CMe3100 Plugin REST API

Innehåll

1 DOCUMENT NOTES 3

1.1 COPYRIGHT AND TRADEMARK 3
1.2 CONTACTS 3

2 USING THIS MANUAL 4

2.1 PURPOSE AND AUDIENCE 4
2.2 MODELS 4
2.3 ADDITIONAL AND UPDATED INFORMATION 4

3 INTRODUCTION 5

3.1 LIST OF SERVICES 5
3.2 USING SERVICES 6

4 REST SERVICES 7

4.1 ACCESS METER DATA 7
4.1.1 Source 7
4.1.2 Serie 7
4.1.3 Data 7
4.1.4 Examples 7
4.2 LIST OF SERVICE METHODS 19
4.2.1 Measurement serie service 19
4.2.2 Measurement serie data service 20
4.2.3 Measurement serie source service 21
4.2.4 Device service 21
4.2.5 Console service 23
4.2.6 Log service 23
4.2.7 User service 24
4.2.8 Config service 25
4.2.9 Device type service 26
4.2.10 Unit service 27
4.2.11 Unit type service 28
4.2.12 Encryption key service 28
4.2.13 Database translation service 29
4.2.14 User link service 30
4.2.15 View mdm serie service 30
View mdm serie service introduces new easier result entity which combines the measurement serie foreign keys to readable data. 31
4.2.16 View mdm data service 31
View mdm data service introduces new easier result entity which combines the measurement serie data foreign keys to readable data 32
4.2.17 Datatable service 32
4.2.18 Mdmtag service 33
4.2.19 System time service 33

DOCUMENT HISTORY 34

4.3 DOCUMENT SOFTWARE APPLIANCE 34

5 REFERENCES 35

5.1 REFERENCES 35
5.2 TERMS AND ABBREVIATIONS 35
5.2.1 Number representation 35

1 Document notes

All information in this manual, including product data, diagrams, charts, etc. represents information on products at the time of publication, and is subject to change without prior notice due to product improvements or other reasons. It is therefore recommended that customers contact Elvaco AB for the latest product information before purchasing a CMe Series product.

The documentation and product are provided on an “as is” basis only and may contain deficiencies or inadequacies. Elvaco AB takes no responsibility for damages, liabilities or other losses by using this product.

1.1 Copyright and Trademark

1.2 Contacts

Elvaco AB Headquarter

Teknikgatan 18
434 37 Kungsbacka
SWEDEN

Phone: +46 300 30250
Fax: +46 300 18440

E-Mail: info@elvaco.com

Elvaco AB Technical Support

Phone: +46 300 434300
E-Mail: support@elvaco.se

Online: http://www.elvaco.com

2 Using this manual

2.1 Purpose and Audience

Elvaco Rest API is for developers and administrators who want script interaction with running web server on CMe3100.

2.2 Models

CNMe3100

2.3 Additional and updated information

Latest documentation version is available on Elvaco web site at http://www.elvaco.com.

3 Introduction

This document describes Elvaco Rest API and interactions within resources provided by CMe3100 product. Elvaco Rest API provides access to resources (data entities) via URL paths. All actions through API are done by using standard HTTP methods GET, POST, PUT, and DELETE. Standard HTTP response codes used to indicate success and error conditions.

