1.5.2 *return_url*

CLASS:	Optional
FORMAT:	String, fully qualified URL
DESCRIPTION:	The URL of the page on the Merchant web site that accepts transaction result data as POST elements. The page may be almost any form of web page, including static HTML pages, CGI scripts, ASP pages, JSP pages, PHP scripts, etc, however cookies or other forms of additional information will not be passed through the Payment Gateway. The “return_url” must be a URL for a publicly visible page on a web server within a domain that is delegated to a public IP number. Internal machine names, such as "localhost", Windows-style machine names, and privately translated IP numbers will fail. If “display_receipt” is set to “yes”, this is the URL of the button on the hosted result page. If “display_receipt” is set to “no”, this is the redirect URL following a transaction. Your web host cannot use Server Name Indicators (SNIs) for determining which SSL certificate to serve. This is not supported by SecurePay’s systems.
TYPICAL USE:	`<input type="hidden" name="return_url" value="http://www.myserver.com.au/result.asp">`

4.1.5.3 *return_url_text*

CLASS:	Optional
FORMAT:	String, length up to 30 characters
DEFAULT:	none

DESCRIPTION	Defines the text on the hosted result page button that takes the card holder to the next step in the process.
TYPICAL USE	`<input type="hidden" name="return_url_text" value="Continue">`

4.1.5.4 *return_url_target*

CLASS:	Optional
FORMAT:	String, values “self”, “top”, “parent” or “new”
DEFAULT:	none
DESCRIPTION	Defines the target of the hosted result page button that takes the card holder to the next step in the process. Useful for iFramed pages to take the process out of the iFrame.
TYPICAL USE	`<input type="hidden" name="return_url_target" value="parent">`

4.1.5.5 *callback_url*

CLASS:	Optional
FORMAT:	String, fully qualified URL
DESCRIPTION:	The URL of the page on the Merchant web site that accepts transaction result data as POST elements. The page may be almost any form of web page, including static HTML pages, CGI scripts, ASP pages, JSP pages, PHP scripts, etc, however cookies or other forms of additional information will not be passed through the Payment Gateway. The “callback_url” must be a URL for a publicly visible page on a web server within a domain that is delegated to a public IP number. Internal machine names, such as "localhost", Windows-style machine names, and privately translated IP numbers will fail. Your web host cannot use Server Name Indicators (SNIs) for determining which SSL certificate to serve. This is not supported by SecurePay’s systems. SHA-1 support is being decommissioned over the coming months and once support is fully dropped, only HMAC SHA-256 will be used. In the transition period, to use HMAC SHA-256, provide “isSHA256=” in the callback URL. It can be appended at the end. Do not provide a value, just leave blank as is. Once support for SHA-1 is stopped, you will not need to provide the parameter in the callback URL but can continue to send as it will not be used by SecureFrame.

TYPICAL USE:	`For HMAC SHA-256 fingerprint`
	`<input type="hidden" name="callback_url" value="http://www.myserver.com.au/result.asp?isSHA256=">`
	`For SHA-1`
	`<input type="hidden" name="callback_url" value="http://www.myserver.com.au/result.asp">`

4.1.5.6 *cancel_url*

CLASS:	Optional
FORMAT:	String, fully qualified URL
DEFAULT:	Uses the return_url if not present.
DESCRIPTION:	The URL of the page on the Merchant web site that accepts transaction result data as POST elements when the card holder clicks a Cancel button on a hosted page.

	The page may be almost any form of web page, including static HTML pages, CGI scripts, ASP pages, JSP pages, PHP scripts, etc, however cookies or other forms of additional information will not be passed through the Payment Gateway. The "cancel_url" must be a URL for a publicly visible page on a web server within a domain that is delegated to a public IP number. Internal machine names, such as "localhost", Windows-style machine names, and privately translated IP numbers will fail. Note: some result parameters may not be populated depending on the point of cancellation of the process. Your web host cannot use Server Name Indicators (SNIs) for determining which SSL certificate to serve. This is not supported by SecurePay's systems.
TYPICAL USE:	`<input type="hidden" name="cancel_url" value="http://www.myserver.com.au/result.asp">`

4.1.5.7 *confirmation*

CLASS:	Optional
FORMAT:	String, values "yes" or "no"
DEFAULT:	yes
DESCRIPTION:	By default, a Confirmation page is displayed to the cardholder, after the card details have been entered. This is to allow the cardholder to review the details (and edit as required) prior to initiating the payment. Setting the "confirmation" field to "no" will bypass this default confirmation step.
TYPICAL USE:	`<input type="hidden" name="confirmation" value="no">`

