

ReST Interface Specification for Cubis MCA Balances

Version 3.0 (valid for Cubis MCA firmware 09-03-09.02.xx)

Author	Free, Michael	2025-08-17	
Reviewed by	Heitmüller, Jan	2025-08-19	
Released by	Free, Michael	2025-08-19	

Transmittal, reproduction, dissemination and / or editing of this document as well as utilization of its contents and communication thereof to others without express authorization are prohibited. Offenders will be held liable for payment of damages. All rights created by patent grant or registration of a utility model or design patent are reserved.

Document History

Ver.	Datum	Author	Changed Sections	Change Description
-	2019-08-28	Kowalski	-	New document
0.1	2019-08-28	Kowalski	All	First Draft version of the document
0.2	2019-10.11	Sluschny, Thomas		QAPP file handling and QAPP ReST functions added, Asynchronous Callbacks added, corrections
0.3	2019-10-21	Sluschny, Thomas		Different draftshield open operations summarized, async column added to overview, corrections
1.0	2019-10-21	Sluschny, Thomas		Reviewed for first implementation
1.1	2019-10-24	Sluschny, Thomas		More ReST like (PUT instead of POST, resource and not query parameter), corrections of SBe, asynclid as uint64_t
1.2	2019-12-02	Beckmann, Steffen	All	Corrections on some commands: - display - storeweight Add new commands - cancelzero - canceltare - cancelzerotare Define behavior on unknown endpoint Change timeout to 60 seconds Correction of typos Several formal adaptations Examples added
1.3	2019-12-09	Thomas Sluschny	All	Review results accepted, Example answers added, Enumeration for "unit" command, change endpoints: net -> value, gross -> gross calc Several formal changes
1.4	2019-12-11	Beckmann, Steffen		Add more examples Changes on DSD web services

Ver.	Datum	Author	Changed Sections	Change Description
1.5	2020-04-01	von Steuben, Jan		Corrected URL for calls: <balance_ip>/modules/... into: <balance_ip>/rest/modules
1.6	2020.04.15	Kowalski, Florian	2.1 API overview	Corrected URL to starting with rest
2.0	2020.04.15	von Steuben, Jan		Adapt layout to new CI and updated version and firmware information
2.1	2020.07.29	Kowalski, Florian	2.12 2.4.9	Added new chapter for Reporting Adapt Response/Body parameters
2.2	2020.12.08	Arnsberger, Daniel	1.3	Changed timeout to 5 minutes
2.3	2022.01.20	Stemme, Gregor		Added: Connector and Task
2.4	2022.03.10	Stemme, Gregor		Added: Job management, lock and unlock device
2.5	2022.04.22	Crepon, Stefan	2.17 / 2.18	Small fixes, Adaption of Ionizer/commands/start
2.6	2022.08.05	Stemme, Gregor		Added chapter Diagnostic file, QAPP center, Firmware update and Audit trail
2.7	2022.09.30	Heitmüller, Jan, Stemme, Gregor	2.7.9, 2.5.4	Added remark to body parameters; added restart of device
2.8	2023.03.17	Heitmüller, Jan	2.1	Added remark to linear and rotational draftshield
2.9	2023.05.18	Free, Michael	2.20.3	Added job status
3.0	2025.08.17	Free, Michael	2.8	Added DSD functions

Contents

1	Introduction	7
1.1	Purpose and scope	7
1.2	Abbreviations	7
1.3	General	8
2	Interface	9
2.1	API endpoint overview.....	9
2.2	General Information.....	12
2.2.1	Header	12
2.2.2	Response Codes.....	12
2.2.3	Types	12
2.2.4	Scheme	12
2.2.5	Asynchronous Callbacks	13
2.3	Command	14
2.3.1	Get Command Status.....	14
2.4	Connector	15
2.4.1	Dynamic Endpoint	15
2.5	Device	17
2.5.1	Locking.....	17
2.5.2	Unlocking	18
2.5.3	Firmware update	18
2.5.4	Restart device	19
2.6	Device Information	21
2.6.1	Serial number.....	21
2.6.2	Package	22
2.6.3	Model	23
2.6.4	APC Version	24
2.6.5	BAC Version	25
2.6.6	WPC Version.....	26
2.6.7	MCU Version.....	27
2.6.8	QAPP-Center	28
2.6.9	Device IDs.....	29
2.6.10	General info	29
2.6.11	Diagnostic file.....	31
2.7	Weighing	32
2.7.1	Tare	32
2.7.2	Cancel tare.....	34

2.7.3	Zero	35
2.7.4	Cancel zero	37
2.7.5	Zero/Tare	38
2.7.6	Cancel zero/tare	40
2.7.7	Value	41
2.7.8	Gross calc	42
2.7.9	Tare storage	43
2.7.10	Display	45
2.7.11	Metrology	46
2.7.12	Unit	48
2.8	Data Storage Device	50
2.8.1	Get records	50
2.8.2	Get single record	51
2.8.3	Get DSD information	53
2.8.4	Store Weight	54
2.8.5	Cancel	56
2.8.6	Get weight	57
2.8.7	Data storage record structure	58
2.9	Leveling	60
2.9.1	Status	60
2.9.2	Start	61
2.9.3	Cancel	61
2.10	Draftshield	62
2.10.1	Open	62
2.10.2	OpenRowiDirect	63
2.10.3	Close All	65
2.10.4	Palm key	66
2.10.5	Learning Degree	67
2.10.6	Status	68
2.11	Calibration Adjustment	69
2.11.1	Start	69
2.11.2	Cancel	70
2.11.3	Status	71
2.12	Ionizer	72
2.12.1	Start	72
2.12.2	Cancel	73
2.12.3	Status	74
2.13	Climate Module	75

2.13.1	Internal	75
2.13.2	External	76
2.13.3	Internal Information	76
2.13.4	External Information	77
2.14	Reporting	79
2.14.1	Takeover	79
2.15	Audit Trail	79
2.15.1	Unarchived Records	79
2.15.2	Records	80
2.16	Screen	82
2.16.1	Overlay Message	82
2.17	Task	82
2.17.1	Start Task	82
2.17.2	Stop Task	83
2.18	QAPP	85
2.18.1	QAPP file handling	85
2.18.2	QAPP ReST functions	86
2.19	QAPP Center	87
2.19.1	License information	87
2.20	Job management	88
2.20.1	Unit operations	88
2.20.2	Job handling	88
2.20.3	Jobs status	90

1 Introduction

1.1 Purpose and scope

This document shall provide the user a documented description of all (external) available ReST web services of the Cubis II balance.

1.2 Abbreviations

Abbreviation	Description
APC	Application Processor Controller
BAC	Balance Controller
HTTPS	Hypertext Transfer Protocol Secure
Json	JavaScript Object Notation
MCU	Micro Controller Unit
ReST	Representational State Transfer
TLS	Transport Layer Security
WPC	Windshield Processor Controller

1.3 General

The ReST communication is encoded with HTTPS/TLS. The default Port is 443 (HTTPS Port).

HTTPS and the certificate are used only for encryption and not for authenticity verification. This means the certificate is generated by the device and self-signed.

The Body is based on JSON and is formatted according to the standard style guide <https://google.github.io/styleguide/jsoncstyleguide.xml>.

The maximal size of body is 1 MByte,

The URL is in spinal case (hyphen separated lower case).
<https://restfulapi.net/resource-naming/>

Communication is handled as UTF-8. The default timeout for every webservice is 5 minutes.

All web services need to include a password within the HTTP header with key “cubispass”. The password can be set in the settings menu of the balance.

All PUT requests have to specify a HTTP header ‘content-type’ with ‘application/json’, an exception is the QAPP file handler.

2 Interface

2.1 API endpoint overview

Name	Endpoint	Get	Post	Del	Put	Async
Command						
Get Status	/rest/commandStatus	X	-	-	-	-
Connector						
Dynamic Endpoint	/rest/modules/connector/dynamic_endpoint	X	X			
Device Information						
SerialNumber	/rest/deviceinfo/serialnumber	X	-	-	-	-
Package	/rest/deviceinfo/package	X	-	-	-	-
Model	/rest/deviceinfo/model	X	-	-	-	-
APC	/rest/deviceinfo/apc	X	-	-	-	-
BAC	/rest/deviceinfo/bac	X	-	-	-	-
WPC	/rest/deviceinfo/wpc	X	-	-	-	-
MCU	/rest/deviceinfo/mcu	X	-	-	-	-
deviceId	/rest/deviceinfo/deviceid	X	-	-	X	-
qappcenter	/rest/deviceinfo/qappcenter	X	-	-	-	-
Weighing						
Tare	/rest/modules/weighing/commands/tare	-	X	-	-	X
Cancel Tare	/rest/modules/weighing/commands/ canceltare	-	X	-	-	-
Zero	/rest/modules/weighing/commands/zero	-	X	-	-	X
Cancel Zero	/rest/modules/weighing/commands/ cancelzero	-	X	-	-	-
Zero/Tare	/rest/modules/weighing/commands/zerotare	-	X	-	-	X
Cancel Zero/Tare	/rest/modules/weighing/commands/ cancelzerotare	-	X	-	-	-
Value	/rest/modules/weighing/data/value	X	-	-	-	-
Gross calc	/rest/modules/weighing/data/grosscalc	X	-	-	-	-
Tare storage	/rest/modules/weighing/data/tare	X	-	X	X	-
Display	/rest/modules/weighing/data/display	X	-	-	-	-
Metrology	/rest/modules/weighing/data/metrology	X	-	-	-	-
Unit	/rest/modules/weighing/data/unit	X	-	-	X	-
Leveling						
Leveling Status	/rest/modules/leveling/data/status	X	-	-	-	-
Leveling start	/rest/modules/leveling/commands/start	-	X	-	-	X
Leveling cancel	/rest/modules/leveling/commands/cancel	-	X	-	-	-