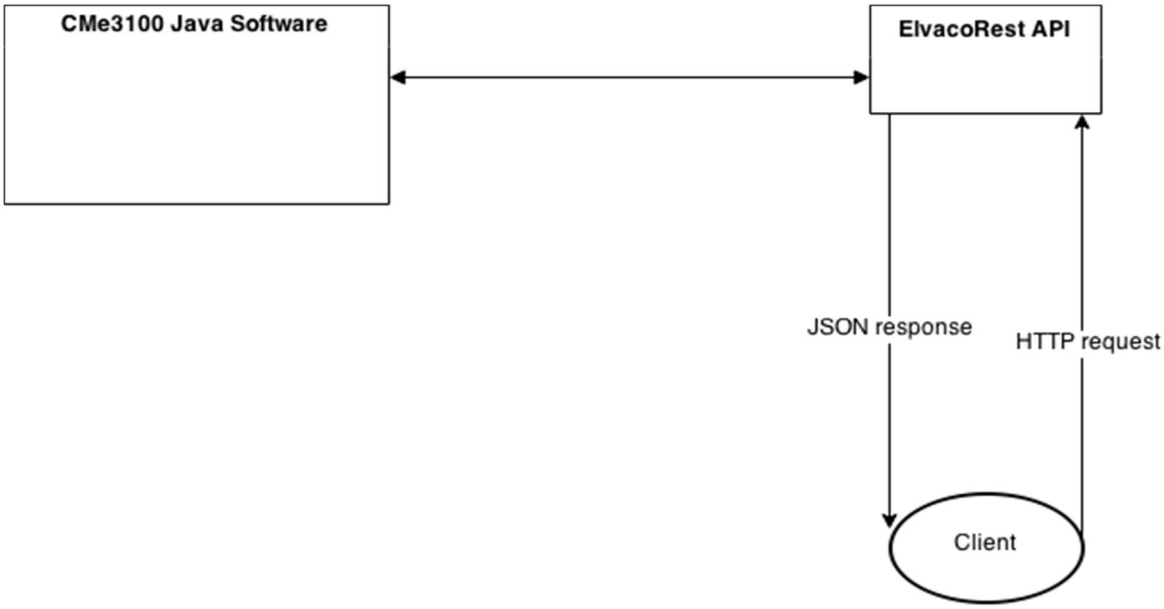

Figure 1- Basic architecture

Request body data is expected to be in JSON, and response body data is returned as JSON. Status object as JSON returns when any operation executed without returning result entity.

{

status: Status text

}

Status text: Record(s) not found / Parameter(s) not found / Success/ Error

HTTP status codes respectively: 404 Not found / 404 Not found / 201 Created / 400 Bad request

Basic HTTP authentication is required.

3.1 List of services

Use the table below to find out the resources of Elvaco Rest API.

Service	Description
/device	Service for handling devices, i.e. M-Bus slaves.
/mdmserie	Service for normalized values/meter data from devices.
/mdmsource	Service for handling data source for mdmseries, i.e. M-Bus, functions etc.
/mdmdata	Service for reading historical or momentary data from different source.

/config	Service for configuring CMe3100, i.e. changing TCP console port, setting e-mail server etc.
/log	Service for monitoring log information for different severity levels, i.e. -2 (Debug), -1 (Unknown), 0 (Info), 1 (Warning), 2 (Error), 3 (Critical), 4 (Fatal), 5 (Exception), 6 (Event).
/deviceType	Service for determining the device types, i.e. electricity, gas, detector, smoke detector etc.
/encryptionkey	Service for determining encryption keys for M-BUS slaves.
/unit	Service for determining units for metering action, i.e. kW, GJ/h, minute(s), day(s) etc.
/unitType	Service for determining type of unit metering action, i.e. date, energy, flow-temp etc.
/user	Service for determining users and their authentication rights.
/userlink	Service for determining user links added by the user.
/dbtranslations	Service for determining database translations, i.e. adding/updating new translations to database.
/console	Service for executing console commands. i.e. execute a console command for requesting momentary reports via e-mail.
/viewmdmserie	Service for easier entity which combines mdmseries foreign key to readable data.
/viewmdmdata	Service for easier entity which combines mdmdata foreign key with readable data.
/datatable	Service to reach consisting datatables data within the system.
/mdmtag	Service for determining tags with related measurementseries..
/systemtime	Service for getting the system time.

Table 1 List of Rest API resources

3.2 Using services

Elvaco Rest API is based on open standards, any web language can be used to access the API. To use Elvaco Rest API, application should make HTTP request and parse the response. Basic authentication required.

Base URL: http://yourserver:port /Elvaco-Rest/rest

4 REST Services

4.1 Access meter data

Data read from devices are normalized in measurement series. The main parts of measurement series are source, data and serie.

4.1.1 Source

The source of a measurement serie defines the source of the data, which can be a specific value read from an M-Bus slave. The source can also be used to define mathematical functions to calculate data from one or more measurement series. A measurement serie can only have one active source.

4.1.2 Serie

The serie defines how data from the source are published to the user or other interface.

4.1.3 Data

The data of the measurement serie holds information about measurement serie's data. By usage of the module; if the data is numeric, data belongs to which measurement serie and data source etc. can be monitored.

4.1.4 Examples

In this section we will demonstrate a use-case scenario to give better understanding for the usage of the Elvaco Rest API. Below are the steps of scenario, and example responses and requests.

4.1.4.1 User wants to list all measurement series on the system.

Request:

```
HTTP GET /Elvaco-Rest/rest/mdmserie/all
```

Response:


```
[
  {
    "storageIntervalCron": "",
    "updateIntervalCron": "",
    "unitTypeId": 134,
    "updateOnNewData": false,
    "description": "Signalstyrka i dBm",
    "expireTimeout": 0,
    "name": "Signalstyrka",
    "measurementSerieId": 241,
    "unitId": 58,
```

```
    "apiIdentifier": "61000002-mbus.dib.rf-level.0.0.0.0"
    "deviceTypeId": "2",
    "calculationOrder": "0",
    "createdFromTemplateId": "9",
    "unitTypeId": 0,
    "priority": "1",
  },
  {
    "storageIntervalCron": "",
    "updateIntervalCron": "",
    "unitTypeId": 167,
    "updateOnNewData": false,
    "description": "Temperatur inomhus",
    "expireTimeout": 0,
    "name": "Temperatur",
    "measurementSerieId": 242,
    "unitId": 13,
    "apiIdentifier": "61000002-mbus.dib.ext-temp.0.0.0.0"
    "deviceTypeId": "2",
    "calculationOrder": "0",
    "createdFromTemplateId": "5",
    "unitTypeId": 5,
    "priority": "1",

  },
 {
  …
 }
]
```


After measurement serie information is listed, user can use measurement serie id to determine measurement serie source.

