UNPLUGG GROUP

PASSION MEETS TECHNOLOGY

U-PAS MPESA API Specification

Author: Vusi Mngomezulu

Date: 13 October 2025

Version: 1.3

Table of Contents

1. Introduction

U-PAS mobile device provides a hosted Receipting and Allocation platform designed specifically for receipting payments towards policies, accounts, or other subscription-based products.
U-PAS makes use of HTTP based web calls to access required member and account information hosted in our clients and partner's administration systems.
Our Integration Partners provide a HTTP service that implements a handler processing GET requests and compiling JSON responses in accordance with this specification.
The combination of the U-PAS web requests, and the JSON responses is an implementation of the UIS APP Specification.

2. Recommendations

After integrations with several companies throughout the industry, U-PAS recommends:

- Although some of our Integration Partners already operate a web-based administration system, U-PAS recommends that our partners implement a new web service project hosted in a separate operating system process space from the main application. This is to facilitate atomic updates to both the web application and the payment interface without affecting either one.
    - On Windows this usually entails creating a separate IIS Application Pool.
    - On Unix/Linux this usually entails creating a separate Apache site.
- An unstructured data store be used to store logging information. This is to facilitate logging the required information without stressing operational databases and possibly allowing logging to become a performance bottleneck.
- A database with good support for column aggregation functions.
- Database developers either:
    - De-normalize the payment collection table.
    - Or store collection, member, policy, and basic service provider info in a specialde-normalized table.
    - The above recommendation is to enable aggregate functions to be computed across thecollections without having to first process what could be an expensive join operation possibly triggering a U-PAS app Client call timeout.
    - This design will reduce table lock contention on most SQL databases engines today.
- Application developers maintain clearly defined collection validation routines separate fromthe collection commit routines. This is to reduce U-PAS app Client response times and allow the easy transition to asynchronous collection commits when collection volume increases.

3. Upgrade to Bank Card Processing

This section was written for Integration Partners that already have implemented the standard cash collection interface and are now upgrading to the unified cash/card/etc., settlement interface.
The intention of this section is to provide a general overview guide towards migrating the current implemented functionality towards supporting the new settlement methods.

There are two areas that this upgrade affects.

I. Recording an insurance collection.

The standard **cash collection** implementation relied on the ***ProcessUISAPPPremiumCollection*** function to both validate and commit the transaction. Due to the requirement to validate collections before thepayment is submitted to a payment processor, the validation routines from the ***ProcessUISAPPPremiumCollection*** function have been moved into their own dedicated function ***PrevalidateUISAPPCollection****;* the ***CommitUISAPPCollection***.function now performs the act of recording the collection to the payment database and updating the policy as required.
In addition to recording the details of the collection, the ***CommitUISAPPCollection*** will also include thesettlement type along with the payment processor's transaction reference (RRN) of each collection.
The settlement type should also be stored for later reference.

II. Cashing up collections processed by an Agent.

The current cash up function totals ALL transactions reported by the U-PAS service. The requirement for this function is to report the details of **cash** collected by Agents. In alignment with this upgrade, the***GetUISAPPCashUpPerVendor*** function should be modified to total all collections performed by an agent with a settlement type of "Cash".

Please note:
The details concerning collections of other settlement types may be added to the ***GetUISAPPCashUpPerVendor*** function in future.

4. Definitions

General:

Integration Partner	The company responsible for implementing the U-PAS Client side of thisintegration specification
Service Provider	A company that offers services to the public in exchange for a premiumor subscription also known as a Broker.
Vendor	A person that is employed or contracted to collect premiums on behalf of aService Provider, also known as an Agent.
Device	The terminal or mechanism used by either the vendor or Member to receiptand allocate premiums.
Member	The private individual that is responsible for payment of premiums.

