Life Energy Motion

E-Mobility solutions

DCBM 400/600 Series - DC Energy Meter

Communication protocols manual

2July2021/Version 0

Table Of Contents

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

1. SAFETY RULES

1.1. Safety warning

In order to guarantee safe operation of the product and to be able to make proper use of all features and functions, please read these instructions thoroughly!

Safe operation can only be guaranteed if the product is used for the purpose it has been designed for and within the limits of the technical specifications. Ensure you get up-to-date technical information that can be found in the latest associated datasheet under www.lem.com.

Terminal protection cover delivered with the product must be installed to obtain proper electrical protection. The data link cable used between the product's elements shall be the one delivered by LEM.

Please note

Electrical equipment should be installed, operated, serviced and maintained only by qualified personnel.

No responsibility is assumed by LEM International SA for any consequences arising out of the use of this material. A qualified person is one who has skills and knowledge related to the construction, installation, and operation of electrical equipment and has received safety training to recognize and avoid the hazards involved.

The meter must be installed inside an enclosure IP51 (indoor) or IP54 (outdoor) according to

EN 50470:2007.

DANGER! Electrical hazard - Fire hazard

When installing or changing the product, the conductor to which the product is connected must be de-energized. Ignoring the warnings can lead to serious injury and/or cause damage!

Notice! Damage or hazards

The appropriate torque is defined by LEM (see Operation manual "Table 2: Torque values for installation" and "Mounting and unmounting" section).

The appropriate crimping of the connection elements is defined by the nationalities in force.

1.2. Important notices

- Time source to set product's time must be provided by the customer. Product must be time synchronized to operate.
- Product's Ethernet interface mustn't be exposed to a public network; network must be private and secured.
- To ensure proper operation, product's logbook completion must be checked periodically; the maximum number of entries is approximately 40 000; product's operation stops if logbook is full.
- The product is designed with IP20, and is intended to be mounted in an enclosure with a suitable IP rating for the final application.

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

Accuracy notices according to PTB type examination certificate

- The direct current meters may only be used for billing purposes in business and official traffic in a charging device and only for measuring the energy supplied to the vehicle.
- The connection line for voltage measurement must be provided with the supplied ferrite so that the measurement reliability is guaranteed in the event of interference.
- For the device types DCBM_N0D_4000_0000, DCBM_N0D_4010_0000, DCBM_N0D_6000_0000 and DCBM_N0D_6010_0000, for which the compensation factor can be selected via the Ethernet interface, the interface must be sealed or a calibrated remote station must be directly connected and secured by means of seals. At the end of the transaction, this remote station must compare the cable ID specified in the data tuple signed by the DC meter with the cable ID originally transferred to the DC meter. If this comparison is valid then the data tuple can be used. The remote station can also directly overwrite the cable ID specified in the data tuple.

1.3. Symbols and conventions

The following symbols point out critical information. They can be found either in this document or on the product.


Instruction manual


Heating


Warning


Electrical hazard


Double insulation

The following symbols aim at improving reader's experience by highlighting sections.


Note


Tip

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

2. Document information

2.1. Document overview

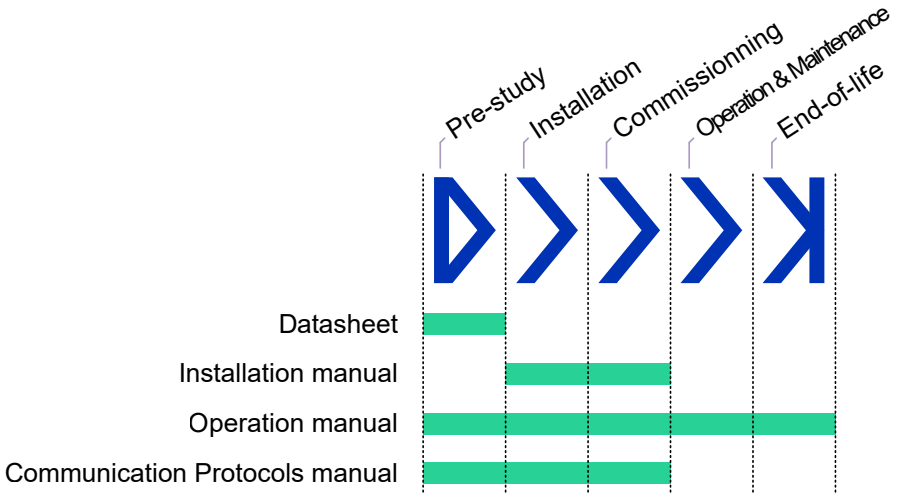
This document relates to the DCBM 400/600 product family. Those products are direct connected energy meters for DC applications.

This manual provides detailed information for interfacing with the products. This includes

- Checking and setting the configuration
- Monitoring status and measurements
- Managing new transactions
- Retrieving the stored data: transaction & event logbooks

The document is structured following the APIs of the products, it explains available fields together with examples.

This document is intended to be used in combination with the DCBM 400/600 Operation manual which describes the relevant concepts. More generally, below illustration depicts the set of documents for DCBM 400/600 product family with associated steps in product lifetime.



LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

2.2. Document issue

Targeted Product DCBM 400/DCBM 600

Release scope Public

Applies to software versions

- APPLICATION_VERSION = 0.1.4.0
- METROLOGY_VERSION = 0.1.4.0

2.3. Company details

LEM International SA

Chemin des Aulx 8

1228 Plan-les-Ouates

Switzerland

2.4. Revision history

Version	Date	Changes
0	2 July 21	First issue

2.5. Related documents

Name	Revision
Datasheet	
Operation manual	
Installation manual	
EN 50470-1	2006
EN 50470-3	2006
IEC 62477-1	
EN 62052-31	2015
IEC 62052-11	2003
IEC 61000-6-2	2016
IEC 61000-6-3	2016
RoHS	
OCPP 1.6	1.6, 2.6
WELMEC 7.2 Software Guide	2015
REA-6A	
PTB 50.7	
VDE-AR-E 2418-3-100	2019-07-16

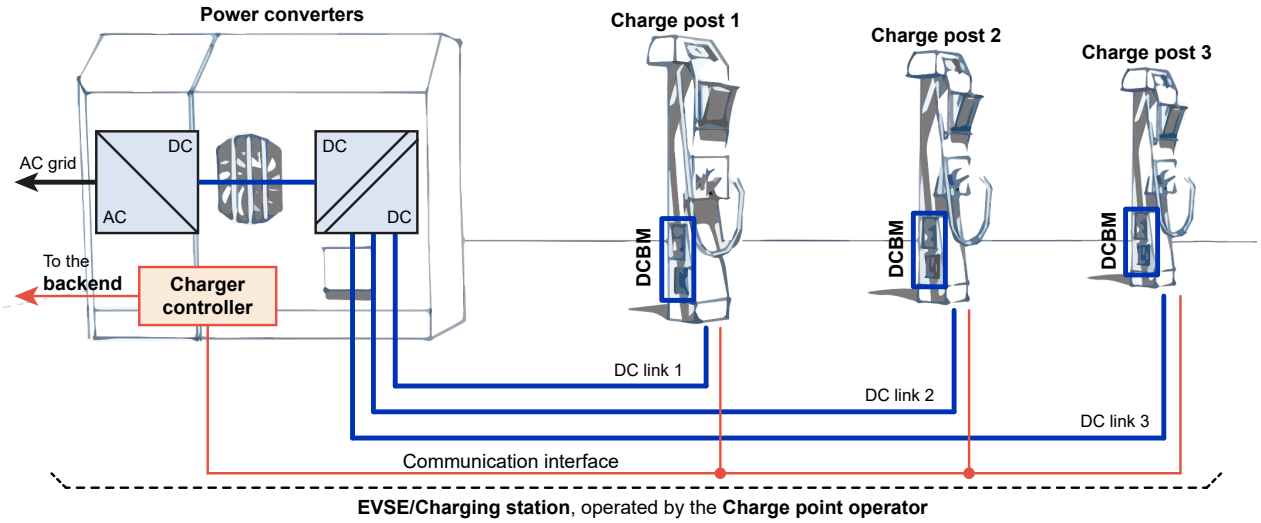
2.6. Abbreviations

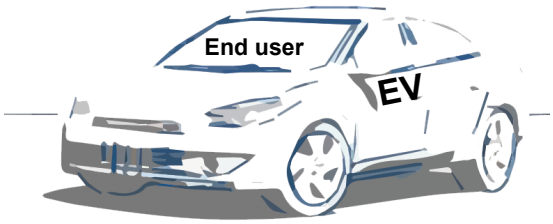
Abbreviation	Description	Abbreviation	Description
ABS	Absolute value	URL / URI	Uniform Resource Locator / Identifier
ADC	Analog to Digital Converter	UTC	Coordinated Universal Time
API	Application Programming Interface	UTF	Universal Character Set Transformation Format
ASCII	American Standard Code for Information Interchange	XTAL	Cristal (oscillator)
CR / LF	Carriage Return / Line Feed		
CRC	Cyclic Redundancy Checksum		
DC	Direct current		
DCBM	DC Billing Meter		
DER	Distinguished Encoding Rules		
DHCP	Dynamic Host Configuration Protocol		
DNS	Domain Name System		
DST	Daylight Saving Time		
ECDSA	Elliptic Curve Digital Signature Algorithm		
EV	Electric Vehicle		
EVSE	Electric Vehicle Supply Equipment		
FF	Fatal Failure		
GMT	Greenwich Mean Time		
GPIO	General Purpose Input / Output		
HTTP[S]	Hypertext Transfer Protocol [Secured]		
ID	Identifier		
IP	Internet Protocol / Ingress Protection Code		
JSON	JavaScript Object Notation		
LEN	Length		
LR / LNR	Legally Relevant / Legally Non-Relevant		
MSB / LSB	Most Significant Bit / Least Significant Bit		
MU	Meter Unit		
N/A	Not Applicable		
NTP	Network Time Protocol		
OBIS	Object Identification System		
OCMF	Open Charge Metering Format		
OCPP	Open Charge Point Protocol		
OVC	Overvoltage Category		
PLMN	Public Land Mobile Network		
REST	Representational state transfer		
RFID	Radio Frequency Identification		
RNG	Random Number Generator		
SHA	Secure Hash Algorithm		
SMS	Short Message Service		
SNTP	Simple Network Time Protocol		
SU	Sensor Unit		
UDP	User Datagram Protocol		
UID / UUID	[Universally] Unique Identifier		

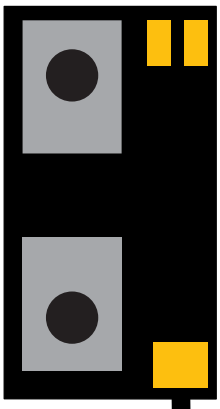
LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

2.7. Glossary

Key words used in this document, designating DCBM construction or DCBM's ecosystem equipment and protagonists, are illustrated in below visuals.

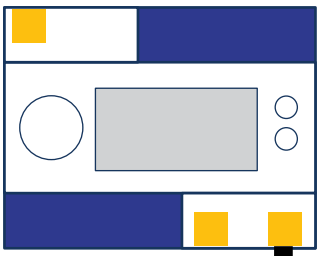






Sensor Unit

The device is connected to busbars or wired systems, and monitors accurately current and voltage to measure energy continuously. Data is securely transferred to the Meter Unit.