4.1.6 *Look and Feel Fields*

4.1.6.1 *template*

CLASS:	Optional
FORMAT:	String, values “default”
DEFAULT:	default
DESCRIPTION	Defines which hosted page template is displayed to card holders.
TYPICAL USE	`<input type="hidden" name="template" value="responsive">`

4.1.6.2 *primary_ref_name*

CLASS:	Optional
FORMAT:	String, length up to 30 characters
DEFAULT:	Invoice Number
DESCRIPTION	Defines the label on the hosted pages for the Primary Reference field.
TYPICAL USE	`<input type="hidden" name="primary_ref_name" value="Order Number">`

4.1.6.3 *page_header_image*

CLASS:	Optional
FORMAT:	String, fully qualified URL. HTTPS method only. Must end in one of gif, jpeg, jpg, png. Valid images only.
DEFAULT:	none
DESCRIPTION	URL of an image to be used as the header of the hosted pages.
TYPICAL USE	`<input type="hidden" name="page_header_image" value="` `http://www.myserver.com.au/result.asp/header.jpg">`

4.1.6.4 *page_footer_image*

CLASS:	Optional
FORMAT:	String, fully qualified URL. HTTPS method only. Must end in one of gif, jpeg, jpg, png. Valid images only.
DEFAULT:	none
DESCRIPTION	URL of an image to be used as the footer of the hosted pages.
TYPICAL USE	`<input type="hidden" name="page_footer_image" value="` `http://www.myserver.com.au/result.asp/header.jpg">`

4.1.6.5 *page_title*

CLASS:	Optional

FORMAT:	String, length up to 30 characters
DEFAULT:	Invoice Payment
DESCRIPTION	Defines the title text on the hosted pages.
TYPICAL USE	`<input type="hidden" name="page_title" value="Pay Your Invoice">`

4.1.6.6 *card_types*

CLASS:	Optional
FORMAT:	String, bar "|" separate values from VISA, MASTERCARD, AMEX, DINERS, JCB, PAYPAL
DEFAULT:	Visa, MasterCard
DESCRIPTION	Defines the card types accepted. This must be a subset of the card types you accept.
TYPICAL USE	`<input type="hidden" name="card_types" value="VISA|MASTERCARD">`

4.1.6.7 *page_style_url*

CLASS:	Optional
FORMAT:	String, fully qualified URL
DESCRIPTION:	The URL of the page on the Merchant web site that provides styling for the hosted page. Styles override the default page styles. Images embedded by be fully qualified URL's. All JavaScript and malicious code will be removed.
TYPICAL USE:	`<input type="hidden" name="page_style_url" value="http://www.myserver.com.au/hppstyles.css">`

4.1.7 *Fraud Guard Fields*

4.1.7.1 *billing_country*

CLASS:	Optional
FORMAT:	String, length 2, country code
DEFAULT:	None
DESCRIPTION	Payee's Country two letter code
TYPICAL USE	`<input type="text" name="billing_country" value="AU">`

4.1.7.2 *delivery_country*

CLASS:	Optional
FORMAT:	String, length 2, country code
DEFAULT:	None

DESCRIPTION	Order delivery country two letter code
TYPICAL USE	`<input type="text" name="delivery_country" value="AU">`

4.1.7.3 *email_address*

CLASS:	Optional
FORMAT:	String, length less than 30
DEFAULT:	none
DESCRIPTION	Payee's email address
TYPICAL USE	`<input type="text" name="email_address" value=”test@test.com">`

4.2 Appendix 2: Result Fields

4.2.1 *Standard Result Fields*

4.2.1.1 *summarycode*

The one-digit summary of the transaction result:

1 = Approved

2 = Declined by the bank

3 = Declined for any other reason

4 = Cancelled by user

Use “rescode” and “restext” for more detail of the transaction result.

4.2.1.2 *rescode*

The primary indicator of the transaction result.

Bank response or internal error code numbers used to determine the transaction result. Rescode's of 00, 08 and 11 indicate approved transactions, while all other codes represent declines. A full list of response codes is available for download from the SecurePay web site.

4.2.1.3 *restext*

The associated text for each "rescode". For bank response codes 00 – 99, this field is generated by the bank's payment systems. All other codes have the "restext" generated by SecurePay.

4.2.1.4 *refid*

The value of the primary_ref parameter from the transactions request. This value is returned to the Merchant's processing system to allow matching of the original transaction request.

4.2.1.5 *txnid*

The bank transaction ID. This string is unique at least per terminal, per bank and per settlement date. This value is required to be re-entered along with other details of the original payment when processing refunds.

4.2.1.6 *settdate*

The bank settlement date. This is the date the funds will be settled into the merchant's account. The date will correspond to today's date until the bank's cut-off time (typically 6-11pm), then roll to the following business day. The settlement date is returned in the format "YYYYMMDD".

