

# SecurePayTech.com

## Three Party Integration

Version 1.1 (updated 3/12/2009)

securepaytech.com

■ SECURE PAYMENT TECHNOLOGIES

## Introduction

3-Party Payments allow the Payment Server to manage the payment pages and collect the cardholder's card details on SecurePayTech's behalf. This means that the bank handles the collecting and processing of card details. However, it is also possible for either the merchant or SecurePayTech to collect these details depending on which is required.

## Prerequisites

Before any 3-Party transactions can take place on your SecurePayTech account, the following permissions must be enabled on your account:

- EPS
- CardDetails
- ThreeParty

Please contact SecurePayTech to arrange for these to be enabled.

## Approaches

There are three ways to integrate with SecurePayTech 3-Party payments:

1. Allow the bank to capture the customer's card details. (*see Bank captures card details below*)
2. Redirect the customer's web browser to a SecurePayTech-branded page for SecurePayTech to capture the customer's card details. (*see SecurePayTech captures card details below*)
3. Capture the card details on your own website and include them in the browser redirect to SecurePayTech. (*see Your web site captures card details below*)

Using any of the approaches, you can optionally have your website notified of the outcome of a transaction even in the event of a communications error. (*see Transaction Result Notification below*)

**Important:** All http requests relating to 3-Party Payments require the `digest` parameter to be specified. Please refer to the *Security* section below for information on this.

## Bank captures card details

In this approach, when the customer is redirected to SecurePayTech with their order details, they will be redirected to a secure bank branded page where they will be prompted to enter their card details.

Be sure not to include any card parameters in the request as this is how SecurePayTech determines that bank branding is desired.

Request URL: <https://tx.securepaytech.com/three-party/take-payment>

Request Parameters:

<b>Parameters</b>	<b>Description</b>	<b>Format</b>
merchantId	Your VPS merchant ID.	Must be supplied
orderReference	An order reference generated by your website	
amount	The amount of the payment expected from the customer.	\$12.50 is represented by 12.5
expectCardSecurityCode	Whether or not to ask the customer for a CSC code.	Must be either "yes" or absent from the request.
returnUrl	The URL of a page on your web site to which the customer is returned after the transaction is complete	A valid HTTPS URL
lateReturnUrl	The URL that SecurePayTech uses to notify your web site of the outcome of the transaction	A valid HTTPS URL (optional)

## SecurePayTech captures card details

This approach will direct the customer to a secure SecurePayTech branded payment form where they will be prompted to enter their credit card details.

Request URL: <https://tx.securepaytech.com/three-party/ready-to-pay>

Request Parameters:

<b>Parameters</b>	<b>Description</b>	<b>Format</b>
merchantId	Your VPS merchant ID.	Must be supplied
orderReference	An order reference generated by your website	
amount	The amount of the payment expected from the customer.	\$12.50 is represented by 12.5
expectCardSecurityCode	Whether or not to ask the customer for a CSC code.	Must be either “yes” or absent from the request.
returnUrl	The URL of a page on your web site to which the customer is returned after the transaction is complete	A valid HTTPS URL
lateReturnUrl	The URL that SecurePayTech uses to notify your web site of the outcome of the transaction	A valid HTTPS URL (optional)

## Your website captures card details

With this approach, your website will collect the customer's card details, and redirect them to SecurePayTech with the purchase details as well as their card details.

**Request URL:** <https://tx.securepaytech.com/three-party/ready-to-pay>

**Request Parameters:**

Parameters	Description	Format
merchantId	Your VPS merchant ID.	Must be supplied
orderReference	An order reference generated by your website	
amount	The amount of the payment expected from the customer.	\$12.50 is represented by 12.5
expectCardSecurityCode	Whether or not to ask the customer for a CSC code.	Must be either "yes" or absent from the request.
returnUrl	The URL of a page on your web site to which the customer is returned after the transaction is complete	A valid HTTPS URL
lateReturnUrl	The URL that SecurePayTech uses to notify your web site of the outcome of the transaction	A valid HTTPS URL (optional)
cardType	The type of Payment Card	Accepted Values: Visa, Mastercard, American Express or Diners Club Card
cardNumber	The number in front of the Payment Card.	Different card types will have different formats. They will be validated by Paymark during the transaction
cardExpiryMonth	The month in which the Payment Card expires	Accepted Values: 01 .. 12
cardExpiryYear	The 4-digit year in which the payment card expires	Accepted Values: 2009 .. 2099
expectSecurityCode	SecurePayTech will expect a CSC if this parameter has a value of yes	Must be either "yes" or absent
cardSecurityCode	The 3 or 4 digit code behind the Payment Card. (a.k.a CSC or CVV2)	Numeric 3 to 4 digits
cardHolderName	The name of the Payment Card holder	