Meter Unit

The device stores, displays, communicates and authenticates measurement data. The Meter Unit delivers power supply to the Sensor Unit.

The Meter Unit offers communication with the charging station through Ethernet /REST APIs.

Data link cable

Delivered with the product, this cable connects Meter Unit to Sensor Unit, both terminals can be sealed.

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

2.8. Disclaimer

LEM cannot be held liable for damage, injury or any legal responsibility incurred directly or indirectly from non product quality issues such as other use of the DCBM than according to LEM written installation instructions (see section “4. Device description and mechanical integration”) or other external factors.

The user shall observe safe and lawful practices, including, but no limited to, those set forth in this document. Before any operation or use, please read “Safety” section carefully.

LEM reserves the right to carry out modifications on its product and documentation at the sole discretion of LEM. Always make sure to have the latest information before placing an order. For up-to-date product information, visit www.lem.com or contact your nearest LEM sales representative.

2.9. Intellectual Property Rights

© Copyright 2020 LEM INTERNATIONAL SA. All rights reserved.

This document shall not be reproduced, copied, adapted, translated, arranged or modified without written permission from LEM, and the content, in whole or in any part, shall not be used for any purpose other than describing LEM DCBM. LEM will retain all intellectual property rights in and to this document. No license is granted by LEM to any intellectual property right. All rights not expressly granted are reserved by LEM.

Disassembly, decompilation, reverse-engineering, decryption, or alteration of the DCBM, including its software, are prohibited.

“LEM” and any related logos are protected trademarks owned by LEM HOLDING SA or trade names and cannot be used for other purposes than LEM usage without specific prior written permission.

Unless otherwise expressly granted by LEM in writing, the sale of the DCBM will not confer any license under any patents, trademarks, trade names, or other proprietary rights owned or controlled by LEM, its affiliates or suppliers, all rights being reserved.

2.10. Third party licensing

The DCBM includes software developped by SEGGER Microcontroller GmbH (SystemView RTT), STMicroelectronics (STM32Cube) and ARM LIMITED (CMSIS), available under BSD license, with following conditions.

Copyright © 2014 - 2020 SEGGER Microcontroller GmbH

Copyright © 2016 STMicroelectronics

Copyright © 2020 ARM LIMITED

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The DCBM includes software developped by Amazon Inc (FreeRTOS), available under MIT license, with following conditions.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

2.11. Warranty

For information about applicable warranty for the DCBM, contact your nearest LEM sales representative.

In the absence of any written agreement with LEM governing the sale of the DCBM to you, LEM general terms and conditions of sale as referred on the order confirmation shall apply. LEM disclaims all warranties of any kind, except as expressly provided in the above agreement or terms and conditions of sale, whether express or implied, relating to the DCBM and its documentation, including any implied warranty of merchantability, fitness for a particular purpose or non-infringement.

3. Introduction

Ethernet is the communication channel for the DCBM. It supports HTTP/REST communication to receive requests and provide measurements and other data.

The REST-compatible API is an application programming interface that uses HTTP requests to obtain (GET), place (PUT) and publish (POST) data. A RESTful API conforms to the Representational State Transfer (called “REST” model). This interface is using JSON format for the data payload.

The DCBM REST interface is structured as follows:

URI with default IP	Methods	Description
192.168.1.2/V1/status	GET	Status of the DCBM
192.168.1.2/v1/settings	GET, PUT	Settings of the DCBM
192.168.1.2/v1/logbook	GET	Event logger of the DCBM
192.168.1.2/v1/livemeasure	GET	Live measurements
192.168.1.2/v1/legal	GET, PUT, POST	Transaction management (start & stop) and transaction data structure, current or stored, in LEM proprietary format
192.168.1.2/v1/ocmf	GET	Transaction data structure, current or stored, in certified, billable, OCMF-compliant format
192.168.1.2/v1/certificate	GET	HTTPS certificate of the DCBM

In this document, request and response headers are described for all types of REST requests supported by the DCBM. Those headers specify how to properly configure a client to ensure functional communication.

The DCBM supports the following methods of `RESTful API`

- `POST`: publish a data
- `GET`: obtain a data
- `PUT`: place a data

In this document the IP address and port are set to the default values:

```
URI: http://192.168.1.2:80/
```

When describing in sections below a REST API header, the following formalism is used:

```
<COMMAND> <PATH> HTTP/1.1
```

With

- `<COMMAND>` = `REST` command (ex: `POST`)
- `<PATH>` = path to add to the `URI` (ex: `/v1/legal`)

Remark: All fields size in this document are given in bytes, without counting ending `'\0'` NULL characters needed for storing a string in C language

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

3.1. Request header

3.1.1. Header formatting

The following request headers shall be formatted as follows:

```
POST /v1/legal HTTP/1.1
Host: <DCBM IP>
Content-Type: application/json
Content-Length: strlen(<BODY>)
<BODY>
```

With `<BODY>` = message sent to the DCBM.

! Close the line above with "`\r\n`" (named "`CRLF`" or carriage return and line feed)

! Extra `CRLF` needed before the `BODY`

3.1.2. GET request

```
GET <PATH> HTTP/1.1
Host: <DCBM IP>
```

3.1.3. PUT request

```
PUT <PATH> HTTP/1.1
Host: <DCBM IP>
Content-Type: application/json
Content-Length: <SIZE>

<BODY>
```

with `<SIZE>` = `strlen(<BODY>)`

3.1.4. POST request

```
POST <PATH> HTTP/1.1
Host: <DCBM IP>
Content-Type: application/json
Content-Length: <SIZE>

<BODY>
```

with `<SIZE>` = strlen(`<BODY>`)

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

3.2. Response header

3.2.1. Header formating

The following response headers are formatted as follows:

```
HTTP/1.1 <ERROR_CODE> <STATUS>
Connection: close
```

With:

- `<ERROR_CODE>` = response code sent by the DCBM.
- `<STATUS>` = `HTTP` status (example : OK / Forbidden / ...)

`HTTP/1.1` Transfer encoding in chunked block is supported, with max size of blocks = `0x100` = 256 bytes

3.2.2. Success case

```
HTTP/1.1 200 OK
Connection: close
Content-Type: application/json
Transfer-Encoding: chunked
```

3.2.3. Failing case

```
HTTP/1.1 400 Bad Request
Connection: close
```

3.2.4. Status codes

Below are the implemented HTTP error codes, used in responses.

Code Number	Meaning	PUT	POST	GET
200	OK	used	used	used
201	Created		used	
308	Permanent redirect	used	used	used
400	Bad request	used	used	used
403	Forbidden		used	
404	Not Found			used
405	Method not allowed	used	used	
412	Preconditions failed	used	used	
500	Internal server error	used	used	used
501	Not implemented	used	used	used

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

3.2.5. Chunked transfer

Typical HTTP chunked response is with max block size of 256 bytes, with length indicated at beginning of the data:

HTTP chunked response

```
100
<BODY_CHUNK>chunk
100
<BODY_CHUNK>
...
```

The last block is identified with:

```
<REMAINING_LENGTH_IN_HEXADECIMAL_FORMAT>
<LAST_BODY_CHUNK>
0
```

Length before chunked body responses are expressed in hexadecimal format, without the "0x" prefix.

Here is a summary of the maximum byte size for storing all the fields (worst cases). Add +1 if it is stored as a string to terminate with "\0" character.

REST interface	Max body size (byte)
status	1395
settings	1339
legal	1272
ocmf	1024
logbook	10 Mbytes
livemeasure	210

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

4. /SETTINGS API

/settings API allows configuration of the DCBM. Some fields are read-only; freely settable fields are `ipAddress`, `dhcp`, `ntp`, `time`, `http`, `ocmf`.

For memory reliability purposes, following rules shall be observed:

- Ensure that the DCBM has been powered **for 2 minutes before writing a setting**.
- Wait for **0.5s before powering-down** the device after writing a setting.

4.1. Overview

```
{
  "meterId": string,
  "cableConf": [
    {
      "cableSpId": integer,
      "cableSpName": string,
      "cableSpRes": integer
    },
    {
      "cableSpId": integer,
      "cableSpName": string,
      "cableSpRes": integer
    },
    {
      "cableSpId": integer,
      "cableSpName": "string,
      "cableSpRes": integer
    },
    {
      "cableSpId": integer,
      "cableSpName": string,
      "cableSpRes": integer
    },
    {
      "cableSpId": integer,
      "cableSpName": string,
      "cableSpRes": integer
    },
    {
      "cableSpId": integer,
      "cableSpName": string,
      "cableSpRes": integer
    },
    {
      "cableSpId": integer,
```


LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

```
      "cableSpName": string,
      "cableSpRes": integer
    },
    {
      "cableSpId": integer,
      "cableSpName": string,
      "cableSpRes": integer
    }
  ],
  "ntp": {
    "servers": [
      {
        "ipAddress": string,
        "port": integer
      },
      {
        "ipAddress": string,
        "port": integer
      }
    ],
    "syncPeriod": integer,
    "ntpActivated": boolean,
    "syncTimeout": integer
  },
  "dhcp": {
    "ipAddress": string,
    "serverPort": integer,
    "clientPort": integer,
    "activation": boolean
  },
  "ipAddress": string,
  "http": {
    "tls_on": boolean,
    "httpPort": integer
  },
  "pulseOutputRate": integer,
  "pulseOutputFreq": integer,
  "time": {
    "utc": string,
    "tz": string,
    "dst": {
      "activated": boolean,
      "offset": integer,
      "start": {
        "order": string,
        "day": string,
```


LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

```
            "month": string,
            "hour": string
          },
          "end": {
            "order": string,
            "day": string,
            "month": string,
            "hour": string
          }
        }
      },
      "ocmfId": {
        "IL": integer,
        "IF": {
          "Rfid": integer,
          "Ocpp": integer,
          "Iso15118": integer,
          "Plmn": integer
        },
        "IT": integer
      }
    }
```


The following fields are read-only:

- /meterId
- /cableConf
- /ntp/servers/syncTimeout
- /pulseOutputRate
- /pulseOutputFreq

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

4.2. Fields description

4.2.1. meterId

The DCBM Serial Number, a pre-set string of max size 37.

Example

```
"meterId": "1202407280"
```

This field is not editable, set in factory.

4.2.2. cableConf

A pre-set array of data to handle cable loss compensation. The concept is to estimate cable losses, to subtract them from energy result of transactions. More details can be found in the Operation manual, section “Handling energy losses”.

This is a 3 fields structure, nested in a 8 entries array.

Example

```
"cableConf": [
   {
      "cableSpId": 0,
      "cableSpName": "no cable",
      "cableSpRes": 0
   },
   {
      "cableSpId": 1,
      "cableSpName": "2 mOhm",
      "cableSpRes": 2
   },
   ...
],
```

- `cableSpId` = cable index value, used for a START of transaction (values from 0 to 7)
- `cableSpName` = cable name (max size 19 bytes)
- `cableSpRes` = cable resistance value in mOhm (value from 0 to 255).

Those fields are not editable, set in factory.

Values of those fields differ among product variants. See Operation manual, section “Product designation”.

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

4.2.3. ntp

Field used for time synchronization through SNTP protocol, allowing synchronization with 2 servers. Details can be found in Operation manual, section “NTP synchronization”.

