`

webAPI – API Developer Guide

for API

Version 2.2.9

Table of Contents

Version control

Rev No:	Changes	Date
0	Initial draft document developed and submitted for review	2015/02/17
1	API change triggered expansion to username / password inclusion for authentication & authorisation.	2015/03/21
2	Additional functions added to API (getFinancialTransactions, getPurchaseHistory).	2015/04/02
3	• Converted to AES encryption @ http://www.movable-type.co.uk/scripts/aes-php.html • Final release of class.GenieAPI_v1.php	2015/04/05
4	• Removed PHP centric information. Preparation for moving to different technology stack. • GeniEngine release v6.0.0 new functionality. • API inheritance expanded for multiple global partitions. • Namespace foundation code implemented. • Additional return fields added. • Switched from QA to DEMO for testing environment. • Added getCategories • GenieAPI_v2 ○ TIME_ZONE & LOCAL_SYSTEM_TIME added ○ Removed old, internally modified AES encryption classes. • All related, client-requested CRS tickets dealt with.	2015/09/20
5	• Upgrade VVS to version 6.0.0. • API (publish) complete rewrite.	2015/10/02
6	Removed non-useful functionality	2016/01/14
7	Launched API v2	2016/02/02
8	Completed API documentation	2016/02/08
9	• Converted to VVS 6.3.0 • Removed all references to USD as all values are now in the account / distributor currency	2016/09/04
10	• Updated to VVS 7.0.5. • Updated protocol to 2.2.0. • Bumped minimum version for PHP to 7.x. • Added IV / OpenSSL encryption. • Added HASH_STUB to CONTENT requirements. • Added field(s) to all responses: ○ ERROR_DESCRIPTION • Added / documented additional calls ○ get_ProductAvailability ○ get_AllProductAvailability ○ get_ErrorDescription	2021-04-17 to 2021-05-26

	• Code refactoring and optimization on several libraries and server environment. • New libraries for: ○ GoLang ○ Ruby on Rails ○ Java ○ C# ○ Python (3x) ○ Perl (updated) • Resolved server-side issues with ENCRYPT_RESPONSE. • Further in-library documentation on some libraries. • Resolved several server-side issues. • Added new field in all responses. • Implemented 7 day call data retention, before dropping the API call log to cold storage. • Added a library section and links to git repositories to document. • Preparation to deprecate AES-CTR encryption in favour of OpenSSL AES-CBC-256-OPENSSL.	
11	• Added NodeJS library (Vincent Seaborne)	2021-07-29
12	• Added voucher output format (Vincent Seaborne)	2021-08-16
13	• Added get_ProductFormats (Vincent Seaborne) • Deprecated get_ErrorDescription (Vincent Seaborne)	2021-08-25
14	• Retrieve last sale (Vincent Seaborne)	2021-09-07
15	• Added UAT section with minimum required calls for UAT checks to succeed. • Added Business Process: Inbound API Integration to download folder. • Added Issuing of Production credentials section • Added clarification regarding product differences between demo and production environments. • Added minimum requirements for production. • Added Updated documentation location	2021-10-06
16	• Implemented EVD Sales Manager, new maximum voucher amount per call = 150. [ND-232] • Added additional return fields to get_Vouchers ○ FINANCIAL_ID ○ CLIENT_REFERENCE_ID ○ CLIENT_REMAINING_BALANCE ○ OEM_PRODUCT_EAN	2021-11-16
17	• Included get_MaxAllowedVouchersPerCall. [GSD-711] • Fix date/time for GetByDateRange calls. [BUGS-288] • REFERENCE_ID *can* now be unique, setting can be applied on retailer. If enabled, throws error if same ID is passed. [CON-543] • To retrieve previous transaction by client reference, use get_SalesTransaction_ByReferenceID. Unique Reference must be enabled for retailer. [CON-567] • get_LastSale to retrieve last sale made by client. Only advisable for single threaded purchase queues.	2022-02-15

18	- get_ProductInfo to retrieve product information about the product passed by ID. [GSD-843]	2022-06-21
19	- get_AccountBalance to show information for RFM - get_Vouchers to show information for RFM	2023-03-06
20	- get_MaxAllowedVouchersPerCall - get_SalesTransaction_ByTransactionId - get_SalesTransaction_ByReferenceId	2024-05-22
21	- Update to all PHP repos. - Add new functionality in repos. - Remove unused code in repos.	
22	- Removed get_SalesTransaction_ByTransactionId duplicate definition	2024-09-26

Related documents

No	Document name	Description
	Business Process: Inbound API Integration	Describes the process for new, inbound integrations.