## Alternative Parameter Names

The following parameters are also accepted to make it easier for merchants who are already integrated with SecurePayTech to migrate their websites into using this 3-Party functionality.

merchantID	An alias for merchantId
orderRef	An alias for orderReference
ccType	Similar to cardType but expecting old codes for types of payment card.
ccNumber	An alias for cardNumber
enableCsc	An alias for expectSecurityCode but accepting any value to indicate that a security code is expected
csc	An alias for cardSecurityCode
ccName	An alias for cardHolderName

## Transaction Result Notification

Sometimes the transaction has completed but the customer's browser is not redirected back to the "transaction complete" page on your website. This means that your website will not know whether payment has been made or not. This can be caused by:

- The customer closing the browser window that displays the payment confirmation page instead of using the return link.
- Network problems experienced by either the customer, SecurePayTech or any of the financial institutions involved in the transaction.

When a transaction has been attempted through SecurePayTech yet SecurePayTech has not been told of the outcome of the transaction, SecurePayTech will periodically check with the financial institutions in the background to determine the outcome of the transaction. SecurePayTech will therefore know the outcome of every transaction eventually even in the event of a communication error. When SecurePayTech discovers the outcome of a transaction in this way, the customer's browser is no longer available to deliver a redirect to your website but SecurePayTech can pass the information to your website if it includes a URL for SecurePayTech to talk to (`lateReturnUrl`). Under these circumstances, your website will then know whether the payment has been made or not. If you do not implement a late-transaction-result URL for SecurePayTech to handle such circumstances, the result of missing transactions is accessible by logging into the SecurePayTech merchant administration site and searching the transaction history.

The following parameters are passed to the late transaction URL:

`TxnResponseCode`  
`Result`  
`OrderReference`  
`Amount`  
`MerchTxnRef`  
`TransactionNumber`  
`ReceiptNumber`  
`AuthorisationID`  
`BatchNumber`  
`AcquirersResponse`

The only parameter always present is `TxnResponseCode`, which is a single alphanumeric character and is '0' on success. Other parameters may be absent on error depending on the error. The error message is in the `Result` parameter.

## Security

### Sniffing

Anyone with a computer on the same network as the customer is able to see and record the information passed between the customer's web browser and your website. This is especially true when the customer is in an Internet café.

To protect against this attack, all redirects are served over HTTPS and are to HTTPS URLs.

### Manipulation

1. When your website redirects the browser to SecurePayTech, it includes in the redirect the amount that should be paid. If the user were to edit this amount before completing the redirect to SecurePayTech and then paying as normal, the transaction would be reported as successful to your website yet the amount paid would be whatever the customer had changed it to.
2. The customer could use the URL that is used by SecurePayTech to notify your web site of the result of a transaction when the communications breaks down to tell your website that payment has been made when it has not.

To defend against these attacks, a secure hash (see *Secure Hashes* below) is used to detect that the request or redirect could only have been sent by SecurePayTech and that redirects have not been tampered with while in transit.

### Secure Hashes

Your website would store a key that is only known between your website and SecurePayTech. This key is our shared secret.

When your website serves a browser redirect to SecurePayTech or when SecurePayTech serves a redirect back to your website or when SecurePayTech uses your late-transaction-result URL (if you choose to specify one), then the shared secret and the HTTP parameter values of the request is put through a one-way hashing algorithm to produce a digest string. The digest string for the same shared secret and parameter values is unique.

The receiver of the request will then gather the parameters and values. The hashing algorithm is then applied to the shared secret and the values of the parameters and then compared with the digest parameter value that was given. If both the results are the same, then it can be confirmed that the request was not tampered with.

It is strongly recommended that your website check any incoming requests this way to defend against attacks described in the *Manipulation* section above.

## SecurePayTech 3-Party Integration

### Generating a Secure Hash

Secure hashes are computed by concatenating the values of the HTTP parameters in the ASCII order of the name of the parameters before prepending the resultant string with the shared secret and feeding the final string to the MD5 algorithm. The Secure hash is the representation of the MD5 digest in hexadecimal.

1. Start by arranging the parameters by parameter name in ASCII order.  
(ie. Amount, lateReturnUrl, merchantId, orderReference, returnUrl)
2. Concatenate the values of these variables.  
(ie. **1.00https://mylatereturnurl.com123456reference01http://myreturnurl.com**)
3. Prepend the shared secret key to the concatenation above. (ie: secret: ABC123)  
(ie. ABC1231.00https://mylatereturnurl.com123456reference01http://myreturnurl.com)
4. Feed string through an MD5 hashing algorithm.  
(MD5: **e51470fbff926ae2d84abc5b622a8dfb**)
5. This MD5 hash should then be posted as the digest parameter if you are sending a request, or compared to the digest parameter if receiving a request.