Example

```
"ntp": {
    "servers": [
      {
        "ipAddress": "192.168.1.1",
        "port": 123
      },
      {
        "ipAddress": "192.168.1.1",
        "port": 123
      }
    ],
    "syncPeriod": 21600,
    "ntpActivated": false,
    "syncTimeout": 172800
  }
```

- `servers` = an array of 2-objects structure
- `ipAddress` = IP address of server offering the SNTP service (UDP protocol):
  - » Using following format : `"W.X.Y.Z"`, with each letter coding a integer of one byte size.
  - » Or an URL (maximum size is 255 chars)
- `port` = server UDP port (max = 65535). Defaut port as per IANA is 123.
- `syncPeriod` = period of NTP synchronization (in seconds , min 60, max 2^32-1)
- `ntpActivated` = boolean to enable/disable SNTP time synchronization. If disabled, “command time synchronization” must be used (see “4.2.9. time”)
- `syncTimeout` = expiration timeout of the synchronization (in seconds). Pre-set value is 48 hours, not editable. Once expired, time synchronization is no longer valid.

The DCBM is rated as `INFO` time (and not `SYSTEM` time).

In case `ntpActivated` = true while time synchronization by command is used, NTP is automatically disabled (i.e. `ntpActivated` = false)

4.2.4. dhcp

DHCP feature can be enabled or disabled with the activation flag:

`"dhcp":{"activation":boolean}` (with true = enabled, false = disabled)

- In case it is disabled (default settings), the IP of the DCBM is the one in the /settings `ipAddress` field.
- In case it is activated, the IP is expected to be received from the network. If no IP is received, the fallback IP address is: `0.0.0.0` (which allows no HTTP communication).

Field `dhcp/ipAddress` allows specifying a specific DHCP server address. Do not confuse it with /settings field `ipAddress` (“4.2.5. ipAddress”)

Current IP address can be displayed in the maintenance screens. See Operation manual, section “Maintenance state”. On the contrary, `ipAddress` at the root of /settings API will not be updated according to dynamic DHCP address (only an input for static addressing, see description below).

Example

```
"dhcp": {
   "ipAddress": "0.0.0.0",
   "serverPort": 67,
   "clientPort": 68,
   "activation": false
 },
```

- `ipAddress` = (optional) the IP address server that offers the DHCP service (UDP protocol), using following format : “W.X.Y.Z”, each letter coding a integer of one byte.
- `serverPort` = UDP port on server side (max = 65535)
- `clientPort` = UDP port on client side (max = 65535)
- `activation` = boolean, activation of the DHCP feature

4.2.5. ipAddress

Field to set the IP address of the DCBM when DHCP is disabled.

Example

```
"ipAddress": "192.168.1.2",
```

`ipAddress` = using following format: “W.X.Y.Z”, each letter coding an integer of max one byte.

The subnet mask is “0.0.0.0”. It cannot be changed. This value allows access from any other IP address (i.e. no subnet mask restriction).

The DCBM can also be reached using its DNS name, just like using its IP address. The DCBM’s DNS name is `"lem-meter"`.

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

4.2.6. http

Field to activate the HTTPS feature and configure the HTTP port.

Example

```
"http": {
    "tls_on": false,
    "httpPort": 80
  }
```

`tls_on` = boolean to enable/disable HTTPS

`httpPort` = port for HTTP usage

Default port numbers as defined by IANA are:

- 80 for HTTP
- 443 for HTTPS

Toggling `tls_on` does not automatically change the port to use; do not forget to set it.

Using HTTPS, the DCBM certificate needs to be accepted by the Charging controller (it is not signed by a Central Authority, as the duration of the certificate is set to 999 years). Otherwise, the DCBM will accept any HTTPS certificate.

4.2.7. pulseOutputRate

This field is not writable, reserved.

Example

```
"pulseOutputRate" : 1
```

4.2.8. pulseOutputFreq

This field is not writable, reserved

Example

```
"pulseOutputFreq" : 50
```


LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

4.2.9. time

Field description

- "utc" = the UTC legal time
- "tz" = the timezone of the location of the DCBM: it can go from -11 to +14 for hour, and 00, 15, 30, 45 for minutes. The timezone is the time shift compared to UTC time.

If tz = "+00:00", then local time matches UTC, so time in /settings and /legal will be displayed as UTC timestamps (i.e. with ending "Z" letter).

Some countries have several time zones and some countries use non-integer timezone (example: Iran is UTC+3:30)

In southern hemisphere, DST starts around October, (i.e. start and end are reversed compared to northern hemisphere).

DST offset is applied between "start" and "end" fields below

- "dst" = the Daylight Saving Time (DST) settings
- "activated" = JSON boolean (true/false) that activates the DST
- "offset" = the number of minutes that consists in the deviation applied for the DST activation. Shall be a positive value (usually 60 minutes, the default value).
  - » "start" = start of DST
    - * "order" = "first", "second", "last"
    - * "day" = "monday"..."sunday"
    - * "month" = "january"..."december"
    - * "hour" = hour when DST starts, expressed in local time (e.g. "01:00") or UTC reference ("T00:00Z")
  - » "end" = end of DST
    - * "order" = "first", "second", "last"
    - * "day" = "monday"..."sunday"
    - * "month" = "january"..."december"
    - * "hour" = hour when DST ends, expressed in local time (e.g. "01:00") or UTC reference ("T00:00Z")

The "hour" fields must not include DST offset.

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

Below is an example of the time JSON struct, commented.

```
"time" : {
  "utc":"2019-07-17T14:46:26Z",   // UTC manual time settings
  "tz":"+01:00",                  // Time zone offset (here: UTC+1)
  "dst" : {                       // Daylight Saving Time fields
    "activated":true,             /* Activation of the Daylight Saving
                                     Time (here: enable DST) */
    "offset":60,                  /* Offset of DST (here: apply 60mn to
                                     time with DST) */
    "start" : {                   /* Start of DST (i.e. when DST offset
                                     starts being applied) */
// Here, DST starts on last Sunday of March at 1 a.m. UTC:
      "order":"last",             // first, second or last
      "day":"sunday",             // day of week when DST start
      "month":"march",            // month to start DST
      "hour":"T01:00Z"            /* hour to start DST (here: "T01:00Z" =
                                     local time "02:00") */
      },
    "end" : {                     /* End of DST (i.e. when DST offset is no
                                     longer applied) */
// Here, DST ends on last Sunday of October at 1 a.m. UTC:
      "order":"last",             // first, second or last
      "day":"sunday",             // day of week when DST ends
      "month":"october",          // month to end DST
      "hour":"T01:00Z"            /* hour to end DST (here: "T01:00Z" =
                                     local time "02:00") */
    }
  }
}
```

The field `time/utc` allows setting time, as “command time synchronization”. Concept is explained in Operation manual section “Command time synchronization”. Below is an example:

```
"time": {
  "utc":"2020-07-17T14:46:26Z"
}
```

Using command time synchronization, the time must be written at least once a day.

If after a period of 48h the time was not set, time synchronization expires (preventing new transactions and invalidating on-going one).

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

4.2.10. ocmfId

The `ocmfId` field is used for /ocmf API for the user identification.

The Charger Controller is responsible for those settings.

This information depends on the protocol used for user identification as well as on the level of security used and assessed.

The Charger Controller (or the user) can configure OCMF settings of the DCBM in /settings API.

When not configured, all the fields are set by default to `"0"`.

```
"ocmfId":{
   "IL": 0,
   "IF":{
      "Rfid": 0,
      "Ocpp": 0,
      "Iso15118": 0,
      "Plmn": 0
   },
   "IT": 0
}
```

Sub-fields are described in following sections.

Due to current software limitation, the OCMF `IT` setting shall by configured once (or kept as default) and not changed during the lifetime of the DCBM.

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

4.2.10.1. IL field: Identification level

Code Number	Meaning	Description
0	"-"	The field is not specified.
1	"NONE"	No user mapping. The other data on user assignment have no significance.
2	"HEARSAY"	The assignment is unsecured; e.g. by reading out an RFID UID.
3	"TRUSTED"	The assignment can be trusted to some extent, but there is no absolute reliability. Example: authorization by backend.
4	"VERIFIED"	The assignment has been verified by the signature component and specific actions.
5	"CERTIFIED"	The mapping was verified by the signature component using a cryptographic signature that certifies the mapping.
6	"SECURE"	The assignment was established by a safe feature (e.g. secure RFID card, ISO15118 with plug and charge, etc.)
7	"MISMATCH"	Error: UIDs don't match.
8	"INVALID"	Error: Certificate not correct.
9	"OUTDATED"	Error: Referenced trust certificate expired.
10	"UNKNOWN"	Error: Certificate could not be verified (no matching trust certificate found).

4.2.10.2. IF field: Identification flags

This section describes identification flag fields, describing detailed user mapping statements.

Field `rfid`

Code Number	Meaning	Description
0	"RFID_NONE"	No assignment by RFID
1	"RFID_PLAIN"	Assignment via external RFID card reader
2	"RFID_RELATED"	Assignment via protected RFID card reader
3	"RFID_PSK"	A previously known common key (pre-shared key) was used, e.g. with a secured RFID card

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

Field `ocpp`

Code Number	Meaning	Description
0	`"OCPP_NONE"`	No user assignment by OCPP
1	`"OCPP_RS"`	Mapping by OCPP RemoteStart method
2	`"OCPP_AUTH"`	OCPP Authorize method mapping
3	`"OCPP_RS_TLS"`	Transport layer security was used to transfer the mapping using the OCPP RemoteStart method
4	`"OCPP_AUTH_TLS"`	Transport layer security was used to transfer the mapping using the OCPP Authorize method
5	`"OCPP_CACHE"`	OCPP authorization cache mapping
6	`"OCPP_WHITELIST"`	Assignment by white-list of OCPP
7	`"OCPP_CERTIFIED"`	A certificate of the backend which certifies the user assignment was used

Field `iso15118`

Code Number	Meaning	Description
0	`"ISO15118_NONE"`	No user assignment by ISO15118
1	`"ISO15118_PNC"`	Plug & Charge was used

Field `plmn`

Code Number	Meaning	Description
0	`"PLMN_NONE"`	No assignment
1	`"PLMN_RING"`	Call
2	`"PLMN_SMS"`	Message

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

4.2.10.3. IT field: Identification type

IT field describes the types of identification data

Code Number	Meaning	Description
0	"NONE"	No assignment available
1	"DENIED"	Assignment is currently unavailable (e.g. two-factor authorization)
2	"UNDEFINED"	Type not specified
3	"ISO14443"	UID of an RFID card according to ISO 14443. Shown as 4 or 7 bytes in hexadecimal notation
4	"ISO15693"	UID of an RFID card according to ISO 15693. Shown as 8 bytes in hexadecimal notation
5	"EMAID"	Electro-Mobility Account ID according to ISO/IEC 15118 (string with length 14 or 15)
6	"EVCCID"	ID of an electric vehicle according to ISO/IEC 15118 (maximum length 6 characters)
7	"EVCOID"	EV contract ID according to DIN 91286.
8	"ISO7812"	Identification card format according to ISO/IEC 7812 (credit and bank cards, etc.)
9	"CARD_TXN_NR"	Card transaction number for a payment with a credit or debit card used in a terminal at the charging point
10	"CENTRAL"	Centrally generated ID. No exact format defined, for example, can be a UUID (OCPP 2.0)
11	"CENTRAL_1"	Centrally generated ID, e.g. started via SMS. No exact format defined (up to OCPP 1.6)
12	"CENTRAL_2"	Centrally generated ID, e.g. started by operator. No exact format defined (up to OCPP 1.6)
13	"LOCAL"	Locally generated ID. No exact format defined, for example, can be a UUID (OCPP 2.0)
14	"LOCAL_1"	Locally generated ID, e.g. ID generated internally by the load point. No exact format defined (up to OCPP 1.6)
15	"LOCAL_2"	Locally generated ID, for other cases. No exact format defined (up to OCPP 1.6)
16	"PHONE_NUMBER"	International telephone number with leading "+"
17	"KEY_CODE"	User-related, private key code. No exact format defined

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

4.3. Allowed requests

4.3.1. PUT - Write /settings

Request format

Settings can be set individually or grouped in a single HTTP request. If several settings are written at once, their order shall be maintained.

