
KUBA
Simplifying payments, transforming mobility

Product Technical Specification

PAD Monitoring Service API

KBP250311-01TS

Document version

Version	Date	Author	Description
01	March 11th, 2025	Kuba, Inc.	Updated document to combine Device and Transaction Monitoring Specifications

Table of Contents

1. Purpose of this document

The purpose of this document is to describe the technical product specification of an external Device Monitoring API with the Proxima Back Office platform.

2. Contact

For more information, please contact Brian Frank (702) 747-2958 (M) or email: brain.frank@kubapay.com

3. Related documents

This Technical Product Specification (KBP240124-02TS) should be read with the Technical Product Specification for EMV Transaction Monitoring Service KBP240124-01TS and our proposal dated February 08th, 2024 (version 1).

4. Terminology

The table below contains a list in alphabetical order of the most important acronyms and/or terms used in this document.

Term	Definition
API	Application Programming Interface
BO	Back Office
cEMV	Contactless EMV
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol over Secure socket
PAD	Payment Acceptance Device
REST	Representational State Transfer
SSL	Secure Socket Layer
TTL	Time To Live
URL	Uniform Resource Locator

5. Architecture Overview

5.1 Scope

Section 5 describes the 3rd party monitoring integration with the Proxima Back-Office system.

The Proxima Back Office system consists of a monitoring 'live dashboard' module that collects transaction, monitoring and diagnostic data from the PADs connected to the system.

The 3rd party monitoring API will be an interface to retrieve relevant monitoring data from the Proxima Back Office.

5.2 Monitoring integration on the Proxima platform

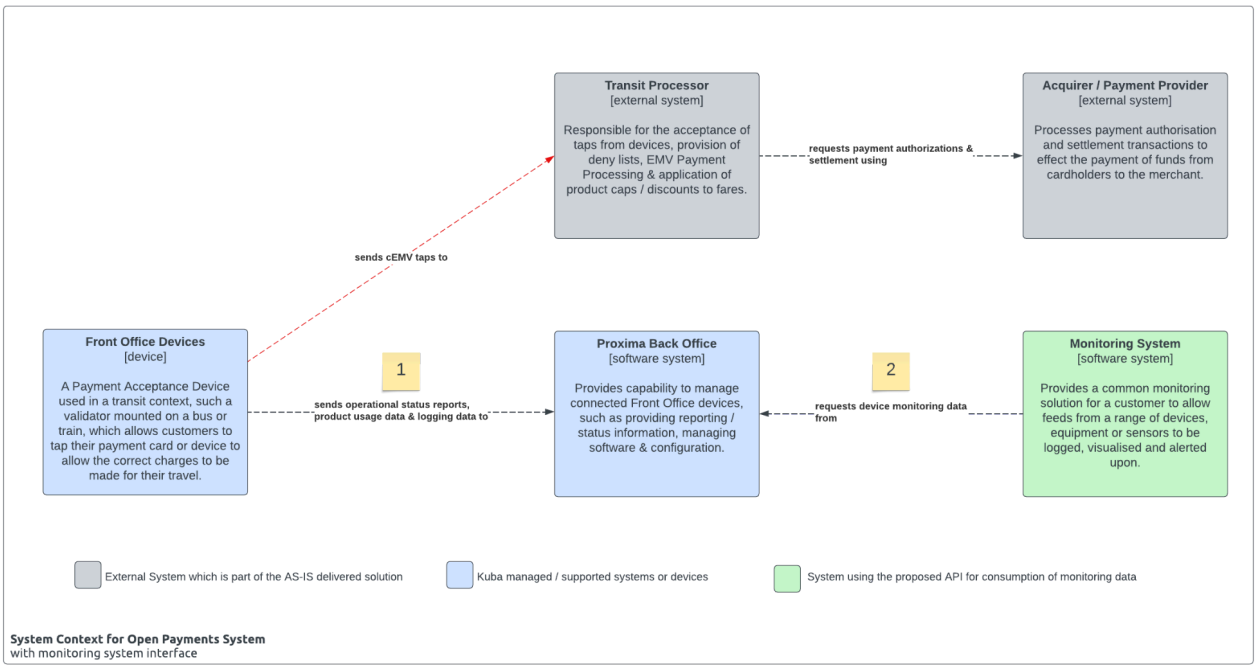

Figure 1: Device monitoring integration on the Proxima platform

As can be seen from Figure 1 above, the main interfaces involved are as follows:

1. Device, transaction, diagnostic & monitoring data is sent from connected Front Office devices to the Proxima Back Office
2. Device monitoring data is exchanged with the Monitoring System

Interface 2 is described in more detail below.

5.3 Authentication

An authentication endpoint is provided which has to be called prior to making calls to the APIs.

For a successful authentication a cookie is returned / short lived session id which is then used in subsequent calls to provide the authentication.

The TTL for the session id is 60 seconds.

Request

Method	URL
POST	<host>/monitoring/authenticate/v1
Parameters	

Body	{
	"UserName": "APIUser",
	"Password": "Password",
	"OperatorIdentifier": 99
	}

- User credentials are configured through the User Management function on the Proxima back office. See the section Creating API Users below.
- OperatorIdentifier will be provided by Kuba.

Response

HTTP Response	

200	{
	"Error": null,
	"UserId": "APIUser",
	"FirstName": "API",
	"LastName": "User",
	"SessionId": "d168d2b7-c744-4d19-8aca-8646c726dfd7"
	}

200 -	"Error": { "ErrorCode": 4,

Unknown User	<pre>"Message": "User does not exist"
	}</pre>

200 - Invalid Password	<pre>"Error": {
	"ErrorCode": 8,
	"Message": "Invalid Password"
	}</pre>

5.4 Authorisation

The HTTP Header (Cookie parameter) has to contain the session-id obtained from the authentication call in section 5.3.

HTTP Header Example

```
--header Cookie:session-id=b53f48e9-b6e3-44db-bed9-1416d56407b9 \
--header Accept: application/json \
--header Content-Type: application/json \
```

5.5 HTTP Responses

Common HTTP responses codes will be provided with further details provided during the implementation of any additional information provided in response to detail any error conditions.

- 2XX values indicate a successful HTTP request:
    - 200 (OK),
    - 201 (created).
- 304 value indicates a not modified HTTP request, and will be returned in combination with the request header 'If- Modified-Since' flag
- 4XX values indicate an unsuccessful HTTP request:
    - 400 (bad request),
    - 403 (forbidden),
    - 404 (not found),
    - 405 (method not allowed),
    - 409 (conflict).
    - 417 (expectation failed) Rate Limit Response
- 5XX values indicate an unsuccessful HTTP request. In case of 5XX errors, the client should retry sending its failed request.

5.6 Rate Limit

A rate limit is applied to calls to the monitoring APIs. HTTP response code 417 will be returned with the following message if the rate limit is exceeded.

```
{
    "ErrorCode": 1,
    "Message": "Last call was less than 1 minute ago"
}
```


6. Device Monitoring APIs

Section 6 describes Interface 2 (Figure 1).

Requests can be made to obtain device monitoring data based on the location or locations, device or devices.

6.1 Get Monitoring Data by Location

- **verb -** requests are GET methods with query parameters.
- **path** - /monitoring/deviceproperties/v1/ForLocations.
- **usage** - get monitoring data for devices currently within one or more locations or all devices within a given location type.