Technical:

U-PAS Client	The administration service that the Service Provider has deployed to managethe member policies and accounts database
U-PAS Service	The Payment Collection service operated and managed by The Unplugg Group.
U-PAS API Key	A cryptographically secure random text string that identifies the relationshipbetween a U-PAS-Service 'Service Provider' and a U-PAS-Client 'Service Provider'. This allows unambiguous, cross-platform identification of Service Providers. This is generated and distributed only by The Unplugg Group.
Vendor PIN	A Vendor PIN is a short (4 – 10 digits) code entered by a vendor on a device to identify themselves to the platform, this may include alphabets. **PINs are required to be unique across aService Provider**
Vendor ID	A Vendor ID is the U-PAS Client generated ID identifying the vendor as an individual on the U-PAS Client software platform. **Vendor IDs are required tobe COMPLETELY UNIQUE across the U-PAS Client**
Datastore Key	This field is ANY field that uniquely identifies a policy in the U-PAS Client Database. Although an integer could be used, it is **recommended** that a UUID isused whenever possible. The U-PAS Service will use this ID as the Policy ID when making calls to the U-PAS Client in association with this policy.
Device ID	Device IDs are short unique identifiers e.g. “12000” used to identify a physicalor virtual transaction terminal. These terminals may either be a hardware pointof sales device, a desktop, cellphone or tablet application or an online form. Device IDs allow a transaction to be grouped into either physical or logicalcategories for example regional branches or a brand.

5. U-PAS App Levels of Integration

Level 1. Suitable for prototyping and testing
Level 2. Suitable for Cash Premium Collection and Cash Ups in the field
Level 3. Extended cash management
Level 4. Extended platform functions
Level 5. Security and Authentication functions

Level 1

To build a minimum interface for testing, timing and prototyping reasons, the following functions should be implemented:

- GetMemberInfoViaIDNumber or Policy Number
- GetServiceProviderInfo
- ProcessUISAPPPremiumCollection
- AuthenticateDeviceVendor

The implementation of these functions will allow a complete test of a collection including theproduction of a receipt.

Level 2

For deployment at clients performing cash collections, it is recommended that at least level 2integration be implemented. This includes all functions from level 1 in addition to:

- GetPolicyStatement
- GetUISAPPCashUpPerVendor
- ReverseUISAPPPremiumCollection

Level 3

Level 3 allows for extended cash and member management from the devices.

- CaptureExistingMember
- GetProductTypeList

Level 4

Level 4 functions improve management and monitoring and reduce manual administration of thesystem.

- RegisterUISAPPDevice
- U-PASClientPing
- AllocateTokenToMember
- RecordCashDeposit

Level 5

Please see the section entitled “Key and Security Description”.

6. Message Transport

The U-PAS Platform is composed of two primary components, the U-PAS Service hosted, managed and developed by The Unplugg Group and developed by third-party software companies.

The U-PAS service makes standard HTTP web calls to a web service hosted as part of the U-PAS Clientsolution. The parameters of the web calls are passed as standard HTTP GET parameters in accordance with the U-PAS specifications included in this document.

Error Response Format

Every request to the U-PAS Client must result in a valid JSON document, or an error message asdescribed here.

E.g.