Name	Endpoint	Get	Post	Del	Put	Async
Draftshield						
Open Doors	/rest/modules/draftshield/commands/open	-	X	-	-	X
Close All	/rest/modules/draftshield/commands/close	-	X	-	-	X
Palm Key	/rest/modules/draftshield/commands/key	-	X	-	-	-
Learning Degree	/rest/modules/draftshield/parameters/openingdegree	X	-	-	X	-
Data Storage Device						
Get records	/rest/modules/devicedatastorage	x	-	-	-	-
Get record	/rest/modules/devicedatastorage/{id}	x	-	-	-	-
Get stats	/rest/modules/devicedatastorage/stats	x	-	-	-	-
Store Weight	/rest/modules/devicedatastorage/commands/storeweight	-	X	-	-	X
Cancel	/rest/modules/devicedatastorage/commands/cancel	-	X	-	-	-
GetAlibi (deprecated)	/rest/modules/devicedatastorage/data/weight	X	-	-	-	-
Calibration Adjustment						
Start	/rest/modules/calibrationadjustment/commands/start	-	X	-	-	X
Cancel	/rest/modules/calibrationadjustment/commands/cancel	-	X	-	-	-
Status	/rest/modules/calibrationadjustment/data/status	X	-	-	-	-
Ionizer						
Start	/rest/modules/ionizer/commands/start	-	X	-	-	X
Cancel	/rest/modules/ionizer/commands/cancel	-	X	-	-	-
Status	/rest/modules/ionizer/state	X	-	-	-	-
Climate Module						
Internal	/rest/modules/internalclimate/data	X	-	-	-	-
External	/rest/modules/externalclimate/data	X	-	-	-	-
Reporting						
Takeover	/rest/modules/reporting/key	-	X	-	-	-
Screen						
Overlay Message	/rest/screen/message	-	X	X	-	-
Task						
Start Task	/rest/start/task	X	X			

Name	Endpoint	Get	Post	Del	Put	Async
QAPP						
QAPP file handling	/qapp/files	X	-	X	X	-
QAPP ReST functions	/qapp/<subURL of specific QAPP>	X	X	X	X	-
Device						
Locking	/rest/device/locking	X	X			
Unlocking	/rest/device/unlocking	X	X			
Restart	/rest/device/restart		X			
Firmware update	/rest/modules/firmware/update		X			
Job Management						
Unit Operations	/rest/unitOperation	X				
Job Handling	/rest/jobManager/job	X	X	X		

2.2 General Information

The following general information is valid for all webservices. However, each webservice could contain additional information.

2.2.1 Header

- cubispass: password for the specified Balance
- Content-Type (valid for POST): application/json

2.2.2 Response Codes

- 200: If the function can be executed successfully
- 202: If action is starting but not finished (Asynchronous)
- 400: If there are missing (mandatory) parameters or empty strings
- 403: If the cubispass is not correct
- 404: If URI or resource not found (e.g. unknown endpoint)
- 405: If a requested method for an endpoint isn't implemented
- 406: If accept header doesn't match
- 409: If another async request is already running or function could not be executed
- 413: If request entity is too large
- 429: if too many HTTP requests are made
- 500: Any server error

2.2.3 Types

The Webservices are divided into asynchronous and synchronous calls.

Synchronous calls return results immediately.

Asynchronous calls return an ID, which can be verified by a global Webservice to return results. These calls do not have a defined time to return results.

Asynchronous call can only be executed one at a time.

2.2.4 Scheme

The same scheme is used throughout all webservices. The format is JSON and composite of the Response parameters of each webservice.

```
{
  "<Response Parameter Name>":<type>,
  "<Response Parameter Name>":<type>
  ....
}
```

For Example:

Response Parameters

name	Type	Mandatory	Example
running	Bool	Yes	True
information	String	Yes	Module information

Json

```
{
  "running":true,
  "information":"Module information"
}
```

The supported data types are:

Type	Json
String	"<property>": "<String>"
Boolean	"<property>": <Boolean>
Number	"<property>": <Number>
Object	"<property>": {<Object Information>}
Array	"<property>": [<Array object>]

2.2.5 Asynchronous Callbacks

On asynchronous calls it is possible for a client to get a callback if the operation is finished. To activate the callback, the client has to transfer the callback URI as "callbackUri" inside the Json data sent in the message body.

For example:

```
{
  "callbackUri" : "https://172.16.244.44/callback"
}
```

In the message body the corresponding async ID is included (see 2.2.3) and the result (according 2.3.1.6).

2.3 Command

2.3.1 Get Command Status

2.3.1.1 Description

Get the command status to the given asyncId (see 2.3.1). The device keeps the result of the last 10 executed asynchronous commands (until power cycle).

2.3.1.2 HTTP Methods

GET

2.3.1.3 Type

Sync

2.3.1.4 Unified Resource identifier

<balance_ip>/rest/commandStatus?asyncId=<Id>

2.3.1.5 Query parameters

name	Type	Mandatory	Example
asyncId	uint64_t	Yes	1234

2.3.1.6 Response Parameters

name	Type	Mandatory	Example
asyncId	uint64_t	Yes	123
running	Bool	Yes	True
result	Object	Yes	Result of asynchronous call

2.3.1.7 Example

Response while request is still running:

```
{
  "asyncId": 123,
  "running": true,
  "result": {}
}
```

2.4 Connector

2.4.1 Dynamic Endpoint

2.4.1.1 Description

- Set HTTP/S dynamic endpoint configuration
- Get HTTP/S dynamic endpoint configuration

Dynamic endpoint will be used in specific connector.

2.4.1.2 HTTP Methods

POST, GET

2.4.1.3 Type

Synchronous

2.4.1.4 Unified Resource identifier

<balance_ip>/rest/modules/connector/dynamic_endpoint

2.4.1.5 Query Parameters

No query parameters

2.4.1.6 Body Parameters

POST:

name	Type	Mandatory	Example
type	String	Yes	"http" or "https"
ip	String	Yes	192.168.10.10
port	Number	Yes	80
endpoint	String	Yes	RestEndpoint

2.4.1.7 Response Parameters

GET:

```
{
  "connector": [
    {
      "type": "HTTP",
      "ip": "192.168.10.10",
      "endpoint": "testEndpoint",
      "port": 3000
    },
    {
```

```
        "type": "HTTPS",
        "ip": "192.168.10.11",
        "endpoint": "HTTPSEndpoint",
        "port": 443.0
    }
]
}
```

2.5 Device

Prevent a user to interact with the device. An overlay is shown to unlock the device. Also, an external system can unlock the balance with the unlock key. Otherwise, a user with the needed role right must enter his credentials on the device.

2.5.1 Locking

2.5.1.1 Description

POST:
Lock the balance with an overlay unlock screen

GET:
Return current device locked state

2.5.1.2 HTTP Methods

POST, GET

2.5.1.3 Type

Synchronous

2.5.1.4 Unified Resource identifier

POST, GET:
<balance_ip>/rest/device/locking

2.5.1.5 Body Parameters

POST:

Name	Type	Mandatory	Example
unlockKey	String	Yes	1234
systemId	String	Yes	Device ID
reason	String	No	This is a reason.

GET: No body parameters

2.5.1.6 Response Parameters

GET: return device locked state

2.5.1.7 Response Codes

POST:
* 400: <Missing Parameter>
* 406: ID/unlock key is empty
* 409: Function not available, Device already locked
* 500: Internal system error. Locking function did not work correctly.

2.5.2 Unlocking

2.5.2.1 Description

POST:

Unlock the balance

GET:

Return current device locked state

2.5.2.2 HTTP Methods

POST, GET

2.5.2.3 Type

Synchronous

2.5.2.4 Unified Resource identifier

POST, GET:

<balance_ip>/rest/device/unlocking

2.5.2.5 Body Parameters

POST:

Name	Type	Mandatory	Example
unlockKey	String	Yes	1234
systemId	String	Yes	Device ID

GET: No body parameters

2.5.2.6 Response Parameters

GET: return device locked state

2.5.2.7 Response Codes

POST:

* 400: <Missing Parameter>

* 403: Forbidden, Unlock key is wrong

* 406: ID/unlock key is empty

* 409: Device is unlocked

* 500: Internal system error. Unlock function did not work correctly

2.5.3 Firmware update

2.5.3.1 Description

Update remotely the firmware version.

2.5.3.2 HTTP Methods

POST

2.5.3.3 Type

Synchronous

2.5.3.4 Unified Resource identifier

POST:

<balance_ip>/rest/firmware/update

2.5.3.5 Body Parameters

Firmware (with *.md5 file) compressed in a *.zip archiv

2.5.3.6 Response Parameters

GET: return device locked state

2.5.3.7 Response Codes

* 409: Not enough space on SD card

* 409: Balance is approved

* 409: Running activity: <Activity>

2.5.4 Restart device

2.5.4.1 Description

Restart device

2.5.4.2 HTTP Methods

POST

2.5.4.3 Type

Synchronous

2.5.4.4 Unified Resource identifier

POST:

<balance_ip>/rest/device/restart

2.5.4.5 Body Parameters

Name	Type	Mandatory	Example
forceReboot	Boolean	No	true false

2.5.4.6 Response Codes

- * 200: OK
- * 403: Password not correct
- * 409: Running activity: <Activity> (forced mode needed)
- * 429: Too many requests

2.6 Device Information

2.6.1 Serial number

2.6.1.1 Description

Get the actual serial number of the balance.