Any writing attempt of settings shall be validated using DCBM acknowledge. The DCBM acknowledge is the `result` field in its response (see “Response format” below).

- Template

```
PUT /v1/settings HTTP/1.1
Content-Type: application/json
Content-Length: strlen(<BODY>)

<BODY>
```

- Example

```
PUT /v1/settings HTTP/1.1
Content-Type: application/json
Content-Length: 41

{“time”: {“utc”: “2019-10-23T15:07:05Z”}}
```

Response format

HTTP chunked response for the previous example.

```
{“meterId”:”1202407280”,”result”:1}
```

When `result` = 1 it indicates that the DCBM acknowledged the request.

When `result` = 0 it indicates that the request was rejected (for instance if a value set is out of range).

Status codes

Code Number	Meaning
200	OK (but `result` must be checked)
400	Bad request

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

4.3.2. GET - Read /settings

Response format

```
GET /v1/settings HTTP/1.1
```

Response format

See section “4.1. Overview”.

Status codes

Code Number	Meaning
200	OK

4.4. Examples

4.4.1. Read settings

- Linux/Windows bash

```
curl -X GET http://192.168.1.2/v1/settings
```

- Windows (PowerShell)

```
Invoke-WebRequest -uri “http://192.168.1.2/v1/settings” | Select-Object
-Expand Content
```

4.4.2. Write all settings at once

In this example we write all the following settings in one command:

- `ntp`
- `dhcp`
- `ipAddress`
- `http`
- `ocmfId`
- `time`

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

- Linux/Windows bash

```
curl -d
'{"ntp":{"servers":[{"ipAddress":"192.168.1.1",
"port":123},{"ipAddress":"192.168.1.1","port":123}],
"syncPeriod":900},"dhcp":{"ipAddress":"0.0.0.0",
"serverPort":67,"clientPort":68,"activation":false},
"ipAddress":"192.168.1.2","http":{"tls_on":false,
"httpPort":80},"ocmfId":{"IL":1,"IF":{"Rfid":0,
"Ocpp":1,"Iso15118":1,"Plmn":0},"IT":5},
"time":{"tz":"+01:00","dst":{"activated":true,
"offset":60,"start":{"order":"last","day":"sunday",
"month":"march","hour":"T01:00Z"},
"end":{"order":"last","day":"sunday",
"month":"october","hour":"T01:00Z"}'
-H "Content-Type: application/json" -X PUT http://192.168.1.2/v1/settings
```

- Windows (PowerShell)

```
Invoke-WebRequest -uri "http://192.168.1.2/v1/settings" -ContentType
"application/json" -Method PUT -Body '
{"ntp":{"servers": [{"ipAddress":"192.168.1.1",
"port":123},{"ipAddress":"192.168.1.1","port":123}],
"syncPeriod":900},"dhcp":{"ipAddress":"0.0.0.0",
"serverPort":67,"clientPort":68,"activation":false},
"ipAddress":"192.168.1.2","http":{"tls_on":false,
"httpPort":80},"ocmfId":{"IL":1,"IF":{"Rfid":0,
"Ocpp":1,"Iso15118":1,"Plmn":0},"IT":5},
"time":{"tz":"+01:00","dst":{"activated":true,
"offset":60,"start":{"order":"last","day":"sunday",
"month":"march","hour":"T01:00Z"},"end":{"order":"last",
"day":"sunday","month":"october","hour":"T01:00Z"}
' | Select-Object -Expand Content
```

4.4.3. Write only one setting

In this example we set the UTC time value to "2020-03-25T14:53:06Z"

- Linux/Windows bash

```
curl -d '{"time":{"utc":"2020-03-25T14:53:06Z"}}' -H 'Content-Type:
application/json' -X PUT http://192.168.1.2/v1/settings
```

- Windows (PowerShell)

```
Invoke-WebRequest -uri "http://192.168.1.2/v1/settings" -ContentType
"application/json" -Method PUT -Body '{"time":{"utc":"2020-03-
25T14:53:06Z"}} ' | Select-Object -Expand Content
```


LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

5. /STATUS API

5.1. Overview

All status fields are read-only (GET method only).

```
{
  "status": {
    "value": integer,
    "bits": {
      "suLinkStatusIsOk": boolean,
      "muFatalErrorOccured": boolean,
      "transactionIsOnGoing": boolean,
      "tamperingIsDetected": boolean,
      "timeSyncStatusIsOk": boolean,
      "overTemperatureIsDetected": boolean,
      "reversedVoltage": boolean,
      "suMeasureFailureOccurred": boolean
    }
  },
  "version": {
    "applicationFirmwareVersion": string,
    "applicationFirmwareAuthTag": string,
    "legalFirmwareVersion": string,
    "legalFirmwareAuthTag": string,
    "sensorFirmwareVersion": string,
    "sensorFirmwareCrc": string
  },
  "time": string,
  "ipAddress": string,
  "meterId": string,
  "errors": {
    "value": integer,
    "bits": {
      "muInitIsFailed": boolean,
      "suStateIsInvalid": boolean,
      "versionCheckIsFailed": boolean,
      "muRngInitIsFailed": boolean,
      "muDataIntegrityIsFailed": boolean,
      "muFwIntegrityIsFailed": boolean,
      "suIntegrityIsFailed": boolean,
      "logbookIntegrityIsFailed": boolean,
      "logbookIsFull": boolean,
      "memoryAccessIsFailed": boolean,
      "muStateIsFailed": boolean,
    }
  },
```


LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

```
  "publicKey": string,
  "publicKeyOcmf": string,
  "indexOfLastTransaction": integer,
  "numberOfStoredTransactions": integer
}
```


5.2. Fields description

5.2.1. status

This field indicates the current status of the DCBM.

Example


```
"status": {
   "value": 17,
   "bits": {
     "suLinkStatusIsOk": true,
     "muFatalErrorOccured": false,
     "transactionIsOnGoing": false,
     "tamperingIsDetected": false,
     "timeSyncStatusIsOk": true,
     "overTemperatureIsDetected": false,
     "reversedVoltage": false,
     "suMeasureFailureOccurred": false
   }
 },
```


- `value` = is the decimal value (max 255) and is calculated as the integer value corresponding to the binary value made of the associated bit flags. More explanations can be found in the Operation manual, section “Status/errors value field description”.

The nominal value is =17 (`suLinkStatusIsOk` is true and `timeSyncStatusIsOk` is true). If value is =1 (`suLinkStatusIsOk`), the time synchronization period has expired (UTC time shall be set again).

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

- `bits` fields, corresponding to value field broken down by unitary flags

Name	Bit index	Meaning	Nominal value	Severity	Associated event logbook entry name
`suLinkStatusIsOk`	0	true: the SU is present and working properly	true	Blocking	`STATUS_SENSOR_LINK`
`muFatalErrorOccured`	1	true: there is a fatal error (FF) detected	false	Blocking	`STATUS_FATAL_ERROR`
`transactionIsOnGoing`	2	true: a transaction is running	false	Blocking	`STATUS_START_STOP_TRANSACTION`
`tamperingIsDetected`	3	true: a tampering attempt is currently seen	false	Informational	`STATUS_TAMPERING`
`timeSyncStatusIsOk`	4	true: the UTC time is properly synchronized (INFO time)	true	Blocking	`STATUS_TIME_SYNC`
`overTemperatureIsDetected`	5	true: the SU is currently in over-temperature (above 80°)	false	Informational	`STATUS_OVERHEAT`
`reversedVoltage`	6	true: the voltage inputs are inverted	false	Informational	`STATUS_REVERSED_VOLTAGE`
`suMeasureFailureOccurred`	7	true: an ADC measurement occurred in the SU	false	Informational	`STATUS_MEASURE_ERROR`

! If `reversedVoltage` = true, this indicates that a voltage level below -50V is seen. The polarity of the voltage sensor probes shall be checked, this may indicate a wrongful connection.

i “Blocking” level indicates that if different from normal value, new transactions are rejected.

5.2.2. version

This field is used to track version and checksum of the DCBM firmware parts.

Example

```
"version": {
   "applicationFirmwareVersion": "0.1.4.0",
   "applicationFirmwareAuthTag": "663A7BA7A685BD6A7C43F136",
   "legalFirmwareVersion": "0.1.4.0",
   "legalFirmwareAuthTag": "E2C03AFCB73E0464827200E5",
   "sensorFirmwareVersion": "0.0.8.0",
   "sensorFirmwareCrc": "540F"
},
```

5.2.3. time

Display the local time and time deviation in ISO8601 extended dateformat.

The "Z" suffix is equivalent to "+00:00". Therefore, if /settings field /time/tz = "+00:00" and DST is not present, the time in this field will be displayed with ending "Z" letter (i.e. as a UTC timestamp).

Example

```
"time": "2019-10-24T15:45:33+02:00",
```

5.2.4. ipAddress

Display the current IP address of the DCBM. Default IP address of the DCBM is 192.168.1.2.

Example

```
"ipAddress": "192.168.1.2",
```

Current IP address can be displayed through the maintenance screens. In case of DCHP activated and no IP address received from the network, the IP displayed is: 0.0.0.0.

5.2.5. meterId

Display the unique identifier of the DCBM.

Example

```
"meterId": "1202407280",
```


LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

5.2.6. errors

Display the error status of the DCBM.

Example


```
"errors": {
   "value": 0,
   "bits": {
     "muInitIsFailed": false,
     "suStateIsInvalid": false,
     "versionCheckIsFailed": false,
     "muRngInitIsFailed": false,
     "muDataIntegrityIsFailed": false,
     "muFwIntegrityIsFailed": false,
     "suIntegrityIsFailed": false,
     "logbookIntegrityIsFailed": false,
     "logbookIsFull": false,
     "memoryAccessIsFailed": false,
     "muStateIsFailed": false,
   }
 },
```


Any error prevents a correct usage of the DCBM: “Fatal Error” level. A new transaction will be refused, DCBM shall be changed if the error persist.