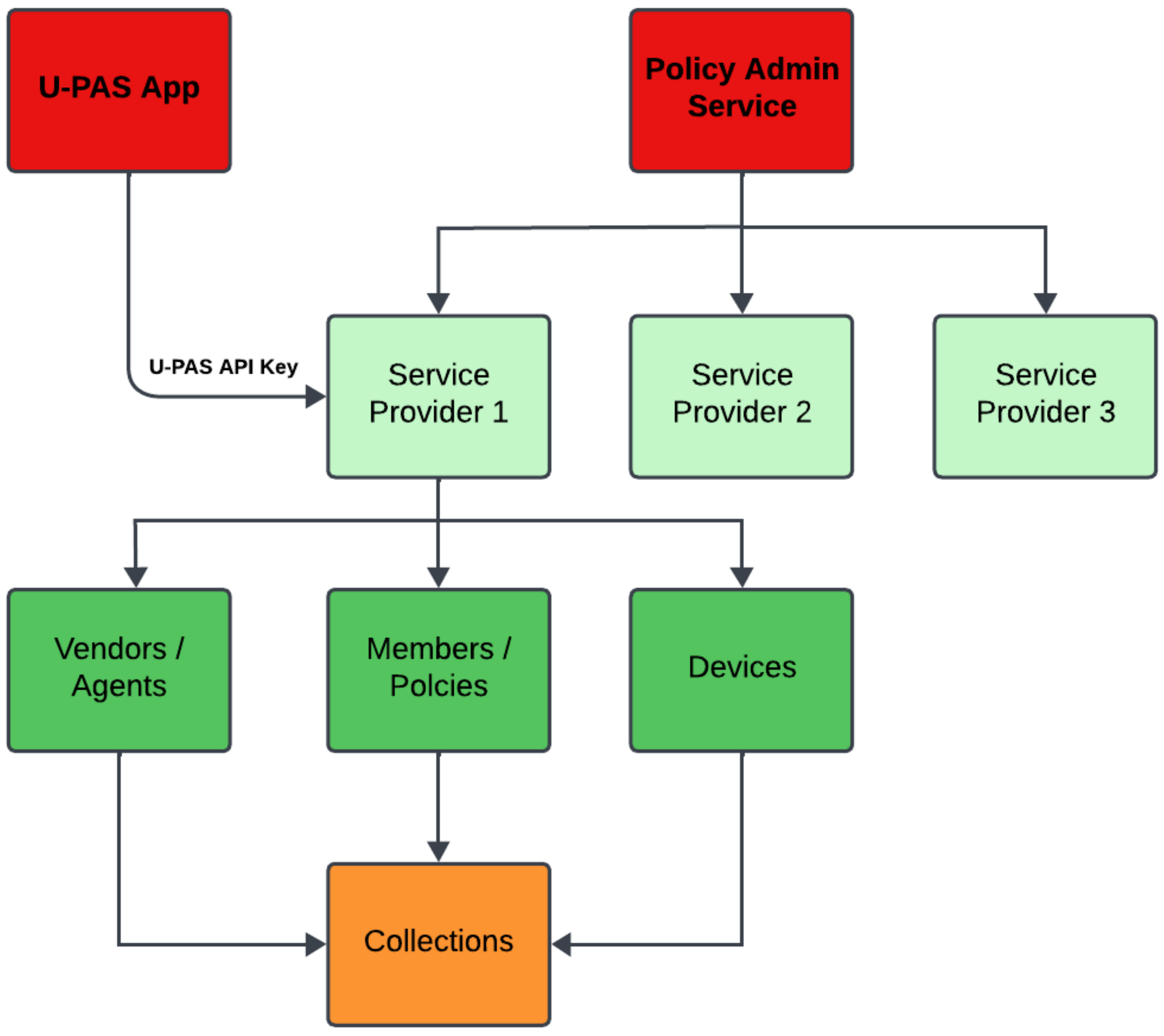
```
{
        'Result':'ERROR',

        'LogLevel':'WARN',
        'VendorMessage':'The Policy has
        lapsed','RefNo':'C15FF83F',

        'SystemMessage':'Policy Lapsed',

        'U-PASRequestID':'d1297113-abfb-4a42-b8bf-a97501675b14'

}
```

LogLevel	*INFO, WARN, ERROR, CRITICAL. Log level is used to determine the severity the error.*
VendorMessage	*The Vendor Message is printed on a receipt for further instruction. This can include information specific to the member or policy and should include an action on what to do next. E.g. The member was notfound, please ask the member to see the office.*
RefNo	*The RefNo is included on the receipt for follow up referencing. Thisreference number may be logged in a ticket or log system on the U-PAS Client for further investigation. The Unplugg Group recommends a short, clear ref number be used, e.g.* ***20140606-82****, or* ***BD743791***
SystemMessage	*The SystemMessage is included in system logs for further diagnostics.This message should be used to identify the TYPE of error encounted and should not change at all for that error type. E.g. Connection to database failed.*
U-PASRequestID	*The U-PASRequestID is included to correlate errors with U-PASService Requests*

Condition	**Log Level**	**System Message**	**Vendor Message**
Technical Issues			
Database Unreachable	ERROR	Database Unreachable	An error occurred while processing your request. Please contactthe software provider for support
Unhandled Application Error	FATAL	Unhandled Application Error	An error occurred while processing your request. Please contactthe software provider for support

All WARN responses are treated as a standard part of the business process and will only be logged for future reference. All ERROR and FATAL responses will trigger an immediate investigation and follow up by The Unplugg Group.

7. Conceptual System Layout



7. Client Functions

7.1 GetMemberInfoViaIDNumber

The GetMemberInfoViaID function is generally called as the first step in the UISAPP collection process.
The purpose of the GetMemberInfoViaID function is to return the required information to collect for apolicy and produce a collection receipt.
For performance and stability reasons the U-PAS Service Caches the results to this function for 24 hours (if NewRegistration is False).
This is to prevent outages at the U-PAS Client side from affecting the UISAPP Service and Member experience. The cache system is arranged:

- UISAPP will first check the internal short-term cache.
- If the member info is not available internally UISAPP will check to see if the U-PAS client isonline if so, UISAPP will send a request to the U-PAS Client.
- If the U-PAS Client is not available, then UISAPP will check the internal long-term cache that isupdated whenever possible.