4.1.4.2 User wants to get the source of the measurement serie.

To be able to get the source information of a measurement serie, request with measurement series id as a parameter should be requested.

Resquest:

```
HTTP GET /Elvaco-Rest/rest/mdmsource/measurementSerieId/{measurementSerieId}
HTTP GET /Elvaco-Rest/rest/mdmsource/measurementSerieId/241
```

Response:


```
{
  "sourceType": "mbus",
  "formatString": null,
  "constant": 1,
  "sourceData": "0.mbus.dib.rf-level.0.0.0.0.value",
  "created": 1418921380184,
  "measurementSeriesId": 241,
  "deactivated": -1,
  "sourceIdentifier": 61000002,
  "sourceId": 1147
}
```


As seen on response body below, source type is mbus (M-Bus slave) for measurement serie with id 241.

Two request examples above let us get mdmserie – measurement serie information, and mdmsource – measurement serie source information. Now mdmdata – measurement serie data information will be listed.

4.1.4.3 User wants to get measurement serie data.

Getting measurement serie data can be determined in five different ways with different parameters.

- Listing all measurement series data with limit and offset.

Request:

```
HTTP GET /Elvaco-Rest/rest/mdmdata/all/limit/{limit}/offset/{offset}
HTTP GET /Elvaco-Rest/rest/mdmdata/all/limit/100/offset/0
```

Note: Requesting too many mdmdata (measurement series data) entities might create heavy loading on application. Using limit and offset with pagination implementation is suggested for this case.

Response:


```
{
  "total_records": 38728,
  "limit": 100,
  "offset": 0,
  "values": [
    {
      "numeric": true,
      "status": 0,
      "created": 1418921232000,
      "value": 62000501,
      "sourceDate": 1418921230641,
      "measurementSerieId": 188,
      "effectiveDate": 1418921220000,
      "measurementSerieDataId": 204,
      "valueAsString": "62000501,000",
      "sourceId": 1094
    },
    {
      "numeric": false,
      "status": 0,
      "created": 1418921235000,
      "value": 0,
      "sourceDate": 1418921230641,
      "measurementSerieId": 189,
      "effectiveDate": 1418921220000,
      "measurementSerieDataId": 205,
      "valueAsString": "1.6.3",
      "sourceId": 1095
    },
    {
      …
    },
```

```
    {
      "numeric": true,
      "status": 0,
      "created": 1418921237000,
      "value": 4,
      "sourceDate": 1418921230641,
      "measurementSerieId": 191,
      "effectiveDate": 1418921220000,
      "measurementSerieDataId": 207,
      "valueAsString": "4,000",
      "sourceId": 1097
    }
 ]
}
```


- Listing measurement series data from specific measurement serie with limit and offset.

Request:

```
HTTP GET /Elvaco-
Rest/rest/mdmdata/measurementserieId/{measurementserieId}/limit/{limit}/offse
t/{offset}
HTTP GET /Elvaco-Rest/rest/mdmdata/measurementserieId/241/limit/100/offset/0
```

Response:


```
 {
  "total_records": 296,
  "limit": 100,
  "offset": 0,
  "values": [
    {
      "numeric": true,
      "status": 0,
      "created": 1418921380000,
      "value": -66,
      "sourceDate": 1418921374435,
      "measurementSerieId": 241,
```

```
      "effectiveDate": 1418921340000,
      "measurementSerieDataId": 257,
      "valueAsString": "-66,000",
      "sourceId": 1147
    },
    {
      …
    },
    {
      "numeric": true,
      "status": 0,
      "created": 1418925604000,
      "value": -66,
      "sourceDate": 1418925601121,
      "measurementSerieId": 241,
      "effectiveDate": 1418925600000,
      "measurementSerieDataId": 523,
      "valueAsString": "-66,000",
      "sourceId": 1147
    }
  ]
}
```


- Listing measurement serie data with latest number of read.

Request:

```
HTTP GET /Elvaco-
Rest/rest/mdmdata/measurementSerieId/{measurementSerieId}/latest/
{numberofread}
HTTP GET /Elvaco-Rest/rest/mdmdata/measurementSerieId/693/latest/5
```

