ZimSwitch vPayments

API and Developer Documentation

ZSS Research and Development

Document Control

Conventions

- Font: Calibri
- Font Size: 11
- Office Template: Office
- Bullets:
  - 🟥 Critical Note
  - 🟩 General Note
  - 🟧 Warning Note

Revision Record

Date	Edited By	Comments	Version
11/13/2013	NA	Public Release	0.0.2

Issue Control

This document is issued by the ZimSwitch Shared Services RnD Manager, to whom any change requests or queries should be directed.

The review life for this document is 1 year.

Distribution

Copy No.	Name	Title	Address
01 Master	Public Release	vPayments API and Developer Documentation	
02			
03			
04			
05			

The master for this document is held electronically and only signed copies are valid. An unsigned, printed document is not copy controlled and is to be used for INFORMATION ONLY, as it will not be automatically updated. It is therefore the responsibility of the reader to ascertain that it is a currently valid copy.

Contents

Scope

Who should read this?

- Merchants who wish to provide vpayments functionality to their customers

The scope of this document is limited to the merchant integration to the vPayments API.

Overview

vPayments is an online payments system (eCommerce) provided by Zimswitch Shared Services to allow "card not present" payments from the customer's bank account directly to the merchant's bank account or into a bank controlled suspense account (BuySafe Option).

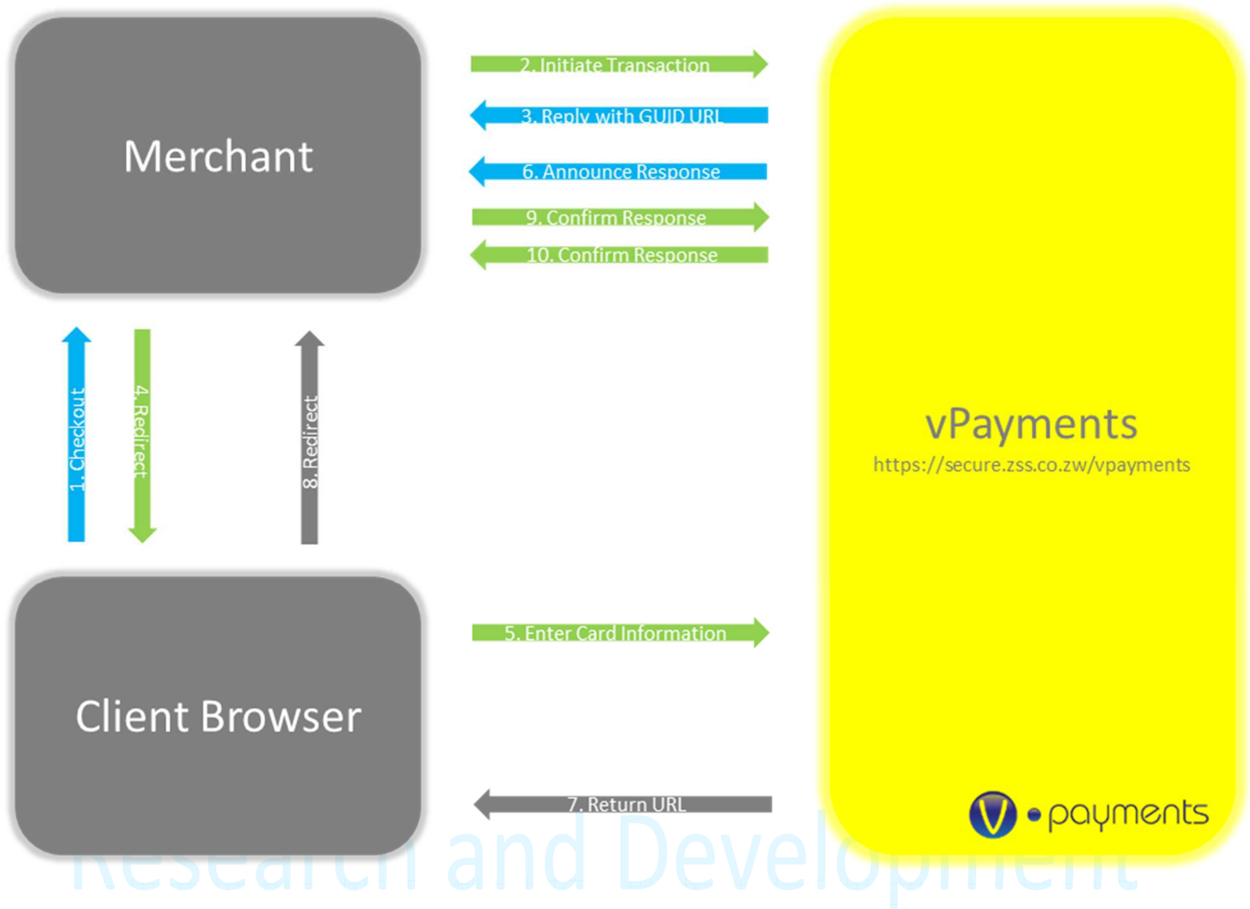
The vPayments Gateway enables merchants to offer their customers the convenience of paying directly from their bank account.