- `value` field: In normal operation, shall be 0. In case it is non-null, the value is displayed on the screen. When bit-parsing the value (0 is LSB) it is possible to know the errors set. More explanations can be found in the Operation manual, section “Status/errors value field description”.
- `bits` fields, corresponding to value field broken down by unitary flags (please note: indexes are not contiguous):

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

Error bit name	Bit index	Description	Frequency of checks	Associated event logbook entry name
muInitIsFailed	0	error when initializing the internal data structure	At startup	EV_INIT_ERROR
suStateInvalid	1	error in the state of the SU	Continuous	EV_SU_INVALID_STATE
versionCheckIsFailed	2	error in SW version (SU version vs MU version pair)	At startup	EV_VERSION_INCONSISTENCY
muRngInitIsFailed	3	error at hardware random number generator initialization	At startup	EV_RNG_INIT_TEST_FAIL
muDataIntegrityIsFailed	4	error with CRC check of the parameter structure of the Meter Unit or in stored transaction	At startup (5s offset) + cyclic (30mn)	EV_LR_FIRMWARE_CHECK_FAILED
muFwIntegrityIsFailed	5	error with CRC check of the Meter Unit firmware	At startup (5s offset) + cyclic (30mn)	EV_LR_FIRMWARE_CHECK_FAILED
suIntegrityIsFailed	6	error with CRC check of the Sensor Unit parameter	Continuous	EV_SU_INTEGRITY_ERROR
suMeasureFailureOccurred	7	failure of an ADC measurement in the SU	Informational	STATUS_MEASURE_ERROR
logbookIntegrityIsFailed	8	error in integrity of the event logbook	At startup (5s offset) + cyclic (30mn)	EV_EXTERNAL_MEMORY_INTEGRITY
logbookIsFull	9	error on event logbook storage (max capability reached)	At startup (5s offset) + on new event	EV_LOGBOOK_FULL
memoryAccessIsFailed	10	error on software interface memory access	Continuous	EV_INVALID_MEMORY_ACCESS
muStateIsFailed	13	legally relevant firmware memory error (stack overflow)	At startup (5s offset) + cyclic (30mn)	EV_STACK_OVERFLOW

! Any raised fatal error prevents further use of the DCBM. Product must be checked.

5.2.7. publicKey

Display the public key of the DCBM in the ASN.1 DER octet string format.

Example

```
"publicKey": "797B79B8E0ACBDA9646ED19B03B85
 C39CCE56F5A179988E874BA75FB8303199C255A4929
 36EE27D58AAAFC0DE53B29931D3022ADD96CB6AD95C
 C59B757C6A154",
```


LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

5.2.8. publicKeyOcmf

Display the public key of the DCBM in the ASN.1 DER octet string format with RFC5480 header.

Example

```
"publicKeyOcmf": "3059301306072A8648CE3D020106
082A8648CE3D03010703420004797B79B8E0ACBDA9646
ED19B03B85C39CCE56F5A179988E874BA75FB8303199C
255A492936EE27D58AAAFC0DE53B29931D3022ADD96CB
6AD95CC59B757C6A154",
```

This is the format expected by the OCMF transparency software (see "7.1. Overview").

5.2.9. indexOfLastTransaction

Indicates the latest transaction storage index in memory (max size 2^32-1). Corresponding transaction is stored at `/v1/legal/<index>`.

Example

```
"indexOfLastTransaction": 14,
```

A value of -1 means that no transaction is stored at all.

5.2.10. numberOfStoredTransactions

Indicates the number of transactions currently stored in the DCBM.

When the transaction buffer is full, wrap around occurs and the oldest transaction is deleted when writing of a new one (see Operation manual section "Memory depth").

Example

```
"numberOfStoredTransactions": 15
```


LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

5.3. Allowed requests

5.3.1. GET - Read /status

Request format

```
GET /v1/status HTTP/1.1
```

Response format

See section “5.1. Overview”.

Status codes

Code Number	Meaning
200	OK

5.4. Examples

- Linux/Windows bash

```
curl -X GET http://192.168.1.2/v1/status
```

- Windows (PowerShell)

```
Invoke-WebRequest -uri “http://192.168.1.2/v1/status” | Select-Object
-Expand Content
```


LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

6. /LEGAL API

6.1. Overview

/legal API allows managing new transactions and retrieving previous ones. Transaction concept is described in Operation manual, section “Transactions”.

This API contains all legal information needed for the billing process and remote display of energy measurement information. It marks the end of the metrological chain of the DCBM. This format is proprietary (LEM) i.e. manufacturer-specific.

A signature allows validating integrity and authentication, refer to Operation manual section “Data authenticity”.

It is possible to read the `/legal` fields during a transaction (intermediate reading), or after a transaction (final reading). A pagination counter increments after each read on /legal.

Below is the JSON type structure for /legal interface.

```
{
  “paginationCounter”: integer,
  “transactionId”: string,
  “evseId”: string,
  “clientId”: string,
  “tariffId”: integer,
  “cableSp”: {
    cableSpName: string
    cableSpId: integer,
    cableSpRes: integer,
  },
  “userData”: string,
  “meterValue”: {
    “timestampStart”: string,
    “timestampStop”: string,
    “transactionDuration”: integer,
    “intermediateRead”: boolean,
    “transactionStatus” : boolean,
    “sampleValue”: {
      “energyUnit”: string,
      “energyImport”: number,
      “energyImportTotalStart”: number,
      “energyImportTotalStop”: number,
      “energyExport”: number,
      “energyExportTotalStart”: number,
      “energyExportTotalStop”: number
    }
  },
  “meterId”: string,
  “signature”: string,
  “publicKey”: string,
}
```


LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

6.2. Fields description

6.2.1. Response description

Below is an example of a transaction result:

- this is the 6th read of the DCBM
- `transactionId` is “azAZ09*-_=:+|,@”
- `evseId` is “+49*DEF*E123ABC”,
- `clientId` is “C12”
- `tariffId` is 2
- cable compensation parameters : `cableId` = 1, name = “`2mR_Comp`”, resistance = 2 mOhm
- transaction occurred the 10th of December of 2020
- transaction has started at 16:39:15 local time
- transaction has ended at 16:40:25 local time
- the transaction duration was 70 seconds
- tampering was detected during the transaction, status is not nominal (i.e. different from 17, see `/status/status API` for more details)
- the energy transfer from the charging station to the EV was 7.637 kWh
- the energy transfer from the EV to the charging station was null

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

```
{
  “paginationCounter”: 6,
  “transactionId”: “azAZ09*-_=:+~|,@”,
  “evseId”: “+49*DEF*E123ABC”,
  “clientId”: “C12”,
  “tariffId”: 2,
  “cableSp”: {
    “cableSpName”: “2mR_Comp”,
    “cableSpId”: 1,
    “cableSpRes”: 2
  },
  “userData”: “”,
  “meterValue”: {
    “timestampStart”: “2020-12-10T16:39:15+01:00”,
    “timestampStop”: “2020-12-10T16:40:25+01:00”,
    “transactionDuration”: 70,
    “intermediateRead”: false,
    “transactionStatus”: 25,
    “sampleValue”: {
      “energyUnit”: “kWh”,
      “energyImport”: 7.637,
      “energyImportTotalStart”: 188.977,
      “energyImportTotalStop”: 196.614,
      “energyExport”: 0.000,
      “energyExportTotalStart”: 0.000,
      “energyExportTotalStop”: 0.000
    }
  },
  “meterId”: “12024072805”,
  “signature”: “304502203DC38FBC722D216568D6ECB4B3
  52577A999B6D184EA6AD48BDCAE7766DB1D628022100A768
  7B4CB5573829D407DD4B17D41C297917B7E8307E5017711B
  5A3A987F6801”,
  “publicKey”: “A80F10D968E1122F8820F288B23C4E1C0D
  A912F35B48481274ADFEFE66D7E87E130C7CF2B8047C45CF
  105041C8C3A57DD242782F755C9443F42DABA9404A67BF”
}
```


LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

6.2.2. paginationCounter

Display the number of times the /legal fields was read.

As this counter is incremented for each read of /legal, the signature is generated and is different even for a past transaction read several times.

Example

```
"paginationCounter":26
```

6.2.3. transactionId - input parameter

This is an input parameter (string) to identify the transaction. Max size = 37 char.

Example

```
"transactionId": "azAZ09*-_=:+~|,@",
```

Authorized characters are: ASCII encoding.

6.2.4. evseId - input parameter

This is an input parameter (string) to identify the charging point. Max size = 37 char.

Example

```
"evseId": "+49*DEF*E123ABC",
```

Authorized characters are: ASCII encoding.

6.2.5. clientId - input parameter

This is an input parameter (string) to identify the end user customer (client). Max size = 37 char.

Example

```
"clientId": "client12657",
```

Authorized characters are: ASCII encoding

The `clientId` is also displayed during a transaction on the screen, so the value used must not be confidential

6.2.6. tariffId - input parameter

This is an input parameter, an integer (from 0 to 255) used for a unique transaction tariff designation.

The DCBM is INFO time (not SYSTEM time), so no tariff changes are possible during a transaction.

Example

```
"tariffId": 2,
```


LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

6.2.7. cableId - input parameter

This field refers to the `/settings/cableConf` table.

This is an input parameter: an integer (from 0 to 7).

The value shall correspond to one of the `cableConf/../cableSpId` array value.

This allows compensating the measurements of the DCBM with a resistance value, selectable within a table.

Example

```
"cableId": 6,
```

6.2.8. cableSp

This field reflects the selected `/settings/cableConf` tuple, selected with `cableId` value (see previous paragraph).

6.2.8.1. cableSpName

A pre-set string (max 19 chars).

This field reflects the selected `/settings/cableConf/cableSpName` table, selected with `cableId` value (see previous paragraph).

6.2.8.2. cableSpId

A pre-set integer from 0 to 7.

This field reflects the selected `/settings/cableConf/cableSpId` table, selected with `cableId` value (see previous paragraph).

6.2.8.3. cableSpRes

A pre-set integer (1 byte) encoding the resistance value in mOhm.

This field reflects the selected `/settings/cableConf/cableSpRes` table, selected with `cableId` value (see previous paragraph).

6.2.9. userData - input parameter

An input parameter (string) that can be used to include specific information within the legal data. Max size = 128 bytes.

i All the UTF-8 and UTF-16 characters set can be used but must fit into 128 byte to be accepted.

Example

```
"userData" : "Contract ID: 563-4*",
```

6.2.10. meterValue

6.2.10.1. timestampStart

Timestamp at the time of the start command, expressed in ISO 8610 date time local format, with timezone information.

i If `/settings/time/tz = "+00:00"` this corresponds to the UTC time zone, then time in this field will be displayed as an UTC timestamp (with terminal "Z" letter), without the +00:00 field.

Example

```
"timestampStart": "2019-10-28T10:41:55+01:00",
```

6.2.10.2. timestampStop

Timestamp at the time of the stop (or the time of the read in case of intermediate reading) command, expressed in ISO 8610 date time local format, with timezone information

If `/settings/time/tz = "+00:00"` this corresponds to the UTC time zone, then time in this field will be displayed as an UTC timestamp (with terminal "Z" letter), without the +00:00 field.

Example

```
"timestampStop": "2019-10-28T11:39:57+01:00",
```

6.2.10.3. transactionDuration

Informational register, expressing the difference between stop timestamp and start timestamp (in seconds).

Example

```
"transactionDuration": 3482,
```

In case of a transaction on-going (see next field) this register increases along with the stop timestamp (`timestampStop` is the timestamp of reading in case of an on-going transaction)

6.2.10.4. intermediateRead

Boolean expressing whether the reading is from a past transaction (=`false`) or for a current transaction (i.e. an intermediate reading) (=`true`).