Response:


```
{
  "total_records": 337,
  "limit": 5,
  "offset": 0,
```

```
"values": [
  {
    "numeric": false,
    "status": -1,
    "created": 1426169671000,
    "value": 0,
    "sourceDate": 1426169671682,
    "measurementSerieId": 693,
    "effectiveDate": 1426169640000,
    "measurementSerieDataId": 71619,
    "valueAsString": "N/A",
    "sourceId": 1599
  },
  {
    "numeric": true,
    "status": 0,
    "created": 1426172404000,
    "value": 0,
    "sourceDate": 1426169667730,
    "measurementSerieId": 693,
    "effectiveDate": 1426172400000,
    "measurementSerieDataId": 71686,
    "valueAsString": "0,000",
    "sourceId": 1599
  },
  {
    "numeric": true,
    "status": 0,
    "created": 1426176005000,
    "value": 0,
    "sourceDate": 1426172401560,
    "measurementSerieId": 693,
    "effectiveDate": 1426176000000,
    "measurementSerieDataId": 71753,
    "valueAsString": "0,000",
    "sourceId": 1599
```

```
    },
    {
      "numeric": true,
      "status": 0,
      "created": 1426179605000,
      "value": 0,
      "sourceDate": 1426176001513,
      "measurementSerieId": 693,
      "effectiveDate": 1426179600000,
      "measurementSerieDataId": 71820,
      "valueAsString": "0,000",
      "sourceId": 1599
    },
    {
      "numeric": true,
      "status": 0,
      "created": 1426183205000,
      "value": 0,
      "sourceDate": 1426179601471,
      "measurementSerieId": 693,
      "effectiveDate": 1426183200000,
      "measurementSerieDataId": 71887,
      "valueAsString": "0,000",
      "sourceId": 1599
    }
  ]
}
```


- Listing measurement series data from when it is created with limit and offset.

Request:

```
HTTP GET /Elvaco-
Rest/rest/mdmdata/created/{created}/limit/{limit}/offset/{offset}
HTTP GET /Elvaco-Rest/rest/mdmdata/created/1420448349000/limit/100/offset/0
```

Response:

```
{
  "total_records": 8,
  "limit": 100,
  "offset": 0,
  "values": [
    {
      "numeric": true,
      "status": 0,
      "created": 1420448349000,
      "value": 16.44,
      "sourceDate": 1420448349043,
      "measurementSerieId": 188,
      "effectiveDate": 1420448340000,
      "measurementSerieDataId": 204,
      "valueAsString": "16,440",
      "sourceId": 1094
    },
    {
      "numeric": true,
      "status": 0,
      "created": 1420448349000,
      "value": 24.03,
      "sourceDate": 1420448349043,
      "measurementSerieId": 189,
      "effectiveDate": 1420448340000,
      "measurementSerieDataId": 205,
      "valueAsString": "24,030",
      "sourceId": 1095
    },
    {
      "numeric": true,
      "status": 0,
      "created": 1420448356000,
      "value": 16.51,
      "sourceDate": 1420448356214,
      "measurementSerieId": 190,
```

```
    "effectiveDate": 1420448340000,
    "measurementSerieDataId": 206,
    "valueAsString": "16,510",
    "sourceId": 1096
  },
  {
    "numeric": true,
    "status": 0,
    "created": 1420448356000,
    "value": 23.990000000000002,
    "sourceDate": 1420448356214,
    "measurementSerieId": 191,
    "effectiveDate": 1420448340000,
    "measurementSerieDataId": 207,
    "valueAsString": "23,990",
    "sourceId": 1097
  },
  {
    "numeric": true,
    "status": 0,
    "created": 1420448426000,
    "value": 16.44,
    "sourceDate": 1420448349043,
    "measurementSerieId": 188,
    "effectiveDate": 1420448400000,
    "measurementSerieDataId": 208,
    "valueAsString": "16,440",
    "sourceId": 1094
  },
  {
    "numeric": true,
    "status": 0,
    "created": 1420448426000,
    "value": 24.02,
    "sourceDate": 1420448425374,
    "measurementSerieId": 189,
```

```
      "effectiveDate": 1420448400000,
      "measurementSerieDataId": 209,
      "valueAsString": "24,020",
      "sourceId": 1095
    },
    {
      "numeric": true,
      "status": 0,
      "created": 1420448426000,
      "value": 16.48,
      "sourceDate": 1420448425374,
      "measurementSerieId": 190,
      "effectiveDate": 1420448400000,
      "measurementSerieDataId": 210,
      "valueAsString": "16,480",
      "sourceId": 1096
    },
    {
      "numeric": true,
      "status": 0,
      "created": 1420448426000,
      "value": 23.990000000000002,
      "sourceDate": 1420448425374,
      "measurementSerieId": 191,
      "effectiveDate": 1420448400000,
      "measurementSerieDataId": 211,
      "valueAsString": "23,990",
      "sourceId": 1097
    }
  ]
}
```


The request will return you measurement serie data from requested created with limit and offset. Created parameter should be in milliseconds.

- Listing measurement serie data with effective date from/to with limit and offset.