Customers will already have a vPayments account that is linked to their bank accounts. The experience to the customer is as follows:

- Select goods and services available on the merchant site and add them to their virtual shopping cart.
- Click the merchant site "Check Out" button.
- Choose a payment method:
  - Cash on delivery
  - RTGS
  - Transfer or Cash deposit
  - vPayments
- If vPayments is selected, the customer will be re-directed from the merchant site to the vPayments site along with additional information as specified in this document. (amount, merchant ID etc)
- The customer will then be able to log in to vPayments using their vPayments username and password and process the payment securely
- Once payment is completed, the customer will be re-directed to the merchant site with the appropriate payment status

For merchants who do not have access to the appropriate development resources, pre-built vPayments payment modules for popular shopping cart systems are available from *our preferred partner (sales@webdevworld.com)* – integration is available at discounted prices for C#, java and PHP. Prices start from $90.

vPayments Model



1. Customer checks out on the Merchant Site.
2. Merchant server initiates a transaction to vPayments.
3. vPayments replies to the merchant with a redirect URL.
4. The Client browser is redirect to vPayments.
5. Client logs in and processes the transaction.
6. * vPayments announces success or failure to Merchant Server.
7. vPayments uses return URL supplied in initiation
8. Client is redirect back to the Merchant site.
9. * Merchant server checks the transaction status with vPayments.
10. * vPayments replies with transaction status.

*Items marked with *, do not have to occur and are in place as fail safes in the event of a communications breakdown.*

BuySafe

BuySafe is a technology intended to protect the customer. It is an extra layer of security to ensure the customer gets their goods delivered on time and in good condition. Not all merchants are BuySafe, on the payment page you will see a notice and logo indicating if the merchant is BuySafe or not. If a merchant is BuySafe certified, funds paid by a customer will be held in a control account at the merchant's bank until the merchant has actually delivered the goods or services. Once the merchant has updated the status of the order to "delivered", the customer is notified and has at least a 24 hour period to dispute the status. If no dispute is raised, the funds are moved from the bank's control account into the merchant's bank account. This extra step is designed to give customers peace of mind when they purchase goods or services that are not delivered or available immediately after the time of purchase. BuySafe is an optional service and is not compulsory for any merchant. The BuySafe process is as follows:

- Customer checks out of the merchant's site.
- Customer makes payment on vPayments.
- If the merchant is registered on BuySafe they will not immediately receive the money, instead it will be held in suspense at their bank.
- The merchant delivers the goods to the customer and then confirm delivery on VPayments, it is vital the merchant also gets a signature to confirm delivery in case of a dispute.
- The customer is notified by email that delivery has been made and has at least 24 hours to dispute delivery.
- If no dispute is lodged before the end of next business day settlement the payment will be transferred to the merchant's account.
- If there is a dispute then the funds are kept in suspense and cannot be accessed by the customer or merchant until the bank has completed a manual dispute resolution process. Note that this process may incur charges from the bank.