2.6.1.2 HTTP Methods

GET

2.6.1.3 Type

Sync

2.6.1.4 Unified Resource identifier

<balance_ip>/rest/deviceinfo/serialnumber

2.6.1.5 Query parameters

No query parameters

2.6.1.6 Response Parameters

name	Type	Mandatory	Example
package	String	Yes	0037302450

2.6.1.7 Example

Response successful:

```
{  
  "serialNumber": "0037302450"  
}
```

2.6.2 Package

2.6.2.1 Description

Get the actual package version of the balance.

2.6.2.2 HTTP Methods

GET

2.6.2.3 Type

Sync

2.6.2.4 Unified Resource identifier

<balance_ip>/rest/deviceinfo/package

2.6.2.5 Response Parameters

name	Type	Mandatory	Example
package	String	Yes	09-03-03.01.10
cn	String	Yes	3A56

2.6.2.6 Example

Response successful:

```
{  
  "package": "09-03-03.01.10",  
  "cn": "3A56"  
}
```

2.6.3 Model

2.6.3.1 Description

Get the model name of the balance.

2.6.3.2 HTTP Methods

GET

2.6.3.3 Type

Sync

2.6.3.4 Unified Resource identifier

<balance_ip>/rest/deviceinfo/model

2.6.3.5 Query parameters

No query parameters

2.6.3.6 Response Parameters

name	Type	Mandatory	Example
model	String	Yes	MCA225S-2S00-I

2.6.3.7 Example

Response successful:

```
{  
  "model": "MCA225S-2S00-I"  
}
```

2.6.4 APC Version

2.6.4.1 Description

Get the APC version of the balance.

2.6.4.2 HTTP Methods

GET

2.6.4.3 Type

Sync

2.6.4.4 Unified Resource identifier

<balance_ip>/rest/deviceinfo/apc

2.6.4.5 Response Parameters

name	Type	Mandatory	Example
apc	String	Yes	11-02-03.01.10

2.6.4.6 Response Codes

- 405: if APC is not connected

2.6.4.7 Example

Response successful:

```
{
  "apc": "11-02-03.01.10"
}
```

2.6.5 BAC Version

2.6.5.1 Description

Get the BAC version of the balance.

2.6.5.2 HTTP Methods

GET

2.6.5.3 Type

Sync

2.6.5.4 Unified Resource identifier

<balance_ip>/rest/deviceinfo/bac

2.6.5.5 Query parameters

No query parameters

2.6.5.6 Response Parameters

name	Type	Mandatory	Example
bac	String	Yes	00-61-03.01.05

2.6.5.7 Response Codes

- 405: if BAC is not connected

2.6.5.8 Example

Response successful:

```
{  
  "bac": "00-61-03.01.05"  
}
```

2.6.6 WPC Version

2.6.6.1 Description

Get the WPC version of the balance.

2.6.6.2 HTTP Methods

GET

2.6.6.3 Type

Sync

2.6.6.4 Unified Resource identifier

<balance_ip>/rest/deviceinfo/wpc

2.6.6.5 Query parameters

No query parameters

2.6.6.6 Response Parameters

name	Type	Mandatory	Example
wpc	String	Yes	05-08-04.01.12

2.6.6.7 Response Codes

- 405: if WPC is not connected

2.6.6.8 Example

Response successful:

```
{  
  "wpc": "05-08-04.01.12"  
}
```

2.6.7 MCU Version

2.6.7.1 Description

Get the MCU version of the balance.

2.6.7.2 HTTP Methods

GET

2.6.7.3 Type

Sync

2.6.7.4 Unified Resource identifier

<balance_ip>/rest/deviceinfo/mcu

2.6.7.5 Query parameters

No query parameters

2.6.7.6 Response Parameters

name	Type	Mandatory	Example
version	String	Yes	05-08-04.01.12

2.6.7.7 Response Codes

- 405: if MCU is not connected

2.6.7.8 Example

Response successful:

```
{  
  "mcu": "05-08-04.01.12"  
}
```

2.6.8 QAPP-Center

2.6.8.1 Description

Get the QAPP Center version of the balance.

2.6.8.2 HTTP Methods

GET

2.6.8.3 Type

Sync

2.6.8.4 Unified Resource identifier

<balance_ip>/rest/deviceinfo/qappcenter

2.6.8.5 Query parameters

No query parameters

2.6.8.6 Response Parameters

name	Type	Mandatory	Example
qappcenter	String	Yes	09-06-04.00.08

2.6.8.7 Example

Response successful:

```
{  
  "qappcenter": "09-06-04.00.08"  
}
```

2.6.9 Device IDs

2.6.9.1 Description

Get or update the device IDs of the balance. The balance provides two configurable Identifications.

2.6.9.2 HTTP Methods

GET, PUT

2.6.9.3 Type

Sync

2.6.9.4 Unified Resource identifier

<balance_ip>/rest/deviceinfo/deviceid/1

<balance_ip>/rest/deviceinfo/deviceid/2

2.6.9.5 Query parameters

No query parameters.

2.6.9.6 Body (PUT)

name	Type	Mandatory	Example
value	String	Yes	This is an example string

The max. length of the value is 80 characters. Otherwise the error code 413 will be returned.

2.6.9.7 Response Parameters

name	Type	Mandatory	Example
value	String	Yes	This is an example string

2.6.10 General info

2.6.10.1 Description

Get general device information like serial number, mac address, device status, pending reports

2.6.10.2 HTTP Methods

GET

2.6.10.3 Type

Sync

2.6.10.4 Unified Resource identifier

<balance_ip>/rest/deviceinfo/generalinfo

2.6.10.5 Query parameters

2.6.10.6 Response Parameters

name	Type	Mandatory	Example
serialNumber	String	Yes	"0000188647"
pendingAudits	Integer	Yes	5
macAddress	String	Yes	"00:30:d6:1a:45:f8"
deviceType	String	Yes	"BALANCE_CUBIS_2"
deviceStatus	String	Yes	"VALID"
deviceName	String	Yes	"Cubis"
deviceModel	String	Yes	"M-MCx22035"
apiVersion	String	Yes	"1"
pendingReports	Integer	Yes	5
groupId	String	Yes	"Lab 1"
specificInfo	Object	Yes	"qappCenterVersion", "packageVersion", "deviceNameTwo"

2.6.10.7 Example

Response successful:

```
{
  "serialNumber": "0000188647",
  "pendingAudits": 5,
  "macAddress": "00:30:d6:1a:45:f8",
  "deviceType": "BALANCE_CUBIS_2",
  "deviceStatus": "VALID",
  "deviceName": "Cubis",
  "deviceModel": "M-MCx2203S",
  "apiVersion": "1",
  "pendingReports": 2,
  "groupId": "Lab 1",
  "specificInfo": {
    "qappCenterVersion": "09-06-99.99.99",
    "packageVersion": "09-03-08.01.02 CN:C9C8",
```

```
    "deviceNameTwo": "Name 2"  
  }  
}
```

2.6.11 Diagnostic file

2.6.11.1 Description

Collect all device information including log files and error reports as a zip archive and send it back, encoded in BASE64

2.6.11.2 HTTP Methods

GET

2.6.11.3 Type

Sync

2.6.11.4 Unified Resource identifier

<balance_ip>/rest/deviceinfo/diagnosticfile

2.6.11.5 Query parameters

name	Type	Mandatory	Example
content	String Enum: STANDARD, EXTENDED	Yes	STANDARD

2.6.11.6 Response Parameters

Plain text, Base64 encoded zip file with diagnostic data

2.7 Weighing

2.7.1 Tare

2.7.1.1 Description

Perform tare action on balance.

2.7.1.2 HTTP Methods

POST

2.7.1.3 Type

Asynchronous

2.7.1.4 Unified Resource identifier

<balance_ip>/rest/modules/weighing/commands/tare

2.7.1.5 Query Parameters

No query parameters

2.7.1.6 Body Parameters

No body parameters

2.7.1.7 Response Parameters

Name	Type	Mandatory	Example
<i>asyncId</i>	uint64_t	Yes	123

2.7.1.8 Response Codes

- 409: Cannot be executed at the moment

2.7.1.9 Async Response

name	Type	Mandatory	Example
state	String	Yes	STANDSTILL
method	String	Yes	doTare
result	String	Yes	SUCCESS
special	String	Yes	ERR_09

2.7.1.10 Example

Response successful:

```
{
  "asyncId": 123,
  "running": false,
  "result": {
    "state": "FINISHED",
    "method": "doTare",
    "result": "SUCCESS",
    "special": ""
  }
}
```

Response, too low for tare:

```
{
  "asyncId": 124,
  "running": false,
  "result": {
    "state": "CANCELED",
    "method": "doTare",
    "result": "SUCCESS",
    "special": "Err. 09"
  }
}
```

2.7.2 Cancel tare

2.7.2.1 Description

Perform cancel of running tare action on balance.