Example

```
"intermediateRead": false,
```

6.2.10.5. transactionStatus

This field indicates the status of the DCBM at time of the reading (intermediate reading case) or time of end of transaction (past transaction case).

Unlike the live status register from the /status API, this one accumulates the status bit changes during the transaction, ensuring traceability of any occurring event. Some of the flags can invalidate the transaction. Refer to 1.3.2.1 status chapter for detail of the bits meanings.

The nominal value, i.e. corresponding to a DCBM ready for a new transaction, is 17.

This corresponds to:

- `"suLinkStatusIsOk"`: true,
- `"timeSyncStatusIsOk"`: true,
- and the rest to false.

Example

```
"transactionStatus": 17,
```


LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

6.2.10.6. sampleValue

JSON fields containing the value of the DC energy measurement.

Field `energyUnit`

Field indicating the unit of the energy register.

This field is static, measurement unit cannot change.

- **Example**

```
"energyUnit": "kWh",
```

Field `energyImport`

Field indicating the difference of imported energy between the stop and the start command, in kWh with 3 decimal digits.

- **Example**

```
"energyImport": 511.994,
```

Field `energyImportTotalStart`

Field indicating the imported energy total register at the time of the start command, in kWh with 3 decimal digits.

i Corresponding OBIS code is “1-0:1.8.0”

- **Example**

```
"energyImportTotalStart": 18.775,
```

Field `energyImportTotalStop`

Field indicating the imported energy total register at the time of the `stop/read` command, in kWh with 3 decimal digits.

i Corresponding OBIS code is “1-0:1.8.0”

- **Example**

```
"energyImportTotalStop": 530.769,
```

i In case of a transaction on-going this register increases along with `timestampStop` (intermediate reading case).

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

Field `energyExport`

Field indicating the difference of exported energy between the stop and the start commands, in kWh with 3 decimal digits.

- **Example**

```
"energyExport": 0.000,
```

Field `energyExportTotalStart`

Field indicating the exported energy total register at the time of the start command, in kWh with 3 decimal digits.

Corresponding OBIS code is “1-0:2.8.0”

- **Example**

```
"energyExportTotalStart": 0.000,
```

Field `energyExportTotalStop`

Field indicating the imported energy total register at the time of the stop/read command, in kWh with 3 decimal digits.

Corresponding OBIS code is “1-0:2.8.0”

- **Example**

```
"energyExportTotalStop": 0.000
```

6.2.11. meterId

Display the `meterId` value of the DCBM (corresponding to its serial number). Max size is 37 characters.

Example

```
"meterId": "1202407280",
```


LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

6.2.12. signature

Signature of the transaction in octet string format, with ASN.1 DER encoding, using ECDSA secp256r1 and SHA256 methods.

The signature is calculated over a rearranged data structure, copying all JSON fields (unlike the OCMF format signature which is calculated over the raw HTTP response body).

Example

```
"signature":
"304502205C7B5B67C012E2691738B4CE5365AEE1
191D0F59AAB81D6C0C0C1BC74303FDB9022100A79
E1BBA77EA6B110E19C81D84D44750C0361A04E566
2783D13D5F1BFDEF66D7"
```

6.2.13. publicKey

Display the public key of the DCBM in the ASN.1 DER octetstring format.

Example

```
"publicKey":
"797B79B8E0ACBDA9646ED19B03B85C39CCE56F5A
179988E874BA75FB8303199C255A492936EE27D58
AAAFC0DE53B29931D3022ADD96CB6AD95CC59B757
C6A154",
```


LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

6.3. Allowed requests

6.3.1. POST - Start a transaction on /legal

Request format

To start a transaction, all fields are needed, and order shall be observed.

- `evseId`
- `transactionId`
- `clientId`
- `tariffId`
- `cableId`
- `userData`

The DCBM accepts multiple transactions with the same `transctionId`. In this case, on retrieval (GET) by `transactionId`, the latest is fetched.

For the transaction start to be accepted, the DCBM time synchronization must be valid. Otherwise, the transaction start will be rejected.

- Template

```
POST /v1/legal HTTP/1.1
Content-Type: application/json
Content-Length: strlen(<BODY>)

<BODY>
```

- Example

```
POST /v1/legal HTTP/1.1
Content-Type: application/json
Content-Length: 91

{"evseId":"evse458877","transactionId":
"transac5000","clientId":"client12",
"tariffId":2,"cableId":2,"userData":""}
```

Response header format

The response header for /legal POST is distinctive, it contains the transaction storage index in the HTTP header location field:

- Success case

```
HTTP/1.1 201 Created
Location: http://192.168.1.2:80/v1/legal/3
Connection: close
Content-Type: application/json
Transfer-Encoding: chunked
```


LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

- Failing case

```
HTTP/1.1 403 Forbidden
Connection: close
```

Response body format

The response body contains all the inputs fields set (except `userData`) and the `running` field.

The `userData` field is not present into the response.

- Example

```
{"evse458877","transactionId":"transac5000",
"clientId":"client12","tariffId":2,"cableId":2,
"running":true}
```

Status codes

Code number	Meaning	Description
201	Created	The start request was accepted, a transaction storage was allocated
308	Permanent redirect	Port was redirected
400	Bad request	An error was detected during fields parsing
403	Forbidden	The transaction start was rejected (a transaction is already running)
405	Method not allowed	Wrong HTTP method used
412	Precondition failed	The DCBM is not ready for a new transaction (ie. a status flag is raised or an error was detected)
501	Not implemented	HTTP invalid format request

6.3.2. PUT - Stop a transaction on /legal

Request format

To stop the on-going transaction, the `transactionId` is required.

The stop command by `transactionId` goes through an URI. Special characters can be sent with or without query-string percent-encoding.

- Template

```
PUT /v1/legal?transactionId=<transactionId> HTTP/1.1
Content-Type: application/json
Content-Length: 18

{"running": false}
```


LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

- Example

```
PUT /v1/legal?transactionId=5000 HTTP/1.1
Content-Type: application/json
Content-Length: 18

{"running": false}
```

Response format

HTTP chunked response content is the same as a read content structure.

- Example


```
{"paginationCounter":14,"transactionId":
"azAZ09*-_=:+~|,@","evseId":"+49*DEF*E123
ABC","clientId":"C12","tariffId":2,"cableSp":
{"cableSpName":"2mR-Comp","cableSpId":1,
"cableSpRes":2},"userData":"","meterValue":
{"timestampStart":"2020-12-10T17:22:54+01:00",
"timestampStop":"2020-12-10T17:27:56+01:00",
"transactionDuration":302,"intermediateRead"
:false,"transactionStatus":17,"sampleValue"
:{"energyUnit":"kWh","energyImport":33.499,
"energyImportTotalStart":96.659,"energyImportTotalStop"
:130.158,"energyExport":0.000,"energyExportTotalStart"
:0.000,"energyExportTotalStop":0.000}},
"meterId":"12024072805","signature":
"304502200C22B3EAB7A27FE60C5DF58B404563843A3
A4C3DB636FCCA42B7D7B8DCDD37FE022100C31D72C47
D7CF565F16EA8ED5820B1F0739781756B55FA3F1B28F
BA4A51E8AB1","publicKey":"D47C8ACBA2E18E93BD57C3
61C2CA7E7BA19157DF7913E20DCECD387DEE5138F
2CE3BCD98CFA51C17D006F6878958C23818EDA88B
3568E0B2F3A6CEC1D04EE44C"}
```


LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

Status codes

Code number	Meaning	Description
200	OK	The start request was accepted, a transaction storage was allocated
308	Permanent redirect	Port was redirected
400	Bad request	An error was detected during fields parsing
405	Method not allowed	Wrong HTTP method used
412	Precondition failed	The DCBM is not ready for a new transaction (i.e. a status flag is raised or an error was detected)
501	Not implemented	HTTP invalid format request

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

6.3.3. GET - Read /legal

Request format

The DCBM can store up to 20399 transactions

- By current or latest

```
GET /v1/legal HTTP/1.1
```

- By `transactionId`

```
GET /v1/legal?transactionId=<transactionId> HTTP/1.1
```

! This solution only gives access to the latest 839 transactions.

! The meter accepts multiple transactions with the same `transactionId`. In this case, on retrieval (GET) by `transactionId`, the latest is fetched.

- By internal transaction index (“absolute” storage index)

```
GET /v1/legal/<index_value> HTTP/1.1
```

i The <index_value> is the one returned when starting a transaction into the HTTP header field (see “6.3.1. POST - Start a transaction on /legal”)

- By internal chronological transaction index value

```
GET /v1/legal/-<index_value> HTTP/1.1
```

i This solution allows requesting the whole list of transactions, successively.

Note:

- “/-1” leads to the penultimate stored transaction.
- “/-20398” leads to the oldest possible transaction (is buffer has been filled).
- “/-20399” can never be reached.

Summary:

Type	Example	Limitation
By `latest`	`curl -X GET http://192.168.1.2/v1/legal`	
By `transactionId`	`curl -X GET http://192.168.1.2/v1/legal?transactionId=”5000”`	On last 839 (~1 week). If same `Id` => latest one
By `absolute index`	`curl -X GET http://192.168.1.2/v1/legal/1234`	Not contiguous. Not chronological. On last 20399 max.
By `chronological index`	`curl -X GET http://192.168.1.2/v1/legal/-1234`	On last 20399 max.

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

Response format

See section “6.1. Overview”.

Status codes

Code Number	Meaning
200	OK

6.4. Examples

6.4.1. Read

Get current or last transaction

- Linux/Windows bash

```
curl -X GET http://192.168.1.2/v1/legal
```

- Windows (PowerShell)

```
Invoke-WebRequest -uri “http://192.168.1.2/v1/legal” | Select-Object
-Expand Content
```

Get a past transaction with the `transactionId`

- Linux/Windows bash

```
curl -X GET http://192.168.1.2/v1/legal?transactionId=transac5000
```

- Windows (PowerShell)

```
Invoke-WebRequest -uri “http://192.168.1.2/v1/
legal?transactionId=transac5000” | Select-Object -Expand Content
```

Get a past transaction with the chronological index

- Linux/Windows bash

```
curl -X GET http://192.168.1.2/v1/legal/-1
```

- Windows (PowerShell)

```
Invoke-WebRequest -uri “http://192.168.1.2/v1/legal/-1” | Select-Object
-Expand Content
```


LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

6.4.2. Start

In the following examples we set:

`evseId`	evse458877
`transactionId`	transac5000
`clientId`	client12657
`tariffId`	2
`cableId`	1
`userData`	“”

- Linux/Windows bash

```
curl -d
'{"evseId":"evse458877","transactionId":
"transac5000","clientId":"client12657",
"tariffId":2,"cableId":1,"userData":""}'
-H 'Content-Type: application/json' -X POST http://192.168.1.2/v1/legal
```

- Windows (PowerShell)

```
Invoke-WebRequest -uri "http://192.168.1.2/v1/legal" -ContentType
"application/json" -Method POST -Body '
{ "evseId":"evse458877","transactionId":
"transac5000","clientId":"client12657",
"tariffId":2,"cableId":1,"userData":"" }
' | Select-Object -Expand Content
```

6.4.3. Stop

The `transactionId` is required to stop the on-going transaction. It can be retrieved with a reading.