4.2.1.7 *pan*

The masked card number of format first six...last three. e.g., 444433...111

4.2.1.8 *expirydate*

The four-digit expiry date entered by the customer. e.g., 0813

4.2.1.9 *merchant*

The merchant_id value used for the transaction

4.2.1.10 *timestamp*

The GMT (UTC) time used for the response fingerprint of the format "YYYYMMDDHHMMSS". This value must be used when generating a string to compare to the response “fingerprint” value to validate the response. The time component must be in 24-hour time format.

4.2.1.11 *amount*

The amount in the base unit of the currency, typically cents for Australian Dollars. e.g., $104.23 will be 10423.

4.2.1.12 *fingerprint*

A string used to validate the transaction output.

SHA-1 support is being decommissioned over the coming months and once support is fully dropped, only HMAC SHA-256 will be used. In the transition period, to use HMAC SHA-256, provide “isSHA256=” in the callback URL. Do not provide a value, just leave blank as is. Once support for SHA-1 is stopped, you will not need to provide the parameter in the callback URL but can continue to send as it will not be used by SecureFrame.A HMAC SHA-256 hash using your transaction password as the secret key of the following fields in order, separated by “|”:

For txn_type 0-7:

merchant, transaction password, reference, amount, timestamp, summarycode

For Example:

ABC0001|txnpassword|MyReference|1000|20220228025627|1

is HMAC SHA-256 hashed to give:

0662c9d11c12d3cb15986c53b95e053691b33e43c40bec5ad70b827c01229771

For txn_type 8:

merchant, transaction password, reference, amount, timestamp, summarycode

For Example:

ABC0001|txnpassword|payor|TestPayorID|20220228025627|1

is HMAC SHA-256 hashed to give:

599562e82101f8202d1965c1124340fa218a76a91fcbdeed0b1060128d16ef6a

4.2.1.13 *cardtype*

The card type used for the transaction. This will be one of:

- “Visa”
- “MasterCard”
- “American Express”
- “Diners”
- “JCB”
- “PayPal”

4.2.2 *Preauthorisations*

Preauth fields are only returned when “txn_type” is set to a preauth transaction type on input.

4.2.2.1 *preauthid*

The bank pre-authorisation ID returned by the payment gateway. This value is used when sending a pre-authorisation complete transaction via XML or Batch.

4.2.3 *FraudGuard Result Fields*

FraudGuard fields are returned in addition to the Standard Result Fields if your account is enabled for FraudGuard by the SecurePay Support team and the “txn_type” includes the FraudGuard option.

4.2.3.1 *afrescode*

FraudGuard response code if “txn_type” includes FraudGuard. Returns “400” if the transaction passes FraudGuard tests. Returns a different string depending on the type of fraud detected.

4.2.3.2 *afrestext*

FraudGuard response text. Used if the “afrescode” is not 000. Contains a description of the FraudGuard result.

4.2.4 *Card Storage Result Fields*

Card storage fields are returned in addition to the Standard Result Fields if the input field “store” “yes” on input.

4.2.4.1 *stsummarycode*

The one-digit summary of the storage result:

1 = Successful

2 = Unsuccessful

4 = Cancelled by user

Use “strescode” and “strestext” for more detail of the storage result.

4.2.4.2 *strescode*

Storage code Returns “800” if the Payor or Token was successfully stored. Returns a different string if the storage failed. The “strestext” describes the failure reason.

4.2.4.3 *strestext*

Storage response text. Contains a description of the storage result.

4.2.4.4 *payor*

If “store_type” is set to “payor” (or absent - default), the “payor” field will be returned in this result field.

4.2.4.5 *token*

If “store_type” is set to “token”, the system-generated token will be returned in this field. If the card has never been stored before, this will be a new value. If the card has been stored previously, the stored value will be returned.

4.2.5 *Surcharge Result Fields*

If “surcharge” is set to “yes” on input or the fields are set in your account, the following additional fields are returned:

4.2.5.1 *baseamount*

The amount passed to the system, prior to the addition of the surcharge, in the base unit of the currency, typically cents for Australian Dollars. E.g. $104.23 will be 10423. This amount added to the “suramount” equals the “amount”.

4.2.5.2 *suramount*

The surcharge amount in the base unit of the currency, typically cents for Australian Dollars. E.g. $104.23 will be 10423. This amount added to the “baseamount” equals the “amount”.

4.2.5.3 *surrate*

The surcharge rate in percent used to calculate the surcharge. E.g. 1% = “1”.

4.2.5.4 *surfee*

The surcharge fee added to the amount amount in the base unit of the currency, typically cents for Australian Dollars. E.g. $104.23 will be 10423.