1. Documentation location

The API documentation is constantly updated as new requirements or features are added.

The only authorized location for this documentation is:

https://downloads.paythem.net/05_API_Libraries/

If you receive this documentation via email or other means, please immediately check the above link to confirm you have received the latest.

2. PayThem VVS System overview

The PayThem VVS system is a JSON based, REST-like API service. It enables 3rd parties to consume web services provided by PayThem via the VVS Platform.

3rd party clients can connect to PayThem's Virtual Voucher System (VVS) to expand their current offerings or build new applications via direct integration with PayThem's available warehouse of vouchers. Through this, the client can retain their own identity and branding while expanding their own product offerings within their own application environment.

2.1.Security and encryption

All communication with the VVS system has multiple layers of encryption, authentication and verification. To secure all transactions, all communications are via a SSL secured URL and must be used with the HTTP POST method (unless the documentation clearly specifies otherwise).

When implementing the API:

- Clients must ensure, when communicating with the VVS system, that the PayThem SSL certificate is valid.
- Only parameters of HTTP POST will be processed. GET / DELETE / PUT are ignored.
- VVS API supports the following encryption protocols:
  - AES-CTR 256.
  - OpenSSL AES-CBC-256-OPENSSL
- For an extra layer of security, we require each API consuming client to provide us with the public (internet) static IP address of the source server from which API calls will be made. Any calls from any other IP addresses will ignored and could lead the account to be blocked and the IP address blacklisted. DDNS and DHCP IP addresses will not be accepted.
- The client's server's time zone and a timestamp in format "yyyy-mm-dd H:i:s" format must be inserted into each query to prevent replay attacks. Please ensure your time zone and time is set correctly, preferably by a NTP service to minimize risk. A maximum of 30 second deviation from our servers will be tolerated before errors are raised.
- All character encoding must be UTF-8 based.
- Each client takes full responsibility for the safeguarding of their encryption keys. Production encryption keys will be shared in two parts, to separate client staff.

- Multiple API accounts can be created per client, each with their own encryption keys, static IP addresses and authentication credentials. Each API account will use the customer’s global account balance for purchases.
- HTTP compression is enabled on our servers and is preferred / recommended.

2.2.Environments

PayThem provides two separate environments for client access.

<u>Products and their related ProductID differs between production and demo environments.</u>

Environment	Purpose
Demo	• For testing connectivity between client system and VVS and testing functionality of VVS. • All information is similar to Production environment, but all vouchers issued are fake. • Product mapping and product ID differ from production. • Not all products are available on Demo. • A balance is loaded onto the accounts for testing. • No static IP required. • Max vouchers per call can be defined, but defaults to 20.
Production	• Production environment where all data is real time and all vouchers are real vouchers. • Each voucher has a real monetary value and account balances reflect real values. • Client must (preferably) provide a static, public IP address for additional security. • Client to provide an estimated number of calls to be performed per hour. • Client to provide an estimated number of vouchers to be retrieved per call. If no information is provided, vouchers returned will be capped at 20.

2.3.Environment endpoints

Environment	URL Endpoint
Demo	https://vvsdemo.paythem.net/API/2824/
Production	https://vvs.paythem.net/API/2824/

When using one of our provided libraries, the library will receive a parameter that defines which environment the current call will use.

2.4.Accounts and users

During client subscription process, each client is provided with an account. Within this account, different API users are created, allowing for different application environments to be created by the client.

For example: API00001 can be used on client server SERVER01 and API00002 can be used on SERVER02.

During API calls, each call will require a username and password to be included in the encrypted "CONTENT" parameter, as described below.

Each API user can have different IP whitelisting and max voucher returned requirements.

2.5.Public & Private tokens

Each environment has its own, unique public key, private key, username and password combinations.

It is crucial to note that the Demo environment's information will not work on Production environment and repeated posting with incorrect details will lead to the account being locked out and the source IP blacklisted.

During an account's API user creation process, PayThem will provide the client with:

- Public key – Passed unencrypted with each query, base64 encoded.
- Private key – used to encrypt JSON parameters before posting to API server.
- Username – encrypted into each post.
- Password – encrypted into each post.

IMPORTANT: It is the client's responsibility to keep all public, private, username and password details secure and hidden from your end users and non-critical staff. If your credentials are compromised (knowingly or unknowingly) PayThem will not be held liable for any damages.