Request:

```
HTTP GET /Elvaco-Rest/rest/mdmdata/effectiveDate/from/{from}/to/{to}/limit/{limit}/offset/{offset}
HTTP GET /Elvaco-Rest/rest/mdmdata/effectiveDate/from/1420448340000/to/1420548400000/limit/100/offset/0
```

Response:


```
{
  "total_records": 204,
  "limit": 100,
  "offset": 0,
  "values": [
    {
      "numeric": true,
      "status": 0,
      "created": 1420448349000,
      "value": 16.44,
      "sourceDate": 1420448349043,
      "measurementSerieId": 188,
      "effectiveDate": 1420448340000,
      "measurementSerieDataId": 204,
      "valueAsString": "16,440",
      "sourceId": 1094
    },
    {
      …
    },
    {
      "numeric": true,
      "status": 0,
      "created": 1420448426000,
      "value": 23.990000000000002,
      "sourceDate": 1420448425374,
      "measurementSerieId": 191,
```

```
      "effectiveDate": 1420448400000,
      "measurementSerieDataId": 211,
      "valueAsString": "23,990",
      "sourceId": 1097
    }
  ]
}
```

The request will return you measurement serie data from requested effective date range with limit and offset. Effective date parameter should be in milliseconds.

4.2 List of service methods

4.2.1 Measurement serie service

HTTP GET

URL	Http Response code
/mdmserie/all	200 – OK 401 - Unauthorized 404 – Not found
/mdmserie/measurementSerieId/{ measurementSerieId }	200 – OK 401 - Unauthorized 404 – Not found
/mdmserie/apiIdentifier/{ apiIdentifier }	200 – OK 401 - Unauthorized 404 – Not found
/mdmserie/tagname/{ tagname }	200 – OK 401 - Unauthorized 404 – Not found
/mdmserie/unitTypeId/{ unitTYpeId }	200 – OK 401 - Unauthorized 404 – Not found
/mdmserie/deviceTypeId/{ deviceTypeId }/unitTypeId/{ unitTypeId }	200 – OK 401 - Unauthorized 404 – Not found

HTTP PUT

/mdmserie/update/measurementserieId/{measurementserieId}	200 – OK 400 – Bad request

	404 – Not found

HTTP DELETE

/mdmserie/delete/all	200 – OK 401 - Unauthorized 404 – Not found
/mdmserie/delete/measurementserieId/{ measurementserieId }	200 – OK 400 – Bad request 401 - Unauthorized 404 – Not found
/mdmserie/delete/tagname/{ tagname }	200 – OK 400 – Bad request 401 - Unauthorized 404 – Not found
/mdmserie/delete/measurmentSerieName/{ measurmentSerieName }	200 – OK 400 – Bad request 401 - Unauthorized 404 – Not found

4.2.2 Measurement serie data service

HTTP GET

/mdmdata/measurementSerieId/{ measurementSerieId }/limit/{ limit }/offset/{ offset } Get mdm data related to specific measurement serie.	200 – OK 401 - Unauthorized 404 – Not found
/mdmdata/effectiveDate/from/{ from }/to/{ to }/limit/{ limit }/offset/{ offset } Get mdm data between provided date parameters.	200 – OK 401 - Unauthorized 404 – Not found
/mdmdata/measurementserieId/ {measurementseriesId}/latest/{numberofread}/limit/{ limit }/offset/{ offset } numberofread: Latest readout number. Ex. 5 for getting latest 5 readout values.	200 – OK 401 - Unauthorized 404 – Not found
/mdmdata/unitTypeId/ {unitTypeId}/latest/all Get latest mdm data for measurementseries which are matching provided unit type.	200 – OK 401 - Unauthorized

	404 – Not found
/mdmdata/deviceTypeId/{deviceTypeId/}unitTypeid/ {unitTypeId}/latest/all Get latest mdm data for measurementseries which are matching provided device and unit type.	200 – OK 401 - Unauthorized 404 – Not found
/mdmdata/measurmentSerieId/{measurementSerieid}/created/{created}/limit/{limit}/offset/{offset} Method returns all data from created to latest readout.	200 – OK 401 - Unauthorized 404 – Not found
/mdmdata/measurmentSerieId/{measurementSerieid}/effectiveDate/from/{ from }/to/{ to }/limit/{ limit }/offset/{ offset } Get mdm data on measurementserie between provided date parameters.	200 – OK 401 - Unauthorized 404 – Not found

4.2.3 Measurement serie source service

HTTP GET

/rest/mdmsource/measurementserieId/{ measurementserieId }	200 – OK 401 - Unauthorized 404 – Not found
/rest/mdmsource/sourceIdentifier/{ sourceIdentifier }	200 – OK 401 - Unauthorized 404 – Not found

HTTP PUT

/rest/mdmsource/update/measurementserieId/{ measurementserieId }	200 – OK 400 – Bad request 401 - Unauthorized 404 – Not found

4.2.4 Device service

HTTP GET

/rest/device/all	200 – OK 401 - Unauthorized 404 – Not found
/rest/device/id/{ id }	200 – OK 401 - Unauthorized 404 – Not found

/rest/device/identification/{ identification }	200 – OK 401 - Unauthorized 404 – Not found
/rest/device/all/limit/{ limit }/offset/{ offset }	200 – OK 401 - Unauthorized 404 – Not found