2.7.2.2 HTTP Methods

POST

2.7.2.3 Type

Synchronous

2.7.2.4 Unified Resource identifier

<balance_ip>/rest/modules/weighing/commands/canceltare

2.7.2.5 Query Parameters

No query parameters

2.7.2.6 Body Parameters

No body parameters

2.7.2.7 Response Parameters

No response parameters

2.7.2.8 Async Response

name	Type	Mandatory	Example
state	String	Yes	STANDSTILL
method	String	Yes	doTare
result	String	Yes	SUCCESS
special	String	Yes	ERR_09

2.7.2.9 Example

Response after request is canceled when checking the command status:

```
{
  "asyncId": 1,
  "running": false,
  "result": {
    "state": "CANCELED",
    "method": "doTare",
    "result": "",
    "special": ""
  }
}
```

2.7.3 Zero

2.7.3.1 Description

Perform zero action on balance.

2.7.3.2 HTTP Methods

POST

2.7.3.3 Type

Asynchronous

2.7.3.4 Unified Resource identifier

<balance_ip>/rest/modules/weighing/commands/zero

2.7.3.5 Query Parameters

No query parameters

2.7.3.6 Body Parameters

No body parameters

2.7.3.7 Response Parameters

Name	Type	Mandatory	Example
<i>asyncId</i>	uint64_t	Yes	123

2.7.3.8 Response Codes

- 409: Cannot be executed at the moment

2.7.3.9 Async Response

name	Type	Mandatory	Example
state	String	Yes	STANDSTILL
method	String	Yes	doTare
result	String	Yes	SUCCESS
special	String	Yes	ERR_09

2.7.3.10 Example

Response successful:

```
{
  "asyncId": 123,
  "running": false,
  "result": {
    "state": "FINISHED",
    "method": "doZero",
```

```
    "result": "SUCCESS",
    "special": ""
  }
}
```

Response, too high for zeroing:

```
{
  "asyncId": 124,
```

```
"running": false,
"result": {
  "state": " CANCELED",
  "method": "doZero",
}
"result": "SUCCESS",
"special": "Err. 08"
```

2.7.4 Cancel zero

2.7.4.1 Description

Perform cancel of running zero action on balance.

2.7.4.2 HTTP Methods

POST

2.7.4.3 Type

Synchronous

2.7.4.4 Unified Resource identifier

<balance_ip>/rest/modules/weighing/commands/cancelzero

2.7.4.5 Query Parameters

No query parameters

2.7.4.6 Body Parameters

No body parameters

2.7.4.7 Response Parameters

No response parameters

2.7.4.8 Example

Response after request is canceled when checking the command status:

```
{
  "asyncId": 1,
  "running": false,
  "result": {
    "state": "CANCELED",
    "method": "doZero",
    "result": "",
    "special": ""
  }
}
```

2.7.5 Zero/Tare

2.7.5.1 Description

Perform zero or tare action on balance.

2.7.5.2 HTTP Methods

POST

2.7.5.3 Type

Asynchronous

2.7.5.4 Unified Resource identifier

<balance_ip>/rest/modules/weighing/commands/cancelzerotare

2.7.5.5 Query Parameters

No query parameters

2.7.5.6 Body Parameters

No body parameters

2.7.5.7 Response Parameters

Name	Type	Mandatory	Example
<i>asyncId</i>	uint64_t	Yes	123

2.7.5.8 Response Codes

- 409: Cannot be executed at the moment

2.7.5.9 Async Response

name	Type	Mandatory	Example
state	String	Yes	STANDSTILL
method	String	Yes	doTare
result	String	Yes	SUCCESS
special	String	Yes	ERR_09

2.7.5.10 Example

Response successful:

```
{
  "asyncId": 123,
  "running": false,
  "result": {
    "state": "FINISHED",
    "method": "doZeroTare",
    "result": "SUCCESS",
    "special": ""
  }
}
```

Response at overload:

```
{
  "asyncId": 124,
  "running": false,
  "result": {
    "state": "CANCELED",
    "method": "doZeroTare",
    "result": "SUCCESS",
    "special": "High"
  }
}
```

2.7.6 Cancel zero/tare

2.7.6.1 Description

Perform cancel of running zero/tare action on balance.

2.7.6.2 HTTP Methods

POST

2.7.6.3 Type

Synchronous

2.7.6.4 Unified Resource identifier

<balance_ip>/rest/modules/weighing/commands/cancelzerotare

2.7.6.5 Query Parameters

No query parameters

2.7.6.6 Body Parameters

No body parameters

2.7.6.7 Response Parameters

No response parameters

2.7.6.8 Example

Response after request is canceled when checking the command status:

```
{
  "asyncId": 1,
  "running": false,
  "result": {
    "state": "CANCELED",
    "method": "doZeroTare",
    "result": "",
    "special": ""
  }
}
```

2.7.7 Value

2.7.7.1 Description

Get the actual weight value of the balance. Independently on the stable flag.

2.7.7.2 HTTP Methods

GET

2.7.7.3 Type

Synchronous

2.7.7.4 Unified Resource identifier

<balance_ip>/rest/modules/weighing/data/value

2.7.7.5 Response Parameters

name	type	Mandatory	Example
number	String	Yes	1.2345
sign	String	Yes	+
mark	String	Yes	N
unit	String	Yes	g
auxiliaryDigits	String	Yes	0
interval	Double	Yes	1.0E-4
isSpecial	Bool	Yes	false
isInvalid	Bool	Yes	false
isStable	Bool	Yes	true
isWarning	Bool	Yes	false
Decimals	Int	Yes	4

2.7.7.6 Example

Response successful:

```
{
  "number": "1.2345",
  "sign": "+",
  "mark": "N",
  "unit": "g",
  "auxiliaryDigits": 0,
  "interval": 0.0001,
  "isSpecial": false,
  "isInvalid": false,
  "isStable": true,
  "isWarning": false,
  "decimals": 4
}
```

2.7.8 Gross calc

2.7.8.1 Description

Get the actual weight value of the balance. Independently on the stable flag.

2.7.8.2 HTTP Methods

GET

2.7.8.3 Type

Synchronous

2.7.8.4 Unified Resource identifier

<balance_ip>/rest/modules/weighing/data/value

2.7.8.5 Response Parameters

name	type	Mandatory	Example
number	String	Yes	1.2345
sign	String	Yes	+
mark	String	Yes	N
unit	String	Yes	g
auxiliaryDigits	String	Yes	0
interval	Double	Yes	1.0E-4
isSpecial	Bool	Yes	false
isInvalid	Bool	Yes	false
isStable	Bool	Yes	true
isWarning	Bool	Yes	false
Decimals	Int	Yes	4

2.7.8.6 Example

Response successful:

```
{
  "number": "1.2345",
  "sign": "+",
  "mark": "N",
  "unit": "g",
  "auxiliaryDigits": 0,
  "interval": 0.0001,
  "isSpecial": false,
  "isInvalid": false,
  "isStable": true,
  "isWarning": false,
  "decimals": 4
}
```

2.7.9 Tare storage

2.7.9.1 Description

Get, set or clear the tare storage 0 (get and clear only), 1 or 2 of balance.

Remark: The unit is not changed. It's that one that were configured within the last started weighing task.

2.7.9.2 HTTP Methods

PUT, GET, DELETE

2.7.9.3 Type

Synchronous

2.7.9.4 Unified Resource identifier

<balance_ip>/rest/modules/weighing/data/tare/0

<balance_ip>/rest/modules/weighing/data/tare/1

<balance_ip>/rest/modules/weighing/data/tare/2

2.7.9.5 Query Parameters

No query parameters

2.7.9.6 Body Parameters (PUT only)

name	Type	Mandatory	Example
value	Double	Yes	23.453

Value should be between: 0.0 ... balance maximum, otherwise 400 "Bad request" returned.

2.7.9.7 Response Parameters (GET only)

name	type	Mandatory	Example
number	String	Yes	100.0000
sign	String	Yes	+
mark	String	Yes	T1
unit	String	Yes	g
auxiliaryDigits	String	Yes	0
interval	Double	Yes	1.0E-4
isSpecial	Bool	Yes	false
isInvalid	Bool	Yes	false
isStable	Bool	Yes	true
isWarning	Bool	Yes	false
Decimals	Int	Yes	4

2.7.9.8 Response Codes

- 404: Not available (valid for tare 0)
- 500: internal Server Error

2.7.9.9 Example

Response successful:

```
{
  "number": "100.0000",
  "sign": "+",
  "mark": "T1",
  "unit": "g",
  "auxiliaryDigits": 0,
  "interval": 0.00001,
  "isSpecial": false,
  "isInvalid": false,
  "isStable": true,
  "isWarning": false,
  "decimals": 5
}
```

Response failure if higher tare storage is already set:

Error during set tare value: XBPIRESP_FUNCTION_LOCKED

Response failure if higher tare storage is still set:

Error during clear tare value: XBPIRESP_FUNCTION_LOCKED

2.7.10 Display

2.7.10.1 Description

Get the current value shown at the display.