In this example we stop the transaction with `transactionId`=”transac5000”

- Linux/Windows bash

```
curl -d '{"running":false}' -H 'Content-Type: application/json' -X PUT
http://192.168.1.2/v1/legal?transactionId=transac5000
```

- Windows (PowerShell)

```
Invoke-WebRequest -uri "http://192.168.1.2/v1/
legal?transactionId=transac5000" -ContentType "application/json" -Method
PUT -Body '{"running": false}'  | Select-Object -Expand Content
```

7. /OCMF API

In parallel to the /legal format used to start, read and stop transactions, the /ocmf API is also available to:

- read the current transaction
- read the last transaction
- read a past transaction

Following section details the used OCMF format, also see Operation Manual section “Transaction readouts”.

OCMF readouts can be verified using official OCMF transparency software. Refer to Operation manual, section “Data authenticity”.

7.1. Overview

The OCMF structure falls into 3 parts:

```
OCMF|JSON1|JSON2
```

where

- OCMF is a fixed header
- JSON1 is the response in JSON format
- JSON2 is the signature in JSON format

See the OCMF specification for detailed description:

http://hers.abl.de/SAFE/Datenformat_OCMF/Datenformat_OCMF_v1.0.pdf

Below is an introduction of reading as an example:

```
OCMF|
{
  “FV”: string,       // ocmf v1.0
  “GI”: string,       // fixed, identify DCBM version
  “GS”: integer,      // Serial number of the DCBM
  “GV”: string,       // fixed, identifies DCBM version
  “PG”: string,       // pagination counter
  “MV”: string,       // fixed, identifies LEM manufacturer
  “MS”: string,       // Serial number of the DCBM
  “MF”: string,       // firmware version
  “IS”: boolean,      // cf /settings ocmf field
  “IL”: string,       // cf /settings ocmf field
  “IF”: [             // cf /settings ocmf field
        string,
        ...
       ],
  “IT”: string,        // cf /settings ocmf field
```


LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

```
  "ID": string,        // transactionId
  "CT": string,        // fixed, label for CI field
  "CI": string,        // evseId
  "RD":                // readings
  [
    {
      "TM": string,    // timestamp of start + time status
      "TX": string,    // B = begin
      "RV": number,    // reading value
      "RI": string,    // obis code imported energy
      "RU": string,    // unit of reading
      "RT": string,    // DC charger
      "EF": string,    // error flag
      "ST": string,    // status flag, G = good
      "UC": {          /* LEM Specific JSON field info on selected cable
                          (compensation applied) */
        "UN": string,     // Name of the cable
        "UI": integer,    /* Id of the cable  (used for selection on
                             start command) */
        "UR": integer     // Resistance value of the cable
      }
    },
    {
      "RV": number,    // reading value
      "RI": string,    // obis code exported energy
      "RU": string,    // unit of reading
      "ST": string     // status flag, G = good
    },
    {
      "TM": string,    /* timestamp of stop/read intermediate + time
                          status */
      "TX": string,    // C = charging, E = end
      "RV": number,    // reading value
      "RI": string,    // obis code imported energy
      "RU": string,    // unit of reading
      "EF": string,    /* error flag display when different from first
                          reading */
      "ST": string     // status flag, G = good
    },
    {
      "RV": number,    // reading value
      "RI": string,    // obis code exported energy
      "RU": string,    // unit of reading
      "ST": string     // status flag, G = good
    }
  ]
}|
{
```


LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

```
    "SA": string,                  // signature type, cf the OCMF spec
    "SD": string                   /* signature value, on JSON1 field
                                      (string) without spaces */

}
```

7.2. Fields description: JSON1

7.2.1. FV field

Type String - Format-Version: = “1.0”

7.2.2. GI field

Type String - Gateway identification= “LEM DCBM”

7.2.3. GS field

Type String - Gateway material = DCBM serial number (string of 37 char max)

7.2.4. GV field

Type String - Gateway version = “v1”, the HTTP REST API version

7.2.5. PG field

Type String - Pagination of the entire dataset = string of “T<value>” with value increased for each read of transaction

7.2.6. MV field

Type String - Meter-Vendor = “LEM”

7.2.7. MS field

Type String - Meter-Serial = DCBM serial number (string of 37 char max)

7.2.8. MF field

Type String - Meter-Firmware: Legal (Metrological) Firmware parts of the DCBM = “MU-0.1.4.0_SU-0.0.8.0” for DCBM400

7.2.9. IS field

Value of this field depends on the `/settings/ocmfId/IL` field value: it has to be set by the Charger Controller.

Type Boolean - Identification status: General status for user assignment:

- `true`: Users successfully assigned,
- `false`: Users not associated

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

Set to true if IL field is set to following values in /settings/ocmfId/IL field, false otherwise :

2	"HEARSAY"	The assignment is unsecured; e.g. by reading out an RFID UID
3	"TRUSTED"	The assignment can be trusted to some extent, but there is no absolute reliability. Example: Authorization by Backend
4	"VERIFIED"	The assignment has been verified by the signature component and specific actions
5	"CERTIFIED"	The mapping was verified by the signature component using a cryptographic signature that certifies the mapping
6	"SECURE"	The assignment was established by a safe feature (e.g. secure RFID card, ISO15118 with plug and charge, etc.)

7.2.10. IL field

Value of this field depends on the /settings/ocmfId/IL field value: it must be set by the charging controller.

Type String - Identification level: JSON Array

See /settings/ocmfId/IL section for corresponding values

7.2.11. IF field

Value of this field depends on the /settings/ocmfId/IF field value: it must be set by the charging controller.

Type Array of String - Identification flags for RFID, OCPP, ISO15118 and PLMN protocol

Set according to settings/ocmfId/IF section

7.2.12. IT field

Value of this field depends on the /settings/ocmfId/IT field value: it must be set by the charging controller.

Type String - Identification-Type: "string"

Set as per settings/ocmfId/IT section

For correct usage the IT fields shall be set once to the corresponding protocol used (or kept to "NONE" as per default value), and not be changed during the lifetime of the DCBM due to current software limitation.

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

7.2.13. ID field

Value of this field depends on the `/settings/ocmfId/IT` field value: it must be set by the charging controller.

Type String - Identification-Data: “string”, set according to

- /legal `START` command
- and `/settings/ocmfId/IT` fields

Table below indicates the redirection of `ID` fields to `/legal/transactionId` or `/legal/clientId`.

0	“NONE”	/legal/transactionId
1	“DENIED”	/legal/transactionId
2	“UNDEFINED”	/legal/transactionId
3	“ISO14443”	/legal/clientId
4	“ISO15693”	/legal/clientId
5	“EMAID”	/legal/clientId
6	“EVCCID”	/legal/clientId
7	“EVCOID”	/legal/clientId
8	“ISO7812”	/legal/clientId
9	“CARD_TXN_NR”	/legal/clientId
10	“CENTRAL”	/legal/transactionId
11	“CENTRAL_1”	/legal/transactionId
12	“CENTRAL_2”	/legal/transactionId
13	“LOCAL”	/legal/transactionId
14	“LOCAL_1”	/legal/transactionId
15	“LOCAL_2”	/legal/transactionId
16	“PHONE_NUMBER”	/legal/clientId
17	“KEY_CODE”	/legal/clientId

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

7.2.14. CT field

Type String - Charge-Point-Identification-Type: “EVSEID”

7.2.15. CI field

Type String - Charge-Point-Identification: string = `/legal/evseId` value

7.2.16. RD fields (Readings)

TM field

Type String - Time: Indication of the system time of reading and synchronization state. = “`<localtime>,000<deviation> <time_sync_status_letter>`”

with `<locatime>` = local time (in datetime format) of the DCBM at the time of the reading, ISO8601 extended format.

with `<deviation>` = signed deviation from local time to UTC, ISO8601 extended format

with `<time_sync_status_letter>` = see table below

Letter	Description
U	Unknown, unsynchronized
I	Informative (Info watch)
S	Synchronized
R	Relative time billing with a quartz timer based on an info watch

The DCBM uses “R” (INFO time) + crystal oscillator timer level

TX field

Type String - Transaction: Reason for reading, reference to the transaction, noted in capital letter:

- 1st reading is start of transaction: “B”
- 3rd reading is stop of transaction: “E” or “C” for intermediate reading

RV field

Type Number - Reading Value: the reading value with 3 fractional digits

- 1st reading tuple: imported energy register at start of transaction
- 2nd reading tuple: exported energy register on stop of transaction
- 3rd reading tuple: imported energy register at end of transaction / during transaction depending on the context (see TX field)
- 4th reading tuple: exported energy register on end of transaction / during transaction depending on the context (see TX field)

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

RI field

Type String - Reading Identification: OBIS code

- 1st reading tuple : “1-0:1.8.0” (Total Imported Energy)
- 2nd reading tuple : “1-0:2.8.0” (Total Exported Energy)
- 3rd reading tuple : “1-0:1.8.0” (Total Imported Energy)
- 4th reading tuple : “1-0:2.8.0” (Total Exported Energy)

RU field

Type String - Reading unit: = “kWh”

RT field

Type String - Reading Current type: = “DC”

EF field

Type String - Error flags

Letter	Description
“”	No error
“E”	Error in the energy register
“t”	Error in the time status
“Et”	Error in the energy registers and the time status

ST field

Type String - Status = the letter is set according to status bit (see following table)

Abbreviation	Identifier	Description
N	NOT_PRESENT	The counter has not been found existent
G	OK	Counter in order (Good)
T	TIMEOUT	Time-crossing while trying to control the counter
M	MANIPULATED	Manipulation detected
E	OTHER_ERROR	Other, unknown errors
F	READ_ERROR	DCBM registers do not read correctly; Value of the reading is not valid

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

UC field

This field reflects the `/settings/cableConf` selected table for the transaction by the `/legal/cableId` input parameter.

This is a LEM specific field, using specific IDs:

UN	string	cable name (max size 9 char)
UI	integer	cable ID (value from 0 to 255)
UR	integer	resistance value of the cable (value from 0 to 255)

7.3. Fields description: JSON2

7.3.1. SA field

Type String - “ECDSA-secp256r1-SHA256”

This field is a fixed value indicating the signature algorithm used and set according to OCMF specification

7.3.2. SD field

Type String - signature performed on JSON1 field , octet string DER format,

Example

```
{
  “SA”: “ECDSA-secp256r1-SHA256”,
  “SD”: “3045022100B3EB273433278F102D4E18EC
  871B575533D4AFC62AC28229FA61428AB74DBA960
  2204A98B7517866F82370EEDF170A8CEF17221759
  146A54FB7A830E7D111C3A30F9”
}
```


LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

7.4. Allowed requests

7.4.1. GET - Read /ocmf

Request format

- By latest

```
GET /v1/ocmf HTTP/1.1
```

- By absolute transaction index

```
GET /v1/ocmf/<index_value> HTTP/1.1
```

- By chronological transaction index

```
GET /v1/ocmf/-<index_value> HTTP/1.1
```

- By transactionId (limited to latest 839 transactions)

! When requesting a past transaction by transactionId, if multiple transactions have the same transactionId, the latest is fetched.

```
GET /v1/ocmf?transactionId=<past_transactionId_value> HTTP/1.1
```

Response format

See section “7.1. Overview”.