2.6.Libraries and Examples

PayThem provides libraries and example implementation for various languages.

Language	Type	Notes	git Repository
PHP 7.x	Classes	Non-composer, class-based for PHP 7.0 to PHP 7.2.	https://bitbucket.org/paythem/php7.x.git
PHP 7.3	Library	Composer, library-based for PHP 7.3 to PHP 7.4. BREAKING CHANGES.	https://bitbucket.org/paythem/inbound_api_php73.git
PHP 7.x	Library	Composer, library-based PHP 8.0 and above. BREAKING CHANGES.	https://bitbucket.org/paythem/inbound_api_php8.git
Perl	Example		https://bitbucket.org/paythem/perl
Python 3	Library		https://bitbucket.org/paythem/python3
Java	Library		https://bitbucket.org/paythem/java
RoR	Library		https://bitbucket.org/paythem/ruby
C#	Library		https://bitbucket.org/paythem/cscharp
GoLang	Library		https://bitbucket.org/paythem/golang
NodeJS	Library		https://bitbucket.org/paythem/nodejs

If a library or example is not available for your environment, please contact PayThem technical support through your distributor representative.

3. API definition

Relates to the creation of a valid call to the PayThem API service.

3.1.Outbound call headers

- X-Public-Key:
  PUBLIC_KEY as provided by PayThem.
- X-Hash:
  HMAC Hash generated from the pre-encrypted, JSON encoded string using sha256 and the PRIVATE_KEY as provided by PayThem.

3.2.Outbound call Parameter explanation

Parameter required:

- M = Mandatory
- O = Optional

Auto? = Auto-inserted by libraries.

Field	Req.	Auto?	Description
ENCRYPT_RESPONSE	M	Y	[Boolean / true] Enable content encryption of response sent from PayThem API service. Response from server will be encrypted with the PRIVATE_KEY of the client.
USERNAME	M	Y	PayThem supplied username.
PASSWORD	M	Y	Paythem supplied password.
PUBLIC_KEY	M	Y	PayThem supplied public key.
API_VERSION	O	Y	[string] The version of the protocol that is being used.
CLIENT_VERSION	O	Y	[string] The version of the client library that is being used.
CLIENT_LANGUAGE	O	Y	[string] The client language being used.
CLIENT_LANGUAGE_VERSION	O	Y	[string] The version of the client language that is being used.
SERVER_URI	M	Y	[string] The environment URL as defined in 2.3.
SERVER_TIMEZONE	M	Y *	[string] Default is set on the PayThem server for each API user. Can be overridden by setting the field in the POST string. Format of: “Region/CityOrCountry”, e.g., “Asia/Qatar”. Please request the correct TZ specifications from us if you have any doubts. Certain libraries will auto-populate this value.
SERVER_TIMESTAMP	M	Y	[string] Local server time for the time zone the server is in. If this is set, TIME_ZONE also needs to be set. Timestamp must be in “CCYY-MM-DD HH:MM:SS” format.

HASH_STUB	M	Y	[string] Randomly generated alpha-numeric string of minimum 10 characters.
FUNCTION	M	N	[string] The requested function, as defined in this document. Using latest libraries, this is automatically inserted.
PARAMETERS	O	N	[array] Default empty array. Named parameter key/value of the function. Documentation per function below. Using latest libraries, this is automatically inserted.

3.3.Type handling

3.3.1. Dates & time

During VVS API posts, dates are converted from and to client's time zone automatically.

IMPORTANT: be sure that your time zone is correctly set and that your server time is not out from international atomic time by more than 29 seconds. Else, determine and pass the time as needed to allow for proper usage.

IMPORTANT: All dates passed from client to server and server to client will be in the format "CCYY-MM-DD HH:MM:SS".

3.3.2. Encoding

All content must be UTF-8.

3.4.Response handling

If no response is returned, there could be a break in communications.

The latest libraries will return an error in the response variable and "ERROR" in the results variable.

The response will contain a JSON string containing the following fields:

- SERVER_TRANSACTION_ID
  The log ID of the client call. This is used to error check with PayThem support in the event of issues. All calls return a unique call ID.
- RESULT
  00000 = No error,
  -00001 = global error, either from calling the server or from server side (unknown error) or else a code representing an error encountered.
- ERROR_DESCRIPTION
  A human-legible error description.
- CONTENT
  A base64 encoded (and optionally encrypted, depending on call parameters) representing a JSON string relevant to the call made.