2.7.10.2 HTTP Methods

GET

2.7.10.3 Type

Synchronous

2.7.10.4 Unified Resource identifier

<balance_ip>/rest/modules/weighing/data/display

2.7.10.5 Query Parameters

No query parameters

2.7.10.6 Response Parameters

These parameters will be returned:

name	type	Mandatory	Examples
number	String	Yes	1.2345
sign	String	Yes	+
mark	String	Yes	G
unit	String	Yes	g
auxiliaryDigits	Int	Yes	0
interval	Float	Yes	0.0001
isSpecial	Bool	Yes	false
isInvalid	Bool	Yes	false
isStable	Bool	Yes	false
isWarning	Bool	Yes	false
decimals	Int	Yes	4

2.7.10.7 Example

Response successful:

```
{
  "number": "1.2345",
  "sign": "+",
  "mark": "G",
  "unit": "g",
  "auxiliaryDigits": 0,
  "interval": 0.00001,
  "isSpecial": false,
  "isInvalid": false,
  "isStable": true,
  "isWarning": false,
  "decimals": 5
}
```

Response at overload:

```
{
  "number": "High",
  "sign": "",
  "mark": "",
  "unit": "g",
  "auxiliaryDigits": 0,
  "interval": 0.00001,
  "isSpecial": true,
  "isInvalid": false,
  "isStable": false,
  "isWarning": false,
  "decimals": 4
}
```

2.7.11 Metrology

2.7.11.1 Description

Get the metrology data of the balance.

2.7.11.2 HTTP Methods

GET

2.7.11.3 Type

Synchronous

2.7.11.4 Unified Resource identifier

<balance_ip>/rest/modules/weighing/data/metrology

2.7.11.5 Query Parameters

No query parameters

2.7.11.6 Body Parameters

No body parameters

2.7.11.7 Response Parameters

name	Type	Mandatory	Example
auxiliaryDigits	Int	Yes	0
accuracyClass	Int	Yes	1
rangeSwitchMode	Int	Yes	0
multiRangeMode	Int	Yes	0
overflowIndex	Int	Yes	0
decimals	Int	Yes	4
unit	String	Yes	g
numRanges	Int	Yes	1
ranges	Array<Range>	Yes	[<Range>,<Range>]

Range Objects

name	Type	Mandatory	Example
min	Float	Yes	0.01
max	Float	Yes	520.0
e	Float	Yes	0.001
d	Float	Yes	0.0001

2.7.11.8 Example

Response successful:

```
{
  "auxiliaryDigits": 0,
  "accuracyClass": 1,
  "rangeSwitchMode": 0,
  "multiRangeMode": 0,
  "overflowIndex": 0,
  "decimals": 4,
  "unit": "g",
  "numRanges": 1,
  "ranges": [
    {
      "min": 0.01,
      "max": 520.0,
      "e": 0.001,
      "d": 0.0001
    }
  ]
}
```

2.7.12 Unit

2.7.12.1 Description

Get or set the current unit selected of the balance.

Remark: The allowed units to set are the ones that were configured within the last started weighing task.

Following values are allowed:

Value	Description	Abbreviation
MILLIGRAM	Milligrams	mg
GRAM	Grams	g
KILOGRAM	Kilograms	kg
CARAT	Carats	ct
POUND	Pounds	lb
OUNCE	Ounces	oz
TROYOUNCE	Troy ounces	ozl
PENNYWEIGHT	Pennyweights	dwl
GRAIN	Grains	GN
TAEI_HONGKONG	Hong Kong tael	tlh
TAEI_SINGAPORE	Singapore tael	tl
TAEI_TAIWAN	Taiwanese tael	tl
TAEI_CHINA	Chinese tael	tl
MOMME	Momme	mm
TOLA	Tola	tl
BATH	Baht	bt
MESGHAL	Mesghal	MS
NEWTON	Newtons	N

2.7.12.2 HTTP Methods

GET, PUT

2.7.12.3 Type

Synchronous

2.7.12.4 Unified Resource identifier

<balance_ip>/rest/modules/weighing/data/unit

2.7.12.5 Query Parameters

No query parameters

2.7.12.6 Body Parameters (PUT only)

name	Type	Mandatory	Example
value	String	Yes	MILLIGRAM

Invalid Unit string results in: 404 Not found

Balance not able to set unit: 500 Internal Server Error

2.7.12.7 Response Parameters (GET only)

name	Type	Mandatory	Example
unit	String	Yes	MILLIGRAM

2.7.12.8 Example

Response GET successful:

```
{  
  "unit": "MILLIGRAM"  
}
```

2.8 Data Storage Device

2.8.1 Get records

2.8.1.1 Description

Gets records from the data storage device.

- Backwards compatible: If only `id` is provided and no other filters, a single record is returned.

- filter by ID range or Unix timestamp range, choose basic vs. full records, and use offset or cursor pagination.

Records limited to 100 per call to avoid inoperability issues.

2.8.1.2 HTTP Methods

GET

2.8.1.3 Type

Synchronous

2.8.1.4 Unified Resource identifier

<balance_ip>/rest/modules/devicedatastorage

2.8.1.5 Query Parameters

name	Type	Mandatory	Description
id	Uint64_t	Yes	Global record id
Count	Integer	No	Number of records to be retrieved; default = 1
meta	Boolean	No	Include record meta data

2.8.1.6 Body Parameters

No body parameters

2.8.1.7 Response Parameters

See 2.8.7 Data storage record structure

2.8.2 Get single record

2.8.2.1 Description

Gets single record from the data storage device.

2.8.2.2 HTTP Methods

GET

2.8.2.3 Type

Synchronous

2.8.2.4 Unified Resource identifier

<balance_ip>/rest/modules/devicedatastorage/{id}

2.8.2.5 Query Parameters

name	Type	Mandatory	Description
Id	Uint64_t	Yes	Provided in path, Global record id
meta	Boolean	No	Include record meta data

2.8.2.6 Body Parameters

No body parameters

2.8.2.7 Response Parameters

See 2.8.7 Data storage record structure

2.8.2.8 Example response

Record contains Net, tare, gross values for DSD ID 3:

```
{
  "timestamp": 1755076091.606,
  "alibiId": 3,
  "comment": "Demo",
  "values": [
    {
      "number": "3.57258",
      "sign": "+",
      "mark": "N",
      "unit": "g",
      "auxiliaryDigits": 0,
      "interval": 0.00001,
      "isSpecial": false,
      "isInvalid": false,
      "isStable": true,
      "isWarning": false,
    }
  ]
}
```

```
    "decimals": 5
  },
  {
    "number": "66.97777",
    "sign": "+",
    "mark": "T",
    "unit": "g",
    "auxiliaryDigits": 0,
    "interval": 0.00001,
    "isSpecial": false,
    "isInvalid": false,
    "isStable": true,
    "isWarning": false,
    "decimals": 5
  },
  {
    "number": "70.55035",
    "sign": "+",
    "mark": "GC",
    "unit": "g",
    "auxiliaryDigits": 0,
    "interval": 0.00001,
    "isSpecial": false,
    "isInvalid": false,
    "isStable": true,
    "isWarning": false,
    "decimals": 5
  }
]
```

2.8.3 Get DSD information

2.8.3.1 Description

Gets general information of the DSD database.

2.8.3.2 HTTP Methods

GET

2.8.3.3 Type

Synchronous

2.8.3.4 Unified Resource identifier

<balance_ip>/rest/modules/devicedatastorage/stats

2.8.3.5 Query Parameters

name	Type	Mandatory	Description
Fields	Array of string	Yes	Fields to retrieve
fromTs	Integer	No	Begin of time span
toTs	Integer	No	End of time span

Following fields are supported:

- **minId**: minimal record ID in database / given time span
- **maxId**: maximum record ID in database / given time span
- **count**: number of records in database / given time span
- **capacityPercent**: percentage of used and not stored database memory that is guaranteed to be stored for a certain amount of time.

2.8.3.6 Body Parameters

No body parameters

2.8.3.7 Response Parameters

Depending on the requested fields; if no fields are given all fields are returned. If a time span is given the time is also returned.

Name	Type	Mandatory	Example
<i>minId</i>	integer	No	1
<i>maxId</i>	integer	No	302153
<i>Count</i>	integer	No	200043
<i>capacityPercent</i>	integer	No	8 = 8%

2.8.4 Store Weight

2.8.4.1 Description

Store a stable measure weight to the data storage device. The command will wait until the measure weight is stable.

2.8.4.2 HTTP Methods

POST

2.8.4.3 Type

Asynchronous

2.8.4.4 Unified Resource identifier

<balance_ip>/rest/modules/devicedatastorage/commands/storeweight

2.8.4.5 Query Parameters

No query parameters

2.8.4.6 Body Parameters

name	Type	Mandatory	Example
comment	String	No	This is a comment

The max. length of the comment is 80 characters. Otherwise the error code 413 will be returned.

2.8.4.7 Response Parameters

Name	Type	Mandatory	Example
<i>asyncId</i>	uint64_t	Yes	123

2.8.4.8 Async Response

See 2.8.7 Data storage record structure

2.8.4.9 Example

Response after request is finished when checking the command status:

```
{
  "asyncId": 1,
  "running": false,
  "result": {
    "timestamp": 1576162077.741,
    "alibiId": 123,
    "comment": "This is a comment",
    "values": [
      {
        "number": "1.2345",
        "sign": "+",
        "mark": "N",
        "unit": "g",
        "auxiliaryDigits": 0,
        "interval": 0.0001,
        "isSpecial": false,
        "isInvalid": false,
        "isStable": true,
        "isWarning": false,
        "decimals": 4
      }
    ]
  }
}
```

2.8.5 Cancel

2.8.5.1 Description

Cancels a running store weight action.