Parameters	**Required**	**Description**
IDType	Y	SAIDNumber, PolicyNumber, IDToken, CellNumber
ID	Y	The ID Value to search
U-PASAPIKey	Y	The key identifying each Service Provider
DeviceID	Y	The MAC number of the device executing this request
RequestID	Y	The ID of this specific request, for reference

Example:

http://127.0.0.1/U-PAS/? Function=GetMemberInfoViaIDNumber&U-PASAPIKey=U-PASAPIKEY&ID=2809010217083&DeviceID=9480&IDType=SAIDNumber&RequestID=9c43430a-ddb9-41a4-8ae6- 582f15febc4c&SigningID=SERVERAPIKEY&ReqDate=201312031721&

Response (Member Registered):

{

'Result':'OK',

# ~ A unique key identifying this ***member***

'DataStoreKey': '05759d82-d803-11e2-8109-a385742a0157',

'MemberName': 'John Doe',

# ~ If available will be presented prominently on the receipt
'SAIDNumber': '',

'CellphoneNumber': '0113210194',

# ~ A 96 character or less message to add to the next collection receipt to be printed for this member.
'ReceiptMessage': 'Thank you for your business!',

```
# ~ The 'Policies' element is a LIST of DICTIONARY ELEMENTS.This is not just a list
'Policies': [

        {

        # ~ The official ID for this Policy on the U-PAS Client. This ID will identify which policy was
        collected for in ProcessUISAPPPremiumCollection.

        'DataStoreKey': '05759d82-d803-11e2-8109-a385742a0157',

        # ~ Either PremiumCollection,
        PolicyInitiation'TransactionType':
        'PremiumCollection',

        # ~ The Policy number and Policy name should not exceed 20 characters combined'PolicyNumber':
        'AP01',



        'PolicyName'                 : 'BM Standard',

        # ~ The 'Premium', 'Recommended' or amount expected to be paid now (Joining fees, Penalties,etc.)

        'Premium': 50,

        # ~ The policy balance should be used to indicate any arrears.
        'Policy Balance'  : 0,

        # ~ Which month should the member be paying for. E.g. '2013-05' for May. UISAPP ensures  that the
        month entered is either on or after this date.

        'PaymentMonth'             : '2013-10',

        # ~Which company underwrites this
        policy'Underwriter': 'Underwriter
        Name',

        # ~Which company underwrites this policy

        'UnderwriterKey': '0d9fa368-dd00-4b4a-8223-68492a9af408',

        # ~Total Policy Cover
        'PolicyCover': 5000.00,

        # ~Underwriting Premium
        'UnderwriterPremium': 5

        }
]
}
```

Notes:

The Service does not support multiple people or policies with the same ID. If multiple people or policies are returned during the **GetMemberInfoViaIDNumber** request, the U-PAS Client is requiredto return an error indicating the next step in resolving this conflict.

The U-PAS client is required to perform an EXACT MATCH on the ID submitted. Closest Match patternsare highly discouraged.

Definitions

Premium	This is the value of a single monthly premium. It does not include any additional fees.
PaymentMonth	The PaymentMonth option enables the U-PAS Client to specify which is the next valid payment month this policy expects to receipt a premium to ensure continuous cover. E.g. if a member's July premium was paid in full of no outstanding fees (see PolicyBalance) and the member returns in July to make another payment towards their policy, the PaymentMonth would be 2015-08.See notes section on **PolicyBalance, Premium and Expected amounts** below
PolicyBalance	This is the current outstanding balance of the policy, excluding the current month's premium. This is the amount that is considered payable to bring the policy up to date and valid. It includes any sign-up premium or penalties and should be positive to reflect any amounts outstanding.
UnderwriterKey	This value identifies the underwriter of the policy independently ofthe underwriter’s name. It must be any unique GUID/UUID that is assigned by the U-PAS Client. The Unplugg Group requires this to unambiguously identify the associated underwriter irrespective ofthe underwriter’s name.
PolicyCover	This is the total cover provided under the policy.
UnderwriterPremium	This value is the total premium to be paid to the underwriter to coverthe risk of this policy.