- In PHP, this will be an associative array.

3.4.1. Fail voucher purchasing

If you are performing get_Vouchers, but received no response, please use get_SalesTransaction_ByReferenceId with your reference ID to check if the transaction has completed and to retrieve the transaction.

3.5.Available API functions summary

API FUNCTION	Description
get_OEMList	Get a list of all the OEM (Original Equipment Manufacturers) in VVS that are assigned to your profile.
get_BrandList	Get a list of all the brands (by OEM).
get_ProductList	Get a list of all products (by brand and OEM).
get_AccountBalance	Get your current account balance.
get_Vouchers	Get vouchers by product. Maximum vouchers that can be requested is currently 20 (default), unless specified and agreed upon otherwise.
get_FinancialTransaction_ByDateRange	Get financial transactions listing for company per date range. Maximum date range is 30 days. No limitations as to how far the history can be retrieved.
get_SalesTransaction_ByDateRange	Get a list of all voucher purchases for your current user. Details include all vouchers and their details related to the transaction. Maximum date range is 15 days.
get_ProductAvailability	Get the current stock availability of a single product.
get_AllProductAvailability	Get the current stock availability of all products.
get_MaxAllowedVouchersPerCall	Return the maximum vouchers a client can purchase per call.
get_ProductInfo	Retrieves a list of all information on a specific product
get_SalesTransaction_ByTransactionId	Returns an array of all voucher sales by the supplied transaction ID
get_SalesTransaction_ByReferenceId	Retrieve all sales transactions by the purchased reference ID
get_ProductFormats	Retrieves a list of all output formats set in the system
get_LastSale	Retrieves the last sale information for the current user

3.6.API call definition

3.6.1. get_OEMList

Description	Get a full list of OEMs in the VVS system.
Parameters	None
Return	Multi-level array, each sub-array contains the following fields.
OEM_ID	The internal VVS ID of the OEM.
OEM_Name	The name of the OEM.
OEM_Website	If recorded on VVS, will contain the website of the OEM.

OEM_ProductCount	The number of active products in VVS for this OEM.

3.6.2. get_BrandList

Description	Get a full list of brands by OEMs in the VVS system.
Parameters	None
Returns	Multi-level array, each sub-array contains the following fields.
OEM_ID	The internal VVS ID of the OEM.
OEM_Name	The name of the OEM.
OEM_BRAND_ID	VVS Brand ID
OEM_BRAND_Name	Brand name.
OEM_MASTER_BRAND_ID	Parent brand ID.
OEM_BRAND_ProductCount	Number of products for this brand.
OEM_BRAND_RedeemInstructions	Redemption instructions for this brand.
OEM_BRAND_PINWording	Wording for the PIN text.
OEM_BRAND_SerialWording	Wording for the Serial text.
OEM_BRAND_Brand_Product_Format_ID	ID of the voucher output format
OEM_BRAND_Brand_Product_Format_Desc	Description of the voucher output format
OEM_BRAND_Brand_Product_Format_Fields	Comma separated list of fields returned when purchasing this voucher or retrieving sales transactions.

3.6.3. get_ProductList

Description	Get a full list of active products in the VVS system by brand by OEM. **Recommended to perform once a day.**
Parameters	None
Returns	Multi-level array, each sub-array contains the following fields.
OEM_ID	The internal VVS ID of the OEM.
OEM_Name	The name of the OEM.
OEM_BRAND_ID	VVS brand ID.
OEM_BRAND_Name	VVS Brand name.
OEM_BRAND_Brand_Product_Format_ID	ID of the voucher output format
OEM_BRAND_Brand_Product_Format_Desc	Description of the voucher output format
OEM_BRAND_Brand_Product_Format_Fields	Comma separated list of fields returned when purchasing this voucher or retrieving sales transactions.
OEM_PRODUCT_ID	Product ID.
OEM_PRODUCT_Name	Product name.
OEM_PRODUCT_VVSSKU	VVS SKU code.
OEM_PRODUCT_BaseCurrency	Base currency full name, e.g. United States Dollar.
OEM_PRODUCT_BaseCurrencySymbol	Base currency symbol.
OEM_PRODUCT_UnitPrice	Base currency product value / price.