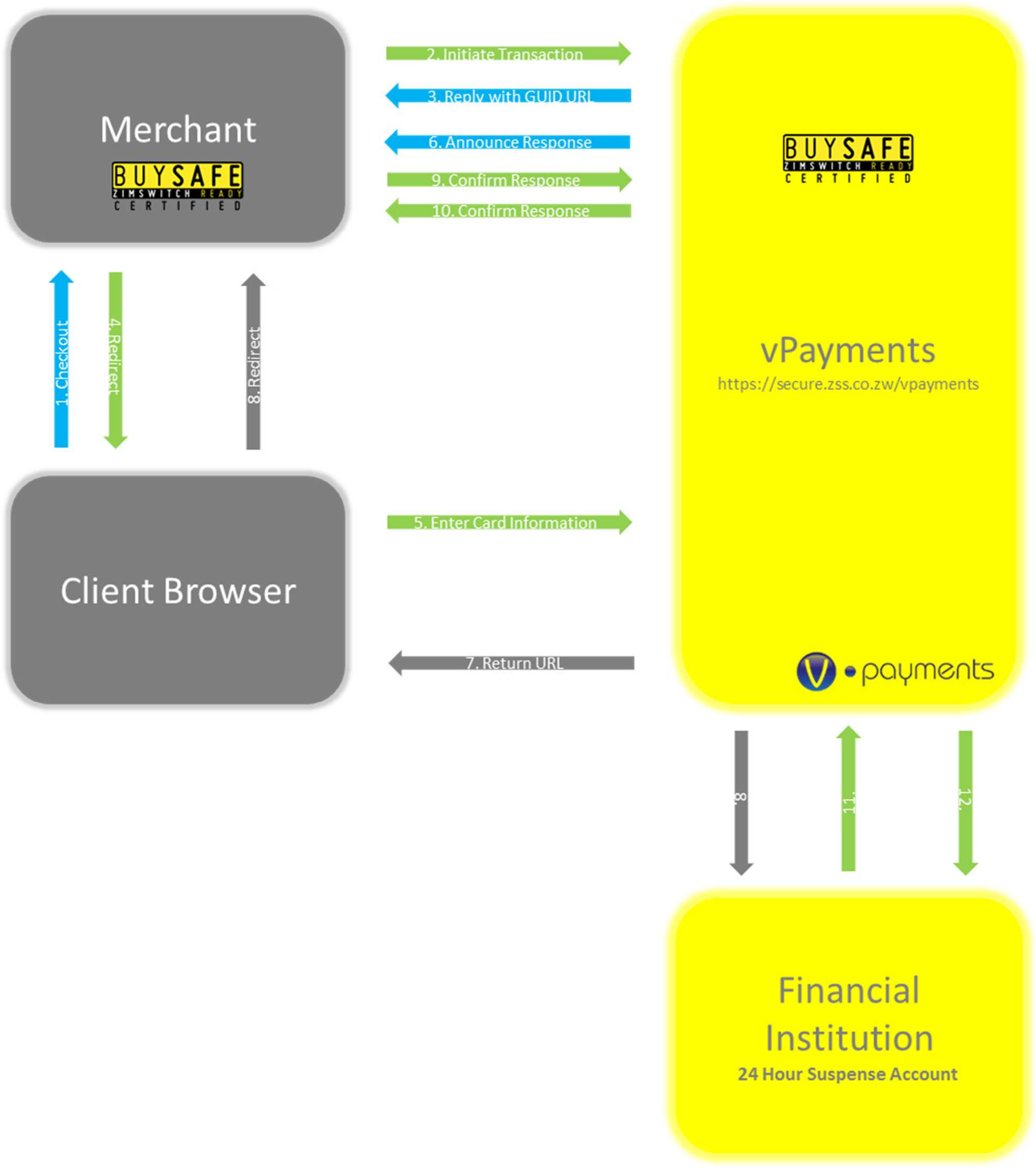
Further disputes

In the event the merchant is not BuySafe registered or the delivery dispute period has expired and the merchant has received their money the customer can still dispute a transaction. But in this case no funds are held in suspense, the merchant still has access to the funds, but the customer, merchant and bank are notified and a manual resolution process is started.

Is it safe to use a non BuySafe merchant?

Not all merchants will be BuySafe registered, for example service providers who do not deliver a physical item. It is still safe to pay such a merchant, but the customer should decide the level of trust for themselves. If the merchant is a well-established company and known to the customer then it is very safe to make payment. It is only a risk if the customer makes payment to an unknown entity that they are not sure will complete delivery, in this case rather use a BuySafe registered merchant who can provide the same service.

vPayments - BuySafe Model



1. Customer checks out on the Merchant Site.
2. Merchant server initiates a transaction to vPayments.
3. vPayments replies to the merchant with a redirect URL.
4. The Client browser is redirect to vPayments.
5. Client logs in and processes the transaction.
6. vPayments initiates a transaction that debits the customer's account and credits the Suspense account at the Financial Institution.

7. vPayments uses return URL supplied in initiation
8. Client is redirected back to the Merchant site.
9. * Merchant server checks the transaction status with vPayments.
10. * vPayments replies with transaction status.
11. vPayments releases the funds in the BuySafe suspense account after 24 hours period has passed.
12. *[1]* If a customer dispute is raised within this period the transaction is placed on hold and the transaction is reversed.