HTTP POST

/rest/device/add
Request body: {
"fabricationNumber": 0,
"deviceTypeId": 0,
"communicationChannel": "",
"telegramCount": 0,
"fcbMode": "",
"secondaryAddress": 0,
"primaryAddress": 0,
"baudrate": 0,
"addressingMode": "",
"deviceType": "",
"manufacturer": "",
"created": 0,
"recordVersion": 0,
"recordType": 0,
"recordId": 0,
"identification": 0,
"position": "",
"updated": 0,
"version": 0,
"status": ""
}
	200 – OK
	400 – Bad request
	401 - Unauthorized
	404 – Not found

HTTP DELETE

/rest/device/delete/id/{ id }	200 – OK 401 - Unauthorized 404 – Not found
/rest/device/delete/identification/{ identification }	200 – OK 401 - Unauthorized

	404 – Not found
/rest/device/delete/all	200 – OK 401 - Unauthorized 404 – Not found

4.2.5 Console service

HTTP POST

/rest/console/command
Request body: {
"params": [" n "],
"command": "command"
}
n: number of devices command: console commands like install, storevalue ect.
	202 – Accepted
	400 – Bad request
	401 –Unauthorized
	404 – Not found

4.2.6 Log service

HTTP GET

/rest/log/all/created/{ created }/limit/{ limit }/offset/{ offset }	200 – OK 401 - Unauthorized 404 – Not found
/rest/log/severity{ severity }/limit/{ limit }/offset/{ offset } Severity: -2 (Debug), -1 (Unknown), 0 (Info), 1 (Warning), 2 (Error), 3 (Critical), 4 (Fatal), 5 (Exception), 6 (Event)	200 – OK 401 - Unauthorized 404 – Not found
/rest/log/source/{ source }/limit/{ limit }/offset/{ offset }	200 – OK 401 - Unauthorized 404 – Not found

HTTTP DELETE

/rest/log/delete/created/{ created }	200 – OK 401 - Unauthorized 404 – Not found
/rest/log/delete/severity/{ severity }	200 – OK 401 - Unauthorized 404 – Not found
/rest/log/delete/source/{ source }	200 – OK 401 - Unauthorized 404 – Not found
/rest/log/delete/id/{ id }	200 – OK

	401 - Unauthorized 404 – Not found

4.2.7 User service

HTTP GET

/rest/user/all	200 – OK 401 - Unauthorized 404 – Not found
/rest/user/id/{ id }	200 – OK 401 - Unauthorized 404 – Not found
/rest/user/username/{ username }	200 – OK 401 - Unauthorized 404 – Not found

HTTP POST

/rest/user/add
Request body: {
"language": "",
"email": "",
"rightsRest": "",
"rightsWeb": "",
"rightsTelnet": "",
"password": "",
"userName": "",
"userId": 0
}
	201 – Created
	401 - Unauthorized
	404 – Not found

HTTP PUT

/rest/user/update/username/{ username } Parameter: JSON object representation of desired update values, ex. { "rightsTelnet": "admin", "email": "", "userName": "admin"}	200 – OK 401 - Unauthorized 404 – Not found
/rest/user/update/id/{ id } Request body: JSON object representation of desired update values, ex. { "rightsTelnet": "admin", "email": "", "userName": "admin"}	200 – OK 401 - Unauthorized 404 – Not found

HTTP DELETE

/rest/user/delete/id/{ id }	200 – OK 400 – Bad request 401 - Unauthorized 404 – Not found
/rest/user/delete/username/{ username }	200 – OK 400 – Bad request 401 - Unauthorized 404 – Not found

4.2.8 Config service

HTTP GET

/rest/config/common/{file} file: configuration file name	200 – OK 401 - Unauthorized 404 – Not found
/rest/config/command/{file} file: configuration file name	200 – OK 401 - Unauthorized 404 – Not found
/rest/config/user/{file}	200 – OK 401 - Unauthorized 404 – Not found
/rest/config/common/rules/{file}	200 – OK 401 - Unauthorized 404 – Not found
/rest/config/common/{file} /{param} file: configuration file name (common.cfg) param: desired value to update	200 – OK 401 - Unauthorized 404 – Not found
/rest/config/command/rules/{file} file: configuration file name	200 – OK 401 - Unauthorized 404 – Not found
/rest/config/command/{file}/{param} file: configuration file name param: desired value to update	200 – OK 401 - Unauthorized 404 – Not found
/rest/config/command/rules/{file}/{param} file: configuration file name param: desired value to update	200 – OK 401 - Unauthorized 404 – Not found

HTTP PUT

/rest/config/common/{file} Request body: JSON representation of desired update values	200 – OK 401 - Unauthorized 404 – Not found
/rest/config/command/{file} Request body: JSON representation of desired update values	200 – OK 401 - Unauthorized 404 – Not found
/rest/config/user/{file} Request body: JSON representation of desired update values	200 – OK 401 - Unauthorized 404 – Not found