OEM_PRODUCT_SellPrice	The price that PayThem sells the voucher to the API consumer / client.
OEM_PRODUCT_RedemptionInstructions	Redemptions instructions for end client. Not all redemption instructions are available for all products.
OEM_PRODUCT_ImageURL	Represents a URL to retrieve a 150px x 150px icon for the product.
OEM_PRODUCT_Available	The amount of stock available for this product. This will be overwritten by the maximum vouchers per transaction value.

3.6.4. get_AccountBalance

Description	Get the company's current account balance. **Balance is returned on each sales transaction. Recommended to perform every 15 minutes.**
Parameters	None
Returns	Array
RFM_ID	ID of RFM type.
RFM_TYPE	Description of current RFM type.
RFM_CONSIGNMENT	*Only on post paid RFM. Consignment amount.
RFM_ACTIVE_BALANCE	*Only on post paid RFM. Current balance before consignment is applied.
RESELLER_TimeOfRequest	Date/Time of request in yyyy-mm-dd hh:mm:ss format. Converted to client time as per timezone of request parameters.
RESELLER_Currency	Currency of the wallet.
RESELLER_Balance	The current balance of the company.

3.6.5. get_ProductAvailability

Description	Get a specific product's availability. **Recommended to perform before each sale with selling product's ID.**
Parameters	Array
PRODUCT_ID	The VVS product ID for which the voucher is to be issued.
Returns	Product availability: integer. This will be overwritten by the maximum vouchers per transaction value.

3.6.6. get_AllProductAvailability

Description	Get all products' availability. **Recommended to perform every 15 minutes.**
Parameters	None
Returns	Array
OEM_PRODUCT_ID	Product ID.
OEM_PRODUCT_Name	Product name.

OEM_PRODUCT_Available	Number of vouchers available for product. This will be overwritten by the maximum vouchers per transaction value.

3.6.7. get_Vouchers

Description	Get a single or multiple vouchers from VVS. It generates a transaction on VVS which can later be queried.
Parameters	Array
PRODUCT_ID	The VVS product ID for which the voucher is to be issued.
QUANTITY	Number of vouchers you would like to purchase during this transaction. Maximum 20, unless otherwise arranged. **If the requested number of vouchers is not available, only the available amount will be dispensed.**
REFERENCE_ID	A transaction reference to be attached to this transaction for reconciliation purposes.
Returns	Array with embedded multi-level sub-array.
RFM_ID	ID of the RFM type.
RFM_TYPE	Description of the RFM type.
RFM_CONSIGNMENT	*Only on post paid RFM. Consignment amount.
RFM_ACTIVE_BALANCE	*Only on post paid RFM. Current balance before consignment is applied.
TRANSACTION_ID	The VVS ID of the transaction.
TRANSACTION_VOUCHER_QUANTITY	Quantity of vouchers returned for this transaction.
TRANSACTION_DATE	Date and time of the transaction. Converted to client time as per timezone of request parameters.
TRANSACTION_VALUE	The total value of the transaction.
FINANCIAL_ID	The financial record ID. Can be linked to a financial transaction.
CLIENT_REFERENCE_ID	The return of the REFERENCE_ID as passed by client.
CLIENT_REMAINING_BALANCE	New balance for client after transaction.
OEM_ID	OEM name.
OEM_Name	Name of the OEM
OEM_BRAND_ID	VVS brand ID.
OEM_BRAND_Name	Brand name.
OEM_BRAND_Brand_Product_Format_ID	PFID = ID of the voucher output format
OEM_BRAND_Brand_Product_Format_Desc	Description of the voucher output format
OEM_BRAND_Brand_Product_Format_Fields	Comma separated list of fields returned when purchasing this voucher or retrieving sales transactions.
OEM_PRODUCT_ID	VVS product ID.
OEM_PRODUCT_Name	VVS product name.
OEM_PRODUCT_SellPrice	Sell price to company.
OEM_PRODUCT_RedemptionInstructions	Redemption instructions for vouchers.

