Vpayments API and Developer Documentation

Vpayments API and Developer Documentation

9/12/2012

Matthew Hood

Document Status	**Final**
Copyright	ZimSwitch Shared Services

Table of Contents

1. SCOPE

This document defines the API (Application Programming Interface) between a third party merchant and the Zimswitch Shared Services (ZSS).

2. Overview

Vpayments is an online payments system provided by Zimswitch to allow card not present payment from the customer's bank account directly to the merchant's bank account.

The system currently supports payments between Zimbabwean banks connected to the Zimswitch platform. At present this does not include VISA, Mastercard or other payment engines, even if the bank offers them.

This document is meant to outline the process of a merchant interfacing with Vpayments regardless of bank or payment mechanism.

3. API

Currently the only supported API interface is the HTTP POST protocols.

As more API protocols are exposed by the Vpayments application they will be added to this document.

3.1 HTTP API

The HTTP API exposes URLs that will accept and announce updates with HTTP POST requests.

All messages form the merchant and successful responses from Vpayments must include a valid hash as explained later.

3.1.1 Merchant Setup

Before a merchant can integrate with Vpayments they need to sign up for the service with their connected Financial Intuition. Once the Financial Institution has complete their setup the merchant will be emailed with a URL to use to setup their first password and their account will be in Test mode.

Test mode means all integration is possible but no funds are transferred and every transaction is automatically failed when the customer clicks make payment. Once the Financial Institution and merchant are happy with their integration they can request that their account gets changed to Live and transactions should then process in full.

After first login the merchant will be able to get their Merchant ID and Merchant Secret Key. It is strongly recommended that the merchant regenerates this key when moving from Test to Live and on a period basis after that. The Merchant Secret Key serves as the unique way of verifying a response is from a merchant and should be kept confidential at all times.

3.1.2 Process Flow

The payment process is multi step and requires interaction between the customer' browser, the merchant server and the Vpayments server. The following flow is for a complete transaction, if the user closes their browser, cancels a payment or the transaction fails at any point the flow may be incomplete.

1. Customer select pay via Vpayments on the merchant website.
2. The merchant server initiates a transaction with Vpayments providing a confirm URL and a return URL for this transaction on their site.
3. Vpayments replies with a process URL and a check URL on Vpayments.
4. The merchant site redirects the customer's browser to the process URL provided in step three.
5. The customer makes payment on Vpayments.
6. Vpayments announces the result of the transaction to the merchant server using the confirm URL provided in step two.
7. Vpayments redirects the client's browser back to the merchant site using the return URL provided in step two.
8. The merchant server polls the check URL provided in step three to confirm the payment status.

Why is it done like this: Because each merchant site may not have HTTPS and the client can farm hash information all correspondence between the merchant site and Vpayments must be done directly. Initiation requests and status checks should not be made through the client browser in anyway. The client should only ever see the redirect URLs which should contain no information about the payment besides IDs and identifying GUIDs.

Why is it 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 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 verifies any hash appended to a message by Vpayments, without this check it is possible that a fake reply has been delivered to the merchant.

3.1.3 Initiate Transaction

The first message sent from the merchant server to Vpayments is to create the transaction and get the subsequent URLs used 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

FIELD NAME	FIELD TYPE	FIELD DESCRIPTION	MANDATORY
confirmurl	URL encoded string	The URL the Vpayments server can use to announce transaction results to the merchant site.	Y
returnurl	URL encoded string	The URL on the merchant site the client will be redirected to after processing the transaction.	Y
reference	URL encoded string	The merchant reference as you would like it to be displayed on Vpayments.	Y
amount	URL encoded string	The total transaction amount in string format to two decimal places.	Y
storefrontid	URL encoded string	The merchant Vpayments ID, this is used to identify the merchant and pull their secret key to check the hash.	Y
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 text, you cannot embed html in this field.	Y
status	URL encoded string	Must be set to "Message"	Y
hash	Uppercase string	The hash for this message used to validate the source as the merchant. Hash generation is explained later in this document.	Y

Example of transaction initialisation 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

The Vpayments server 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

FIELD NAME	FIELD TYPE	FIELD DESCRIPTION	MANDATORY
status	URL encoded string	On a successful transaction initialisation status will be set to "Ok"	Y
processurl	URL encoded string	The URL the client's web browser must be redirected to process the payment on Vpayments	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 successful transaction initialisation 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-abc5-47a0fdddbe87&Hash=8EC843AA03005AEFEB02E175E0EC818E6A1B2ACAADA97B613E20D643A14BAFFE30D58D3EC528AF9063CF63FAFF0F2AAB5E941B1FFEB7EB9E09E96045DFC2C373

A failed response will contain the follow fields

FIELD NAME	FIELD TYPE	FIELD DESCRIPTION	MANDATORY
status	URL encoded string	On a failed transaction initialisation status will be set to "Error"	Y
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 detailed errors.	Y

Example of failed transaction:

Status=Error&Error=Invalid+amount+field.

3.1.4 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.

The Vpayments server will post a status update message to the confirm URL provided by the merchant at payment initialisation.

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 whenever they want this should only be done at the following times.

- 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 is 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 the future.
- When a user wants to manually verify a transaction status from the merchant site.

To poll for the current transaction status the merchant server needs to make an empty HTTP POST to the check URL provided by the Vpayments server in the initialisation response message. Vpayments will reply with a message in the same format as before. An 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 to the server it the message will contain the following fields.

FIELD NAME	FIELD TYPE	FIELD DESCRIPTION	MANDATORY
reference	URL encoded string	The Vpayments internal reference number.	Y
amount	URL encoded string	The total transaction amount in string format to two decimal places.	Y
status	URL encoded string	The status of the transaction, the possible options are:	Y

		• Cancelled • Created but not Paid • Awaiting Redirect • Paid • Failed	
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.	Y
bankreference	URL encoded string	The Financial Institution's reference for the transaction. If the transaction has not been paid successfully the field 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.

3.1.5 Unprocessed Transactions

In the event a transaction has not been completed 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 transactions from the previous day with a pending status the merchant must poll the Vpayments server for the transaction status as network issues 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 redirect could get back to the site to confirm. As such a merchant should not automatically cancel all old payment but keep them open until they have been able to poll for a status update.

Why is it done like this: It is important that a merchant knows the status of pending transactions, as network problems could have interfered with status updates. As such it is mandated that a

transaction can only be processed on the same day it is created. This means the merchant site can confirm the status on any old payment and perform house keeping their side the next day.

3.1.6 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 the data and the Merchant Secret Key. Vpayments will not process any message that does not include a valid hash value. It is also vital that the merchant check all messages coming from Vpayments to ensure that it contains a valid hash.

Why is it done like this: The hash serves two very important functions. The first is to ensure data integrity; if the value of any of the fields in the message has changed then the hash will be invalid. The second reason is to validate the source of the message. As the hash also includes 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 should know the secret key.

The process for generating the hash is as follow:

1. Concatenate all message values (but not field name), before URL encoding and in the same order they appear in the message.
2. Append merchant secret key.
3. UTF8 encode to generate byte array. (If applicable to your development language)
4. SHA512 hash.
5. Output byte array in uppercase hex.
6. Add back to message.

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.