4.2.9 Device type service

HTTP GET

/rest/deviceType/all	200 – OK 401 - Unauthorized 404 – Not found
/rest/deviceType/deviceTypeId/{ deviceTypeId }	200 – OK 401 - Unauthorized 404 – Not found
/rest/deviceType/value/{ value }	200 – OK 401 - Unauthorized 404 – Not found

HTTP POST

/rest/deviceType/add
Request body: {
"updated": 0,
"deviceIcon": "",
"value": "",
"devicetypeId": 0
}
	200 – OK
	400 – Bad request
	401 –Unauthorized

HTTP DELETE

/rest/deviceType/delete/deviceTypeid/{ deviceTypeid }	200 – OK 401 - Unauthorized 404 – Not found
/rest/deviceType/delete/value/{ value }	200 – OK 401 - Unauthorized

	404 – Not found

4.2.10 Unit service

HTTP GET

/rest/unit/all	200 – OK 401 - Unauthorized 404 – Not found
/rest/unit/unitId/{ unitId }	200 – OK 401 - Unauthorized 404 – Not found
/rest/unit/value/{ value }	200 – OK 401 - Unauthorized 404 – Not found

HTTP POST

/rest/unit/add
Request body: {
"display": false,
"value": "",
"unitId": 0
}
	201 – Created
	400 – Bad request
	401 -Unauthorized

HTTP PUT

/rest/unit/update/unitId/{ unitId } Request body: JSON representation of desired update values, ex. { "value": "new value"}	200 – OK 400 – Bad request 401 –Unauthorized

HTTP DELETE

/rest/unit/delete/unitId/{ unitId }	201 – Created 401 –Unauthorized 404 – Not found
/rest/unit/delete/value/{ value }	201 – Created 401 –Unauthorized 404 – Not found

4.2.11 Unit type service

HTTP GET

/rest/unitType /all	201 – Created 401 –Unauthorized 404 – Not found
/rest/unitType/unitType/{ unitType }	201 – Created 401 –Unauthorized 404 – Not found
/rest/unitType /value/{value}	201 – Created 401 –Unauthorized 404 – Not found

HTTP POST

/rest/unitType/add	201 – Created 400 –Bad request 401 -Unauthorized

HTTP DELETE

/rest/unitType/delete/unitTypeId/{ unitTypeId }	200 – OK 401 –Unauthorized 404 – Not found
/rest/unitType/delete/value/{ value }	200 – OK 401 –Unauthorized 404 – Not found

4.2.12 Encryption key service

HTTP GET

/rest/encryptionkey/all	200 – OK 401 –Unauthorized 404 – Not found
/rest/encryptionkey/id/{id}	200 – OK 401 –Unauthorized 404 – Not found

HTTP POST

/rest/encryptionkey/add Request body: {	201 – Created 400 –Bad request

"key": "", "address": 0, "version": 0, "deviceType": 0, "manufacturer": "", "keyId": 0 }	401 -Unauthorized

HTTP PUT

/rest/encryptionkey/update/id/ Request body: JSON object representation of desired update values.	200 – OK 400 – Bad request 401 –Unauthorized

HTTP DELETE

/rest/encryptionkey/delete/all	200 – OK 400 – Bad request 401 –Unauthorized
/rest/encryptionkey/delete/id/{id}	200 – OK 400 – Bad request 401 –Unauthorized

4.2.13 Database translation service

HTTP GET

/rest/dbtranslation/all	200 – OK 401 –Unauthorized 404 – Not found
/rest/dbtranslation/language/{ language }	200 – OK 401 –Unauthorized 404 – Not found
/rest/dbtranslation/key/{ key }	200 – OK 401 –Unauthorized 404 – Not found

HTTP PUT

/rest/dbtranslation/update/ language /{language}/ key /{key}	200 – OK 400 – Bad request 401 –Unauthorized

4.2.14 User link service

HTTP GET

/rest/userlink/username/{ username }	200 – OK 401 –Unauthorized 404 – Not found
/rest/userlink/userLinkId/{ userLinkId }	200 – OK 401 –Unauthorized 404 – Not found

HTTP POST

/rest/userlink/add Request body: { "name": "Some name", "userId": 0, "newTab": true, "url": "" }	201 – Created 400 –Bad request 401 -Unauthorized

HTTP PUT

/rest/userlink/update/userLinkId/{userLinkId}	200 – OK 400 – Bad request 401 –Unauthorized

HTTP DELETE

/rest/userlink/delete/all/username/{username}	200 – OK 400 – Bad request 401 –Unauthorized
/rest/userlink/delete/all/userLinkId/{userLinkId}	200 – OK 400 – Bad request 401 –Unauthorized