OEM_PRODUCT_VVSSKU	VVS SKU code.
OEM_PRODUCT_EAN	EAN as registered for this retailer.
VOUCHERS - Sub-array	Multi-level sub-array of vouchers. Empty if no vouchers are available.
OEM_VOUCHER_ID	VVS internal voucher ID.
OEM_VOUCHER_SALES_ID	Sales reference for voucher. Separate from SERVER_TRANSACTION_ID and TRANSACTION_ID.
OEM_VOUCHER_EXPIRATION_DATE	Expiration date of voucher.
OEM_VOUCHER_SERIAL	Serial number
OEM_VOUCHER_PIN	PIN (Returned if PFID = 1 or 4)
OEM_VOUCHER_USERNAME	Username (Returned if PFID = 2)
OEM_VOUCHER_PASSWORD	Password (Returned if PFID = 2 or 3)
OEM_VOUCHER_CARD_NUMBER	Card number (Returned if PFID = 3 or 4)

3.6.8. get_LastSale

Description	Get all vouchers purchased on the last sale
Parameters	None
Returns	Array
TRANSACTION_ID	The VVS ID of the transaction.
TRANSACTION_VOUCHER_QUANTITY	Quantity of vouchers returned for this transaction.
TRANSACTION_DATE	Date and time of the transaction. Converted to client time as per timezone of request parameters.
TRANSACTION_VALUE	The total value of the transaction.
OEM_ID	OEM name.
OEM_Name	Name of the OEM
OEM_BRAND_ID	VVS brand ID.
OEM_BRAND_Name	Brand name.
OEM_BRAND_Brand_Product_Format_ID	PFID = ID of the voucher output format
OEM_BRAND_Brand_Product_Format_Desc	Description of the voucher output format
OEM_BRAND_Brand_Product_Format_Fields	Comma separated list of fields returned when purchasing this voucher or retrieving sales transactions.
OEM_PRODUCT_ID	VVS product ID.
OEM_PRODUCT_Name	VVS product name.
OEM_PRODUCT_SellPrice	Sell price to company.
OEM_PRODUCT_RedemptionInstructions	Redemption instructions for vouchers.
OEM_PRODUCT_VVSSKU	VVS SKU code.
VOUCHERS - Sub-array	Multi-level sub-array of vouchers for sale.
OEM_VOUCHER_ID	VVS internal voucher ID.
OEM_VOUCHER_EXPIRATION_DATE	Expiration date of voucher.
OEM_VOUCHER_SERIAL	Serial number
OEM_VOUCHER_PIN	PIN (Returned if PFID = 1 or 4)
OEM_VOUCHER_USERNAME	Username (Returned if PFID = 2)
OEM_VOUCHER_PASSWORD	Password (Returned if PFID = 2 or 3)
OEM_VOUCHER_CARD_NUMBER	Card number (Returned if PFID = 3 or 4)

3.6.9. get_SalesTransaction_ByDateRange

Description	Get all vouchers purchased during a specific period for API user.
Parameters	Array
FROM_DATE	The start date, including time (2020-01-01 13:56:00)
TO_DATE	The end date, including time (2020-01-01 13:56:00) May not be more than 30 days from start date.
Returns	Array
TRANSACTION_ID	The VVS ID of the sales transaction. Not to be confused with SERVER_TRANSACTION_ID.
TRANSACTION_VOUCHER_QUANTITY	Quantity of vouchers in the transaction.
TRANSACTION_DATE	Date and time of the transaction. Converted to client time as per timezone of request parameters.
TRANSACTION_USD_VALUE	Ignore. To be deprecated.
TRANSACTION_Reference	The reference as supplied as parameter during original, originating get_Vouchers call.
TRANSACTION_CurrentStatus	The current status of the transaction. • Pending • Revoked - Insufficient funds in branch • Revoked - Insufficient funds in company • Completed successfully
OEM_ID	OEM name.
OEM_Name	Name of the OEM
OEM_BRAND_ID	VVS brand ID.
OEM_BRAND_Name	Brand name.
OEM_BRAND_Brand_Product_Format_ID	PFID = ID of the voucher output format
OEM_BRAND_Brand_Product_Format_Desc	Description of the voucher output format
OEM_BRAND_Brand_Product_Format_Fields	Comma separated list of fields returned when purchasing this voucher or retrieving sales transactions.
OEM_PRODUCT_ID	VVS product ID.
OEM_PRODUCT_Name	VVS product name.
OEM_PRODUCT_SellPrice	Sell price to company.
OEM_PRODUCT_RedemptionInstructions	Redemption instructions for vouchers.
OEM_VOUCHER_TransactionStatus	The status of this voucher. • Sold • Reversed • Complaint
OEM_VOUCHER_ID	VVS internal voucher ID.
OEM_VOUCHER_SALES_ID	Sales reference for voucher. Separate from SERVER_TRANSACTION_ID and TRANSACTION_ID.
OEM_VOUCHER_EXPIRATION_DATE	Expiration date of voucher.
OEM_VOUCHER_SERIAL	Serial number
OEM_VOUCHER_PIN	PIN (Returned if PFID = 1 or 4)