Notes on **PolicyBalance, Premium and Expected amounts:**

MPIA = Months Paid in Advance

	U-PAS Response		Calculated on the device		
	Premium	Policy Balance	MPIA	Expected Amount	Minimum Amount
Policy in Arrears	Premium	> 0	3	(Premium * MPIA) + Balance	Premium * MPIA
Policy Up To Date	Premium	= 0	3	0	Premium * MPIA
Policy in Credit	Premium	< 0	3	Premium * MPIA	(Premium * MPIA) + Balance
Policy inExcessive Credit	Premium	< 0	1	0	Premium * MPIA

Arrears payments have a positive balance for Policy Balance, I.e. the member is expected to make a'positive' payment to the Service Provider.

Example:

Currency symbol added for clarification only.

U-PAS App Response				Calculated on the device	
	Premium	Policy Balance	Months Paid in Adv	Expected Amount	Minimum Amount
Policy in Arrears	R50	100	3	(R50 * 3mon) + R100= **R250**	R50 * 3mon = **R150**
Policy Up To Date	R50	= 0	3	**R0**	R50 * 3mon = **R150**
Policy in Credit	R50	-20	3	**R150**	R50 * 3mon - R20 = **R130**
Policy in Excessive Credit	R50	-100	1	**R0**	**R50**

The above standard means that agents will always be displayed the precalculated amounts for either accepting a full payment, for a full month, for every month the member would like to pay and/or including any outstanding debits or accrued credits. Accrued credits are deducted from any amounts expected to be receipted by the U-PAS Service if the policy is in credit only. The reason is members are expected to pay a full month's premium for every month they wish to pay for in advance.

Although the devices do provide the functionality to capture an amount that is not some combinationof Full months premium, and balance outstanding, the use of this functionality is discouraged on the U-PAS service, and this is reflected on the devices. It is however still up to the U-PAS client to either accept the payment or throw a WARN (See Errors section) informing the member only full payments are accepted.

	E.g. Current Date	Payment Month	Month Paid From	MPIA	Period Covered
1 Month Paid	2015-07-24	2015-08	2015-07	1	August
2 Months Paid	2015-07-24	2015-08	2015-07	2	Aug - Sep
2 Month Paid	2015-08-05	2015-07	2015-07	2	Jul-Aug
1 Month Paid	2015-08-05	2015-08	2015-08	1	Aug

The U-PAS Service will always ensure that the member will pay either in or after the payment month returned in PaymentMonth. It is up to the U-PAS Client to ensure that the correct month is specified in PaymentMonth.

Errors and Warnings

The **GetMemberInfoViaIDNumber** function is expected to return a member, request a new member registration, or return an error or warning. Please see the table below for examples of errors and warnings

Condition	**Log Level**	**System Message**	**Vendor Message**
Admin and Policy Issues			
Member not found	WARN	Member not found	The member or policy was not found. If this policy has been registered before, please report this errorto the office
Policy Lapsed	WARN	Policy Lapsed	The member has a lapsed policy. Please direct the member to the office to resolve thisissue.
Policy Cancelled	WARN	Policy Cancelled	The policy has been cancelled. Please directthe member to the office to resolve this issue.

GetMemberInfoViaIDNumber “Member Capture” Response:

The Member Capture response below triggers the start of the member capture / sign up process onthe devices.

```
{
        'Result':'OK',
        'Action'                        : 'CaptureMember'
}
```

7.2 GetPolicyStatement

The GetPolicyStatement function is called 'Sometime After' the **GetMemberInfoViaID** function.

This function should return a list of the last 6 collections. This list will exclude the collection the vendoris either currently collecting or has just done. The UISAPP Service will automatically add the current collection info to the receipt where required.

Parameters

Parameters	**Required**	**Description**
PolicyID	Y	DataStoreKey from **GetMemberInfoViaID.** This is the ID as used bythe U-PAS Client.
DeviceID	N	The MAC number of the device executing this request
VendorID	N	The ID as returned By **AuthenticateDeviceVendor** for this Vendor.This variable should not be required, only recorded if supplied
U-PASAPIKey	Y	The key identifying each Service Provider
RequestID	Y	The ID of this specific request, for reference

Example:

http://127.0.0.1/U-PAS/? Function=GetPolicyStatement&U-PASAPIKey=U-PASAPIKEY&DeviceID=9480&VendorID=&PolicyID= UNPSOW00005&RequestID=9e86d6ee-1144-49eb-b155- 821fb13c1b10&SigningID=SERVERAPIKEY&ReqDate=201312031817&