4.2.15 View mdm serie service

HTTP GET

/viewmdmserie/all	200 – OK 400 – Bad request 401 –Unauthorized

/viewmdmserie/measurementId/{measurementId}/	200 – OK 400 – Bad request 401 –Unauthorized
/viewmdmserie/apiIdentifier/{apiIdentifier}	200 – OK 400 – Bad request 401 –Unauthorized
/viewmdmserie/tagName/{tagName}	200 – OK 400 – Bad request 401 –Unauthorized
/viewmdmserie/untitTypeId/{unitTypeId}	200 – OK 400 – Bad request 401 –Unauthorized
/viewmdmserie/deviceTypeId/{deviceTypeId}/ unitTypeId/{unitTypeId}	200 – OK 400 – Bad request 401 –Unauthorized

View mdm serie service introduces new easier result entity which combines the measurement serie foreign keys to readable data.

```
ViewMdmMeasurementSerie{
      "sourcePosition": String,
      "sourceData": String,
      "sourceIdentifier": String,
      "sourceType": String,
      "unitTypeString": String,
      "unitTypeId": int,
      "unitString": String,
      "unitId": int,
      "deviceTypeString": String
      "apiIdentifier ": String,
      "deviceTypeId": int,
      "description ": String,
      "name ": String,
      "measurementSerieId ": int
    }
```

4.2.16 View mdm data service

HTTP GET

/rest/viewmdmdata/measurementserieId/{ measurementserieId }/limit/{limit}/offset/{offset}	200 – OK 401 - Unauthorized 404 – Not found
/rest/viewmdmdata/latest/{latestNumberofreads}	200 – OK 401 - Unauthorized 404 – Not found
/rest/viewmdmdata/latest/{latestNumberofreads}/offset/{ offset }	200 – OK 401 - Unauthorized 404 – Not found
/rest/viewmdmdata/effectiveDate/from/{ from }/to/{ to }/limit/{ limit }/offset/{offset}	200 – OK 401 - Unauthorized 404 – Not found
/rest/viewmdmdata/measurementSerieId/{ measurementSerieId}/effectiveDate/from/{ from }/to/{ to }/limit/{ limit }/offset/{offset}	200 – OK 401 - Unauthorized 404 – Not found
/rest/viewmdmdata/unitTypeId/{ unitTypeId }/latest/all	200 – OK 401 - Unauthorized 404 – Not found
/rest/viewmdmdata/devicetypeId/{ devicetypeId }/unitTypeId/{ unitTypeId }/latest/all	200 – OK 401 - Unauthorized 404 – Not found

View mdm data service introduces new easier result entity which combines the measurement serie data foreign keys to readable data

```
ViewMdmMeasurementSerieData{
      "status": String,
      "valueAsString": String,
      "effectiveDate": int,
      "measurementSerieId": int,
    }
```

4.2.17 Datatable service

HTTP GET

/rest/datatable/measurementseries/offset/{offset}/limit/{limit}/language/{language}	200 – OK 401 - Unauthorized 404 – Not found

/rest/datatable/device/offset/{offset}/limit/{limit}/language/{language}	200 – OK 401 - Unauthorized 404 – Not found
/rest/datatable/valueTable/secondaryAddress/{secondaryAddress}/start/{start}/end/ {end}/offset/{offset}/limit/{limit}/orederByCreated/{orederByCreated}	200 – OK 401 - Unauthorized 404 – Not found
/rest/datatable/deviceMeasurementserie/secondaryAddress/ {secondaryAddress}/ language/ {language}	200 – OK 401 - Unauthorized 404 – Not found
/rest/datatable/systemLog/startDate/{startDate}/endDate/{endDate} /severity/ {severity}/offset/{offset}	200 – OK 401 - Unauthorized 404 – Not found
/rest/datatable/deviceLog/secondaryAddress/{secondaryAddress}/start /{start}/end/ {end}/severity/{severity}/offset/{offset}/limit/{limit}	200 – OK 401 - Unauthorized 404 – Not found

4.2.18 Mdmtag service

HTTP GET

/rest/mdmtag/all	200 – OK 401 - Unauthorized 404 – Not found

4.2.19 System time service

HTTP GET

/rest/systemtime	200 – OK 401 - Unauthorized 404 – Not found

Document History

Version	Date	Description	Author
1.0.0	2014-12-22	First draft	Aydan Halilov
1.0.1	2014-01-05	First draft update	Aydan Halilov
1.0.2	2015-01-16	Service methods list	Aydan Halilov
1.0.3	2015-04-01	New methods added	Aydan Halilov
1.0.4	2015-04-07	Changes and update	Aydan Halilov

4.3 Document software appliance

Type	Version	Date	Comments
CMe3100	1.0.*	2014-12-22	
CMe3100	1.2.0	2015-04-07	

5 References

5.1 References

[1]

[2]

5.2 Terms and Abbreviations

Abbreviation	Description

5.2.1 Number representation

Decimal numbers are represented as normal number, i.e. 10 (ten).

Hexadecimal numbers are represented with prefix 0x, i.e. 0x0A (ten)

Binary numbers are represented with prefix 0b, i.e. 0b00001010 (ten)