Query Parameters

Name	Optional	Description / Usage
if_modified_since	Yes	if set, only returns devices which have communicated after this timestamp specified as yyyymmddhhmm
all	Yes	If set, location_type must be specified
location_type	No	Indicates selected location types to monitor • 1 = Bus
location_ids	Yes (if all is not set)	Location ids to monitor • single location = *location_ids=123* • multiple *location_ids=123&location_ids=456*

Response Data Elements

Element Name	Type	Description
response_date_time	String	date time of server response (ISO8601 format)
list	Collection	Contains 1 or more of the following objects
device	Object	
fo_device_logical_id	String	Logical device Id of the linked FO device
fo_device_type	String	validator, gate, pos, tvm, inspection
fo_device_type_model	String	ABT3000, MV3000
fo_device_serial_number	String	Serial number of the linked FO device
fo_device_description	String	Device description of linked FO device
fo_device_location_id	String	Device location id of linked FO device
fo_device_location	String	Device location description of linked FO device
fo_device_last_connection	String	Last replicator connection timestamp (ISO8601 format)
device_replicator_info	Object	Monitoring data for the device
software_version	String	Software version active on the device
software_last_connection	String	timestamp (ISO8601 format)
cd_version	String	CD version active on the device
cd_last_connection	String	timestamp (ISO8601 format)
dataset_version	String	version active on the device
dataset_last_connection	String	timestamp (ISO8601 format)
denylist_version	String	Deny list version active on the device
denylist_last_connection	String	timestamp (ISO8601 format)

Element Name	Type	Description
acceptlist_version	String	Accept list version active on the device
acceptlist_last_connection	String	timestamp (ISO8601 format)
binlist_version	String	BIN list version active on the device
binlist_last_connection	String	timestamp (ISO8601 format)
asset_last_connection	String	timestamp (ISO8601 format)
monitoring_last_connection	String	timestamp (ISO8601 format)
ud_last_transaction_time	String	timestamp (ISO8601 format)
device_monitor_info	Collection	Collection of key-value pairs
key	String	key for this data item
value	String	value for this data item

Device Monitor Key Pairs

device_monitor_info consists of the following key-value pairs.

Key	Type	Description
application::isinservice	String	**true** or **false** indicating the device is in service
application::isdisabled	String	**true** or **false** indicating the device is disabled
os::uptime	String	uptime of the device, format: d hh:mm:ss.mmm

Other key-value pairs can be added in the future.

Example Requests

Method	Request Parameters
Single Bus Location	<host>/monitoring/deviceproperties/v1/ForLocations?location_type=**1**&location_ids=**1732**

Multiple Bus Devices	<host>/monitoring/deviceproperties/v1/ForLocations?location_type=**1**&location_ids=**1739**&location_ids=**1731**
All Bus Devices	<host>/monitoring/deviceproperties/v1/ForLocations/all?location_type=**1**
All Bus Devices since timestamp	<host>/monitoring/deviceproperties/v1/ForLocations/all?location_type=1&if_modified_since=20250 3121116