OEM_VOUCHER_USERNAME	Username (Returned if PFID = 2)
OEM_VOUCHER_PASSWORD	Password (Returned if PFID = 2 or 3)
OEM_VOUCHER_CARD_NUMBER	Card number (Returned if PFID = 3 or 4)

3.6.10. get_FinancialTransaction_ByDateRange

Description	Get all financial transactions (deposits, credits, reversals, purchases, etc) for company.
Parameters	Array
FROM_DATE	The start date, including time (2020-01-01 13:56:00)
TO_DATE	The end date, including time (2020-01-01 13:56:00) May not be more than 30 days from start date.
Returns	Multi-level array, each sub-array contains the following fields.
TRANSACTION_ID	The VVS financial transaction ID. Not to be confused with SERVER_TRANSACTION_ID.
TRANSACTION_Type	The type of transaction, e.g. credit, cash, EFT, reversal, voucher reversal, purchases.
TRANSACTION_Value	The value of the transaction. Negative is debit, positive credit.
TRANSACTION_Date	Date of transaction.
TRANSACTION_DateCaptured	Date and time the transaction was completed. Converted to client time as per timezone of request parameters.
TRANSACTION_Reference	Transaction reference / description.
TRANSACTION_CurrencyName	The currency in which the transaction was captured.
TRANSACTION_CurrencySymbol	The currency symbol.
TRANSACTION_AccountBalance	The account balance after the transaction was completed.
OEM_VOUCHER_SALES_ID	If the represented transaction was a purchase, this will include the voucher purchase reference as returned by get_Vouchers.

3.6.11. get_ProductFormats

Description	Retrieves all voucher formats for products
Parameters	None
Returns	Multi-level array, each sub-array contains the following fields.
FORMAT_ID	ID of the output format
FORMAT_DESCRIPTION	A description of fields that will be returned for voucher(s)
FORMAT_FIELDS	Comma separated value of applicable fields for voucher(s)

3.6.12. get_SalesTransaction_ByReferenceId

Description	Retrieves all transactions for a given reference ID

Parameters	Array
REFERENCE_ID	The original reference ID supplied during get_Vouchers
Returns	Multi-level array, each sub-array contains the following fields.
TRANSACTION_ID	The VVS ID of the transaction.
TRANSACTION_VOUCHER_QUANTITY	Quantity of vouchers returned for this transaction.
TRANSACTION_DATE	Date and time of the transaction. Converted to client time as per timezone of request parameters.
TRANSACTION_VALUE	The total value of the transaction.
FINANCIAL_ID	The financial record ID. Can be linked to a financial transaction.
CLIENT_REFERENCE_ID	The return of the REFERENCE_ID as passed by client.
CLIENT_REMAINING_BALANCE	New balance for client after transaction.
OEM_ID	OEM name.
OEM_Name	Name of the OEM
OEM_BRAND_ID	VVS brand ID.
OEM_BRAND_Name	Brand name.
OEM_BRAND_Brand_Product_Format_ID	PFID = ID of the voucher output format
OEM_BRAND_Brand_Product_Format_Desc	Description of the voucher output format
OEM_BRAND_Brand_Product_Format_Fields	Comma separated list of fields returned when purchasing this voucher or retrieving sales transactions.
OEM_PRODUCT_ID	VVS product ID.
OEM_PRODUCT_Name	VVS product name.
OEM_PRODUCT_SellPrice	Sell price to company.
OEM_PRODUCT_RedemptionInstructions	Redemption instructions for vouchers.
OEM_PRODUCT_VVSSKU	VVS SKU code.
OEM_PRODUCT_EAN	EAN as registered for this retailer.
VOUCHERS - Sub-array	Multi-level sub-array of vouchers. Empty if no vouchers are available.
OEM_VOUCHER_ID	VVS internal voucher ID.
OEM_VOUCHER_SALES_ID	Sales reference for voucher. Separate from SERVER_TRANSACTION_ID and TRANSACTION_ID.
OEM_VOUCHER_EXPIRATION_DATE	Expiration date of voucher.
OEM_VOUCHER_SERIAL	Serial number

3.6.13. get_ProductInfo

Description	Retrieves information about the requested product.
Parameters	Array
PRODUCT_ID	The VVS product ID for which the voucher is to be issued.
Returns	Single level array containing information about the product passed.
OEM_ID	OEM name.
OEM_Name	Name of the OEM
OEM_BRAND_ID	VVS brand ID.