Response:

```
{
        'Result':'OK',
        'MemberName’: 'John Smith',
        'SAIDNumber’: '8007315180083',
        'PolicyBalance’: 0,
        'Statement' : [
                {'ReceiptNumber’: '106', 'ReceiptDate’: '2012-02-23 14:56:20', 'ReceiptVendor’: 'A.Mata',
                        'ReceiptAmount’: 320},
                {'ReceiptNumber’: '105', 'ReceiptDate’: '2012-02-23 14:56:20', 'ReceiptVendor’: 'A.Mata',
                        'ReceiptAmount’: 320},
                {'ReceiptNumber’: '104', 'ReceiptDate’: '2012-02-23 14:56:20', 'ReceiptVendor’: 'A.Mata',
                        'ReceiptAmount’: 320},
                {'ReceiptNumber’: '103', 'ReceiptDate’: '2012-02-23 14:56:20', 'ReceiptVendor’: 'A.Mata',
                        'ReceiptAmount’: 320},
                {'ReceiptNumber’: '102', 'ReceiptDate’: '2012-02-23 14:56:20', 'ReceiptVendor’: 'A.Mata','
                        ReceiptAmount’: 320},
                {'ReceiptNumber’: '101', 'ReceiptDate’: '2012-02-23 14:56:20', 'ReceiptVendor’: 'A.Mata','
                        ReceiptAmount’: 320}
        ]
}
```

7.3 ReverseUISAPPPremiumCollection

The ReverseUISAPPPremiumCollection enables the U-PAS Service to reverse a collection due to technical errors. The U-PAS service identifies the collection to be removed using the *CollectionID* parameter below. Please note, the *OriginalVendorID*, *ReceiptNumber* and *PolicyID* fields are provided for convenience and logging purposes as The Unplugg Group cannot guarantee these fields will be completely unique.

Parameters	**Required**	**Description**
U-PASAPIKey	Y	The key identifying each Service Provider
CollectionID	Y	The unique system-wide ID for this collection
Reason	Y	The reason why this collection has been reversed
OriginalVendorID	N	
ReceiptNumber	N	
PolicyID	N	
RequestID	Y	The ID of this specific request, for reference

Example:

Please Note: This example is a little out of date. Please see the parameters listed in the table above.

http://127.0.0.1/U-PAS/?

Function=ReverseUISAPPPremiumCollection&U-PASAPIKey=U-PASAPIKEY&CollectionID=861fa67d-

111c-4592-bd0f-

ba8878c10af0&Reason=CollectionTimedout&SigningID=SERVERAPIKEY&ReqDate=201312031953&

Response:

```
{
        'Result':'OK'
}
```

Errors and Warnings

Condition	**Log Level**	**System Message**	**Vendor Message**
Collection not found (Either because it was not recorded, or not reported)	WARN	Collection Not Found	The collection specified was not recorded.
Collection not reversed (Certain collections maynot be reversed, e.g. bank card)	ERROR	Collection Not Reversed	The collection was unable to be reversed.

7.4 AuthenticateDeviceVendor

The AuthenticateDeviceVendor function is called when a vendor tries to authenticate their PIN on a device.

The UISAPP platform will cache this information for 24 hours and may refer to this information later should the U-PAS client go offline.

Parameters

Parameters	**Required**	**Description**
U-PASAPIKey	Y	The key identifying each Service Provider
DeviceID	Y	The MAC Number of the device that processed this collection
RequestID	Y	The ID of this specific request, for reference
VendorPin	Y	The pin as entered on the devices to authorize Vendors

Example:

http://127.0.0.1/U-PAS/? U-PASAPIKey=U-PASAPIKEY&Function=AuthenticateDeviceVendor&DeviceID=9480&VendorPin=14752&SigningID=SERVERAPIKEY&ReqDate=201312031752&

Response:

```
{
        'Result'                :'OK',
        'Enabled'               : True,
        'VendorID'              : '691e73a2b178',
        'VendorReceiptName'     : 'A.Mata',      #~ maximum of 10 char, due to space on the receipt
        'VendorStatus'          : 'Operator'     #~ Either Normal, Super or Operator. Anything else
        will
                                fail
}
```

Errors and Warnings

Condition	**Log Level**	**System Message**	**Vendor Message**
Vendor not found	WARN	Vendor Not Found	The specified code is not authorized

8. Key and Security Description

This section describes the function and use of the U-PAS Keys and security mechanism.
The U-PAS interface relies on multiple levels of security, incorporating two-way authentication,authorization and transport security.