2.8.5.2 HTTP Methods

POST

2.8.5.3 Type

Synchronous

2.8.5.4 Unified Resource identifier

<balance_ip>/rest/modules/devicedatastorage/commands/cancel

2.8.5.5 Query Parameters

No query Parameters

2.8.5.6 Body Parameters

No body parameters

2.8.5.7 Response Parameters

No Response Parameters

2.8.5.8 Example

Response after request is canceled when checking the command status:

```
{
  "asyncId": 1,
  "running": false,
  "result": {
    "state": "CANCELED",
    "alibiId": -1,
    "weightBlock": {}
  }
}
```

2.8.6 Get weight

2.8.6.1 Description

Gets a stored measured weight of the data storage device.

Kept for compatibility reasons. Use 0

Get single record

2.8.6.2 HTTP Methods

GET

2.8.6.3 Type

Synchronous

2.8.6.4 Unified Resource identifier

<balance_ip>/rest/modules/devicedatastorage/data/weight?id=<alibild>

2.8.6.5 Query Parameters

name	Type	Mandatory	Example
alibild	int64_t	Yes	123

Valid values for alibild are 1 ... max (int64_t).

2.8.6.6 Body Parameters

No body parameters

2.8.6.7 Response Parameters

See 2.8.7 Data storage record structure

2.8.6.8 Response Codes

- 404: not found

2.8.6.9 Example

Response when tare is used:

```
{
  "timestamp": 1576162077.741,
  "alibiId": 123,
  "comment": "This is a comment",
  "values": [
    {
      "number": "1.2345",
      "sign": "+",
      "mark": "N",
      "unit": "g",
      "auxiliaryDigits": 0,
      "interval": 0.0001,
      "isSpecial": false,
      "isInvalid": false,
      "isStable": true,
      "isWarning": false,
      "decimals": 4
    },
    {
      "number": "1.0000",
      "sign": "+",

```

```

        "mark": "T",
        "unit": "g",
        "auxiliaryDigits": 0,
        "interval": 0.0001,
        "isSpecial": false,
        "isInvalid": false,
        "isStable": true,
        "isWarning": false,
        "decimals": 4
    }
    {
        "number": "2.2345",
        "sign": "+",
        "mark": "GC",
        "unit": "g",
        "auxiliaryDigits": 0,
        "interval": 0.0001,
        "isSpecial": false,
        "isInvalid": false,
        "isStable": true,
        "isWarning": false,
        "decimals": 4
    }
    ]
}
```

2.8.7 Data storage record structure

General structure

Name	Type	Mandatory	Example
<i>timestamp</i>	Integer	Yes	1755076091
<i>alibild</i>	Integer	Yes	23
<i>comment</i>	String	Yes	Comment
<i>values</i>	Array of values	Yes	[]

Value structure

Name	Type	Mandatory	Example
<i>number</i>	Double as string	Yes	3.57258
<i>sign</i>	String	Yes	“+” or “-”
<i>mark</i>	String	Yes	“N”, “T”, “G”, “GC”
<i>unit</i>	String	Yes	“mg”, “g”, “kg”
<i>auxiliaryDigits</i>	Integer	Yes	0, 2
<i>interval</i>	Double	Yes	0.00001 (resolution of balance)
<i>isSpecial</i>	Boolean	Yes	
<i>isInvalid</i>	Boolean	Yes	
<i>isStable</i>	Boolean	Yes	
<i>isWarning</i>	Boolean	Yes	
<i>decimals</i>	Integer	Yes	3, 4, 5

2.9 Leveling

2.9.1 Status

2.9.1.1 Description

Get the actual status of elise.

2.9.1.2 HTTP Methods

GET

2.9.1.3 Type

Synchronous

2.9.1.4 Unified Resource identifier

<balance_ip>/rest/modules/leveling/data/status

2.9.1.5 Response Parameters

name	type	Mandatory	Example
isLeveled	Bool	Yes	true
pRadius	Double	Yes	106.83164325236227
pAngle	Double	Yes	0.7522976746821101
xCoordinate	Double	Yes	78,0
yCoordinate	Double	Yes	108
criticalRadius	Double	Yes	65
maxRadius	Double	Yes	108

2.9.1.6 Response Codes

- 409: Cannot be executed at the moment

2.9.2 Start

2.9.2.1 Description

Start leveling action. Independent of automatic or manual footsteps.

2.9.2.2 HTTP Methods

POST

2.9.2.3 Type

Asynchronous

2.9.2.4 Unified Resource identifier

<balance_ip>/rest/modules/leveling/commands/start

2.9.2.5 Response Parameters

Name	Type	Mandatory	Example
<i>asyncId</i>	uint64_t	Yes	123

2.9.2.6 Response Codes

- 409: Cannot be executed at the moment

2.9.3 Cancel

2.9.3.1 Description

Cancel leveling function if the action is actually performing.

2.9.3.2 HTTP Methods

POST

2.9.3.3 Type

Synchronous

2.9.3.4 Unified Resource identifier

<balance_ip>/rest/modules/leveling/commands/cancel

2.9.3.5 Query Parameters

No query parameters

2.9.3.6 Body Parameters

No body parameters

2.9.3.7 Async Response

name	Type	Mandatory	Example
State	String	Yes	FINISHED_AUTOM ATIC LEVELING_ERROR

2.10 Draftshield

2.10.1 Open

2.10.1.1 Description

Open doors of linear or rotation or both draftshields. Possible doors are right, left and top (only linear) or combinations of them. Supported commands are:

Command	Linear draftshield	Rotation draftshield	Linear&Rotation draftshield
OPEN_LEFT	X	X	X
OPEN_RIGHT	X	X	X
OPEN_TOP	X	-	X
OPEN_LEFT_RIGHT	X	-	X
OPEN_LEFT_TOP	X	-	X
OPEN_RIGHT_TOP	X	-	X
OPEN_ALL	X	-	X

Alternatively, rotation draftshield door opens to a given opening degree on the shortest way.

2.10.1.2 HTTP Methods

POST

2.10.1.3 Type

Asynchronous

Unified Resource identifier

<balance_ip>/rest/modules/draftshield/commands/open

2.10.1.4 Query Parameters

No query parameters

2.10.1.5 Body Parameters

For balance with linear draftshield

Possible Strings for doors are RIGHT, LEFT and TOP

name	Type	Mandatory	Example
door	String	Yes	OPEN_RIGHT

2.10.1.6 Response Parameters

Name	Type	Mandatory	Example
<i>asynclid</i>	uint64_t	Yes	123

2.10.1.7 Response Codes

- 409: Cannot be executed at the moment

2.10.1.8 Async Response

name	Type	Mandatory	Example
State	String	Yes	STANDSTILL

2.10.2 OpenRowiDirect

2.10.2.1 Description

Rotate the draftshield to the specific angle.

2.10.2.2 HTTP Methods

POST

2.10.2.3 Type

Asynchronous

Unified Resource identifier

<balance_ip>/rest/modules/draftshield/commands/openrotation

2.10.2.4 Query Parameters

No query parameters

2.10.2.5 Body Parameters

The parameter 'openingDegree' can be set in a range of 45 to 315.

name	Type	Mandatory	Example
openingDegree	Int	Yes	160

2.10.2.6 Response Parameters

Name	Type	Mandatory	Example
<i>asyncId</i>	uint64_t	Yes	123

2.10.2.7 Async Response

name	Type	Mandatory	Example
State	String	Yes	STANDSTILL

2.10.3 Close All

2.10.3.1 Description

Cancel leveling function if the action is actually performing and close draftshield.

2.10.3.2 HTTP Methods

POST

2.10.3.3 Type

Asynchronous

2.10.3.4 Unified Resource identifier

<balance_ip>/rest/modules/draftshield/commands/close

2.10.3.5 Query Parameters

No query parameters

2.10.3.6 Body Parameters

No body parameters

2.10.3.7 Response Parameters

Name	Type	Mandatory	Example
<i>asynclid</i>	uint64_t	Yes	123

2.10.3.8 Response Codes

409: Cannot be executed at the moment

2.10.3.9 Async Response

name	Type	Mandatory	Example
State	String	Yes	STANDSTILL

2.10.4 Palm key

2.10.4.1 Description

Performs click event for Right or Left Palm Key.

2.10.4.2 HTTP Methods

POST

2.10.4.3 Type

Synchronous

2.10.4.4 Unified Resource identifier

<balance_ip>/rest/modules/draftshield/commands/key

2.10.4.5 Query Parameters

No query parameters

2.10.4.6 Body Parameters

name	Type	Mandatory	Example
palmKey	String	Yes	RIGHT

Valid PalmKey values are LEFT and RIGHT.

2.10.5 Learning Degree

2.10.5.1 Description

Learning degree stands for the opening angle of the rotational draftshield. It is possible to set the learning degree for both palm keys separately.

2.10.5.2 HTTP Methods

PUT, GET

2.10.5.3 Type

Synchronous

2.10.5.4 Unified Resource identifier

<balance_ip>/rest/modules/draftshield/parameters/openingdegree/left
<balance_ip>/rest/modules/draftshield/parameters/openingdegree/right

2.10.5.5 Query Parameters

No query parameters.

2.10.5.6 Body Parameters (PUT only)

name	Type	Mandatory	Example
openingDegree	Int	Yes	140

Valid values for openingDegree are 45 ... 315. Otherwise the error code 400 will be returned.

2.10.5.7 Response Parameters (GET only)

name	Type	Mandatory	Example
openingDegree	Int	Yes	140

2.10.6 Status

2.10.6.1 Description

Get the current status of the draftshield.