OEM_BRAND_Name	Brand name.
OEM_BRAND_Brand_Product_Format_ID	PFID = ID of the voucher output format.
OEM_BRAND_Brand_Product_Format_Desc	Description of the voucher output format.
OEM_BRAND_Brand_Product_Format_Fields	Comma separated list of fields returned when purchasing this voucher or retrieving sales transactions.
OEM_PRODUCT_ID	VVS product ID.
OEM_PRODUCT_Name	VVS product name.
OEM_PRODUCT_VVSSKU	VVS SKU code.
OEM_PRODUCT_SellPrice	Sell price to company.
OEM_PRODUCT_RedemptionInstructions	Redemption instructions for vouchers.
OEM_PRODUCT_BaseCurrency	Name of the currency of the product value.
OEM_PRODUCT_BaseCurrencySymbol	ISO Symbol of the currency of the product value.
OEM_PRODUCT_UnitPrice	Value in local currency of the product.
OEM_PRODUCT_SellPrice	The price in which the product will be sold at.
OEM_PRODUCT_RedemptionInstructions	Instructions on how to redeem the product.
OEM_PRODUCT_ImageURL	URL to the image of the product.

3.6.14. get_MaxAllowedVouchersPerCall

Description	Retrieve the maximum allowed voucher purchase per call
Parameters	NONE
Returns	Maximum amount.

3.6.15. get_SalesTransaction_ByTransactionId

Description	Retrieves all transactions for a given VVS transaction ID.
Parameters	Array
REFERENCE_ID	The original Sales Transaction ID returned during get_Vouchers
Returns	Multi-level array, each sub-array contains the following fields.
TRANSACTION_ID	The VVS ID of the transaction.
TRANSACTION_VOUCHER_QUANTITY	Quantity of vouchers returned for this transaction.
TRANSACTION_DATE	Date and time of the transaction. Converted to client time as per timezone of request parameters.
TRANSACTION_VALUE	The total value of the transaction.
FINANCIAL_ID	The financial record ID. Can be linked to a financial transaction.
CLIENT_REFERENCE_ID	The return of the REFERENCE_ID as passed by client.
CLIENT_REMAINING_BALANCE	New balance for client after transaction.
OEM_ID	OEM name.
OEM_Name	Name of the OEM
OEM_BRAND_ID	VVS brand ID.
OEM_BRAND_Name	Brand name.
OEM_BRAND_Brand_Product_Format_ID	PFID = ID of the voucher output format
OEM_BRAND_Brand_Product_Format_Desc	Description of the voucher output format

OEM_BRAND_Brand_Product_Format_Fields	Comma separated list of fields returned when purchasing this voucher or retrieving sales transactions.
OEM_PRODUCT_ID	VVS product ID.
OEM_PRODUCT_Name	VVS product name.
OEM_PRODUCT_SellPrice	Sell price to company.
OEM_PRODUCT_RedemptionInstructions	Redemption instructions for vouchers.
OEM_PRODUCT_VVSSKU	VVS SKU code.
OEM_PRODUCT_EAN	EAN as registered for this retailer.
VOUCHERS - Sub-array	Multi-level sub-array of vouchers. Empty if no vouchers are available.
OEM_VOUCHER_ID	VVS internal voucher ID.
OEM_VOUCHER_SALES_ID	Sales reference for voucher. Separate from SERVER_TRANSACTION_ID and TRANSACTION_ID.
OEM_VOUCHER_EXPIRATION_DATE	Expiration date of voucher.
OEM_VOUCHER_SERIAL	Serial number

4. UAT - User Acceptance Testing procedure

Before a client can be issued production credentials, an acceptance test by PayThem will be conducted to assure that all required features have been integrated.

Minimum required calls to proceed to UAT phase:

- get_ProductList
- get_Vouchers
- get_SalesTransactions_ByDateRange
- get_AccountBalance
- get_ProductAvailability
- get_AllProductAvailability

Once all these calls have been confirmed to be processed successfully, production credentials will be issued.

5. Issuing of production credentials

Once UAT is complete, only your designated Distributor administrator can request.

Please contact them directly and they will submit the relevant tickets to the correct department.

Credentials will be shared in two parts:

- Public key / username – via email.
- Private key / password – any agreed, non-email communication method.

For production environment, a static IP address is required. No user will be created unless a static IP is provided.