Two-way Authentication and Authorization

At least two security keys are involved in every request to enable authenticating and authorizingrequests:

U-PASAPIKey	Identifies to which Service Provider this request is targeted
SigningID	Identifies the security key of the U-PAS Service making the request

U-PAS Service Authentication

U-PASAPIKey

The U-PASAPIKey allows multi-tenant U-PAS Client Software to redirect the request to the ultimate Service Provider data-source for the current request.
This key is issued by The Unplugg Group on behalf of each Service Provider operating on the UISAPP platformduring the Service Provider sign up process.
Due to the security requirements of the U-PAS Key, the U-PAS Client is required to allow this key to be regularly changed. A mapping process internal to the U-PAS client should map the U-PAS Key to a Service Provider ID for each U-PAS request.

U-PAS Key Example: *c0a28785-61eb-46a2-b731-8cb756081170*

Devices should be scoped to the same Service Provider that owns the supplied key for additionalsecurity on the U-PAS Client.
The U-PASAPIKey WILL change during the lifetime of the account and so should be accessible toupdate by the correctly authorized user.

SigningID

The SigningID (and associated encryption key) allows the U-PAS Client software to preventunauthorized access to the U-PAS interface.

The SigningID and the associated encryption key is issued by The Unplugg Group upon the integration of theU-PAS Client. The encryption key is changed regularly and should be easy for the system administrators to update.

SigningID example: *cd1da206-1e6a-487d-89c5-71677631fc27*

Encryption Key example: *65f56070-9663-4305-bd70-1f80d74b36e*

Authentication

It is the responsibility of the U-PAS Client to ensure that each request is authenticated and authorized.

The U-PAS Service provides the following mechanism to the client to perform this process:

The request authentication mechanism combines the above keys, request parameters, current time (ReqDate) and the encryption key into a standard HMAC-SHA1 function.

Example HMAC("http://127.0.0.1/U-PAS/? U-PASAPIKey=U-PASAPIKEY&Function=AuthenticateDeviceVendor&DeviceID=9480&VendorPin=14752&SigningID=SERVERAPIKEY&ReqDate=201312031752&", "ENCYPTKEY")

The result of this funcion is then fed into a standard Base64 encoding functionand appended to the HTTP header parameter *x-UISAPP-SecurityHash*

Example: x-UISAPP-SecurityHash: YmZkMTJmYjQtMD...GI4MmVhYzY2MDMzCg== (shortened here for bevity)

The U-PAS Client is then expected to load the associated keys and duplicate this process, ensuring thatthe result of the Base64 function matches the *x-UISAPP-SecurityHash* Header.

Note: The *ReqDate* GET parameter should be compared to the U-PAS Client's local system clock, all requests outside of a 5-minute window either side should be rejected.

U-PAS Client Authentication

The U-PAS Service supports three options of client authentication:

- Standard SSL
- OpenVPN certificates
- IP Address specification

These are listed in decreasing priority of convenience/security. The highest level is the combination of all three.

SSL

The U-PAS Service supports privately signed certificates as we import and store the public keysinternally, not relying on external providers for validation.

OpenVPN

U-PAS App runs an online OpenVPN Service, allowing U-PAS Clients to create secure tunnels to an internal service network. As U-PAS App carefully monitors the connections on this network, possiblesecurity breaches are dramatically reduced. This network also provides the option for Clients to connect to the U-PAS Service without having a public static IP address or opening extra ports on a public firewall.

IP Address specification

The U-PAS client may provide a specific IP address. This is the least secure of the three methods however prevents most DNS related outages. Whenever possible, U-PAS App would prefer an IPaddress over a DNS address.

Please Note: Data may be transmitted over the public internet if SSL or OpenVPN is not used. This maybe a violation of the Service Provider's security and privacy policies.

Auditing and Logging

To fulfill the UISAPP platforms auditing and logging mandate, the U-PAS Client is requested toimplement the following logging functions:

General Logging

For each request:

- The URL requested.
- The x-UISAPP-Security HTTP header
- The Date and Time
- RequestID / CollectionID

These are specifically important when errors are encountered.

Specific Logging

ProcessUISAPPPremiumCollection

- PolicyID
- VendorID
- AmountInCents

Errors

- Any error reference numbers returned
- Error message
- U-PASRequestID
- Vendor