2.10.6.2 HTTP Methods

GET

2.10.6.3 Type

Synchronous

2.10.6.4 Unified Resource identifier

<balance_ip>/rest/modules/draftshield/state

2.10.6.5 Query Parameters

No query parameters

2.10.6.6 Body Parameters

No body parameters

2.10.6.7 Response Parameters

name	Type	Mandatory	Example
status	String	Yes	W 000180

2.11 Calibration Adjustment

2.11.1 Start

2.11.1.1 Description

Start specific calibration function.

The possible actions are:

- CAL_EXTERN_DEFAULT
- CAL_EXTERN_USER
- CAL_INTERN
- LIN_INTERN
- SET_PRELOAD
- DELETE_PRELOAD
- DET_INT_LIN_WEIGHT
- DET_INT_CAL_WEIGHT

2.11.1.2 HTTP Methods

POST

2.11.1.3 Type

Asynchronous

2.11.1.4 Unified Resource identifier

<balance_ip>/rest/modules/calibrationadjustment/commands/start

2.11.1.5 Query Parameters

No query parameters

2.11.1.6 Body Parameters

name	Type	Mandatory	Example
function	String	Yes	CAL_INTERN

2.11.1.7 Response Parameters

Name	Type	Mandatory	Example
<i>asyncId</i>	uint64_t	Yes	123

2.11.1.8 Response Codes

- 409: Cannot be executed at the moment

2.11.1.9 Async Response

name	Type	Mandatory	Example
State	String	Yes	FINISHED / REPORT

2.11.2 Cancel

2.11.2.1 Description

Cancels the current calibration action.

2.11.2.2 HTTP Methods

POST

2.11.2.3 Type

Synchronous

2.11.2.4 Unified Resource identifier

<balance_ip>/rest/modules/calibrationadjustment/commands/cancel

2.11.2.5 Query Parameters

No query parameters

2.11.2.6 Body Parameters

No body parameters

2.11.2.7 Response Parameters

No Response Parameters

2.11.3 Status

2.11.3.1 Description

Get the current status of the calibration adjustment.

Possible functions are:

- INITIAL
- IDLE
- ACTIVE
- PREISOCAL
- DOING
- PREINTERN
- PREEXTERN
- CANCEL
- PRELOAD
- REPORT
- ERROR

2.11.3.2 HTTP Methods

GET

2.11.3.3 Type

Synchronous

2.11.3.4 Unified Resource identifier

<balance_ip>/rest/modules/calibrationadjustment/state

2.11.3.5 Query Parameters

No query parameters

2.11.3.6 Body Parameters

No body parameters

2.11.3.7 Response Parameters

name	Type	Mandatory	Example
function	String	Yes	CAL_EXTERN_DEFAULT

2.12 Ionizer

2.12.1 Start

2.12.1.1 Description

Start ionizer with defined time and Intensity. If no information's will be sent the values of the settings menu are taken. The time is in seconds defined.

2.12.1.2 HTTP Methods

POST

2.12.1.3 Type

Asynchronous

2.12.1.4 Unified Resource identifier

<balance_ip>/rest/modules/ionizer/commands/start

2.12.1.5 Query Parameters

No query Parameters

2.12.1.6 Body Parameters

name	Type	Mandatory	Example
time	Int	No	2
intensity	String	No	SOFT

Possible intensities are WEAK, MEDIUM and STRONG.

The range for time is 1 ... 60 seconds.

2.12.1.7 Response Parameters

Name	Type	Mandatory	Example
<i>asyncId</i>	uint64_t	Yes	123

2.12.1.8 Response Codes

- 409: Cannot be executed at the moment

2.12.1.9 Async Response

name	Type	Mandatory	Example
State	String	Yes	OFF / ERROR

2.12.2 Cancel

2.12.2.1 Description

Cancels the current calibration action.

2.12.2.2 HTTP Methods

POST

2.12.2.3 Type

Synchronous

2.12.2.4 Unified Resource identifier

<balance_ip>/rest/modules/ionizer/commands/cancel

2.12.2.5 Query Parameters

No query parameters

2.12.2.6 Body Parameters

No body parameters

2.12.2.7 Response Parameters

No Response Parameters

2.12.3 Status

2.12.3.1 Description

Get the current status of the ionizer.
The given time is in seconds defined.

2.12.3.2 HTTP Methods

GET

2.12.3.3 Type

Synchronous

2.12.3.4 Unified Resource identifier

<balance_ip>/rest/modules/ionizer/data/status

2.12.3.5 Query Parameters

No query parameters

2.12.3.6 Body Parameters

No body parameters

2.12.3.7 Response Parameters

name	Type	Mandatory	Example
status	String	Yes	I001

2.13 Climate Module

2.13.1 Internal

2.13.1.1 Description

Get the environment information of the internal climate module.

2.13.1.2 HTTP Methods

GET

2.13.1.3 Type

Synchronous

2.13.1.4 Unified Resource identifier

<balance_ip>/rest/modules/internalclimate/data

2.13.1.5 Query Parameters

No query parameters

2.13.1.6 Body Parameters

No body parameters

2.13.1.7 Response Parameters

name	Type	Mandatory	Example
temperature	Double	Yes	23.3
temperatureUnit	String	Yes	°C
humidity	Double	Yes	25.1
humidityUnit	String	Yes	%
airPressure	Double	Yes	998.0
pressureUnit	String	Yes	hPa
calibrationDate	Timestamp	Yes	123456
nextCalibration	Timestamp	No	123455

2.13.2 External

2.13.2.1 Description

Get the environment information from the external climate module.

2.13.2.2 HTTP Methods

GET

2.13.2.3 Type

Synchronous

2.13.2.4 Unified Resource identifier

<balance_ip>/rest/modules/externalclimate/data

2.13.2.5 Query Parameters

No query parameters

2.13.2.6 Body Parameters

No body parameters

2.13.2.7 Response Parameters

name	Type	Mandatory	Example
temperature	Float	Yes	23.3
temperatureUnit	String	Yes	°C
humidity	Float	Yes	25.1
humidityUnit	String	Yes	%
airPressure	Float	Yes	998.0
pressureUnit	String	Yes	hPa
calibrationDate	Int	Yes	123456
nextCalibration	Int	No	123455

2.13.3 Internal Information

2.13.3.1 Description

Get the device information of the internal climate module.

2.13.3.2 HTTP Methods

GET

2.13.3.3 Type

Synchronous

2.13.3.4 Unified Resource identifier

<balance_ip>/rest/modules/internalclimate/data/information

2.13.3.5 Query Parameters

No query parameters

2.13.3.6 Body Parameters

No body parameters

2.13.3.7 Response Parameters

name	Type	Mandatory	Example
calCertificate	Timestamp	Yes	123456
calLaboratory	Timestamp	Yes	123456
calibrationDate	Timestamp	Yes	123456
nextCalibration	Timestamp	Yes	123455

2.13.4 External Information**2.13.4.1 Description**

Get the device information of the external climate module.

2.13.4.2 HTTP Methods

GET

2.13.4.3 Type

Synchronous

2.13.4.4 Unified Resource identifier

<balance_ip>/rest/modules/externalclimate/data/information

2.13.4.5 Query Parameters

No query parameters

2.13.4.6 Body Parameters

No body parameters

2.13.4.7 Response Parameters

name	Type	Mandatory	Example
calCertificate	Timestamp	Yes	123456
calLaboratory	Timestamp	Yes	123456
calibrationDate	Timestamp	Yes	123456
nextCalibration	Timestamp	Yes	123455

2.14 Reporting

2.14.1 Takeover

2.14.1.1 Description

Perform takeover function on Weighing Screen. Will be only performed if the takeover button is available on Screen.

2.14.1.2 HTTP Methods

Post

2.14.1.3 Type

Synchronous

2.14.1.4 Unified Resource identifier

<balance_ip>/rest/modules/reporting/key

2.14.1.5 Query Parameters

No query parameters

2.14.1.6 Body Parameters

name	Type	Mandatory	Example
function	String	Yes	TAKEOVER

possible Functions are: TAKEOVER

2.14.1.7 Response Parameters

No Response Parameters

2.15 Audit Trail

2.15.1 Unarchived Records

Get all unarchived audit trail records from the device and set audit trail records as archived

2.15.1.1 HTTP Methods

Post

2.15.1.2 Type

Synchronous

2.15.1.3 Unified Resource identifier

<balance_ip>/rest/modules/audit/records

2.15.1.4 Query Parameters

No query parameters

2.15.1.5 Body Parameters

name	Type	Mandatory	Example
method	String	Yes	ARCHIVED

GET:

No body parameters

2.15.1.6 Response Parameters

GET: returns a list with all unarchived audit trail records from the device and set audit trail records as archived

2.15.2 Records

2.15.2.1 HTTP Methods

POST, GET

2.15.2.2 Type

Synchronous

2.15.2.3 Unified Resource identifier

<balance_ip>/rest/ modules/audit/records

2.15.2.4 Query Parameters

No query parameters

2.15.2.5 Body Parameters

POST:

Formatted JSON object, containing following properties:

Name	Type	Mandatory	Example
title	String	Yes	Test Title
content	String	Yes	Test Message

2.15.2.6 Response Parameters

2.15.2.7 No Response Codes

POST:

* 403: No valid audit trail (QAPP102) license available

* 409: Conflict

GET:

* 403: No valid audit trail (QAPP102) license available

* 409: Conflict

2.16 Screen

2.16.1 Overlay Message

2.16.1.1 Description

Opens a message box on the balance screen.