Status codes

Code Number	Meaning
200	OK

7.5. Examples

Get current or last transaction

- Linux/Windows bash

```
curl -X GET http://192.168.1.2/v1/ocmf
```

- Windows (PowerShell)

```
Invoke-WebRequest -uri “http://192.168.1.2/v1/ocmf” | Select-Object
-Expand Content
```


LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

Get a past transaction by `transactionId` (limited to latest 839 transactions)

When requesting a past transaction by `transactionId`, if multiple transactions have the same `transactionId`, the latest is fetched.

- Linux/Windows bash

```
curl -X GET http://192.168.1.2/v1/ocmf?transactionId=transac5000
```

- Windows (PowerShell)

```
Invoke-WebRequest -uri
"http://192.168.1.2/v1/ocmf?transactionId=transac5000" | Select-Object
-Expand Content
```

Get a past transaction by storage index

- Linux/Windows bash

```
curl -X GET http://192.168.1.2/v1/ocmf/0
```

- Windows (PowerShell)

```
Invoke-WebRequest -uri "http://192.168.1.2/v1/ocmf/0" | Select-Object
-Expand Content
```

Get a past transaction by chronological index

- Linux/Windows bash

```
curl -X GET http://192.168.1.2/v1/ocmf/-12
```

- Windows (PowerShell)

```
Invoke-WebRequest -uri "http://192.168.1.2/v1/ocmf/-12" | Select-Object
-Expand Content
```


LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

8. /LOGBOOK API

Read-only, contains the legally relevant and legally non-relevant events during the lifetime of the DCBM.

A signature allows validating integrity and authentication, refer to Operation manual section “Data authenticity”.

Legally non-relevant events are prefixed by EV_APP.

8.1. Overview

The “`logbook`” field is a JSON array of events. All fields are provided as strings.

The logbook can store approximately 40 000 events. When the logbook is full, the DCBM ceases to operate (FF error, event: “EV_LOGBOOK_FULL”). **The DCBM shall be changed.**

```
{
  “meterId”: string,
  “logbook”: [
    {
      “timestamp”: string,
      “eventCode”: string,
      “status”: [
        string,
        string,
        ...
      ]
    },
    ...
  ],
  “signature”: string
}
```

8.2. Fields description

8.2.1. meterId

Displays the `meterId` value (serial number) of the DCBM. Max size is 37 characters.

Example

```
“meterId”: “1202407280”,
```


LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

8.2.2. logbook

The `logbook` field is a JSON array that contains event tuples. An example is given below.

```
"logbook": [
    {
     "timestamp": "2020-10-28T09:40:07Z",
     "eventCode": "EV_TIME_SYNC_SUCCEEDED",
     "status": [
       "STATUS_SENSOR_LINK",
       "STATUS_TIME_SYNC"
     ]
   },
   ...
 ],
```

8.2.2.1. timestamp

UTC timestamp of the event, in ISO8601 extended datetime format

Unlike /legal or /ocmf APIs, the timestamp of the logbook is expressed in UTC time (“Z” suffix) and does not depends on local time settings

Example

```
"timestamp": "2019-10-28T09:40:07Z",
```


LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

8.2.2.2. eventCode

The `eventCode` of the event, one of following list.

Some of the events are linked to a fatal error, preventing new transactions permanently. Some are linked to blocking statuses, preventing new transactions temporarily.

Name	Description	Incidence and links
`EV_LNR_FIRMWARE_UPDATE_SUCCEEDED`	Application firmware update succeeded	
`EV_LNR_FIRMWARE_UPDATE_FAILED`	Application firmware update failed	
`EV_TIME_SYNC_FAILED`	Time synchronization expired	Is blocking until `EV_TIME_SYNC_SUCCEEDED` Logged on falling status `timeSyncStatusIsOk`
`EV_TIME_SYNC_SUCCEEDED`	Time synchronization succeeded	Logged on raised status `timeSyncStatusIsOk`
`EV_TAMPERING`	Cover was detected open	Logged on status `tamperingIsDetected`
`EV_LR_FIRMWARE_CHECK_FAILED`	Metrology firmware check failed	Is linked to a fatal error Logged on errors `muDataIntegrityIsFailed` and `muFwIntegrityIsFailed`
`EV_LNR_FIRMWARE_CHECK_FAILED`	Application firmware check failed	
`EV_MEMORY_LOW`	Allocated stack overflew	
`EV_MEASUREMENT_CONSISTENCY_FAILING`	Measurement error occurred, data on data link was missed	Is blocking Logged on falling status `suLinkStatusIsOk` (not the only cause of this status)
`EV_POWER_SUPPLY_FAILURE`	Power fail occurred	
`EV_BATTERY_MONITORING_LOW`	*Obsolete*	
`EV_ECSDA_KEY_CHANGED`	ECDSA private/public keyset was changed	
`EV_PAIRING_KEY_CHANGED`	SU/MU pairing key was changed	
`EV_MANUF_KEY_CHANGED`	CRC on production parameters was changed	
`EV_INVALID_MEMORY_ACCESS`	Invalid memory access was detected	Reboots the DCBM Is linked to a fatal error Logged on error `memoryAccessIsFailed`
`EV_LOGBOOK_FULL`	The logbook is full	Is linked to a fatal error Logged on error `logbookIsFull`
`EV_SU_INVALID_STATE`	The Sensor Unit is in an invalid state	Is linked to a fatal error Logged on error `suStateIsInvalid` and falling status `suLinkStatusIsOk`
`EV_SU_INTEGRITY_ERROR`	Integrity error was detected inside the Sensor Unit	Is linked to a fatal error Logged on error `suIntegrityIsFailed`
`EV_SU_MEASURE_ERROR`	Error occurred in the measurement module of the Sensor Unit	Logged on raised status `suMeasureFailureOccurred`
`EV_VERSION_INCONSISTENCY`	The Sensor Unit firmware version is not the one the Meter Unit expects	Is linked to a fatal error Logged on error `versionCheckIsFailed`
`EV_STACK_OVERFLOW`	Allocated stack overflew in metrology firmware	Is linked to a fatal error Logged on error `muStateIsFailed`

LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

Name	Description	Incidence and links
EV_INIT_ERROR	Error occurred during initialization of the DCBM	Is linked to a fatal error Logged on error muInitIsFailed
EV_TEMPERATURE_OVERHEAT	Detected temperature is outside acceptable range	Logged on raised status overTemperatureIsDetected
EV_TEMPERATURE_NORMAL	Detected temperature reached back the acceptable range	Logged on falling status overTemperatureIsDetected
EV_RNG_INIT_TEST_FAIL	Random Number Generator initialization test has failed	Is linked to a fatal error Logged on error muRngInitIsFailed
EV_VOLTAGE_REVERSED	Voltage connection of the Sensor Unit is reversed	Logged on status reversedVoltage
EV_COMMISSIONING	Commissioning event of the DCBM	
EV_EXTERNAL_MEMORY_INTEGRITY	The logbook internal CRC check is wrong	Is linked to a fatal error Logged on error logbookIntegrityIsFailed
EV_LNR_FIRMWARE_UPDATE_ATTEMPT	Application firmware update was initiated	
EV_RTC_FALLBACK	RTC fallback at startup	
EV_APP_NO_EVENT	*Event codes separator*	
EV_APP_DHCP_CONFIGURATION_CHANGED	DHCP configuration changed and DHCP was activated	
EV_APP_IP_ADDRESS_CHANGED	Dynamic or static IP address changed	
EV_APP_TIME_SERVER_CONFIGURATION_CHANGED	SNTP configuration changed	

Example

```
"eventCode": "EV_TIME_SYNC_SUCCEEDED",
```

8.2.2.3. status

`status` field is a copy of the status bit at the time of the event. The status names are displayed when the corresponding status bits are set. Refer to /status API description.

Example

```
"status": [
 "STATUS_SENSOR_LINK",
 "STATUS_TIME_SYNC"
]
```

8.2.3. signature

Signature of the logbook in octet string format, with ASN.1 DER encoding, using ECDSA secp256r1 and SHA256 methods.

Example

```
"signature":
 "304502205C7B5B67C012E2691738B4CE5365AEE
 1191D0F59AAB81D6C0C0C1BC74303FDB9022100A
 79E1BBA77EA6B110E19C81D84D44750C0361A04E
 5662783D13D5F1BFDEF66D7"
```


LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

8.3. Allowed requests

8.3.1. GET - Read /logbook

Request format

```
GET /v1/logbook HTTP/1.1
```

Response format

See section “8.1. Overview”.

Status codes

Code Number	Meaning
200	OK

8.4. Examples

- Linux/Windows bash

```
curl -X GET http://192.168.1.2/v1/logbook
```

- Windows (PowerShell)

```
Invoke-WebRequest -uri “http://192.168.1.2/v1/logbook” | Select-Object
-Expand Content
```


LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

9. /LIVEMEASURE API

9.1. Overview

All `/livemeasure` fields are read-only (`GET` method).

```
{
  "voltage": number,
  "current": number,
  "power": number,
  "temperatureH": number,
  "temperatureL": number,
  "energyImportTotal": number,
  "energyExportTotal": number,
  "timestamp" : string
}
```

9.2. Fields description

Fields are **read-only**

All `/livemeasure` measurements are updated every second, and correspond to an average over the last 100 ms only.

9.2.1. voltage

Voltage measurement, in Volt unit, with 3 decimal digits. Can be negative (with status `reversedVoltage` under -50 V).

Example

```
"voltage": 995.079,
```

9.2.2. current

Current measurement, in Ampere unit, with 3 decimal digits. Can be negative.

Example

```
"current": 401.152,
```

9.2.3. power

Power measurement, in kiloWatt unit, with 3 decimal digits. Can be negative.

It corresponds to abs(`voltage`) * `current` / 1000.

Example

```
"power": 399.177,
```


LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

9.2.4. temperatureH

Temperature measurement on side “I1” of the Sensor Unit (“T°2” on the display), in celcius degrees unit, with 1 decimal digit.

Example

```
“temperatureH”: 50.7,
```

9.2.5. temperatureL

Temperature measurement on side “I2” of the Sensor Unit (“T°1” on the display), in celcius degrees unit, with 1 decimal digit.

Example

```
“temperatureL”: 52.6,
```

9.2.6. energyImportTotal

Total energy register of imported DC energy, in kiloWatt Hour (kWh) unit (OBIS = 1-0:1.8.0) with 3 decimal digits.

Example

```
“energyImportTotal”: 36.739,
```

9.2.7. energyExportTotal

Total energy register of exported DC energy, in kiloWatt Hour (kWh) unit (OBIS = 1-0:2.8.0) with 3 decimal digits.

Example

```
“energyExportTotal”: 0.000,
```

9.2.8. timestamp

Timestamp of the current livemeasure data set, in UTC time in ISO8601 extended dateformat (ending with the “Z” suffix).

Example

```
“timestamp”: “2020-10-12T09:55:13Z”
```


LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com

9.3. Allowed requests

9.3.1. GET - Read /livemeasure

Request format

```
GET /v1/livemeasure HTTP/1.1
```

Response format

See section “9.1. Overview”.

Status codes

Code Number	Meaning
200	OK

9.4. Examples

- Linux/Windows bash

```
curl -X GET http://192.168.1.2/v1/livemeasure
```

- Windows (PowerShell)

```
Invoke-WebRequest -uri “http://192.168.1.2/v1/livemeasure” | Select-Object -Expand Content
```


LEM International SA
Chemin des Aulx 8
1228 PLAN-LES-OUATES Switzerland
www.lem.com