Example Response (Single Bus Location = 1732)


```
{
    "Response_date_time": "2025-03-12T11:08:50.2862245Z",
    "List": [
        {
            "Device": {
                "Fo_device_logical_id": "66017",
                "Fo_device_type": "Validator",
                "Fo_device_type_model": "ABT3000",
                "Fo_device_serial_number": "100107231057301512",
                "Fo_device_description": null,
                "Fo_device_location_id": "1732",
                "Fo_device_location": "Bus - 1732",
                "Fo_device_last_connection": "2025-03-12T05:34:17.0970000Z"
            },
            "Device_replicator_info": {
                "Software_version": "5.12.70.5349",
                "Software_last_connection": "2025-03-12T05:21:32.4570000Z",
                "Cd_version": "34",
                "Cd_last_connection": "2025-03-12T05:33:14.8870000Z",
                "Dataset_version": "DatasetConfig - 3",
                "Dataset_last_connection": "2025-03-12T05:31:52.9170000Z",
                "Denylist_version": null,
                "Denylist_last_connection": null,
                "Acceptlist_version": "0",
                "Acceptlist_last_connection": "2025-03-12T05:23:25.5530000Z",
```

```
                "Binlist_version": "2753",
                "Binlist_last_connection": "2025-03-12T05:21:53.4100000Z",
                "Asset_last_connection": "2024-12-30T06:10:07.3300000Z",
                "Monitoring_last_connection": "2025-03-12T05:34:07.6870000Z",
                "Ud_last_transaction_time": "2025-03-12T02:47:35.4830000Z"
            },
            "Device_monitor_info": {
                "application::isdisabled": "false",
                "application::isinservice": "true",
                "application::servicestatus": "{\"Rows\":[]}",
                "gps::position": "{\n    \"altitude\": 0,\n    \"dateTime\": \"\",\n
\"direction\": 0,\n    \"gpsFix\": 0,\n    \"groundSpeed\": 0,\n    \"hasAltitude\":
false,\n    \"hasDateTime\": false,\n    \"hasDirection\": false,\n    \"hasGpsFix\":
false,\n    \"hasGroundSpeed\": false,\n    \"hasLatitude\": false,\n    \"hasLongitude\":
false,\n    \"latitude\": 0,\n    \"longitude\": 0,\n    \"numberSatelites\": 0,\n
\"properties\": {\n    }\n}\n",
                "location::location::info": "{\n    \"current\": {\n        \"properties\":
{\n        },\n        \"stopinfo\": {\n            \"abbreviation\": \"\",\n
\"farematrixreference\": 9999,\n            \"name\": \"MST\",\n            \"reference\":
9999,\n            \"shortname\": \"\",\n            \"type\": 0\n        }\n    },\n
\"dataid\": \"\",\n    \"dataversion\": 0,\n    \"locationProviderSource\":
\"location::devicesettings::provider\",\n    \"properties\": {\n    },\n    \"type\":
65535\n}\n",
                "location::location::serviceStatus": "",
                "location::location::source": "location::devicesettings::provider",
                "os::uptime": "0 01:10:59.010"
            }
        }
    ]
}
```


6.2 Get Monitoring Data by Device

- **verb -** requests are GET methods with query parameters.
- **path** - /monitoring/deviceproperties/v1/ForDevices
- **usage** - get monitoring data for specific devices or all devices for a given device type.

Note : responses are the same as Get Monitoring Data by Location, so details of the responses are not repeated here.

Query Parameters

Name	Optional	Description / Usage
if_modified_since	Yes	if set, only returns devices which have communicated after this timestamp specified as yyyymmddhhmm
all	Yes	If set, device_type must be specified
device_type	No	Indicates selected device types to monitor • 1 = Validator
device_ids	Yes (if all is not set)	Device ids to monitor • single device = *device_ids=66001* • multiple *device_ids=66001&location_ids=66002*

Example Requests

Method	Request Parameters
Single Device	<host>/monitoring/deviceproperties/v1/ForDevices?device_type=**1**&device_ids=**66001**
• Retrieves data for a single device of type **1** (validator) with id **66001**	
Multiple Devices	<host>/monitoring/deviceproperties/v1/ForDevices?device_type=**1**&device_ids=**66002**&device_ids=**66003**
• Retrieves data for two type **1** devices of type (validators) with ids **66002** & **66003**	
All Devices	<host>/monitoring/deviceproperties/v1/ForDevices/all?device_type=**1**
• Retrieves data for all type **1** (validator) devices	
All Devices after given timestamp	<host>/monitoring/deviceproperties/v1/ForDevices/all?device_type=**1**&if_modified_since=**202503121132**
• Retrieves data for all type **1** (validator) devices where the device has communicated after	

the given timestamp **202503121132** (yyyymmddhhmm)

7. Transaction Monitoring API

Requests can be made to obtain transaction monitoring data.

7.1 Get Transaction Data

- **verb -** requests are GET methods with query parameters.
- **path** - /monitoring/cicoemvtransactions/v1
- **usage** - get transaction data based on the specified starting transaction id, and limited to a number of transactions as specified in limit parameter.

Query Parameters

Name	Optional	Description / Usage
seq_start	No	Id of the transaction number to start at.
limit	No	The number of transactions to return.