*Items marked with *, do not have to occur and are in place as fail safes in the event of a communications breakdown.*

[1] – This step never happens if a dispute does not occur.

API

Currently the only supported API Interface is the HTTP POST Protocol. As more supported API Protocols are exposed by the vPayments application they will be added to this document.

HTTP API

The HTTP API exposes URL's that will accept and announce updates with the HTTP POST requests. All messages from the merchant and successful responses from vPayments must include a valid hash (This will be explained further).

Encoding and Formatting

- All server to server communication is done using a single message specification.
- All messages are done by HTTP Post and all values are URL Encoded.
- All replies are done in plain text in the same format as an HTTP Post and all values are URL Encoded.

Merchant Setup

Before a merchant can integrate with vPayments they need to sign up for the service with the connected Financial Institution. Once the Financial Institution has completed their setup the merchant will be emailed with a URL to setup their first account. At this point the account will be in *Test Mode (sandbox).*

Test Mode

Test Mode means that all integration is possible but no funds are transferred and every transaction is automatically failed when the customer clicks make payment. Once the merchant is complete with their integration they can request that their account is moved out of Test Mode and moved into Live. At this point live transactions will be sent through and funds will move.

Migrating to Live

After first login the merchant will be able to retrieve their Merchant ID and Merchant Secret Key. It is strongly recommended that the merchant regenerates this key when moving from Test Mode to

Live, and every annual quarter thereafter. The Merchant Secret Key serves as unique identifier for verifying a response from the merchant. ***It is to be kept confidential at all times.***

Process Flow

The payment process is multi step and requires interaction between the customer's browser, the merchant server and vPayments. The following flow is for a complete transaction, if the user closes their browser, there is a communications breakdown, cancels the payments authorization, or the transaction fails at any point, the flow is incomplete.

- Customer selects pay with vPayments on the merchant website.
- The merchant server initiates a transaction with vPayments providing a confirmation URL and a return URL for this transaction on their website.
- vPayments replies with a process URL and a check URL on vPayments.
- The merchant site redirects the customer's browser to the process URL provided in step three.
- The customer authorizes the payment on vPayments.
- vPayments announces the result of the transaction to the merchant server using the confirm URL provided in step two.
- vPayments redirect the client's browser back to the merchant site using the return URL provided in step two.
- The merchant server polls the check URL provided in step three to confirm payments status

- *Why it is done like this*

  Step six only needs to occur so that the merchant site has been announced the status before the client is returned to it. There is the possibility this will arrive after the redirect or fail to arrive at all, even if it does arrive on time it is strongly recommended that the merchant server polls vPayments for confirmation anyway on the client redirect. This way you can ensure the transaction status.
- *It is **vital** that the merchant verified any hash appended to a message by vPayments, without this check it is possible that fake a reply has been delivered to the merchant.*

Initiate Transaction

The first message sent from the merchant server to vPayments is to create the transaction and get the subsequent URL's use in processing.

The URL to post the message to is

- https://secure.zss.co.zw/vpayments/Interface/InitiateTransaction

The post should include the following fields.

Initiate Transaction Fields

FIELD NAME	FIELD TYPE	FIELD DESCRIPTION	MANDATORY
confirmurl	URL Encoded String	The URL the vPayments server can use to	YES

		announce transaction results to the merchant site	
returnurl	URL Encoded String	The URL on the merchant website the client will be redirected to after processing the transaction	YES
reference	URL Encoded String	The merchant reference as you would like it to be displayed on vPayments	YES
amount	URL Encoded String	The total transaction amount in string format to two decimal places.	YES
storefrontid	URL Encoded String	The merchant vPayments ID, this is used to identify the merchant and pull the secret key to check the hash	YES
additionalinfo	URL Encoded String	Additional display info vPayments will display to the client on the transaction page. For example a list of products. This must be preformatted test, you cannot embed html in this field.	YES
status	URL Encoded String	Must be set to “Message”	YES
hash	Uppercase String	*[1]* The hash for this message used to validate the source as the merchant.	

[1] – Hash generation is explained later in this document.