2.16.1.2 HTTP Methods

POST, DELETE

2.16.1.3 Type

Synchronous

2.16.1.4 Unified Resource identifier

<balance_ip>/rest/screen/message

2.16.1.5 Query Parameters

No query parameters

2.16.1.6 Body Parameters

name	Type	Mandatory	Example
title	String	Yes	Test Title
content	JSON	YES	{"data" : "example"}

2.16.1.7 Response Parameters

No Response Parameters

2.17 Task

2.17.1 Start Task

2.17.1.1 Description

- Get available tasks for the active user
- Start task
- Start task with workflow data

2.17.1.2 HTTP Methods

POST, GET

2.17.1.3 Type

Synchronous

2.17.1.4 Unified Resource identifier

<balance_ip>/rest/task/start

2.17.1.5 Query Parameters

No query parameters

2.17.1.6 Body Parameters

POST:

name	Type	Mandatory	Example
taskId	String	Yes	212b738b-08b0...
workflowData	JSON	YES	{"data" : "example"}

2.17.1.7 Response Parameters

GET:

```
{
  "tasks": [
    {
      "taskName": "Weighing",
      "taskId": "212b738b-08b0-4af1-8f93-a979d4d36951"
    }
  ]
}
```

POST (start task without workflow data):

```
{
  "taskId" : "212b738b-08b0-4af1-8f93-a979d4d36951",
  "workflowData" : {}
}
```

2.17.2 Stop Task

2.17.2.1 Description

- Get running task
- Stop task

2.17.2.2 HTTP Methods

POST, GET

2.17.2.3 Type

Synchronous

2.17.2.4 Unified Resource identifier

<balance_ip>/rest/task/stop

2.17.2.5 Query Parameters

No query parameters

2.17.2.6 Body Parameters

POST:

name	Type	Mandatory	Example
taskId	String	Yes	QAPP402
filename	String	Yes	job_samples.json

Example:

```
curl -X PUT --insecure --data '{"data":2}'
"https://172.16.244.44/qapp/files?ordernumber=QAPP500&filename=test" --header "cubispass:123456"
```

2.17.2.7 Body Parameters

PUT:

Any valid JSON content

2.17.2.8 DELETE, Response Parameters

GET:

```
{
  "tasks": [
    {
      "taskName": "Weighing",
      "taskId": "212b738b-08b0-4af1-8f93-a979d4d36951"
    }
  ]
}
```

POST (start task without workflow data):

```
{
  "taskId" : "212b738b-08b0-4af1-8f93-a979d4d36951",
  "workflowData" : {}
}
```

2.18 QAPP

2.18.1 QAPP file handling

2.18.1.1 Description

Storing data for QAPPs with a specific order number in files. Add or delete a list of calibration weights and/or add or delete list of weighing samples., the payload of this request contains the samples to execute.

The maximum used memory is 20MB for the sum of all QAPPs. The number of files is irrelevant.

Path on Cubis balance:

<secured_qapp_management_folder>/<ordernumber>/<filename>

PUT:

Creates the file with given filename and payload.

The content-type will not be checked.

Possibly overwrites existing file.

GET:

Return content of file name as given

DELETE:

Deletes file as given in file name

2.18.1.2 HTTP Methods

PUT, DELETE, GET

2.18.1.3 Type

Synchronous

2.18.1.4 Unified Resource identifier

<balance_ip>/qapp/files?ordernumber=<ordernumber>&filename=<filename>

2.18.1.5 Query Parameters

Name	Type	Mandatory	Example
ordernumber	String	Yes	QAPP402

Example:

```
curl -X PUT --insecure --data '{"data":2}'
```

```
"https://172.16.244.44/qapp/files?ordernumber=QAPP500&filename=test" --header "cubispass:123456"
```

2.18.1.6 Body Parameters

PUT:

Any valid JSON content

DELETE, GET:

No body parameters

2.18.1.7 Response Parameters

PUT, DELETE:

No Response Parameters

GET:

return content of the given file

2.18.1.8 Response Codes

PUT:

* 400: content-length of request payload smaller than 0+

* 404: The given QAPP with order number is not installed (licensed)

GET, DELETE:

* 404: the given file name does not exist

2.18.2 QAPP ReST functions

2.18.2.1 Description

QAPPs can provide REST end points. These endpoints are accessible from outside e.g. external systems with connector-QAPPs.

All unhandled URIs of type "/qapp/<subURL>" are forwarded to QAPP with <subURL> information.

2.18.2.2 HTTP Methods

GET, POST, DELETE and PUT are supported

2.18.2.3 Type

Synchronous

2.18.2.4 Unified Resource identifier

<balance_ip>/qapp/<subURL>

2.18.2.5 Query Parameters

Depends on QAPP and configuration

2.18.2.6 Body Parameters

Depends on QAPP and configuration

2.18.2.7 Response Parameters

Depends on QAPP and configuration

2.19 QAPP Center

2.19.1 License information

2.19.1.1 Description

- Get license status from QAPP or Package

2.19.1.2 HTTP Methods

GET, POST

2.19.1.3 Type

Synchronous

2.19.1.4 Unified Resource identifier

<balance_ip>/rest/qappcenter/license

2.19.1.5 Query Parameters

GET:

Name	Type	Mandatory	Example
Ordernumber	String	Yes	123-567-abc

POST: none

2.19.1.6 Body Parameters

GET: none

POST:

Formatted JSON object, containing following properties:

Name	Type	Mandatory	Example
ordernumber	String	Yes	"QAPP100"
license	String	Yes	"1234567890"

2.19.1.7 Response Parameters

GET: return ordernumber and license status

POST: none

2.19.1.8 Response Codes

POST:

* 400: <Missing Parameter>

* 403: Unit operation doesn't exist or Unit operation QAPP doesn't exist

* 409: Job name is not unique or Job can't be created

GET:

- * 400: ID is missing or empty
- * 404: Job does not exist

2.20 Job management

To be able to use the job management, at least the firmware version 07.02.xx must be installed on the balance.

2.20.1 Unit operations

2.20.1.1 Description

Get information about the installed and licensed unit operations.

GET:

Returns information for all installed and licensed unit operation QAPPs.

HTTP Methods:

GET

2.20.1.2 Type

Synchronous

2.20.1.3 Unified Resource identifier

<balance_ip>/rest/unitOperation

2.20.1.4 Query Parameters

No query parameters

2.20.1.5 Body Parameters

No body parameters

2.20.1.6 Response Parameters

GET:

return QAPP order number and version of all installed unit operation QAPPs

2.20.1.7 Response Codes

No exceptional response codes

2.20.2 Job handling

2.20.2.1 Description

Storing a job and ensures that a job can be executed with the balance. Add, delete a job from the balance or get the status and QAPP order number from supported unit operation.

POST:

Creates a job on the balance.

GET:

Return job status and order number from an existing job.

DELETE:

Deletes an existing job from the balance.

2.20.2.2 HTTP Methods

POST, DELETE, GET

2.20.2.3 Type

Synchronous

2.20.2.4 Unified Resource identifier

DELETE, GET:

<balance_ip>/rest/jobManager/job?id=<jobid>

POST:

<balance_ip>/rest/jobManager/job

2.20.2.5 Query Parameters

GET, DELETE:

Name	Type	Mandatory	Example
id	String	Yes	123-567-abc

POST: No query parameters

2.20.2.6 Body Parameters

POST:

Formatted JSON object, containing following properties:

Name	Type	Mandatory	Example
name	String	Yes	Job name
description	String	No	Whatever you want
assignees	List of strings	No	["John.Doe","Operator"]
workflowData	Json	Yes	{"data": "data"}
qappOrderNumber	String	Yes	QAPP402
blocking	Bool	Yes	true
jobInfo	Object	No	See below

Additional, optional job information

Name	Type	Mandatory	Example
application	String	Yes	Microsoft Excel

host	String	Yes	192.168.1.1
user	String	Yes	John Doe

Remarks:

The list of assignees should name their unique login name.

qappOrderNumber must reference an instantiated task of a unit operation.

DELETE, GET:

No body parameters

2.20.2.7 Response Parameters

DELETE: No Response Parameters

GET: return job status and order number

POST: return job identifier

2.20.2.8 Response Codes

POST:

* 400: <Missing Parameter>

* 403: Unit operation doesn't exist or Unit operation QAPP doesn't exist

* 409: Job name is not unique or Job can't be created

GET:

* 400: ID is missing or empty

* 404: Job does not exist

DELETE:

* 400: ID is missing or empty

* 404: Job does not exist

* 409: Job is not open

2.20.3 Jobs status**2.20.3.1 Description**

Returns a list of all stored jobs, their state

GET:

Return job status and order number from an existing job.

2.20.3.2 HTTP Methods

GET

2.20.3.3 Type

Synchronous

2.20.3.4 Unified Resource identifier

2.20.3.5 Query Parameters

GET: No query parameters

2.20.3.6 Body Parameters

GET:

No body parameters

2.20.3.7 Response Parameters

GET: return list of all jobs

Name	Type	Example
id	String	50db2657-6c1a-4c93-9e0d-f52dd9798a04
name	String	Example Job name
orderNumber	String	QAPP506
status	String	OPEN

2.20.3.8 Response Codes

No exceptional response codes