Response Data Elements

Element Name	Type	Description
response_date_time	String	date time of server response (ISO8601 format)
seq_next_trx_id	Integer	indication next start transaction_id, 0 if no more transactions are currently available
list	Collection	Contains 1 or more of the following objects
transaction_header	Object	Mandatory data available for each transaction
trx_id	Integer	unique transaction id of the linked transaction
trx_insert_time	String	Date, time transaction is received in BO (ISO8601 format)

Element Name	Type	Description
trx_time	String	Date, time transaction happened (ISO8601 format)
trx_type	String	One of • check-in • check-out • psp
trx_device_type	String	validator, pos, tvm
trx_device_id	Integer	device id on which transaction happened
trx_location	Integer	stop id where transaction happened
check-in	Object	Optional element returned when when trx_type == check-in
ci_result	Integer	is valid, co_result > 0 indicates error
ci_media_print_serialnr	String	media print serial number
ci_product_id	Integer	product identifie r, available when ci_result == 0
ci_value	Integer	ci trx value in cents, valid when ci_result == 0
check-out	Object	Optional element returned when trx_type == check-out
co_result	Integer	0 is valid, co_result > 0 indicates error
co_media_print_serialnr	String	media print serial number
co_product_id	Integer	product identifier, available when co_reslult == 0
co_value	Integer	co trx value in cents, valid when result == 0
co_linked_ci_trx_id	Integer	valid when co_result == 0, linked ci trx_id
psp	Object	Optional element returned when trx_type == psp
psp_result	Integer	0 valid, psp_result > 0 indicates error
psp_external_trx_id	String	valid when psp_result == 0,

Element Name	Type	Description
		indicates the external psp transaction id
psp_linked_cico_trx_id	String	valid when psp_result == 0, linked trx_id of ci trx or co trx

Example Response (/monitoring/cicoemvtransactions/v1?seq_start=1039301&limit=1)


```
{
    "Response_date_time": "2025-03-12T12:51:20.7057468Z",
    "Seq_next_trx_id": 1039305,
    "List": [
        {
            "Transaction_header": {
                "Trx_id": 1039304,
                "Trx_insert_time": "2024-06-18T22:33:32.0000000Z",
                "Trx_time": "2024-06-18T22:33:18.0000000Z",
                "Trx_type": "check-in",
                "Trx_device_id": 66050,
                "Trx_device_type": "Validator",
                "Trx_location": 5272
            },
            "Check_in": {
                "Ci_result": 0,
                "Ci_media_serial_number": "48157760",
                "Ci_product_id": 2101,
                "Ci_value": 200
            },
            "Check_out": null,
            "Psp": {
                "Psp_result": 0,
                "Psp_external_trx_id": "4b4c911f-45a5-4213-8f44-9f527e086683",
                "Psp_linked_cico_trx_id": 1039304
            }
        }
    ]
}
```


Example Requests

Method	Request Parameters
10 Transactions	<host>/monitoring/cicoemvtransactions/v1?seq_start=**1038891**&limit=**10**
• Returns **10** transactions starting at transaction id **1038891**	

7.2 Usage Notes

- Values of check-in (ci_value) and check-out (co_value) are in cents. (2000 is 20,00)
- Total journey value is the sum of linked check-in value (ci_value) + check-out value (co_value).
- Transaction_ids are not sequential, there can be caps between different transactions!
- A PSP transaction needs to follow a check-in and check-out transaction in the following conditions:
    - After a valid check-in or check-out, this is when (ci_result == 0) or (co_result == 0)
    - Invalid check-in or invalid check-out with error hotlist, this is when (ci_result == 430) or (co_result == 430)

Most common returned result codes: (Ci_result / Co_result)

- Valid = 0
- Hotlist = 430
- Passback = 601
- Local support not available = TBD
- Declined = 413, 414
- Medium not within the acceptance list = 419

7.3 Limitations

- Maximum number of transactions that can be requested in one call is 1000.
- “limit” parameter in the URL should be set maximum to 1000.

END