Example of a transaction initialization message:

confirmurl=http%3a%2f%2fwww.mysite.me%2fconfirm&returnurl=http%3a%2f%2fwww.mysite.me%2freturn&reference=12345&amount=10.00&storefrontid=1&additionalinfo=Some+other+info&status=Message&Hash=71FA7362C603E3265E8DAE315703BC64CF48874E3D736D63DFDBB9CA61533AEF715266FAE84D9D116598A4113278F651E00B5D6D0A443DC2380FA3D13AA8A743

vPayments will respond with a String formatted as if it were an HTTP POST i.e. an equals (=) sign between fields and fields separated by an ampersand (&).

A successful response will contain the following fields:

Successful Response Fields

FIELD NAME	FIELD TYPE	FIELD DESCRIPTION	MANDATORY
status	URL Encoded String	On a successful transaction initialization, status will be set to “Ok”	YES
processurl	URL Encoded String	The URL the clients web browser must be redirect to process the payment on vPayments	YES
checkurl	URL Encoded String	The URL the merchant site can poll to check for the results of the transaction	YES
hash	URL Encoded String	The total transaction amount in string format to two decimal places.	YES

Example of a successful initialization message:

Status=Ok&ProcessUrl=https%3a%2f%2fsecure.zss.co.zw%2fvpayments%2fInterface%2fMakePayment%2f%3fguid%3d4c538282-469e-4238-abc5-47a0fdddbe87&CheckUrl=https%3a%2f%2fsecure.zss.co.zw%2fvpayments%2fInterface%2fCheckPayment%2f%3fguid%3d4c538282-469e-4238-abc547a0fdddbe87&Hash=8EC843AA03005AEFEB02E175E0EC818E6A1B2ACAADA97B613E20D643A14BAFFE30D58D3EC528AF9063CF63FAFF0F2AAB5E941B1FFEB7EB9E09E96045DFC2C373

A failed response will contain the following fields:

Failed Response Fields

FIELD NAME	FIELD TYPE	FIELD DESCRIPTION	MANDATORY
status	URL Encoded String	On a failed transaction initialization, status will be set to "Error"	YES
error	URL Encoded String	A text description of the error, once the merchant account has been set to live it will only reply with "Invalid Payment Initialization". In testing it will return more verbose errors.	YES

Example of Failed Transaction:

Status=Error&Error=Invalid+amount+field

Checking Transaction Status

The merchant site can update their pending transaction status in two ways, it is advisable you use both for extra security and redundancy. vPayments will post a status update message to the confirm URL provided by the merchant at payments initialization.

Alternatively the merchant server can poll the check URL provided by vPayments to check the status of their transaction. While the merchant server can check a transaction status at any time, this should only be at following times however

- The user has been redirected back to the merchant site. Regardless of whether the merchant server has been notified of a transaction status change or not it advisable to poll for the latest transaction status to confirm it is valid.
- Before any unpaid transaction is deleted from the merchant server. Before removing a pending transaction from the merchant site it is vital you poll to confirm its status, it is possible that the transaction has been paid but the update was never received by the merchant server. If the transaction is still pending on vPayments do not delete it from the merchant server as it is possible it will still be processed in future.
- When a user wants to manually verify a transaction from the merchant site.

To poll for the current transaction status the merchant server needs to make an empty HTTP POST to check the URL provided by vPayments in the initiation response message. vPayments will reply with a message in the same format as before. As example check URL might be

https://secure.zss.co.zw/vpayments/Interface/CheckPayment/?guid=bf885024-3087-48ca-9cc9-eef251faa741

Whether vPayments is replying to a poll or posting the results the merchant server, the message will contain the following fields.

FIELD NAME	FIELD TYPE	FIELD DESCRIPTION	MANDATORY
reference	URL Encoded String	The vPayments internal reference	YES
amount	URL Encoded String	The total transaction amount in String Format to Two Decimal Places	YES
status	URL Encoded String	The status of the transaction, the possible return values are: 🟩 Cancelled 🟩 Created but not Paid 🟩 Awaiting Redirect 🟩 Paid 🟩 Failed	YES
processdate	URL Encoded String	The date and time when the transaction was processed in universal sortable format ("'yyyy'-'MM'-'dd'-'HH':'mm':'ss'Z"). If the transaction is not yet processed the field will be present but empty.	YES
bankreference	URL Encoded String	The Financial Institutions reference for the transaction. If the transaction has not been paid successfully the filed will be present but empty	Y
checkurl	URL Encoded String	The URL the Merchant site can poll to check for the results of the transaction	Y
hash	Uppercase String	The hash for this message used to validate the source as vPayments. Hash generation is explained later in this document.	Y

Example of a transaction status update:

Reference=570&Amount=0.15&Status=Paid&ProcessDate=2012-08-21+14%3a53%3a05Z&BankReference=000233-00000001&CheckUrl=https%3a%2f%2fsecure.zss.co.zw%2fvpayments%2fInterface%2fCheckPayment%2f%3fguid%3dbf885024-3087-48ca-9cc9-eef251faa741&Hash=673932DCDA36547E3D8C4BD6749E81EDE8B79F0E4C95442545F6A01F9995991EDAF4655CD6AE63D698AE2DCADA26CB98C64FB4B9FE8A61E4E5753C4751D2EEE7

An invalid check URL GUID will reply with an error message as explained above

Unprocessed Transactions

In the event a transaction has not been complete by the customer vPayments will cancel any transaction from that day at midnight that night and announce the failure to the merchant server.

If by the next morning the merchant server still has transaction from the previous day with a pending status the merchant must poll vPayments for the transaction status as network issue have probably resulted in a failure to update the merchant server overnight? It is also possible some of these transactions may have been successful but neither vPayments nor the customer could get back to the site to confirm. As such a merchant should not automatically cancel old payments but keep them open until they have been able to poll for a status update.

Why it is done like this:

It is important that a merchant knows the status of pending transactions, as network problem could have interfered with status updates. As such it is mandated that a transaction can only be processed on the same that it is created. This means the merchant site can confirm the status of any old payments and perform house keeping the side the next day.

The Hash Variable

Every transaction message, except for errors, is accompanied by a hash field. The value is a not easily reversible hash of all data and the Merchant Secret Key, vPayments will not process any message that does not include the a valid hash.

Why it is done like this:

The hash serves two very important functions.

- First, too ensure data integrity; if the value of any of the fields in the message is changed then the hash will be invalid.
- Second, is to validate the source integrity, As the hash also include the Merchant Secret Key, it is possible for the merchant and vPayments to confirm each other’s identity as they are the only ones that know the secret key.

The process for generating the hash is as follows:

- Concatenate all the message values (but not the field names), before URL encoding and in the same order as they appear in the message.
- Append the Merchant Secret Key.
- UTF8 encode to generate byte array. (If applicable to your development language)
- SHA512 Hash
- Output byte array in Uppercase Hexadecimal
- Add back to the message as the Hash variable.

Under the merchant login on vPayments there is a test centre which the developer can use to enter dummy data and generate the appropriate hash to check their implementation.

Example Message with Hash

(Hash generated with Storefront ID 2 and Secret Key 700b76db-473b-4945-93bf-46622f2c1671)

```
POST /Interface/InitiateTransaction HTTP/1.1
Method: POST
Content-Type: application/x-www-form-urlencoded
User-Agent: Java/1.6.0_29
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: keep-alive
Content-Length: 387

confirmurl=http%3A%2F%2F10.91.20.174%3A9000%2FApplication%2FconfirmReturn&returnurl=http%3A%2F
%2F10.91.20.174%3A9000%2FApplication%2FpaymentReturn&reference=From+PlayFramework&amount=10.0&
storefrontid=2&additionalinfo=Hopefully+this+works&status=MesSagE&hash=0463A23BED35B4E9155EC3A
5DEFCDB7F3E214CC2D05057B89099086DC6F69197768461404676D0414791995DD2DFE16870F69ACEB99B9BDBDA91C
831A71B656E



HTTP/1.1 200 OK
Date: Thu, 08 Dec 2011 09:18:54 GMT
X-AspNet-Version: 4.0.30319
X-AspNetMvc-Version: 3.0
Cache-Control: private
Content-Type: text/html; charset=utf-8
Content-Length: 400
Connection: Close

Status=Ok&ProcessUrl=https%3a%2f%2fsecure.zss.co.zw%2fvpayments%2fInterface%2fMakePayment%2f%3
fguid%3de8c86898-3f9a-4d2b-9608-f31663d5119d&CheckUrl=https%3a%2f%2fsecure.zss.co.zw%2fvpaymen
ts%2fInterface%2fCheckPayment%2f%3fguid%3de8c86898-3f9a-4d2b-9608-f31663d5119d&Hash=F8363F8F2B
49B4B3E870433C5357A15DDC02C510BB5C9D462BCF178D2861E410A06050C7DA0BEEE08AAC3C23574783695EBE304D
EB236C45026F63B53FF3CBAB
```

On behalf of ZimSwitch Shared Services, we hope you found this document informative.